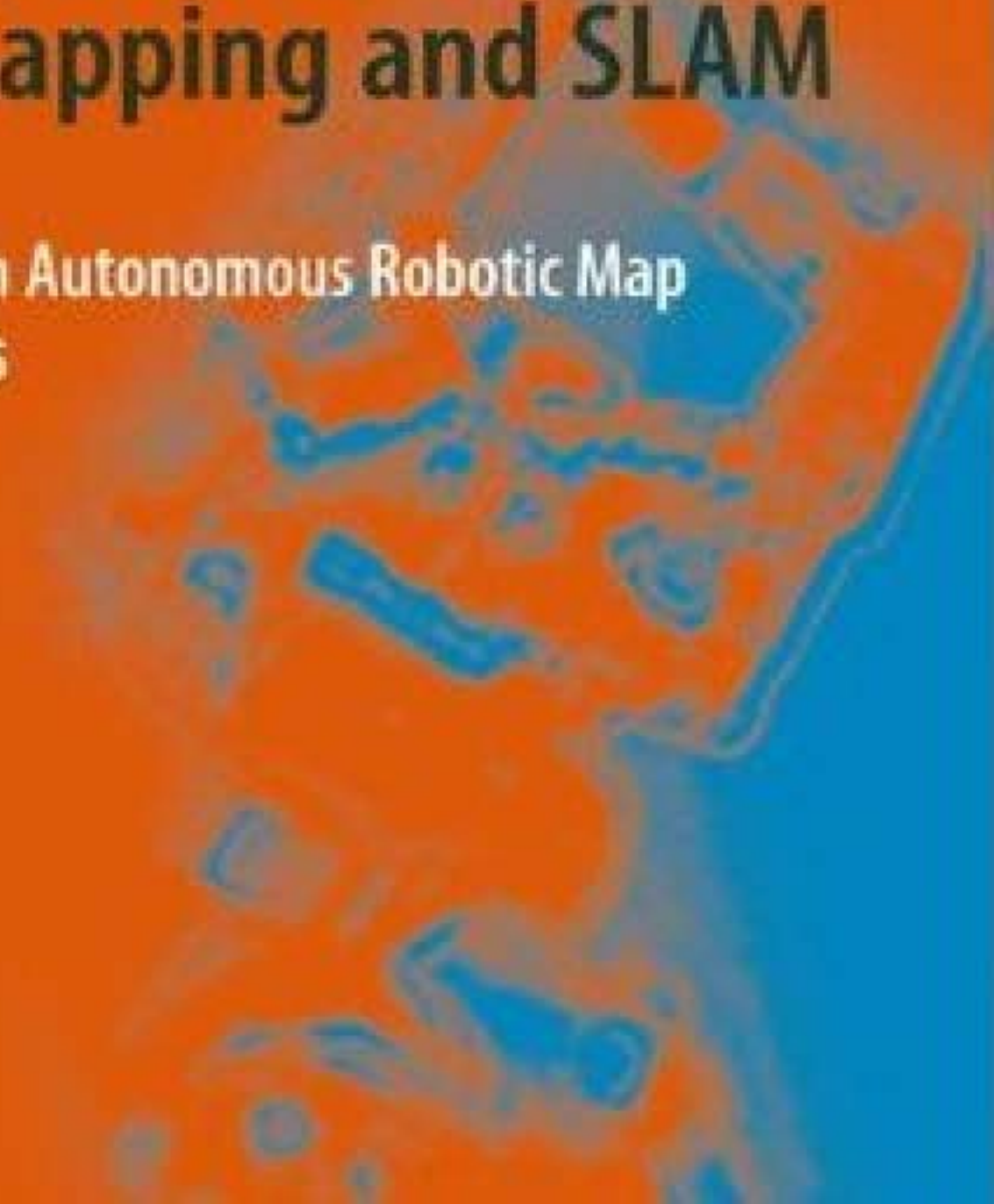
springer tracts in advanced robotics 72

John Mullane
Ba-Ngu Vo
Martin Adams
Ba-Tuong Vo

Random Finite Sets for Robot Mapping and SLAM

New Concepts in Autonomous Robotic Map Representations



Springer

Springer Tracts in Advanced Robotics

Volume 72

John Mullane, Ba-Ngu Vo, Martin Adams,
and Ba-Tuong Vo

Random Finite Sets for Robot Mapping and SLAM

New Concepts in Autonomous Robotic Map Representations

Springer

Professor Bruno Siciliano, Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy, E-mail: siciliano@unina.it

Professor Oussama Khatib, Artificial Intelligence Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305-9010, USA, E-mail: khatib@cs.stanford.edu

Professor Frans Groen, Department of Computer Science, Universiteit van Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands, E-mail: groen@science.uva.nl

Authors

Dr. John Mullane
Nanyang Technological University
School of Electrical & Electronic Engineering
50 Nanyang Avenue
Singapore 639798
Singapore
E-mail: jsmullane@ntu.edu.sg

Prof. Martin Adams
Universidad de Chile
Department of Electrical Engineering
Av. Tupper 2007
837-0451 Santiago
Chile
E-mail: martin@ing.uchile.cl

Prof. Ba-Ngu Vo
The University of Western Australia
School of Electrical, Electronic &
Computer Engineering
35 Stirling Highway
Crawley 6009
Western Australia
E-mail: bnvo@ee.uwa.edu.au

Dr. Ba-Tuong Vo
The University of Western Australia
School of Electrical, Electronic &
Computer Engineering
35 Stirling Highway
Crawley 6009
Western Australia
E-mail: btv@ee.uwa.edu.au

ISBN 978-3-642-21389-2 e-ISBN 978-3-642-21390-8

DOI 10.1007/978-3-642-21390-8

Springer Tracts in Advanced Robotics ISSN 1610-7438

Library of Congress Control Number: 2011928065

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

5 4 3 2 1 0

springer.com

STAR (Springer Tracts in Advanced Robotics) has been promoted under the auspices of EURON (European Robotics Research Network)

To our families, Ngoc, Ba-Nang, Ba-Huu, Hui-Shin and James.

Foreword

Robotics is undergoing a major transformation in scope and dimension. From a largely dominant industrial focus, robotics is rapidly expanding into human environments and vigorously engaged in its new challenges. Interacting with, assisting, serving, and exploring with humans, the emerging robots will increasingly touch people and their lives.

Beyond its impact on physical robots, the body of knowledge robotics has produced is revealing a much wider range of applications reaching across diverse research areas and scientific disciplines, such as: biomechanics, haptics, neurosciences, virtual simulation, animation, surgery, and sensor networks among others. In return, the challenges of the new emerging areas are proving an abundant source of stimulation and insights for the field of robotics. It is indeed at the intersection of disciplines that the most striking advances happen.

The *Springer Tracts in Advanced Robotics (STAR)* is devoted to bringing to the research community the latest advances in the robotics field on the basis of their significance and quality. Through a wide and timely dissemination of critical research developments in robotics, our objective with this series is to promote more exchanges and collaborations among the researchers in the community and contribute to further advancements in this rapidly growing field.

The monograph, written by John Mullane, Ba-Ngu Vo, Martin Adams and Ba-Tuong Vo, is devoted to a field of autonomous robotic systems, which has received a great deal of attention by the research community in recent years. The contents are focused on the problem of representing the environment and its uncertainty in terms of feature based maps. Random finite sets are adopted as the fundamental tool to represent a feature map, and a general framework is proposed which eliminates the need for feature management and data association, and propagates both feature state and number estimates in a joint Bayesian framework. The approaches are tested in a number of experiments on both ground based and marine based facilities.

STAR is proud to welcome yet another volume in the series dedicated to the popular area of SLAM!

Naples, Italy
April 2011

Bruno Siciliano
STAR Editor

Preface

This book is intended to demonstrate advances in the field of autonomous navigation and to serve as an essential text for academics, researchers, industrial scientists and general practitioners involved in robotic mapping, sensor modelling and the popular Simultaneous Localisation and Map building (SLAM) problem. The book focusses on a critical area of autonomous robotics research - the representation of the environment and its uncertainty, referred to as *the map*. Probabilistic maps can be primarily categorised into three popular representations: Feature based; occupancy grid and topological. This book focusses on, arguably the most popular and widely used of these, the feature based map. Two areas which are essential in designing successful autonomous vehicles are addressed: *Feature Based Robotic Mapping (FBRM)* which assumes a known robot trajectory, and an area which has been referred to as the "Holy grail" of autonomous robotics research: *Feature Based - Simultaneous Localisation and Map Building (FB-SLAM)* [1]. Rather than the commonly used framework, which stacks feature estimates and their measurements into a vector, this book advocates that a more appropriate representation for the map, in both FBRM and FB-SLAM, hereafter referred to as simply SLAM, is the *Random Finite Set* (RFS).

In both FBRM and SLAM, it is necessary to estimate the location of an initially unknown number of features, which represent the environment. In current, vector based methods the number of features and their locations, are represented by a vector of varying size. Methods are then introduced which augment this vector when new features are detected. Data association techniques are then necessary to determine which feature elements of this vector correspond to which elements of the total current observation, which is also typically represented as a vector, containing the measured attributes of the currently sensed features. Only then can a Bayesian update of the total feature map take place. Hence there is an implicit assumption that *immediately before the update, the number of map states is known.* Therefore, for a given vehicle trajectory, with error-free data association, linear process and measurement models and white Gaussian noise, optimal estimates

of a known number of feature locations are realizable using current, vector based, approaches. However, when the intrinsic properties of the map are considered (unknown number of insignificantly ordered features), this work will demonstrate that Bayes optimality has not yet been established.

This book therefore proposes generalisations of the classical vector-based frameworks for both FBRM and SLAM. These address the concept of Bayes optimality for estimation with unknown feature number by formulating both FBRM and SLAM as random set estimation problems. The proposed formulations unify the currently independent concepts of feature management, data association and state estimation adopted by previous solutions. In the case of FBRM, this occurs through the recursive propagation of a distribution of a random finite set of features, when the vehicle's trajectory is known. In the case of SLAM, the joint estimation of the vehicle location and the random finite set of features is derived.

In the case of FBRM, the RFS approach yields the propagation of the FB map density and leads to optimal map estimates in the presence of unknown map size, spurious measurements, feature detection and data association uncertainty. The proposed framework further allows for the joint treatment of error in feature number and location estimates as it jointly propagates both the estimate of the number of features and their corresponding states. In the case of SLAM, the vehicle's pose state is also jointly estimated. In both cases, under the RFS framework, the need for feature management and data association algorithms is eliminated.

An RFS is simply a finite-set-valued random variable. Similar to random vectors, the probability density (if it exists) is a very useful descriptor of an RFS, especially in filtering and estimation. However, the space of finite sets does not inherit the usual Euclidean notion of integration and density. Hence, standard tools for random vectors are not appropriate for random finite sets. Mahler's Finite Set Statistics (FISST) provide practical mathematical tools and principled approximations for dealing with RFSs [2], [3], based on a notion of integration and density that is consistent with point process theory [4]. This approach has attracted substantial research interest in the tracking community, [5], [6], [3] and this book develops these tools for both FBRM and SLAM.

Finally, in any estimation problem, the notion of *estimation error* is of utmost importance. In all FBRM and SLAM experiments, the measure of success should be a clearly defined concept. In much of the vector based, SLAM research to date, successful performance is evaluated based on the location error of a *sub-set* of the estimated features. Even if the spatial estimation errors of *all* of the estimated features were used to estimate the performance, the concept is meaningless if the *number* of features has not been estimated correctly, since the goal of SLAM is to estimate both the trajectory and the map. Therefore, for the sake of gauging the performance of

all the FBRM and SLAM results presented in this book, a consistent metric for the evaluation of feature map estimation error is presented. This metric takes into account the error in the cardinality of the map estimate in terms of the number of feature estimates, as well as their spatial locations.

Singapore — John Mullane
Perth, Australia — Ba-Ngu Vo
Santiago, Chile — Martin Adams
Perth, Australia — Ba-Tuong Vo

March 2011

Acknowledgements

The application of Finite Set Statistics (FISST) to robot mapping and SLAM has been inspired by the multi-source – multi-target work of Ronald P.S. Mahler, Lockheed Martin Tactical Systems, Minnesota, USA.

All of the experimental work demonstrated in this book was carried out in Singapore. In particular, the land based experimental work was funded under the Academic Research Fund (AcRF) Grant, RG 10/01, Singapore, and the marine based research described in this book was funded in part by the Singapore National Research Foundation (NRF) through the Singapore-MIT Alliance for Research and Technology (SMART), Centre for Environmental Sensing and Modelling (CENSAM). The theoretical work was supported in part by the Australian Research Council under the discovery grants DP0880553 and DP0989007.

The authors are indebted to Nick Patrikalakis, Franz Hover and Anthony Yeo all of whom, during their time at the SMART Centre, helped in the coordination of the coastal sea trials used to generate the results in Chapter 6. We are also grateful for the help of Akshay Rao in gathering marine data during the sea trials. The committed technical support of Chai Soon Tan and Chiu Kiat Chia at Nanyang Technological University (NTU), Singapore is also gratefully acknowledged.

The authors would like to thank their respective universities for the provision of the facilities for completing this book, namely NTU, University of Western Australia and University of Chile.

Finally we would like to acknowledge our families for their support and patience during the writing, correcting and editing of this book.

Contents

Part III: Random Finite Set Based Simultaneous Localisation and Map Building

Acronyms

FB	Feature Based
FBRM	Feature Based Robotic Mapping
OG	Occupancy Grid
SLAM	Simultaneous Localisation and Map Building
PDF	Probability Density Function
IID	Independent and Identically Distributed
std.	Standard Deviation
RFS	Random Finite Set
FISST	Finite Set Statistics
PHD	Probability Hypothesis Density
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
NN	Nearest Neighbour
MH	Multiple Hypothesis
MHT	Multiple Hypothesis Tracking
FastSLAM	Factored Solution to SLAM
RB	Rao-Blackwellised
CPHD	Cardinalised PHD
MeMBer	Multi-target Multi-Bernoulli
CMeMBer	Cardinalised MeMBer
MSE	Mean Squared Error
OMAT	Optimal Mass Transfer (Metric)
OSPA	Optimal Sub-pattern Assignment (Metric)
MAP	Maximum A Posteriori
EAP	Expected A Posteriori
MaM	Marginal Multi-Object
JoM	Joint Multi-Object
FoV	Field of View
SMC	Sequential Monte Carlo
GM	Gaussian Mixture
GMM	Gaussian Mixture Model
JCBB	Joint Compatibility Branch and Bound
MMW	Millimetre Wave

Nomenclature

k	Current time index
M_k	Feature map random vector at time k
$\widehat{M}_k$	Estimate of M_k
X_k	Vehicle pose random vector at time k
$X_{0:k}$	Vehicle trajectory random vector
$\widehat{X}_{0:k}$	Estimate of $X_{0:k}$
U_k	System control input vector at time k
$U_{0:k}$	History of system control inputs
$\mathcal{M}$	Feature map random finite set (RFS)
$\mathcal{M}_k$	Explored map RFS up to k
$\widehat{\mathcal{M}}_k$	Estimate of $\mathcal{M}_k$
$\mathcal{Z}_k$	RFS sensor measurement
$\mathcal{Z}_{0:k}$	History of RFS measurements
$\mathcal{D}_k$	Feature measurement RFS
$\mathcal{C}_k$	Clutter measurement RFS
$\mathcal{B}_k$	RFS of the new features
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$p_{k\|k}(\mathcal{M}_k, X_{0:k}\|\cdot)$	Conditional pdf of RFS-SLAM
$p_{k\|k}(\mathcal{M}_k\|\cdot)$	Conditional pdf of $\mathcal{M}_k$
$p_{k\|k-1}(\mathcal{M}_k\|\cdot)$	Predicted conditional pdf of $\mathcal{M}_k$
$g_k(\mathcal{Z}_k\|\cdot)$	Conditional pdf of $\mathcal{Z}_k$
$f_{\mathcal{M}}(\cdot\|\mathcal{M}_{k-1})$	Transition density of the map RFS
$f_X(\cdot\|X_{k-1})$	Vehicle transition density
$f_{\mathcal{D}}(\cdot)$	Density of the feature measurement RFS
$f_{\mathcal{C}}(\cdot)$	Density of the clutter measurement RFS
$f_{\mathcal{B}}(\cdot)$	Density of the new feature RFS
$v_{k\|k}(m\|\cdot)$	PHD of the explored map RFS, $\mathcal{M}_k$
$v_{k\|k-1}(m\|\cdot)$	PHD of the predicted map, $\mathcal{M}_{k\|k-1}$
$c_k(z\|\cdot)$	PHD of $\mathcal{C}_k$
$b_k(m\|\cdot)$	PHD of $\mathcal{B}_k$
$\mathfrak{m}_k$	Number of features in $\mathcal{M}_k$

$\widehat{\mathfrak{m}}_k$	Estimated number of features in $\mathcal{M}_k$
m^i	i^{th} feature in $\mathcal{M}_k$
$\mathfrak{z}_k$	Number of measurements in $\mathcal{Z}_k$
z^i	i^{th} measurement in $\mathcal{Z}_k$
$g_k(z\|m)$	Likelihood of z given feature m
$P_D(m)$	Detection probability of feature m
N	Number of particles
$\mathfrak{n}_k$	$N \times \mathfrak{m}_k$
$X_{0:k}^{(i)}$	i^{th} sample trajectory up to time k
$\eta_k^{(i)}$	Weight of i^{th} sample trajectory at time k
$v_k^{(i)}(m\|X_{0:k}^{(i)})$	i^{th} trajectory conditioned PHD of $\mathcal{M}_k$
$J_k^{(i)}$	Number of Gaussian components in the i^{th} trajectory conditioned PHD of $\mathcal{M}_k$
$\omega_k^{(i,j)}$	Weight of the j^{th} Gaussian component of the i^{th} trajectory conditioned PHD of $\mathcal{M}_k$
$\mu_k^{(i,j)}$	Mean of the j^{th} Gaussian component of the i^{th} trajectory conditioned PHD of $\mathcal{M}_k$
$P_k^{(i,j)}$	Covariance of the j^{th} Gaussian component of the i^{th} trajectory conditioned PHD of $\mathcal{M}_k$
$\bar{d}^{(c)}(\widehat{\mathcal{M}}_k, \mathcal{M}_k)$	Error between $\widehat{\mathcal{M}}_k$ and $\mathcal{M}_k$, with cut-off parameter c
$\rho_k(\mathfrak{m})$	Distribution of the size of the feature map
P_E	Probability of existence of a single feature
$\bar{m}$	Feature selected according to a given strategy

Chapter 1
Introduction

Machines which perceive the world through the use of sensors, make computational decisions based on the sensors' outputs and then influence the world with actuators, are broadly labelled as "Robots". Due to the imperfect nature of all real sensors and actuators, the lack of predictability within real environments and the necessary approximations to achieve computational decisions, robotics is a science which is becoming ever more dependent on probabilistic algorithms. Autonomous robot vehicles are examples of such machines, which are now being used in areas other than the factory floors, and which therefore must operate in unstructured, and possibly previously unexplored environments. Their reliance on probabilistic algorithms, which can interpret sensory data and make decisions in the presence of uncertainty, is increasing. Therefore, mathematical interpretations of the vehicle's environment which consider all the relevant uncertainty are of a fundamental importance to an autonomous vehicle, and its ability to function reliably within that environment. While a universal mathematical model which considers the vast complexities of the physical world remains an extremely challenging task, stochastic mathematical representations of a robots operating environment are widely adopted by the autonomous robotic community. Probability densities on the chosen map representation are often derived and then recursively propagated in time via the Bayesian framework, using appropriate measurement likelihoods.

Of crucial importance in autonomous navigation is the computational representation of a robot's surroundings and its uncertainty, referred to as the map. This book directly addresses this issue, initially in the area of *Feature Based Robotic Mapping (FBRM)* which assumes known robot location, and then in the area of *Feature Based - Simultaneous Localisation and Map Building (FB-SLAM*[1]*)*. The book demonstrates that the commonly used vector based methods for FBRM and SLAM suffer many fundamental disadvantages when applied to realistic situations. Such situations occur in environments in

[1] From here on, simply referred to as SLAM.

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 1–8.
springerlink.com

which an a-priori unknown number of features are to be estimated and in the presence of realistic sensor defects such as missed detections and false alarms. This book therefore takes a step back to the basic estimation principles and aims of the FBRM and SLAM problems, and shows that a more appropriate representation for the map, in both cases, is the *Random Finite Set* (RFS). An RFS is a random variable that take values as finite sets. It is defined by a discrete distribution that characterises the number of elements in the set, and a family of joint distributions which characterise the distribution of the element's values, conditioned on the cardinality [3].

To date, three fundamentally different approaches, namely occupancy grids [7], FB maps [8] and topological maps [9] have been applied in autonomous mapping research. Of these, occupancy grids and FB maps have emerged as the most popular means of probabilistic, environmental representation. Numerous examples of impressive localisation, mapping and navigation algorithms which adopt these environment models can be seen both in indoor [10], [11], [12], [13], [14] and outdoor [15], [16], [17], [1], [18] environments.

The Occupancy Grid approach propagates estimates of landmark existence on a grid with a fixed, predetermined number of cells. In environmental representations of this type, the number of map states is therefore predefined, and constant and therefore, only the cells' "contents", which typically correspond to the likelihood of an objects existence at that cell's coordinates, need to be updated. Hence, the grid, which fully represents the environment, can be represented mathematically by either a vector or matrix of predefined, *fixed* dimensions.

Grid based approaches however suffer from many disadvantages. Standard occupancy grid maps do not maintain the dependencies of the occupancy probability of each cell. Also, a full map posterior is generally intractable, due to the huge combinatorial number of maps which can be represented on a grid. Further, the grid size, and its resolution (cell size) must be once and for all determined prior to navigation, thus restricting the area which can be mathematically represented by the robot.

FB mapping approaches offer the advantage that the sensor data is compressed into features (such as point clusters, circles, lines, corners etc.). The origins of the feature map can be traced back to the seminal work of Smith et. al. [8], in which the environment is assumed to consist of these simplified representations of the physical landmarks – the features. The feature map representation has since gained wide spread popularity, particularly in the outdoor robotics domain due to its ability to estimate large scale maps and provide correction information for simultaneous vehicle pose estimation. The work of Smith *et. al.* also first established the "*vector of all the spatial variables, which we call the system state vector*", i.e. $M = [m^1, m^2, \cdots, m^{\mathrm{m}}]$.

FB approaches can be computationally appealing, since few features need to be detected and updated per sensor scan, and feature to feature and feature to vehicle correlations can be maintained. They fundamentally differ

from their grid based counterparts, in that they are required to estimate the location of an initially unknown number of objects, represented as features. Hence, the number of features and their locations, which fully represent the environment, are typically represented by *varying* the size of a vector. Methods are then introduced which augment this vector when new features are detected. Data association techniques are then necessary to determine which feature elements of this vector correspond to which elements of the total current observation, which is also typically represented as a vector, containing the measured attributes of the currently sensed features. Only then can a Bayesian update of the feature map take place. This concept, which summarises the current state of the art in FBRM and SLAM, is shown in Figure 1.1. It can be seen in the figure that, there is an implicit assumption that immediately before the update, the number of map states (p in Figure 1.1) to be estimated, is determined by the map management heuristics/filters just described. Therefore, with error-free data association and optimal feature initialisation routines, optimal estimates of a *predefined number* of feature locations are realizable using current, vector based, linear Gaussian approaches. However, when the intrinsic properties of the map are considered (unknown number of insignificantly ordered features), Bayes optimality of the true problem has not yet been established. As noted in the field of multi-target filtering by Mahler ([3], page 571):

"...having a good estimate of target number is half the battle in multitarget tracking. If one has 1,000 measurements but we know that roughly 900 of them are false alarms, then the problem of detecting the actual targets has been greatly simplified."

This book advocates that the same principle applies to feature maps in robotics. Realistic feature detection algorithms produce false alarms (as well as missed detections), and estimating the true number of features is therefore central to the FBRM and SLAM problems. This book therefore addresses the concept of Bayes optimality for estimation with unknown feature number, by formulating it as a random set estimation problem. The proposed formulation unifies the independent filters adopted by previous solutions, and high-lighted in Figure 1.1, through the recursive propagation of a distribution of a random finite set of features. This allows for the joint propagation of the FB map density and leads to optimal map estimates in the presence of unknown map size, spurious measurements, feature detection and data association uncertainty. The proposed framework further allows for the joint treatment of error in feature number and location estimates as it jointly propagates both the estimate of the number of landmarks and their corresponding states, and consequently eliminates the need for feature management and data association algorithms. The RFS approach to FBRM and SLAM is depicted in Figure 1.2.

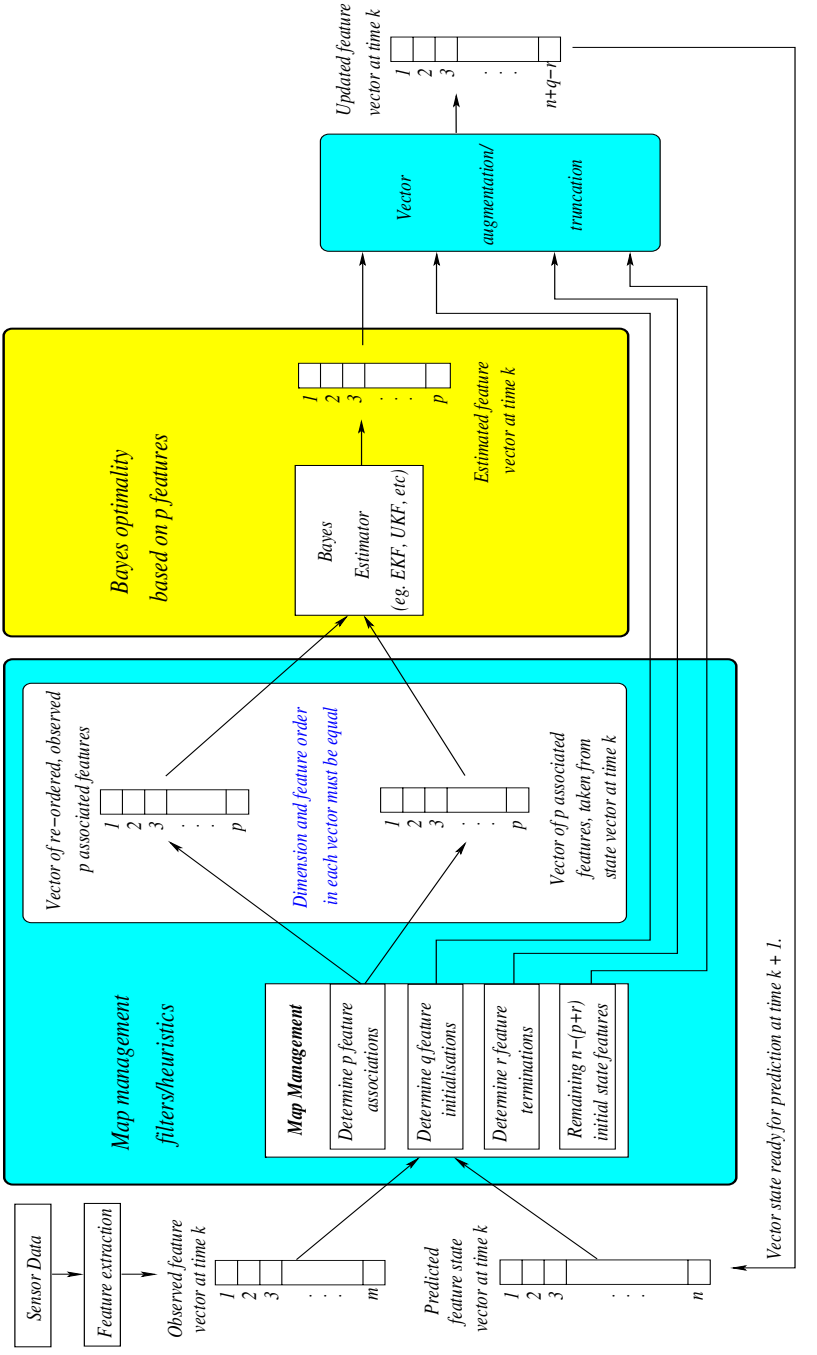


Fig. 1.1 Vector based feature mapping, for a single update cycle, in both FBRM and SLAM.

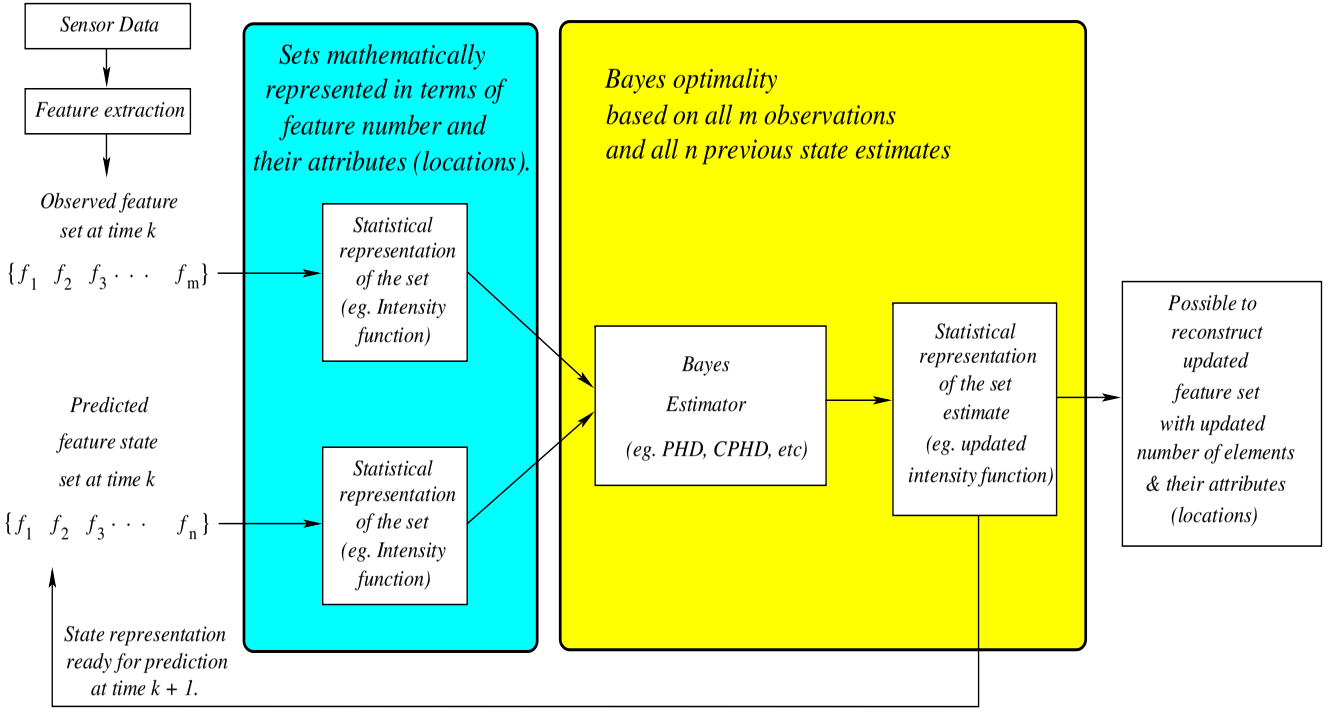


Fig. 1.2 RFS based feature mapping, for a single update cycle, in both FBRM and SLAM. Notice that no map management or data association filters/heuristics are necessary.

The main focus of the book applies the concepts of RFS mapping to the well known SLAM problem. For a FB map, SLAM requires the joint estimation of the vehicle location and the map. As in vector based robotic mapping algorithms, vector based SLAM algorithms also require 'feature management' as well as data association hypotheses and an estimator to generate the joint posterior estimate. Hence, in the final part of this book, RFS based recursive filtering algorithms are presented which jointly propagate both the estimate of the number of landmarks, their corresponding states, and the vehicle pose state, again without the need for explicit feature management and data association algorithms.

1.1 Structure of the Book

The book is divided into three parts. In part I, the question "Why use random finite sets?" is addressed. Chapter 2 summarises fundamental differences between RFS and vector based representations of features. The fundamental mathematical relationships between map states, observations and vehicle pose are examined under both vector and RFS based frameworks. Essential components of robot navigation algorithms which are mathematically inconcise, when modelled under the vector based framework, are shown to be concisely realisable under the RFS framework. The issues of map representation, data association, map management, map error quantification and the concise application of Bayes theorem will be summarised in this chapter.

Chapter 3, introduces mathematical representations which can be used for RFSs. As in the case of vector based approaches, full Bayesian estimation in the space of features and robot trajectory is intractable for all realistic scenarios. This chapter poses the fundamental question, "Given the posterior distribution of the map (and trajectory in the case of SLAM), what is the Bayes optimal estimate?" Principled estimators are therefore presented here, which are capable of representing RFSs in a Bayes optimal manner. The *probability hypothesis density* (PHD) filter is introduced as one of the simplest approximations to Bayesian estimation with RFSs. This chapter therefore provides the foundations for most of the filtering algorithms for both FBRM and SLAM, used throughout the book.

Although much of the current literature advocates that the "Mapping only problem", addressed in Part II is now a solved problem, Chapter 4 presents new insights to motivate an RFS approach to mapping. By focussing on the mapping only problem, an estimation framework which yields Bayes optimal map estimates in the general case of unknown feature number, spurious sensor measurements, feature detection and data association uncertainty is developed. Further, Chapter 4 examines in more detail, the concept of FB map estimation error, for the general case of an unknown number of features. This leads to the application of an error metric which is defined on the state space of all possible feature maps.

Part III addresses the full SLAM problem. Chapter 5 offers a "brute force" solution to the SLAM problem using RFSs, as it models the joint vehicle trajectory and map as a singular RFS, and recursively propagates its first order moment. A first order approximation of the RFS state recursion is implemented which utilises the PHD filter. Under Gaussian noise assumptions, an extended Kalman Gaussian mixture implementation is used to implement the PHD-SLAM filter. The filter jointly estimates the vehicle pose, feature number in the map and their corresponding locations. Assuming a mildly non-linear Gaussian system, an extended-Kalman Gaussian Mixture implementation of the recursion is then tested for SLAM. Simulations demonstrate SLAM in the presence of data with a high rate of spurious measurements, and comparisons with vector based SLAM are shown.

In Chapter 6 a Rao-Blackwellised (RB) implementation of the PHD-SLAM filter is proposed based on the Gaussian mixture PHD filter for the map and a particle filter for the vehicle trajectory. This applies a trajectory conditioned, PHD mapping recursion to the SLAM problem, in a similar manner to the well known FastSLAM algorithm [17]. In this sense, each particle, representing a single, hypothesised vehicle trajectory, maintains its own, conditionally independent PHD map estimate. It will be shown in this chapter that the EKF approximation used to represent each trajectory-conditioned map in FastSLAM is not valid under the RFS framework. Therefore the likelihood of the measurement set, conditioned on the robot's trajectory, but not the map, is derived in closed form. This allows the weight for each particle (representing a possible robot trajectory) to be calculated correctly. This chapter further demonstrates that, under the PHD map representation, the unique ability of *map averaging* can be achieved, in a principled manner. Simulated as well as real experimental results are shown. Also, comparisons with classical vector based SLAM algorithms and their various feature management routines, demonstrate the merits of the proposed approach, particularly in situations of high clutter and data association ambiguity.

Finally, Chapter 7 demonstrates that the RFS-FBRM and SLAM frameworks allow other approximations and implementations, besides those of the basic PHD filter, to be made. PHD-SLAM estimates the PHD of the map, encompassing the expected number of features, and the vehicle trajectory. Chapter 7 will show that the estimated PHD can be appended with the *distribution* of the number of features, as opposed to just its mean value. In this chapter, the *Cardinalised-PHD* (CPHD) filter will be introduced in which both the PHD and the feature cardinality distribution are estimated in predictor - corrector form. The map can then be readily extracted from the posterior cardinality distribution. Further, a Multi-Bernoulli representation of an RFS will be introduced, which allows each map feature to have its own probability of existence, yielding a valid density function which jointly captures its existence as well as spatial uncertainty. Known as the Cardinalised *Multi-target Multi-Bernoulli* (CMeMBer) SLAM filter, Chapter 7 shows how the existence probability and spatial density of each feature, within a robot

trajectory's map, can be propagated forward in time as measurements arrive. Under the suggested improved RFS approximations, the accuracy of the map and trajectory estimates would be expected to out perform those of the standard PHD-SLAM estimators of Chapters 5 and 6.

Chapter 2
Why Random Finite Sets?

2.1 Introduction

We begin the justification for the use of RFSs by re-evaluating the basic issues of feature representation, and considering the fundamental mathematical relationship between environmental feature representations, and robot motion. We further the justification for the use of RFSs in FBRM and SLAM by considering an issue of fundamental mathematical importance in any estimation problem - estimation error.

2.2 Environmental Representation: Fundamentals

2.2.1 FBRM and SLAM New Concepts

Consider a simplistic, hypothetical scenario in which a mobile robot traverses three different trajectories, amongst static objects, as shown in Figure 2.1. If the trajectory taken by the robot were X^1 (red), then it would seem logical that an on board sensor, with a limited range capability, may sense feature m_1 followed by m_2 followed by m_3 etc. Hence after completing trajectory X^1, if a vector M is used to represent the map, then the estimated map could be

$$\widehat{M} = [m_1 \;\; m_2 \;\; m_3 \;\; m_4 \;\; m_5 \;\; m_6 \;\; m_7]^T \tag{2.1}$$

Alternatively, had the robot pursued trajectory X^2 (blue) instead, the order in which the features would be sensed would likely be very different, and the resulting estimated map could be

$$\widehat{M} = [m_4 \;\; m_2 \;\; m_3 \;\; m_1 \;\; m_5 \;\; m_7 \;\; m_6]^T \tag{2.2}$$

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 11–25.
springerlink.com

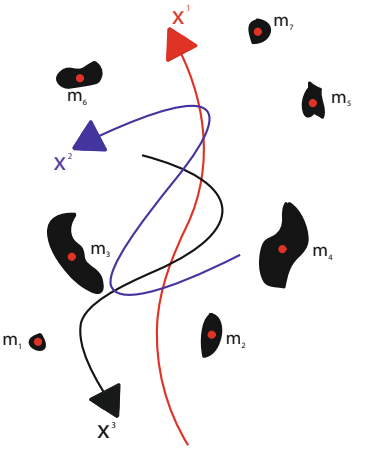


Fig. 2.1 A hypothetical scenario in which a mobile robot executes three different trajectories X^1, X^2, X^3, amidst static objects (features) m_1 to m_7.

and had the robot pursued trajectory X^3 (black), the following estimated map vector could result

$$\widehat{M} = [m_6 \;\; m_7 \;\; m_5 \;\; m_4 \;\; m_3 \;\; m_2 \;\; m_1]^T. \tag{2.3}$$

Since the order of the elements within a vector is of importance (a change in the order yields a different vector), three different map vectors result. However, since the map features themselves were assumed static, it seems odd that this estimated vector is actually dependent on the vehicle's trajectory. In a strict mathematical sense, the order of the features within the map estimate should not be significant, as any permutation of the vectors results in a valid representation of the map. By definition, the representation which captures all permutations of the elements within the vector, and therefore the features in the map, is a *finite set* $\mathcal{M}$, whose estimate $\widehat{\mathcal{M}}$ would be as shown in representation 2.4.

$$\begin{aligned} \widehat{M} &= [m_1 \; m_2 \; m_3 \; m_4 \; m_5 \; m_6 \; m_7]^T \\ \widehat{M} &= [m_4 \; m_2 \; m_3 \; m_1 \; m_5 \; m_7 \; m_6]^T \\ &\vdots \\ \underbrace{\widehat{M} = [m_6 \; m_7 \; m_5 \; m_4 \; m_3 \; m_2 \; m_1]^T}& \\ \widehat{\mathcal{M}} &= \{m_1 \; m_2 \; m_3 \; m_4 \; m_5 \; m_6 \; m_7\} \end{aligned} \tag{2.4}$$

Note that for notational purposes, we denote vector representations in italics (e.g. for the map M) and set representations in mathcal format (e.g. for the map $\mathcal{M}$).

2.2.2 Eliminating the Data Association Problem

For most sensors/sensor models considered in SLAM, the order in which sensor readings are recorded at each sampling instance, simply depends on the direction in which the vehicle/sensor is oriented, and has no significance on the state of the map, which typically evolves in a globally defined coordinate system, independent of the vehicle's pose. This is illustrated in Figure 2.2 in which a measurement to state assignment problem is evident. It can be seen

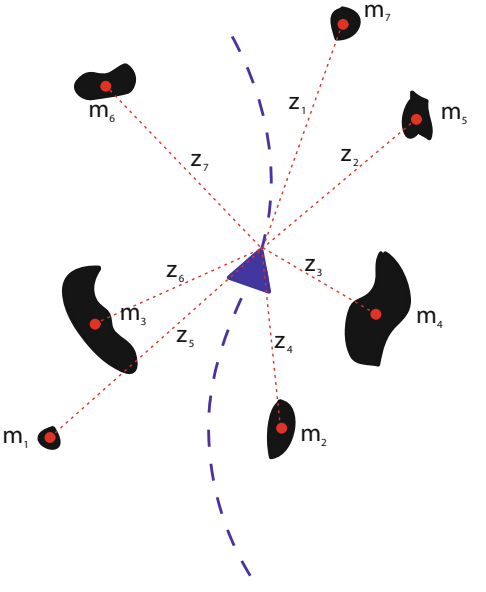


Fig. 2.2 The order in which observations (features) z_1 to z_7 are detected/extracted from the sensor data is usually different from the order of the currently estimated features m_1 to m_7 in the state vector.

in the figure, that even for an ideal sensor, which is always able to detect all of the features, all of the time, under the vector based representation, a re-ordering of the observed feature vector Z is necessary. This is because, in

general, observed feature z_1 will not correspond to the current estimate m_1 etc. (Figure 2.2) and the correct feature associations must be determined - i.e.:

$$\begin{aligned} Z &= [z_1 \;\; z_2 \;\; z_3 \;\; z_4 \;\; z_5 \;\; z_6 \;\; z_7]^T \\ &\swarrow \;\; \downarrow \;\; \searrow \;\; ??? \qquad \text{Feature Association} \\ \widehat{M} &= [m_1 \;\; m_2 \;\; m_3 \;\; m_4 \;\; m_5 \;\; m_6 \;\; m_7]^T \end{aligned} \tag{2.5}$$

It can be seen in Figure 1.1, that this data association step is necessary, before any vector based, Bayesian update can take place. Hence, current vector based formulations of the FBRM and SLAM problems require this feature association problem to be solved *prior* to the Bayesian (EKF, UKF etc) update. This is because, feature estimates and measurements are rigidly ordered in their respective finite vector valued, map states.

The proposed RFS approach on the other hand, represents both features and measurements as finite valued sets $\mathcal{M}$ and $\mathcal{Z}$ respectively, which assume no distinct ordering of the features, as shown in representations 2.4 and 2.6.

$$\begin{aligned} Z &= [z_1 \;\; z_2 \;\; z_3 \;\; z_4 \;\; z_5 \;\; z_6 \;\; z_7]^T \\ Z &= [z_4 \;\; z_2 \;\; z_3 \;\; z_1 \;\; z_5 \;\; z_7 \;\; z_6]^T \\ &\vdots \\ \underbrace{Z} &= \underbrace{[z_6 \;\; z_7 \;\; z_5 \;\; z_4 \;\; z_3 \;\; z_2 \;\; z_1]^T} \\ \mathcal{Z} &= \{z_1 \;\; z_2 \;\; z_3 \;\; z_4 \;\; z_5 \;\; z_6 \;\; z_7\} \end{aligned} \tag{2.6}$$

Since the finite set representations 2.4 and 2.6 naturally encapsulate all possible permutations of the feature map and measurement, feature association assignment does not have to be addressed. This will be demonstrated throughout the book.

2.2.3 Eliminating the Map Management Problem

For the more realistic case of non-ideal sensors/feature extraction algorithms, the number of measurements, $\mathfrak{z}_k$, at any given time is not fixed due to detection uncertainty, spurious measurements and unknown number of true features. As the robot moves through its environment, more and more features are detected, as they enter the field of view (FoV) of its sensors. Hence the map size grows monotonically as shown in Figure 2.3. In the figure it can be seen that 5 features have been detected, although there are seven features in the environment shown. Objects m_5, m_6 and m_7 lie out of range of the sensor, in the robot's current position. Due to sensor and/or feature detection

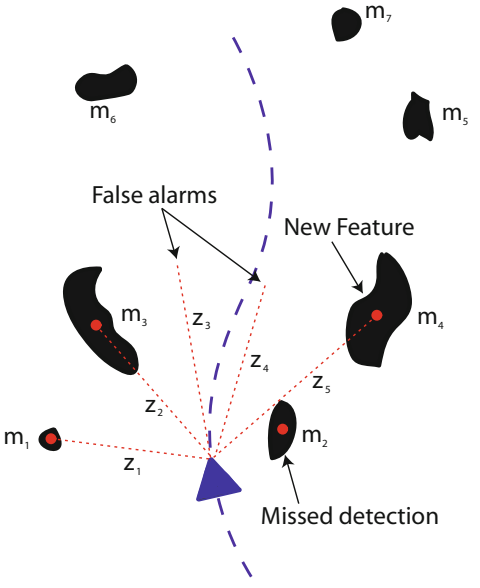


Fig. 2.3 Feature detection with a more realistic sensor. As the robot moves, new features will enter the FoV of the sensor(s). In general, some features may be undetected (missed detections), and some falsely detected features (false alarms) may be declared, due to less than ideal sensor/feature detection algorithm performance.

algorithm imperfections, two false alarms z_3 and z_4 have occurred. These can originate from clutter, sensor noise or incorrect feature detection algorithm performance. Notice also, that although object m_2 lies within the FoV of the sensor, it has not been detected, and constitutes a *missed detection*.

Suppose that features m_1, m_2 and m_3 already exist at time $k-1$ in a vector based map representation, and that feature m_4 now falls into the robot's sensor(s) FoV. Feature m_4 is to now be initialised and included in the state estimate at time k. From a strict mathematical point of view, it is unclear, using a vector based framework, how this should be carried out, as shown in equation 2.7.

$$\begin{aligned} \widehat{M}_{k-1} &= [m_1 \;\; m_2 \;\; m_3]^T \\ \widehat{M}_k &\stackrel{?}{=} [m_1 \;\; m_2 \;\; m_3]^T \text{“}+\text{”} [m_4] \end{aligned} \tag{2.7}$$

where $\stackrel{?}{=}$ is used here to mean "how do we assign $\widehat{M}_k$?" M_{k-1} represents the vector based state at time $k-1$ and M_k the corresponding state at

time k. A clear mathematical operation for combining vectors of different dimensions is not defined. To date, many FBRM and SLAM techniques use vector augmentation methods. However if the map is defined as a set, then a set based map transition function can be mathematically defined as

$$\begin{aligned} \widehat{\mathcal{M}}_{k-1} &= \{m_1 \;\; m_2 \;\; m_3\} \\ \widehat{\mathcal{M}}_{k} &= \{m_1 \;\; m_2 \;\; m_3\} \cup \{m_4\} \end{aligned} \tag{2.8}$$

Another fundamental component of any FBRM or SLAM framework is a necessity to relate observations to the estimated state. As can be seen in equations 2.9, the relationship between observations and the estimated state is not clearly defined under a vector based framework.

$$\begin{aligned} Z_k &= h([m_1 \;\; m_2 \;\; m_3 \;\; m_4], X_k) + \text{noise} \\ \text{i.e.}: \;\; [z_1 \;\; z_2 \;\; z_3 \;\; z_4 \;\; z_5]^T &\stackrel{?}{=} h([m_1 \;\; m_2 \;\; m_3 \;\; m_4], X_k) + \text{noise} \end{aligned} \tag{2.9}$$

where Z_k represents the observation vector at time k, and here the observation example of Figure 2.3 is used. X_k represents the vehicles pose at time k and $h()$ is the (typically non-linear) function relating map feature locations and the vehicle pose, to the observations. Equation 2.9 highlights the problem of relating, for example, five observations to just four feature locations, and the robot's pose. The extra observed features are clearly the result of clutter in this case, and one feature has been missed (undetected). How such "clutter" observations, and missed detections can be incorporated into the vector based measurement equation is undefined. Clearly, assuming that single features give rise to at most single observations, there is an inconsistency, due to the mismatch in the map state and observation vectors' dimensions. In the case of vectors, map management heuristics are typically used to first remove one of the observed features so that the equation can be "forced to work".

If set based measurements and state map estimates are used, a strict mathematical relationship is possible as shown in equation 2.10

$$\mathcal{Z}_k = \bigcup_{m \in \mathcal{M}_k} \mathcal{D}_k(m, X_k) \cup \mathcal{C}_k(X_k) \tag{2.10}$$

where $\mathcal{D}_k(m, X_k)$ is the RFS of measurements generated by a feature at m, and dependent on X_k and $\mathcal{C}_k(X_k)$ is the RFS of the spurious measurements at time k, which may depend on the vehicle pose X_k. Therefore $\mathcal{Z}_k = \{z_k^1, z_k^2, \ldots, z_k^{\mathfrak{z}_k}\}$ consists of a random number, $\mathfrak{z}_k$, of measurements, whose order of appearance has no physical significance *with respect to the estimated map of features*.

As a result of the data association methods and map management rules which are necessary when the vector based representation is used for FBRM and SLAM, it is clear that the uncertainty in the *number of features* is not

modelled. Typically, post-processing (outside of the Bayesian estimation component) filters are required to estimate the feature number [14]. If an RFS approach is used however, the uncertainty in both the feature state values (typically locations) *and* number can be modelled in a consistent, joint mathematical manner.

2.3 FBRM and SLAM Error Quantification

Fundamental to any state estimation problem is the concept of estimation error. While the concept of error quantification is well established in the occupancy grid literature [16, 19], in the feature-based literature the topic is less rigorously addressed. Current error evaluations of feature-based frameworks typically analyse the consistency of a subset of the feature location estimates [1], [15]. While this may illustrate the consistent spatial state estimates of the selected features, it gives no indication as to the quality of the estimate of the joint multi-feature map state. Qualitative analysis, in which estimated map features and robot location are superimposed on satellite images [17], is also not mathematically consistent and overlooks the underlying estimation problems of the feature map, namely that of the error in the estimated *number and location* of features in the map.

Whilst the majority of autonomous navigation work focuses on the localisation accuracy that can be achieved, the accuracy of the resulting map estimate is of equal importance. A precise measurement of the robots surroundings is essential to any task or behaviour the robot may be required to perform. A broad range of exteroceptive sensors are generally deployed on autonomous vehicles to acquire information about the surrounding area. Many sensors, such as laser range finders, sonars and some types of radar, measure the relative range and bearing from the vehicle to environmental landmarks and are used to update the time predicted map state. Such measurements are however subject to uncertainty such as range and bearing measurement noise, detection uncertainty, spurious measurements and data association uncertainty [20], [16].

This section demonstrates that, in the context of jointly evaluating the error in the estimated number of features and their locations, and their true values, the collection of features, should be represented by a finite set. The rationale behind this representation traces back to a fundamental consideration in estimation theory - estimation error. Without a meaningful notion of estimation error, estimation has very little meaning. Despite the fact that mapping error is equally as important as localisation error, its formal treatment has been largely neglected.

To illustrate this point, recall that in existing SLAM formulations the map is constructed by stacking features into a vector, and consider the simplistic scenarios depicted in figure 2.4. Figure 2.4a depicts a scenario in which there are two true features at coordinates $(0, 0)$ and $(1, 1)$. The true map, M, is

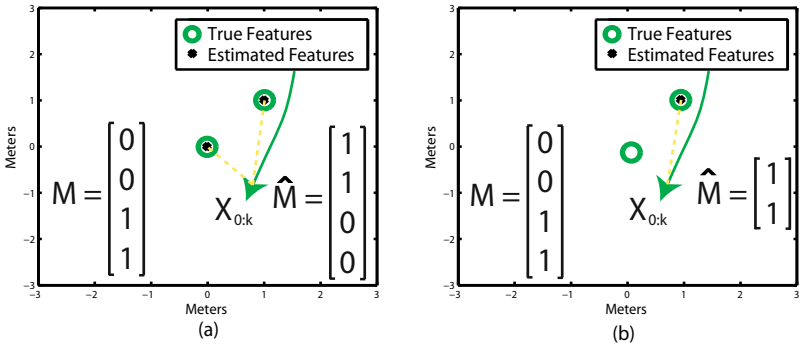


Fig. 2.4 A hypothetical scenario showing a fundamental inconsistency with vector representations of feature maps. If M is the true map, how should the error be assigned when the number of features in the map estimate, $\widehat{\mathrm{M}}$, is incorrect?

then represented by the vector $M = [0\ 0\ 1\ 1]^T$. If features are stacked into a vector in order of appearance then, given a vehicle trajectory $X_{0:k}$ (e.g. as shown in the figure) and perfect measurements, the estimated map may be given by the vector $\widehat{M} = [1\ 1\ 0\ 0]^T$. Despite a seemingly perfect estimate of the map, the Euclidean error of the estimated map, $||M - \widehat{M}||$, is 2. This inconsistency arises because the ordering of the features in the representation of the map should not be relevant. A vector representation however, imposes a mathematically strict arrangement of features in the estimated map based on the order in which they were detected [21], [1]. Intuitively, the elements of $\widehat{M}$ could be permuted to obtain a zero error, however, the representation of all possible permutations of the elements of a vector is, by definition, a set. Hence, such a permuting operation implies that the resulting error distance is no longer a distance for vectors but a distance for sets, and thus this book derives a set based approach to SLAM. Another problem is depicted in figure 2.4b, in which there are again two features at (0,0) and (1,1), but due to a missed detection (for instance), the estimated map comprises only one feature at (1,1). In such a situation, it is difficult to define a mathematically consistent error metric (Euclidean error, Mean Squared Error) between the vectors M and $\widehat{M}$ since they contain a different number of elements. It is evident from these examples that stacking individual features into a single vector does not lead to a natural notion of mapping error, in general.

A *finite set* representation of the map, $\mathcal{M}_k = \{m_k^1, \ldots, m_k^{\mathfrak{m}_k}\}$, where $m^1, \ldots, m^{\mathfrak{m}_k}$ are the $\mathfrak{m}_k$ features present at time k, admits a mathematically consistent notion of estimation error since the 'distance', or error between sets, is a well understood concept. Examples of such 'distance' metrics include the Hausdorff, Optimal Mass Transfer (OMAT) [22] and Optimal Sub-pattern Assignment (OSPA) [23] distances.

To the authors' knowledge, despite widespread quantification of localisation estimation error, the absolute difference between all estimated and actual features in the map is rarely jointly considered[1]. As an example, Figure 2.5 shows a hypothetical posterior map estimate returned by two separate feature mapping filters. If the true feature map, $M = \{m^1, \ldots, m^{\mathfrak{m}_k}\}$ (shown as green circles) and the estimated map $\widehat{M} = \{\widehat{m}^1, \ldots, \widehat{m}^{\hat{\mathfrak{m}}_k}\}$ (shown as black crosses), where $\mathfrak{m}_k$ is the total number of features in the map and $\hat{\mathfrak{m}}_k$ is the estimated number of features in the map, which map estimate is closer to M?. While visual perception may indicate that the left-hand map estimate is superior, an accepted metric to answer this fundamental question is lacking in the mobile robotics community. Suitable error metrics to address this problem, will be the subject of Chapter 4.

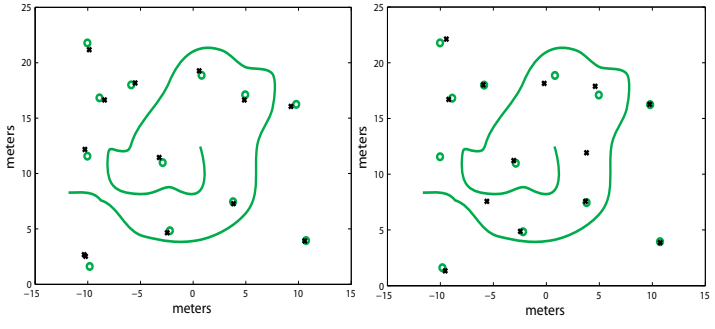


Fig. 2.5 Hypothetical posterior estimates from a feature mapping filter, $\widehat{M}_{left}$ and $\widehat{M}_{right}$, with true feature locations (green circles) and estimated feature locations (black crosses) shown.

2.4 Bayesian FBRM and SLAM with Vectors and Sets

This section outlines the Bayesian recursion which is central to the majority of FBRM and SLAM stochastic mapping algorithms. Let M denote a generic mathematical representation of the environment to be mapped, $Z_k = \{z_k^1, ..., z_k^{\mathfrak{z}_k}\}$ denote the collection of $\mathfrak{z}_k$ sensor measurements at time k and X_k be the vehicle pose, at time k. In the case of FBRM, the aim is to then propagate the posterior density of the map from the measurement and pose history, $Z_{0:k} = [Z_1, ..., Z_k]$ and $X_{0:k} = [X_1, ..., X_k]$ respectively. Maximum a posteriori (MAP) or expected a posteriori (EAP) estimates may then be extracted from the posterior density at each time k.

[1] Approaches examining the consistency of a subset of feature estimates are common however [1, 15].

Assuming that such a density exists, from an optimal Bayesian perspective the posterior Probability Density Function (PDF)

$$p_{k|k}(M|Z_{0:k}, X_{0:k})$$

captures all the relevant statistical information about the map, up to and including time k. The posterior PDF of the map can in principle be propagated in time via the well-known Bayes recursion,

$$p_{k|k}(M|Z_{0:k}, X_{0:k}) \propto g_k(Z_k|M, X_k)p_{k|k-1}(M|Z_{0:k-1}, X_{k-1}). \tag{2.11}$$

For clarity of exposition, this static mapping only problem is adhered to in the first part of this book. This formulation can however be easily extended to the SLAM problem in which the full posterior $p_{k|k}(X_{0:k}, M_k|Z_{0:k})$ can be propagated in time. The formulation can be further extended to incorporate dynamic maps and multiple vehicle SLAM, which will be the subject of the final chapter of this book.

A mathematical representation of the map, M, is required before the likelihood, $g_k(Z|M, X_k)$, and prior, $p_{k|k-1}(M|Z_{0:k-1}, X_{k-1})$, can be well defined. Bayesian based estimation of both occupancy grid (OG), vector FB and RFS FB map representations are now addressed. The following sections highlight the advantages of RFS over vector based formulations, in terms of Bayes optimality.

2.4.1 Bayesian Estimation with Occupancy Grids

Since its inception by Moravec and Elfes [7], the occupancy grid map, denoted $M = [m^1, m^2, \cdots, m^{\mathfrak{m}}]$, has been widely accepted as a viable mathematical representation of a given environment. In the context of an Occupancy Grid, $\mathfrak{m}$, represents a fixed number of spatial cells, usually distributed in the form of a lattice, which are obtained via an *a priori* tessellation of the spatial state space. Each grid cell is then denoted 'Occupied', if a landmark[2] exists in the cell, and 'Empty', if the cell is empty of landmarks. The recursion of equation 2.11, then propagates the posterior density on the occupancy grid, typically by invoking a grid cell independence assumption,

$$p_{k|k}(M|Z_{0:k}, X_{0:k}) = \prod_{i=1}^{i=\mathfrak{m}} p_{k|k}(m^i|Z_{0:k}, X_{0:k})$$

with $p_{k|k}(m^i|Z_{0:k}, X_{0:k})$ denoting the probability, α, of a landmark existing in cell m^i. The occupancy grid environment representation is attractive due

[2] Note in this work, a 'landmark' refers to any physical object in the environment. A 'feature' then refers to a simplified representation of a landmark.

to its ability to model arbitrary landmarks as the cell number tends to infinity. An important, and rarely examined aspect of the grid approach however, is that the *number* of grid cells, $\mathfrak{m}$, is inherently known *a priori*. This has a fundamental impact on the optimality of the Bayesian recursion since it means that only the occupancy of each cell needs to be estimated and not the *number* of grid cells. Thus a vector-valued map state can be used to represent the grid cells since, in this case, it is not necessary to encapsulate uncertainty in the number of states. Given the existence estimation state space of the representation, stochastic detector dependent measurement likelihoods are also required [16]. Much of the grid based mapping literature distributes occupancy uncertainty in the spatial space to model the uncertainty of the sensing and map estimation process [24], [20]. However, while this environmental representation deals with detection and spurious measurements to propagate the landmark existence estimate, such a representation in its mathematical structure does not inherently encapsulate and propagate the spatial uncertainty of sensor measurements [16]. This will be explained further in Section 2.5. A true spatial state space is explicitly considered in the feature map representation described next.

2.4.2 Bayesian Estimation with a Vector Feature Map

While defining a vector-valued feature map representation may appear to be a trivial case of terminology, in fact it has already been demonstrated that it has numerous mathematical consequences [3], namely an inherent rigid ordering of variables and a fixed length. The feature map approach has long been recognised as a "*a state estimation problem involving a variable number of dimensions (features)*" [25], however a vector representation for a feature map can only represent a *fixed* number of features. That is, the posterior vector feature map density,

$$p_{k|k}(M = [m^1, m^2, \cdots, m^{\hat{\mathfrak{m}}_k}]|Z_{0:k}, X_{0:k})$$

represents the spatial density of $\hat{\mathfrak{m}}_k$ features only, and does not encapsulate uncertainty in feature number. This limitation of vector representations is not new to robotics researchers and the sub-optimal map management methods mentioned in Section 2.2 and shown in Figure 1.1 are subsequently adopted to adjust the estimate of $\mathfrak{m}_k$ through 'augmenting' and 'pruning' filtering/heuristic based operations [17], [1]. More advanced methods, which allow reversible data association across a finite window of time frames have also been considered [18], [26]. Furthermore, the order of the features $1, \ldots, \hat{\mathfrak{m}}_k$ in the vector is fixed, coupled with a vector-valued measurement equation (also of rigid order), which leads to the need for costly data association algorithms to decide the measurement-feature assignment. This can be seen in the case

of SLAM, as applying Bayes theorem (equation 2.11) to a vector valued map involves the following steps:

- Predicted time update, based on the previous vehicle pose and previous inputs to the robot (typically speed, steering commands):

$$p_{k|k-1}(\ X_{0:k}, M_k|Z_{0:k-1}, U_{0:k-1}, X_0) = \int f_X(X_{0:k}, M_k|X_{0:k-1}, M_{k-1}, U_{k-1}) \times p_{k-1|k-1}(X_{0:k-1}, M_{k-1}|Z_{0:k-1}, U_{0:k-2}, X_0)dX_{k-1} \quad (2.12)$$

- Acquire the measurement vector Z_k.
- Carry out feature associations *before* the application of Bayes theorem.
- Perform the measurement update:

$$p_{k|k}(X_{\ 0:k}, M_k|Z_{0:k}, U_{0:k-1}, X_0) = \frac{g_k(Z_k|M_k, X_k)p_{k|k-1}(X_{0:k}, M_k|Z_{0:k-1}, U_{0:k-1}, X_0)}{\int\int g_k(Z_k|M_k, X_k)p_{k|k-1}(X_{0:k}, M_k|Z_{0:k-1}, U_{0:k-1}, X_0)dX_k dM_k} \quad (2.13)$$

- Perform independent map management.

It is important to note that when both the measurement likelihood[3] $g_k(Z_k|M_k, X_k)$ and the predicted SLAM state $p_{k|k-1}(X_{0:k}, M_k|Z_{0:k-1}, U_{0:k-1},\ X_0)$, in the numerator of equation 2.13, are represented by random vectors, they must be of compatible dimensions *before* the Bayes update can be carried out. This is why the independent data association step is necessary. It is also of importance to note that the SLAM state and feature number are not jointly propagated or estimated.

The next section introduces the finite set representation for a feature map, which yields the joint encapsulation of the feature number and spatial uncertainty as well as their optimal joint estimation.

2.4.3 Bayesian Estimation with a Finite Set Feature Map

Inconsistencies in the classical vector feature map representation can be demonstrated through a simple question: How is a map with no features represented by a vector? A set can represent such a case through the null set. Furthermore, due to the unknown number of features in a map and the

[3] Note the notational change for the measurement likelihood. Throughout this book, the p_k notation is used only on the densities from which state estimates are to be extracted via a suitable Bayes optimal estimator. While not commonly used, we believe that denoting the measurement likelihood by g_k, adds clarity and improves readability.

physical insignificance of their order, the feature map can be naturally represented as a finite set, $\mathcal{M} = \{m^1, m^2, \cdots, m^{\mathfrak{m}}\}$. A random finite set (RFS) then encapsulates the uncertainty in the finite set, i.e. uncertainty in feature number and their spatial states. Thus, an RFS feature map can be completely specified by a discrete distribution that models the uncertainty in the number of features, and continuous joint densities that model their spatial uncertainty, conditioned on a given number estimate. In a similar vein to the previous vector feature map (for FBRM), an RFS can be described by its PDF

$$p_{k|k}(\mathcal{M} = \{m^1, m^2, \cdots, m^{\hat{\mathfrak{m}}_k}\} | \mathcal{Z}_{0:k}, X_{0:k})$$

and propagated through a Bayesian recursion as follows:

- Predicted time update, based on the previous vehicle pose and previous inputs to the robot:

$$\begin{aligned} p_{k|k-1}(\, X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0) = \\ \int f_X(X_{0:k}, \mathcal{M}_k | X_{0:k-1}, \mathcal{M}_{k-1}, U_{k-1}) \times \\ p_{k-1|k-1}(X_{0:k-1}, \mathcal{M}_{k-1} | \mathcal{Z}_{0:k-1}, U_{0:k-2}, X_0) dX_{k-1} \end{aligned} \quad (2.14)$$

- Acquire the measurement set $\mathcal{Z}_k$.
- Perform the measurement update:

$$p_{k|k}(\, X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k}, U_{0:k-1}, X_0) = \quad (2.15)$$
$$\frac{g_k(\mathcal{Z}_k | \mathcal{M}_k, X_k) p_{k|k-1}(X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0)}{\int \int g_k(\mathcal{Z}_k | \mathcal{M}_k, X_k) p_{k|k-1}(X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0) dX_k \delta \mathcal{M}_k}$$

  where δ implies set integration.

Contrary to the vector based implementation of Bayes theorem in equation 2.13, it is important to note that the measurement likelihood $g_k(\mathcal{Z}_k | \mathcal{M}_k, X_k)$ and predicted SLAM state $p_{k|k-1}(X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0)$ in the numerator of equation 2.15, are finite set statistics (FISST) representing the RFS, which do not have to be of compatible dimensions.

Integration over the map in the denominator of equation 2.15 requires integration over all possible feature maps (all possible locations *and* numbers of features). By adopting an RFS map representation, integrating over the map becomes a set integral. This feature map recursion therefore encapsulates the inherent feature number uncertainty of the map, introduced by detection uncertainty, spurious measurements and vehicle manoeuvres, as well as the feature location uncertainty introduced by measurement noise. Features are not rigidly placed in a map vector, nor are measurements simply a direct function of the map state, due to the explicit modelling of clutter. Therefore, contrary to previous vector represented approaches, no explicit measurement-feature assignment (the data association problem) is required.

Hence, by adopting an RFS representation of the mapped and observed features, Bayes theorem can be applied to jointly estimate the feature state, number and vehicle pose for SLAM.

2.5 Further Attributes of the RFS Representation

To date, a map representation which unifies the existence filtering state-space of the occupancy map representation and the spatial state-space of the feature map representation remains elusive. While previous researchers generally adopt independent filters to propagate the spatial and existence posteriors of a vector feature map, such an approach leads to some theoretical inconsistencies. For instance, consider the posterior density for a single feature map, $p_{k|k}(M{=}[m]|Z_{0:k})$. In order for the Bayesian recursion of equation 2.11 to be valid, the density must be a PDF, i.e. $\int p_{k|k}(M|Z_{0:k})dM = 1$. This however implicitly implies that the feature definitely exists somewhere in the map. By using a separate existence filter to obtain an existence probability of α, the implication is that $\int p_{k|k}(M|Z_{0:k})dM = \alpha$, which subsequently violates a fundamental property of a PDF $\forall\ \alpha \neq 1$. For such a case, it is evident that a vector-valued feature map representation cannot jointly incorporate feature existence and location uncertainty.

An RFS framework can readily overcome these issues. For instance, an analogous joint recursion can be obtained by adopting a Poisson RFS to represent the feature map. This approach does not maintain an existence estimate on each feature, but instead propagates a density which represents the mean number of features in the map as well as their spatial densities. An alternative RFS map model is a multi-Bernoulli RFS, as will be shown in Chapter 3 (equation 3.1), which can jointly encapsulate the positional and existence uncertainty of each individual feature under a valid PDF, which can be subsequently propagated and estimated via the so called MeMBer Filter.

2.6 Summary

This chapter has provided several motivations for the theoretical representation of feature based maps to take the form of RFSs as opposed to the classically used random vectors. Indeed it has been demonstrated that a vector representation of the map introduces many algorithmic/mathematical consequences, in the forms of the ordering of features within the estimated map and observation vectors; the data association problem; the map management problem; feature map error quantification and the problems of vector dimensionality differences within a vector based, Bayes recursion. It was demonstrated that these mathematical consequences result in algorithmic routines which typically augment or truncate vectors outside of the Bayesian FBRM/SLAM recursions, resulting in Bayes optimality only being possible on a predetermined subset of the detected features.

The RFS representation has been conceptually introduced as a means in which the Bayes optimal estimation of both feature number and spatial state, is achievable without the need for such pre-Bayesian augmenting/truncating methods. Indeed, it was highlighted that no data association is necessary at all, under the RFS framework. This naturally leads us to the scope of Chapter 3, in which mathematically tractable, RFS based approximations are derived, for Bayes optimal FBRM and SLAM.

Chapter 3
Estimation with Random Finite Sets

3.1 Introduction

The previous chapter provided the motivation to adopt an RFS representation for the map in both FBRM and SLAM problems. The main advantage of the RFS formulation is that the dimensions of the measurement likelihood and the predicted FBRM or SLAM state do not have to be compatible in the application of Bayes theorem, for optimal state estimation. The *implementation* of Bayes theorem with RFSs (equation 2.15) is therefore the subject of this chapter. It should be noted that in any realistic implementation of the *vector* based Bayes filter, the recursion of equation 2.13 is, in general, intractable. Hence, the well known extended Kalman filter (EKFs), unscented Kalman filter (UKFs) and higher order filters are used to approximate multi-feature, vector based densities. Unfortunately, the general RFS recursion in equation 2.15 is also mathematically intractable, since multiple integrals on the space of features are required. This chapter therefore introduces principled approximations which propagate approximations of the full multi-feature posterior density, such as the expectation of the map. Techniques borrowed from recent research in point process theory known as the *probability hypothesis density* (PHD) filter, *cardinalised probability hypothesis density* (C-PHD) filter, and the *multi-target, multi-Bernoulli* (MeMBer) filter, all offer principled approximations to RFS densities. A discussion on Bayesian RFS estimators will be presented, with special attention given to one of the simplest of these, the PHD filter. In the remaining chapters, variants of this filter will be explained and implemented to execute both FBRM and SLAM with simulated and real data sets.

The notion of Bayes optimality is equally as important as the Bayesian recursion of equation 2.15 itself. The following section therefore discusses optimal feature map estimation in the case of RFS based FBRM and SLAM, and once again, for clarity, makes comparisons with vector based

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 27–42.
springerlink.com

estimators. Issues with standard estimators are demonstrated, and optimal solutions presented.

3.2 Classical State Estimators

In this section, we pose a fundamental question: "Given the posterior distribution of the map/SLAM state, what is the Bayes "optimal" estimate?" While an RFS map representation can jointly encapsulate feature number and location uncertainty, the problem of extracting the optimal estimate from the posterior density (in the case of RFS SLAM), $p_{k|k}(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0)$ of equation 2.15, is not straight forward. This section therefore outlines certain technical inconsistencies of traditional estimators, leading to summaries of principled approaches in Section 3.3 (for more details see [3], [27]).

3.2.1 Naive Estimators

The difficulty of applying standard estimators to RFSs arises because they represent information on the number of their elements (features) which is a dimensionless quantity, and the elements themselves, which can have dimensions (in the case of features – their location, in units of distance from a globally defined origin). To demonstrate some of the difficulties in deriving useful estimators for RFSs, consider the following example in which a PDF $p()$ on the RFS $\mathcal{M}$, representing an entire, unknown map, is assumed to be available. Intuitive, and standard, expected a posteriori (EAP) and maximum a posteriori (MAP) estimators are applied to a seemingly simple estimation problem [3].

Consider a simplistic situation in which there is at most one feature located in the map. Suppose that a corresponding feature existence filter [28] reports a 0.5 probability of the feature being present. Suppose also that, if the feature is present, the posterior density of the corresponding spatial state, $p(\mathcal{M})$, indicates that it is equally likely to be found anywhere in the one-dimensional interval $[0, 2]$, with the unit of distance given in meters. It should be noted here that, already at this simplistic level, vector approaches cannot jointly model this feature state. Under an RFS representation however, the map state $\mathcal{M}$ can be defined as Bernoulli RFS, with probability density,

$$p(\mathcal{M}) = \begin{cases} 0.5 & \mathcal{M} = \emptyset \\ 0.25 & \mathcal{M} = \{m\}, 0 \leq m \leq 2 \\ 0 & \text{otherwise} \end{cases} . \tag{3.1}$$

Note that the density, $p(\mathcal{M})$, is still a valid PDF, since its integral, with respect to $\mathcal{M}$, equates to unity. However, in this case, it is difficult to define an expected a posteriori (EAP) estimate since the addition of sets is not meaningfully defined. Instead, a naive maximum a posteriori (MAP) estimate could be constructed as,

$$\widehat{\mathcal{M}}^{MAP} \stackrel{?}{=} \arg\sup_{\mathcal{M}} p(\mathcal{M}) = \emptyset, \tag{3.2}$$

(where $\stackrel{?}{=}$ represents a question "Is it equal to?"), since $p(\emptyset) > p(\{m\})$ $(0.5 > 0.25)$. If the unit of distance is changed from meters to kilometres, the spatial probability density consequently becomes $p(m) = U(0, 0.002)$, and the probability density of the map state $\mathcal{M}$ is,

$$p(\mathcal{M}) = \begin{cases} 0.5 & \mathcal{M} = \emptyset \\ 250 & \mathcal{M} = \{m\}, 0 \leq m \leq 0.002 \\ 0 & \text{otherwise} \end{cases},$$

and the naive MAP estimate then becomes,

$$\widehat{\mathcal{M}}^{MAP} \stackrel{?}{=} \arg\sup_{\mathcal{M}} p(\mathcal{M}) = \{m\} \tag{3.3}$$

for any $0 \leq m \leq 0.002$ since $p(\{m\}) > p(\emptyset)$ $(250 > 0.5)$. This leads to the conclusion, that a target is now present, even though only the units of measurement have changed. This arises since the naive MAP yields a mathematical paradigm which compares a dimensionless quantity $p(\mathcal{M})$ (when $\mathcal{M} = \emptyset$) to a quantity $p(\mathcal{M})$ with dimensions (when $\mathcal{M} = m$). Such an MAP estimate is not well-defined since a change in the units of measurement results in a dramatic change in the estimate. In fact the MAP is only defined if the units of all possibilities are the same, such as in discretised state spaces, divided into cells.

This example has shown that standard estimators (EAP and MAP) are not well defined in the presence of non-unity target existence probability. It is therefore the aim of the next section to introduce new multi-target state estimators which are well behaved.

3.3 Bayes Optimal RFS Estimators

Several principled solutions to performing multi-object state estimation are now presented, in the form of two statistical estimators and a first order moment technique (the PHD filter) with desirable properties.

We begin by opening the discussion on the full SLAM problem in terms of joint Bayes optimal estimators for the vehicle trajectory and the map. The

Bayes risk is then defined for the map along with corresponding feature map estimators. Finally, Bayes optimal estimation approximations for FBRM and SLAM are derived.

This section discusses various Bayesian estimators for the SLAM problem and their optimality, based on a vector representation of the vehicles trajectory, and a finite-set representation of the map. The notion of Bayes optimal estimators is fundamental to the Bayesian estimation paradigm. In general, if the function $\hat{\theta} : z \mapsto \hat{\theta}(z)$ is an estimator of a parameter θ, based on a measurement z, and $C(\hat{\theta}(z), \theta)$ is the penalty for using $\hat{\theta}(z)$ to estimate θ, then the Bayes risk $R(\hat{\theta})$ is the expected penalty over all possible realisations of the measurement and parameter, i.e

$$R(\hat{\theta}) = \int \int C(\hat{\theta}(z), \theta) p(z, \theta) d\theta dz \tag{3.4}$$

where $p(z, \theta)$ is the joint probability density of the measurement z and the parameter θ. A Bayes optimal estimator is any estimator that minimises the Bayes risk.

In the SLAM context, relevant Bayes optimal estimators are those for the vehicle trajectory and the map. The posterior densities[1] $p_k(X_{1:k}) \triangleq p_k(X_{1:k}|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0)$ and $p_k(\mathcal{M}_k) \triangleq p_k(\mathcal{M}_k|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0)$ for the vehicle trajectory and map, can be obtained by marginalising the joint posterior density, $p_k(\mathcal{M}_k, X_{1:k}|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0)$. For the vehicle trajectory, the posterior mean, which minimises the mean squared error (MSE), is a widely used Bayes optimal estimator. However, since the map is a finite set, the notion of MSE does not apply. Moreover, standard Bayes optimal estimators are defined for vectors and subsequently do not apply to finite-set feature maps. To the best of authors' knowledge, there is no work which establishes the Bayes optimality of estimators for finite-set feature maps (and consequently feature-based SLAM). Therefore the following sections propose frameworks for Bayes optimal estimation with RFSs, which assume varying degrees of approximation to the statistical representations of sets.

3.3.1 Bayes Risk in Feature Map Estimation

The convergence of the vehicle location estimation aspect, of feature-based frameworks, has received a great deal of attention to date [1]. However, to the authors' knowledge, the convergence of the corresponding map estimate, particularly with regards to converging to the true number of features, has never before been addressed or proven. Therefore, to address the optimal map estimation problem, as before, let $\mathcal{M}_k$ denote the feature-based map state at time k comprising $\mathfrak{m}_k$ features and $p_k(\mathcal{M}_k)$ denote its posterior density.

[1] Note that henceforth for compactness, $p_k(\cdot) = p_{k|k}(\cdot)$.

If $\widehat{\mathcal{M}}_k : \mathcal{Z}_{1:k} \mapsto \widehat{\mathcal{M}}_k(\mathcal{Z}_{1:k})$ is an estimator of the feature map $\mathcal{M}_k$, and $C(\widehat{\mathcal{M}}_k(\mathcal{Z}_{1:k}), \mathcal{M}_k)$ is the penalty for using $\widehat{\mathcal{M}}_k(\mathcal{Z}_{1:k})$ to estimate $\mathcal{M}_k$, then the Bayes risk for mapping is given by

$$R(\widehat{\mathcal{M}}_k) = \int \int C(\widehat{\mathcal{M}}_k(\mathcal{Z}_{1:k}), \mathcal{M}_k) p_k(\mathcal{M}_k, \mathcal{Z}_{1:k}) \delta\mathcal{M}_k \delta\mathcal{Z}_{1:k}.$$

where $p_k(\mathcal{M}_k, \mathcal{Z}_{1:k})$ is the joint density of the map and measurement history. Note that since the map and measurements are finite sets, standard integration for vectors is not appropriate for the definition of the Bayes risk. Subsequently the Bayes risk above is defined in terms of set integrals. Several principled solutions to performing feature map estimation are presented next, with the main focus of attention being on the PHD filter in Section 3.3.4, which is used widely throughout this book. The following estimators are Bayes optimal given the definition of an appropriate Bayes risk as just described.

3.3.2 Marginal Multi-Object Estimator

The Marginal Multi-Object (MaM) estimator is defined via a two-step estimation procedure. The number of features is first estimated using a maximum a posterior (MAP) estimator on the posterior cardinality distribution, ρ,

$$\hat{\mathfrak{m}}_k = \arg\sup_m \rho_k(|\mathcal{M}_k| = m). \tag{3.5}$$

Second, the individual feature states are estimated by searching over all maps with cardinality $\hat{\mathfrak{m}}_k$, using a MAP criteria,

$$\widehat{\mathcal{M}_k}^{MaM} = \arg \sup_{\mathcal{M}:|\mathcal{M}_k|=\hat{\mathfrak{m}}_k} p_k(\mathcal{M}). \tag{3.6}$$

It has been shown that the MaM estimator is Bayes optimal, however convergence results are not currently known.

3.3.3 Joint Multi-Object Estimator

In contrast to the MaM estimator, which first estimates the number of features and restricts its feature state estimation process to maps with only that number of features, the Joint Multi-Object (JoM) estimator executes its feature state estimation process on maps of all possible feature number.

The JoM estimator is defined as

$$\widehat{\mathcal{M}}_{k,s}^{JoM} = \arg\sup_{\mathcal{M}_k} \left(p_k(\mathcal{M}_k) \frac{s^{|\mathcal{M}_k|}}{|\mathcal{M}_k|!} \right), \tag{3.7}$$

where s is a constant with units of volume in the feature space, arg sup denotes the argument of the supremum, and $|\mathcal{M}_k|$ denotes the cardinality of $\mathcal{M}_k$. Notice that the fundamental difference between this estimator and the MAP estimator of equations 3.2 and 3.3 is that the factor $s^{|\mathcal{M}_k|}/|\mathcal{M}_k|!$ allows target based attributes of differing dimensions (e.g. spatial and non-spatial) to be "compared" in a principled manner.

First, to consider all possible sizes $\mathfrak{m}$ of the feature map for each $\mathfrak{m} \geq 0$, determine the MAP estimate,

$$\widehat{\mathcal{M}}^{(\mathfrak{m})} = \arg \sup_{\mathcal{M}:|\mathcal{M}|=\mathfrak{m}} p_k(\mathcal{M}|\mathcal{Z}_{0:k}). \tag{3.8}$$

Second, set

$$\widehat{\mathcal{M}}_s^{JoM} = \widehat{\mathcal{M}}^{(\hat{\mathfrak{m}})} \quad \text{where} \quad \hat{\mathfrak{m}} = \arg\sup_{\mathfrak{m}} p_k(\widehat{\mathcal{M}}^{(\mathfrak{m})}|\mathcal{Z}_{0:k}) \frac{s^{\mathfrak{m}}}{\mathfrak{m}!}. \tag{3.9}$$

It has been shown that the JoM estimator is Bayes optimal and is statistically consistent i.e. the feature map error distance (to be discussed in Section 4.3), between the optimal estimate and the true map, tends to zero as data accumulates [23], [3], [27]. Hence,

- "The JoM estimator determines the number $\hat{\mathfrak{m}}$ and the locations $\widehat{\mathcal{M}}$ of features optimally and simultaneously without resorting to optimal data association." [3].

Additionally, the value of s in equation 3.9 should be made equal to the desired accuracy for the state estimate. The smaller s is, the greater the accuracy of the estimate, but the rate of convergence of the estimator will be compromised. Because of this, while JoM is a theoretically attractive estimator, it is computationally expensive.

3.3.4 The Probability Hypothesis Density (PHD) Estimator

A simple approach to set-based estimation, is to exploit the physical intuition of the first moment of an RFS, known as its PHD or *intensity function*. This corresponds to the multi-feature equivalent of an expected value – the expectation of an RFS.

This section starts by giving an explanation of what the PHD is, and how it should be statistically interpreted in Section 3.3.4.1. This is followed by two intuitive derivations of the PHD in Sections 3.3.4.2 and 3.3.4.3.

3.3.4.1 Interpretation of the PHD

The intensity function at a point, gives the *density* of the expected number of features occurring at that point and therefore the mass (integral of the density over the volume of the space) of the PHD gives the expected number of features. The peaks of the intensity function indicate locations with relatively high concentration of expected number of features, in other words locations with high probability of feature existence. To provide an intuitive interpretation, consider a simple 1D example of two targets located at $x = 1$ and $x = 4$ each with spatial variance $\sigma^2 = 1$ taken from page 569, [3]. A corresponding Gaussian mixture representation of the PHD for this problem is:

$$\text{PHD}(x) = \frac{1}{\sqrt{2\pi}\sigma}\left[\exp\left(-\frac{(x-1)^2}{2\sigma^2}\right) + \exp\left(-\frac{(x-4)^2}{2\sigma^2}\right)\right]. \tag{3.10}$$

PHD(x) versus x is plotted in figure 3.1. Note that the maxima of PHD(x)

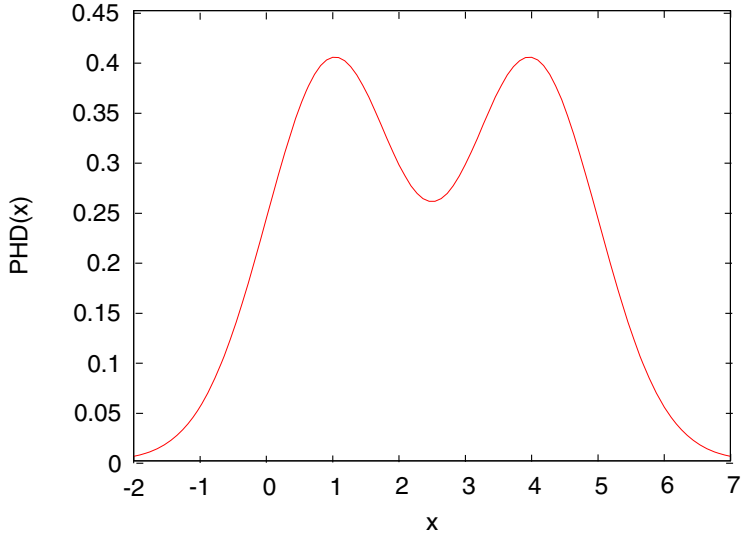


Fig. 3.1 A PHD for a 1D, 2 target problem of equation 3.10

occur near the target locations $(x = 1,\ 4)$). The integral of PHD(x) is $\mathfrak{m}$ where

$$\mathfrak{m} = \int \text{PHD}(x)dx = \int \mathcal{N}(1, \sigma^2)dx + \int \mathcal{N}(4, \sigma^2)dx \tag{3.11}$$
$$= \quad 1 \quad + \quad 1 \quad = 2$$

i.e. $\mathfrak{m}$ equals the actual number of targets. Here we note that a PHD is not a PDF, since its integral over the space of its variable is not, in general, unity.

For a 2D, robotic feature based map, graphical depictions of posterior PHDs after two consecutive measurements, approximated by Gaussian mixtures, are shown in figures 3.2 and 3.3. In each figure the intensity function

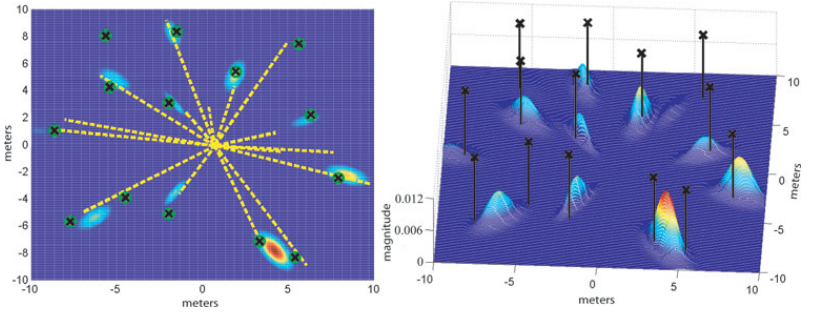


Fig. 3.2 A sample map PHD at time $k-1$, with the true map represented by black crosses. The measurement at $k-1$ is represented by the yellow dashed lines. The peaks of the PHD represent locations with the highest concentration of expected number of features. The local PHD mass in the region of most features is 1, indicating the presence of 1 feature. The local mass close to some unresolved features (for instance at (5,-8)) is closer to 2, demonstrating the unique ability of the PHD function to jointly capture the number of features.

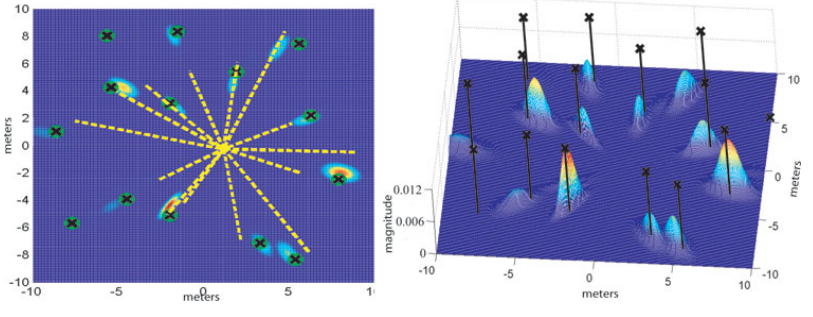


Fig. 3.3 A sample map PHD and measurement at time k. Note that the features at (5,-8) are resolved due to well separated measurements, while at (-3,-4), a lone false alarm close to the feature measurement contributes to the local PHD mass. At (-5,-4) a small likelihood over all measurements, coupled with a moderate $c_k(z|X_k)$ results in a reduced local mass.

is plotted as a function of the spatial coordinates. Since the integral of the intensity function (or PHD) is, by definition, the estimated number of features in the map, the mass (or integral) of each Gaussian can be interpreted as

the number of features it represents. In the case of closely lying features (and large measurement noise), the PHD approach may not be able to resolve the features, as demonstrated for the right hand feature of Figure 3.2 at approximate coordinates (5, -8). However the PHD will represent the spatial density of L features by a singular Gaussian with a corresponding mass of L, which may improve the feature number estimate. This is only theoretically possible using the RFS framework. A graphical example for $L = 2$ is illustrated in Figure 3.2, which is then resolved through measurement updates into individual Gaussian components for each feature of mass $L \approx 1$, as shown in Figure 3.3 (the two peaks at approximate coordinates (5, -8)).

The PHD estimator has recently been proven to be Bayes optimal [29] and has been proven to be powerful and effective in multi-target tracking [3].

3.3.4.2 The PHD as the Limit of an Occupancy Probability

Intuitively, the PHD can be derived as a limiting case of the occupancy probability used in grid based methods. Following [30], consider a grid system and let m_i denote the centre and $B(m_i)$ the region defined by the boundaries of the ith grid cell. Let $P^{(occ)}(B(m_i))$ denote the occupancy probability and $\lambda(B(m_i))$ the area of the ith grid cell. Assume that the grid is sufficiently fine so that each grid cell contains at most one feature, then the expected number of features over the region $S_J = \bigcup_{i \in J} B(m_i)$ is given by,

$$\begin{aligned} \mathbb{E}\left[|\mathcal{M} \cap S_J|\right] &= \sum_{i \in J} P^{(occ)}(B(m_i)) \\ &= \sum_{i \in J} v(m_i)\lambda(B(m_i)). \end{aligned} \tag{3.12}$$

where $v(m_i) = \frac{P^{(occ)}(B(m_i))}{\lambda(B(m_i))}$. Intuitively any region S_J can be represented by $\bigcup_{i \in J} B(m_i)$, for some J. As the grid cell area tends to zero (or for an infinitesimally small cell), $B(m_i) \to dm$. The sum then becomes an integral and the expected number of features in S becomes,

$$\mathbb{E}\left[|\mathcal{M} \cap S|\right] = \int_S v(m)dm. \tag{3.13}$$

$v(m)$ defines the PHD and it can be interpreted as the occupancy probability density at the point m. The (coordinates of the) peaks of the intensity are points (in the space of features) with the highest local concentration of expected number of features and hence can be used to generate optimal estimates for the elements of $\mathcal{M}$. The integral of the PHD gives the expected number of features and the peaks of the PHD function can be used as estimates of the positions of the features.

3.3.4.3 The PHD as the Density of the Expectation of a Point Process

An alternative derivation of the PHD now follows. An analogous notion to the 'expectation' of an RFS can be borrowed from point process theory. This construct treats the random set as a random counting measure or a point process (a random finite set and a simple finite point process are equivalent [31]).

Let $p(\mathcal{M})$ be the multi-feature probability distribution of the map RFS $\mathcal{M}$. A somewhat naive interpretation of its expected value $\widehat{\mathcal{M}}$ would then be

$$\widehat{\mathcal{M}}^{\text{naive}} \triangleq \int \mathcal{M}_i p(\mathcal{M}) \delta\mathcal{M}_i. \tag{3.14}$$

where $\mathcal{M}_i$ represents the ith subset of $\mathcal{M}$. Since the addition of finite subsets of $\mathcal{M}$ is undefined, the above integral is also undefined. It can be solved by defining a transformation which maps finite subsets $\mathcal{M}_i$ into vectors M_i in some vector space. This transformation must maintain the set theoretic structure by transforming unions into sums - i.e. $\mathcal{M}_i \cup \mathcal{M}_j = M_i + M_j$, if $\mathcal{M}_i \cap \mathcal{M}_j = \emptyset$. In this case, an expected value can be defined in terms of the "equivalent" vectors

$$\widehat{\mathcal{M}} \triangleq \mathbb{E}[M] = \int M_i p(\mathcal{M}) dM_i \tag{3.15}$$

The point process literature [32] uses a transformation $M_i = \delta_{\mathcal{M}_i}$ where

$$\begin{aligned} \delta_{\mathcal{M}_i} &\triangleq 0 && \text{if } \mathcal{M}_i = \emptyset \\ \delta_{\mathcal{M}_i} &\triangleq \sum_{m \in \mathcal{M}_i} \delta(x - m) && \text{otherwise.} \end{aligned} \tag{3.16}$$

where x is the vector space of the features and $\delta(x - m)$ is the Dirac delta density concentrated at each feature m within the random finite subset $\mathcal{M}_i$. Taking the expectation of equation 3.16 gives

$$v(m) \triangleq \mathbb{E}[\delta_{\mathcal{M}}] = \int \delta_{\mathcal{M}_i} p(\mathcal{M}) \delta\mathcal{M}_i \tag{3.17}$$

which is the multi-feature equivalent of the expected value. This is called the probability hypothesis density (PHD), also known as the intensity density, or intensity function $v(m)$ of $\mathcal{M}$.

Note that while $v(m)$ is a density, it is *not* a PDF, since it may not necessarily integrate to 1. This is clear, as the integral of $v(m)$ over any region S gives the expected number of features in that region - i.e.

$$\int_S v(m) dm = \mathbb{E}\left[|\mathcal{M} \cap S|\right] \tag{3.18}$$

Hence, the integral of the non-negative intensity function $v(m)$ over $\mathcal{M}$ gives the expected number of features in the map.

Note that we have arrived at the same result as equation 3.13, in which the PHD was considered to be the limit of an occupancy probability.

3.3.4.4 Recovering the Map from the PHD Intensity Function

Since $v(m)$ is a density, it can be readily approximated by standard Sequential Monte Carlo (SMC) or Gaussian Mixture methods as described later in Chapter 4. The PHD filter recursion therefore propagates the intensity function $v(m)$ of the RFS state and uses the RFS measurement in the update stage. Since the intensity is the first order statistic of a random finite set, the PHD filter is analogous to the constant gain Kalman filter, which propagates the first order statistic (the mean) of the vector-based state. However, the *intensity*, $v(m)$, can be used to estimate both the number of features in the map, and their corresponding states (along with the uncertainty in the state estimates) [2].

If the RFS, $\mathcal{M}_k$, is Poisson, i.e. the number of points is Poisson distributed and the points themselves are independently and identically distributed (IID), then the probability density of $\mathcal{M}_k$ can be constructed exactly from the PHD.

$$p_k(\mathcal{M}_k) = \frac{\prod\limits_{m \in \mathcal{M}_k} v_k(m)}{\exp(\int v_k(m) dm)}. \tag{3.19}$$

where $v_k(m)$ is the map intensity function at time k and $\mathcal{M}_k$ is the RFS map which has passed through the field of view (FoV) of the vehicle's on board sensor(s) up to and including time k. In this sense, the PHD can be thought of as a 1st moment approximation of the probability density of an RFS.

Under these approximations, it has been shown [2] that, similar to standard recursive estimators, the PHD recursion has a *predictor-corrector* form.

3.4 The PHD Filter

As defined in Section 3.2.1, $\mathcal{M}$ is the RFS representing the entire unknown map. Let $\mathcal{M}_{k-1}$ be the RFS representing the explored map, with trajectory $X_{0:k-1} = [X_0, X_1, \ldots, X_{k-1}]$ at time $k-1$, i.e.

$$\mathcal{M}_{k-1} = \mathcal{M} \cap FoV(X_{0:k-1}). \tag{3.20}$$

Note that $FoV(X_{0:k-1}) = FoV(X_0) \cup FoV(X_1) \cup \cdots \cup FoV(X_{k-1})$. $\mathcal{M}_{k-1}$ therefore represents the set on the space of features which intersects with the union of individual FoVs, over the vehicle trajectory up to and including time $k-1$. Given this representation, $\mathcal{M}_{k-1}$ evolves in time according to,

$$\mathcal{M}_k = \mathcal{M}_{k-1} \cup \left(FoV(X_k) \cap \bar{\mathcal{M}}_{k-1} \right) \tag{3.21}$$

where $\bar{\mathcal{M}}_{k-1} = \mathcal{M} - \mathcal{M}_{k-1}$ represents the unexplored map (note the difference operator used here is the set difference, sometimes referred to as $\mathcal{M} \backslash \mathcal{M}_{k-1}$ or the relative complement of $\mathcal{M}_{k-1}$ in $\mathcal{M}$), i.e the set of features that are not in $\mathcal{M}_{k-1}$. Let the newly explored features which have entered the FoV be modelled by the independent RFS, $\mathcal{B}_k(X_k)$. In this case, the RFS map transition density is given by,

$$f_{\mathcal{M}}(\mathcal{M}_k|\mathcal{M}_{k-1}, X_k) = \sum_{\mathcal{W} \subseteq \mathcal{M}_k} f_{\mathcal{M}}(\mathcal{W}|\mathcal{M}_{k-1}) f_{\mathcal{B}}(\mathcal{M}_k - \mathcal{W}|X_k) \tag{3.22}$$

where $f_{\mathcal{M}}(\mathcal{W}|\mathcal{M}_{k-1})$ is the transition density of the set of features that are in $FoV(X_{0:k-1})$ at time $k-1$ to time k, and $f_{\mathcal{B}}(\mathcal{M}_k - \mathcal{W})|X_k)$ is the density of the RFS, $\mathcal{B}(X_k)$, of the new features that pass within the field of view at time k. To define the PHD filter in a form general enough to be applied to FBRM and SLAM, we now define a state variable Γ_k which corresponds to the state of interest to be estimated. In the case of FBRM, Γ_k would be replaced by "$m|X_k$" i.e. the feature at m, given the vehicle location X_k. This implementation of the PHD filter will be the subject of Chapter 4. In the case of SLAM a "brute force" approach is implemented in Chapter 5 which replaces Γ_k with each feature augmented with a hypothesised vehicle trajectory. Its implementation is shown to demonstrate a viable, and theoretically simple, SLAM implementation. A more elegant, Rao-Blackwellised implementation of the PHD filter is implemented in Chapter 6 which propagates N conditionally independent PHDs, based on each of the N hypothesised trajectories, represented as particles. In this case, Γ_k is effectively replaced by "$m|X_{0:k}$" i.e. the feature at m conditioned on the vehicle trajectory $X_{0:k}$. This will demonstrate a more computationally efficient SLAM implementation, which allows the Bayes optimal, expected trajectory and expected map to be evaluated.

In terms of the general state variable Γ_k the prediction of the map intensity function $v_{k|k-1}(\Gamma_k)$, is given by

$$v_{k|k-1}(\Gamma_k) = v_{k-1|k-1}(\Gamma_{k-1}) + b(\Gamma_k) \tag{3.23}$$

where $b(\Gamma_k)$ is the PHD of the new feature RFS, $\mathcal{B}(X_k)$. Note that $v_{k-1|k-1}()$ corresponds to the estimate of the intensity function at time $k-1$, given all observations up to, and including, time $k-1$. For ease of notation however, this will be referred to as $v_{k-1}()$ in future instances.

The PHD corrector equation is then [2],

$$v_k(\Gamma_k) = v_{k|k-1}(\Gamma_k)\Bigg[1 - P_D(\Gamma_k) + \sum_{z\in\mathcal{Z}_k} \frac{P_D(\Gamma_k)g_k(z|\Gamma_k)}{c_k(z) + \int P_D(\xi_k)g_k(z|\xi_k)v_{k|k-1}(\xi_k)d\xi_k}\Bigg] \quad (3.24)$$

where $v_{k|k-1}(\Gamma_k)$ is the predicted intensity function from equation 3.23, ξ is a subset of $\mathcal{M}_k$ and,

$P_D(\Gamma_k)$ = the probability of detecting a feature at m, from vehicle location X_k (encapsulated in Γ_k),
$g_k(z|\Gamma_k)$ = the measurement model of the sensor at time k,
$c_k(z)$ = intensity of the clutter RFS $\mathcal{C}_k(X_k)$ (in equation 2.10) at time k.

3.4.1 Intuitive Interpretation of the PHD Filter

An intuitive interpretation of the PHD filter equations 3.23 and 3.24 is given in Chapter 16 of [3]. The predictor equation 3.23 comprises the addition of the previous PHD correction and the PHD of the set of features hypothesised to enter the sensor's FoV. The corrector equation 3.24, can be more clearly interpreted in its integrated form since, by definition

$$\int v_k(\Gamma_k)d\Gamma_k = \mathfrak{m}_k \quad (3.25)$$

where $\mathfrak{m}_k$ is the number of estimated features at time k. To simplify the interpretation further, a constant (state independent) probability of detection is assumed in this section - i.e.

$$P_D(\Gamma_k) = P_D. \quad (3.26)$$

Therefore, from equation 3.24,

$$\begin{aligned}\mathfrak{m}_k &= \int v_k(\Gamma_k)d\Gamma_k \\ &= (1 - P_D)\mathfrak{m}_{k|k-1} + \\ &\quad P_D \sum_{z\in\mathcal{Z}_k} \frac{\int g_k(z|\Gamma_k)v_{k|k-1}(\Gamma_k)d\Gamma_k}{c_k(z) + P_D\int g_k(z|\xi_k)v_{k|k-1}(\xi_k)d\xi_k}\end{aligned} \quad (3.27)$$

Notice that the integrals in the numerator and denominator of the final term within the summation of equation 3.27 are identical and to simplify the equation we introduce

$$D_{k|k-1}(g_k, v_{k|k-1}) \triangleq \int g_k(z|\Gamma_k) v_{k|k-1}(\Gamma_k) d\Gamma_k = \int g_k(z|\xi_k) v_{k|k-1}(\xi_k) d\xi_k \tag{3.28}$$

where g_k abbreviates $g_k(z|\Gamma_k)$ and $v_{k|k-1}$ abbreviates $v_{k|k-1}(\Gamma_k)$. Therefore the integral of the PHD corrector equation 3.24, with constant P_D, can be written as the feature number corrector equation

$$\mathfrak{m}_k = (1 - P_D)\mathfrak{m}_{k|k-1} + P_D \sum_{z \in \mathcal{Z}_k} \frac{D_{k|k-1}(g_k, v_{k|k-1})}{c_k(z) + P_D D_{k|k-1}(g_k, v_{k|k-1})} \tag{3.29}$$

Equation 3.29 is useful for intuitively interpreting the PHD corrector equation, and is governed by the following effects:

1. *Probability of detection* P_D. If the map feature at m is not in the FoV of the sensor, it could not have been observed, thus $P_D = 0$. Therefore, from equation 3.29

$$\mathfrak{m}_k = (1 - 0)\mathfrak{m}_{k|k-1} + 0 = \mathfrak{m}_{k|k-1} \tag{3.30}$$

i.e. the updated number of features simply equals the predicted number, since no new information is available. Similarly from from equation 3.24,

$$v_k(\Gamma_k) = v_{k|k-1}(\Gamma_k)[1 - 0 + 0] = v_{k|k-1}(\Gamma_k) \tag{3.31}$$

i.e. the updated PHD will simply equal the predicted value.
On the other hand, if m is within the sensor FoV and if $P_D \approx 1$, the summation in equation 3.24, tends to dominate the PHD update and

$$v_k(\Gamma_k) \quad \approx \quad v_{k|k-1}(\Gamma_k) \left[\sum_{z \in \mathcal{Z}_k} \frac{g_k(z|\Gamma_k)}{c_k(z) + \int g_k(z|\xi_k) v_{k|k-1}(\xi_k) d\xi_k} \right] \tag{3.32}$$

Then the predicted PHD is modified by the sum of terms dependent on the measurement likelihood and clutter PHD.

2. *False alarms* $c_k(z)$. A particular feature observation could have originated from a feature or as a false alarm. Assume that the number of false alarms λ (represented by its intensity $c_k(z)$) is large and uniformly distributed in some region R. If the observed feature is in R, the term within the summation of equation 3.29 becomes

$$\begin{aligned} & P_D \frac{D_{k|k-1}(g_k, v_{k|k-1})}{c_k(z) + P_D D_{k|k-1}(g_k, v_{k|k-1})} \\ & = P_D \frac{D_{k|k-1}(g_k, v_{k|k-1})}{\frac{\lambda}{|R|} + P_D D_{k|k-1}(g_k, v_{k|k-1})} \approx P_D \frac{|R|}{\lambda} D_{k|k-1}(g_k, v_{k|k-1}) \approx 0 \end{aligned} \tag{3.33}$$

since λ is so large that it dominates the denominator. Therefore, if an observation originates from R, it is likely to be a false alarm, and it contributes almost nothing to the total posterior feature count, as it should.

On the other hand if a measurement originates from a region other than R, with low clutter, then it is unlikely to be a false alarm. This means that $c_k(z) \approx 0$ and

$$P_D \frac{D_{k|k-1}(g_k, v_{k|k-1})}{c_k(z) + P_D D_{k|k-1}(g_k, v_{k|k-1})} \approx P_D \frac{D_{k|k-1}(g_k, v_{k|k-1})}{0 + P_D D_{k|k-1}(g_k, v_{k|k-1})} = 1 \tag{3.34}$$

so that the measurement now contributes one feature to the total feature number. In general, if the number of false alarms, governed by the clutter PHD $c_k(z|X_k)$, is high, this increases the denominator of the summation, thus lowering the effect of the sensor update, as it should.

3. *Prior information* $P_D g_k(z|\Gamma_k)$. Assume that the sensor model is good, and $P_D g_k(z|\Gamma_k)$ is large for a particular state Γ_k which produces z. If z is consistent with prior information (the observation model), $P_D D_{k|k-1}(g_k, v_{k|k-1})$ will tend to dominate the denominator of the summation in equation 3.29, and the term corresponding to that feature in the summation will become

$$P_D \frac{D_{k|k-1}(g_k, v_{k|k-1})}{c_k(z) + P_D D_{k|k-1}(g_k, v_{k|k-1})} \approx 1 \tag{3.35}$$

Hence, a feature which is consistent with the observation model tends to contribute one feature to the total feature count.

Conversely if the observation z is inconsistent with prior information (is unlikely according to the sensor model), then the product $P_D g_k(z|\Gamma_k)$ will be small, and its corresponding term in the summation in equation 3.29 will tend to be ignored.

Equations 3.23 and 3.24 which comprise the PHD filter have been shown to be Bayes optimal, assuming that the RFS observation and map statistics can be represented by their first moments only [3].

3.5 Summary

This chapter addressed the issues of estimation with RFSs. Initially, the traditional MAP and EAP estimators were applied to a simple, single feature problem with both feature existence and spatial uncertainty. It was demonstrated that such estimators are not suitable in such applications, and new multi-feature estimators were defined, which minimised the Bayes risk in feature map estimation.

The main focus of attention of the chapter was on the PHD filter. An RFS map density can be represented by its first moment, the intensity function. Brief derivations for the PHD estimator (intensity function) were shown based on the PHD as the limit of an occupancy probability, and the density of the expectation of a point process.

The PHD recursion is far more numerically tractable than propagating the RFS map densities of equation 2.15. In addition, the recursion can be readily extended to incorporate multiple sensors/vehicles by sequentially updating the map PHD with the measurement from each robot.

Chapter 4
An RFS Theoretic for Bayesian Feature-Based Robotic Mapping

4.1 Introduction

Estimating a FB map requires the joint propagation of the FB map density encapsulating uncertainty in feature number and location. This chapter addresses the joint propagation of the FB map density and leads to an optimal map estimate in the presence of unknown map size, spurious measurements, feature detection and data association uncertainty. The proposed framework further allows for the joint treatment of error in feature number and location estimates. As a proof of concept, the first-order moment recursion, the PHD filter, is implemented using both simulated and real experimental data. The feasibility of the proposed framework is demonstrated, particularly in situations of high clutter density and large data association ambiguity. This chapter establishes new tools for a more generalised representation of the FB map, which is a fundamental component of the more challenging SLAM problem, to follow in Part II.

In this chapter, FB map only estimation is addressed, i.e. the vehicle trajectory is assumed known. The chapter presents new insights to motivate an RFS approach, under which Bayes optimality is examined. Further, the concept of FB map estimation error (in the general case of an unknown number of features) is addressed through a mathematically consistent[1] error metric.

The chapter is organised as follows. Section 4.2 reiterates the natural representation of the feature map as a *set* of features, and formulates the Bayesian estimation problem for estimating the feature locations and number in a joint manner. Current FB error evaluation methods are discussed and a mathematically consistent error metric introduced in Section 4.3. Section 4.4 explicitly relates the PHD filter concepts of section 3.4 in Chapter 3 to the FBRM problem. As a proof-of-concept, the first-order moment PHD recursion for

[1] A mathematically consistent metric is one that is defined on the state-space of interest (the space of all possible feature maps) and satisfies the necessary metric axioms. See Section 4.3 and [23] for further details.

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 45–76.
springerlink.com

FB maps is further outlined in Section 4.4, followed by contrasting filter implementations in Section 4.5. Simulation and real experimental results, using the new framework, are demonstrated in Section 4.6 with comparisons to classical approaches. Finally Section 4.8 offers some bibliographical comments on related robotic mapping work and summarises the latest developments in a related field of multi-object filtering.

4.2 The Feature-Based Map Estimation Framework

As introduced in Chapters 2 and 3, in contrast to the classical vector representation, a RFS map state, $\mathcal{M}_k$, can jointly encapsulate feature number and location uncertainty. Equivalently, a RFS measurement, $\mathcal{Z}_k$, can model uncertainty in measurement number and value.

Given the current vehicle state, X_k, and the feature RFS map $\mathcal{M}_k$, as mentioned in Chapter 2 (equation 2.10), and repeated here for convenience, the measurement consists of a set union,

$$\mathcal{Z}_k = \bigcup_{m \in \mathcal{M}_k} \mathcal{D}_k(m, X_k) \cup \mathcal{C}_k(X_k) \tag{4.1}$$

where $\mathcal{D}_k(m, X_k)$ is the RFS of a measurement generated by a feature at m and $\mathcal{C}_k(X_k)$ is the RFS of the spurious measurements at time k, which may be dependent on the vehicle pose, X_k. Therefore $\mathcal{Z}_k$ consists of a random number of measurements. In this chapter, and throughout the book, the measurements take the form of range and bearing. It should be noted however that the RFS framework can be readily extended to other measurement types. Note that the number of detected measurements may differ from the number of features in $\mathcal{M}_k$ due to detection uncertainty, occlusion and spurious measurements. It is also assumed that $\mathcal{D}_k(m, X_k)$, and $\mathcal{C}_k(X_k)$ are independent RFSs.

The RFS of a measurement generated by a feature at m is assumed to be a Bernoulli RFS[2] given by, $\mathcal{D}_k(m, X_k) = \emptyset$ with probability $1 - P_D(m, X_k)$ and $\mathcal{D}_k(m, X_k) = \{z\}$ with probability density $P_D(m, X_k) g_k(z|m, X_k)$. For a given robot pose X_k, $P_D(m, X_k)$ is the probability of the sensor detecting a feature at m and, when conditioned on $g_k(z|m, X_k)$, is the likelihood that a feature at m generates the measurement z. The concept of detection probability is important to the measurement model, and indeed multiple models such as uniform or exponential mixture [33] can be accommodated in to the framework. In particular, state-dependant detection probabilities are most useful in a mobile robotics framework, since the ability of the sensor (or feature extraction algorithm) to detect a given object can be highly influenced

[2] The Bernoulli RFS is empty with a probability $1 - \alpha$ and is distributed according to a density p with probability α. The example in Section 3.2.1 was that of a Bernoulli RFS.

by its relative location. For instance, occlusions would effectively result in a P_D of zero, and it is also possible that the P_D would taper off with increasing distance from the sensor. While such approaches can be readily incorporated into the measurement model and subsequent filter [33], [34], throughout the context of this book a binary approach is taken. That is $P_D(m, X_k) = P_D$ when the feature is within the sensor field of view and $P_D(m, X_k) = 0$ otherwise.

The measurement generated by the sensor at time k is modelled by the RFS of equation 4.1. The RFS $\mathcal{Z}_k$ therefore encapsulates all measurement uncertainty such as measurement noise, sensor field of view (i.e. state-dependent probability of detection) and spurious measurements. The probability density that the sensor produces the measurement $\mathcal{Z}_k$, given the vehicle state X_k and map $\mathcal{M}_k$ at time k, is then given by the convolution [2],

$$g_k(\mathcal{Z}_k|X_k, \mathcal{M}_k) = \sum_{\mathcal{W} \subseteq \mathcal{Z}_k} g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k) g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W}). \tag{4.2}$$

Here, $g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k)$ denotes the density of the RFS $\mathcal{D}_k$ of measurements generated by features in $\mathcal{M}_k$, given the state of the vehicle, and $g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W})$ denotes the density of the RFS $\mathcal{C}_k$ of spurious measurements where, as in Section 3.4, the difference operation used in equation 4.2 is the set difference. $g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k)$ describes the likelihood of receiving a measurement from the elements of the map, and incorporates detection uncertainty and measurement noise. $g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W})$ models the spurious measurements of the sensor and is typically *a priori* assigned [18], [14].

If $f_{\mathcal{M}}(\mathcal{M}_k|\mathcal{M}_{k-1}, X_{k-1})$ then represents the RFS feature map state transition density (which typically incorporates an increasing number of map features as the vehicle moves, as shown in Section 3.4), and the RFS measurement is as in equation 4.1, the generalised Bayesian RFS FBRM recursion can be written in a form similar to equations 2.14 and 2.15 in Chapter 2. The difference here is that only the RFS map density is to be estimated:

$$p_{k|k-1}(\mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_k) = \int f_{\mathcal{M}}(\mathcal{M}_k|\mathcal{M}_{k-1}, X_k) \times p_{k-1}(\mathcal{M}_{k-1}|\mathcal{Z}_{0:k-1}, X_{0:k-1}) \delta\mathcal{M}_{k-1} \tag{4.3}$$

$$p_k(\mathcal{M}_k|\mathcal{Z}_{0:k}, X_{0:k}) = \frac{g_k(\mathcal{Z}_k|X_k, \mathcal{M}_k) p_{k|k-1}(\mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_k)}{\int g_k(\mathcal{Z}_k|X_k, \mathcal{M}_k) p_{k|k-1}(\mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_k) \delta\mathcal{M}_k} \tag{4.4}$$

where δ once again implies a set integral. Note again here, that for ease of notation, p_{k-1} in equation 4.3 actually corresponds to $p_{k-1|k-1}$ and p_k in equation 4.4 actually corresponds to $p_{k|k}$. This simplified notation will be adopted throughout this book. Integration over the map, requires integration over all possible feature maps (all possible locations *and* numbers of features).

By adopting an RFS map representation, integrating over the map becomes a set integral, which can be defined as [3],

$$\int p_{k|k-1}(\mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_k)\delta\mathcal{M}_k = \sum_{\mathfrak{m}_k=0}^{\infty} \frac{1}{\mathfrak{m}_k!} \int p_{k|k-1}(\{m^1, \ldots, m^{\mathfrak{m}_k}\}|\mathcal{Z}_{0:k-1}, X_k) dm^1 dm^2 \ldots dm^{\mathfrak{m}_k}. \quad (4.5)$$

This feature map recursion therefore encapsulates the inherent feature number uncertainty of the map introduced by detection uncertainty, spurious measurements and vehicle manoeuvres, as well as the feature location uncertainty introduced by measurement noise. Features are not rigidly placed in a map vector (as is the case in previous approaches), nor are measurements simply a direct function of the map state, due to the explicit modelling of clutter. Therefore, as introduced in Chapter 2, contrary to previous vector represented approaches, no explicit measurement to feature assignment (the data association problem) is required.

4.3 FB Map Estimation Error

This book focuses on arguably the most popular and widely studied mobile robotics problem, localisation and mapping in a feature-based map. While in practice, the form a map takes would mostly depend on its application (path planning for instance would likely require higher resolution Grid Based (GB) maps), feature based maps form the basis of numerous mobile robotic frameworks. Whether for the given application, the feature represents a fixed static object for localisation, a target for defence or an injured human for search & rescue, the problem can in general be formulated as one of estimating an unknown number of objects at unknown locations. As such, a measure of the accuracy of any given mapping filter is of critical importance. To the authors' knowledge, there is currently no work which presents a well defined mathematical distance to gauge feature-based mapping error.

The primary difficulty in determining map estimation error is due to differences between the estimated and true number of features, and the need to satisfy the 4 metric axioms [23][3]. Error metrics for fixed dimension problems, such as a sum of the squared error can be readily obtained from grid-based robotic frameworks [7] or metrics based on the Hausdorff distance [35] from the template matching literature. Such metrics however, cannot encapsulate

[3] The 4 metric axioms can be defined as follows. Let $\mathcal{X}$ be an arbitrary, non-empty set. Then the function d is a metric iff: 1) $d(x, y) >= 0$, for all $x, y \in \mathcal{X}$; 2) $d(x, y) = 0$ iff $x = y$, $x \in \mathcal{X}$ (identity axiom); 3) $d(x, y) = d(y, x)$, for all $x, y \in \mathcal{X}$ (symmetry axiom); 4) $d(x, y) \leq d(x, z) + d(y, z)$, for all $x, y, z \in \mathcal{X}$ (triangle inequality axiom).

cardinality estimation error and are thus not suited to the evaluation of FB map estimates.

A finite set representation of the map naturally allows for a metric to gauge map estimation error [23]. While there are several metrics for finite-set-valued estimation [23], the feature map estimation error metric used here (which jointly considers errors in feature location estimate, and feature number estimate) is based on a 2^{nd}-order Wasserstein construction. If $|\mathcal{M}| > |\widehat{\mathcal{M}}|$, it is given by,

$$\bar{d}^{(c)}(\widehat{\mathcal{M}}, \mathcal{M}) := \left(\frac{1}{|\mathcal{M}|} \left(\min_{j \in \{1,\ldots,|\mathcal{M}|\}} \sum_{i=1}^{|\widehat{\mathcal{M}}|} d^{(c)}(\widehat{m}^i, m^j)^2 + c^2(|\mathcal{M}| - |\widehat{\mathcal{M}}|) \right) \right)^{1/2} \tag{4.6}$$

where,

$$d^{(c)}(\widehat{m}^i, m^j) = \min(c, ||\widehat{m}^i - m^j||) \tag{4.7}$$

is the minimum of the cut-off parameter, c, and the Euclidean distance between the estimated feature location, $\widehat{m}^i$ and the true feature location m^j. If $|\mathcal{M}| < |\widehat{\mathcal{M}}|$ the metric is obtained through $\bar{d}^{(c)}(\mathcal{M}, \widehat{\mathcal{M}})$. Note that the standard Euclidean distance used in the definition of $d^{(c)}(\widehat{m}^i, m^j)$ is arbitrary. To incorporate orientated features, other vector distances such as a Mahalanobis distance could be adopted where,

$$d^{(c)}(\widehat{m}^i, m^j) = \min\left(c, \left((\widehat{m}^i - m^j)^T \widehat{P}_i^{-1} (\widehat{m}^i - m^j)\right)^{1/2}\right) \tag{4.8}$$

with $\widehat{P}_i$ being the estimated covariance of the i^{th} feature. The Wasserstein construction is commonly adopted in theoretical statistics as a measure of similarity between probability distributions. It has been shown previously that for distributions of equal dimensionality, the Wasserstein distance reduces to an optimal assignment problem [23]. Minimisation of a global distance between the estimated and ground truth maps inherently considers the problem in its joint-estimation framework, as opposed to analysis of individual estimates of features assigned through nearest neighbour methods [1]. The metric of equation 4.6 explicitly accounts for the dimensionality estimate errors through the term $c^2(|\mathcal{M}| - |\widehat{\mathcal{M}}|)$, which assigns a user-defined cost, c, for each unassigned feature state. This threshold decides whether a given feature estimate is an incorrectly declared feature, or is a correctly declared feature that is poorly localised. The effects of varying the c parameter are illustrated later in Section 4.6.5. For example, given the estimated maps of Figure 2.5 in Chapter 2, with the parameter choice $c = 5$, $\bar{d}^{(5)}(\widehat{\mathcal{M}}_{left}, \mathcal{M}) = 1.48m$ whereas $\bar{d}^{(5)}(\widehat{\mathcal{M}}_{right}, \mathcal{M}) = 2.02m$, with $\bar{d}^{(5)}(\mathcal{M}, \mathcal{M}) = 0m$. With such a metric, the quality of estimated maps can be compared in a quantitative manner.

Equation 4.6 will be used to gauge the FBRM estimation performance, in the cases of known ground truth maps, in the results section. The remaining sections of this chapter are devoted to a solution to the FBRM problem using

the PHD filter. Contrasting RFS FBRM filters are outlined and derived for both static and pseudo-static maps, coupled with analysis and results using benchmark approaches adopted by the robotics community.

4.4 The PHD-FBRM Filter

The recursion of equations 4.3 and 4.4, which are the mapping only versions of the general Bayes update equations 2.14 and 2.15, constitutes a Bayesian approach to the FBRM problem. However, as is the case for the vector recursion, a direct implementation is generally computationally intractable due to multiple integrals on the space of features. This is evident from the set integration result of equation 4.5. Therefore, as an approximation, the PHD filter, introduced in Chapter 3 (Section 3.4), is used to propagate the expectation of the map as opposed to the full multi-feature posterior density.

As introduced in Chapters 2 and 3, the newly explored features which have entered the FoV of the vehicle's sensor(s) are modelled by the RFS, $\mathcal{B}_k(X_k)$ and the spurious measurements at time k, which may depend on the vehicle pose X_k, are modelled by the RFS $\mathcal{C}_k(X_k)$. Then, the state of interest in the FBRM case is the feature position m given the robot pose X_k. Hence in the general state update equations 3.23 and 3.24

$$\Gamma_k \longrightarrow m|X_k \tag{4.9}$$

and from equation 3.23 the prediction of the map intensity function $v_{k|k-1}(\Gamma_k)$ becomes $v_{k|k-1}(m|X_k)$, given the robot location (i.e. FBRM), and is given by

$$v_{k|k-1}(m|X_k) = v_{k-1}(m|X_{k-1})^4 + b_k(m|X_k) \tag{4.10}$$

where $b_k(m|X_k)$ is the PHD of the new feature RFS, $\mathcal{B}(X_k)$.

The PHD filter corrector equation 3.24, $v_k(\Gamma_k)$ then becomes $v_k(m|X_k)$, where

$$v_k(m|X_k) = v_{k|k-1}(m|X_k)\Bigg[1 - P_D(m|X_k) + \sum_{z\in\mathcal{Z}_k} \frac{P_D(m|X_k)g_k(z|m,X_k)}{c_k(z|X_k) + \int P_D(\xi|X_k)g_k(z|\xi,X_k)v_{k|k-1}(\xi|X_k)d\xi}\Bigg] \tag{4.11}$$

where $v_{k|k-1}(m|X_k)$ is the predicted intensity function and,

[4] As mentioned in chapter 3, $v_{k-1}()$ is a shortened notation, actually referring to $v_{k-1|k-1}()$ i.e. the estimated intensity function at time $k-1$, given all observations up to, and including, time $k-1$.

$$\begin{aligned} P_D(m|X_k) &= \text{the probability of detecting a feature at} \\ &\quad\ m, \text{ from vehicle location } X_k, \\ g_k(z|m, X_k) &= \text{the measurement model of the sensor at time } k, \\ c_k(z|X_k) &= \text{intensity of the clutter RFS } \mathcal{C}_k(X_k) \text{ at time } k. \end{aligned}$$

Hence the FBRM filter estimates $v_k(m|X_k)$ which, under a Gaussian mixture (GM) implementation, has the form of the PHD (intensity function), shown in figures 3.2 and 3.3. In the FBRM case, under this representation, each Gaussian is conditioned on the known vehicle state X_k.

While equation 4.11 defines the PHD measurement update, the following subsections describe differing interpretations of the map state, $\mathcal{M}_k$. One interpretation is as a static (in both feature location and number) unknown state, while the other is as a static (in location), but monotonically increasing in number as the vehicle manoeuvres.

4.4.1 Static Map State

The PHD recursion of equation 4.11 estimates the map by fusing the RFS measurement likelihood with the prior map intensity function. This represents a direct implementation of the FBRM problem, where a static map is assumed as $\mathcal{M} = \{m^1, \cdots, m^{\mathfrak{m}_k}\}$, where $\mathfrak{m}_k$ is the true number of features in the *entire* map state. The problem is to estimate the entire number of map features $\mathfrak{m}_k$, as well as their respective locations, m. Since the map, $\mathcal{M}$, does not evolve in time, the predicted multi-feature density is simply obtained by,

$$p_{k|k-1}(\mathcal{M}|\mathcal{Z}_{0:k-1}, X_k) = p_{k-1}(\mathcal{M}|\mathcal{Z}_{0:k-1}, X_{k-1}) \tag{4.12}$$

and consequently the predicted intensity required in equation 4.11 is given by,

$$v_{k|k-1}(m|X_k) = v_{k-1}(m|X_{k-1}) \tag{4.13}$$

where $v_{k-1}(m|X_{k-1})$ is the posterior intensity at time $k-1$. This "brute force" approach to the FBRM problem requires an intuitive prior on the map state (analogous to the OG mapping problem), which is then refined through successive measurement updates.

4.4.2 Pseudo-static Map State

A second contrasting interpretation of the map is to consider a map state with stationary features, but which grows indefinitely over time due to the limited sensor field of view. This pseudo-static map model concept was already introduced in Section 3.4, and is repeated here for completeness. Let

the map state represent the subset of $\mathcal{M}$, which has been observed by the on-board sensor, i.e.

$$\mathcal{M}_{k-1} = \mathcal{M} \cap FoV(X_{0:k-1}) \tag{4.14}$$

with $FoV(X_{0:k-1}) = FoV(X_0) \cup FoV(X_1) \cup \cdots \cup FoV(X_{k-1})$ where FoV refers to the field of view of the sensor. $\mathcal{M}_{k-1}$ therefore represents the set on the space of features which intersects the union of the individual FoVs over the vehicle trajectory up to and including time $k-1$. Given this representation, although the features in the map state are assumed static, the map itself evolves in time according to,

$$\mathcal{M}_k = \mathcal{M}_{k-1} \cup \left(FoV(X_{0:k}) \cap \bar{\mathcal{M}}_{k-1} \right) \tag{4.15}$$

where, $\bar{\mathcal{M}}_{k-1}$ represents the unexplored map[5]. This predictive map model was introduced in Chapter 3, Section 3.4. This models the map state as static in value, but monotonically increasing in cardinality with time, due to new features in $\mathcal{M}$ entering the field of view of the sensor as the vehicle traverses the terrain. Therefore, $\mathcal{M}_k = \{m^1, \cdots, m^{\mathfrak{m}_k}\}$, where $\mathfrak{m}_k$ is the time varying cardinality of $\mathcal{M}_k$ and $\mathfrak{m}_k \leq \mathfrak{m}$, with $\mathfrak{m}$ being the total number of features in the entire map, $\mathcal{M}$.

As introduced in section 3.4, the new features which have entered the FoV can be modelled by the RFS $\mathcal{B}(X_k)$ and the RFS map transition likelihood required in the recursion equation 4.3 can be written,

$$f_{\mathcal{M}}(\mathcal{M}_k|\mathcal{M}_{k-1}, X_k) = \sum_{\mathcal{W} \subseteq \mathcal{M}_k} f_{\mathcal{M}}(\mathcal{W}|\mathcal{M}_{k-1}) f_{\mathcal{B}}(\mathcal{M}_k - \mathcal{W}|X_k) \tag{4.16}$$

where $f_{\mathcal{M}}(\mathcal{W}|\mathcal{M}_{k-1})$ is the transition density of the set of features that are in $FoV(X_{0:k-1})$ at time $k-1$ to time k, and $f_{\mathcal{B}}(\mathcal{M}_k - \mathcal{W}|X_k)$ is the density of the RFS, $\mathcal{B}(X_k)$, of the new features that pass within the field of view at time k.

It can then be shown that the intensity of the multi-feature predicted density $p(\mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_k)$ is given by [3],

$$v_{k|k-1}(m|X_k) = v_{k-1}(m|X_{k-1}) + b_k(m|X_k) \tag{4.17}$$

where $b_k(m|X_k)$ is the intensity of the new feature RFS, $\mathcal{B}(X_k)$.

4.5 PHD-FBRM Filter Implementations

This section outlines suitable implementations of the previously presented PHD-FBRM frameworks. The static map state assumption is analogous to

[5] Mathematically: all features in the entire map, $\mathcal{M}$, not already in $\mathcal{M}_{k-1}$.

grid mapping approaches, in that an intuitive prior is set over the entire map, and inferences are drawn on both the observed and unobserved map. A Sequential Monte Carlo (SMC) implementation for this map interpretation is adopted. The pseudo-static approach is analogous to the approach adopted by the majority of FB map estimation algorithms [1], [36], [17], which uses Kalman filters for map estimation. For accurate comparison with previous vector based methods, a Kalman-based, GM implementation is thus adopted.

4.5.1 The Static Map: An SMC PHD-FBRM Implementation

Recall from section 4.4.1 that the the static feature map filter is analogous to an occupancy grid in that the number of map elements is fixed a priori. Dispersing particles in a grid (or any arbitrary) layout, represents all the possible regions of features occurring, whereas the total weight in a given region represents the a priori estimate on the number of features in that region. Thus in this map representation, the particles model both the multitude of the number of features in the map, as well as their locations.

This section presents a SMC implementation of the static map PHD-FBRM recursion. Assume at time $k-1$, a set of weighted particles $\{\omega_{k-1}^{(j)}, m_{k-1}^{(j)}\}_{j=1}^{J_{k-1}}$ representing the prior intensity, $v_{k-1}(m|X_{k-1})$ is available, i.e.

$$v_{k-1}(m|X_{k-1}) \approx \sum_{j=1}^{J_{k-1}} \omega_{k-1}^{(j)} \delta_{m_{k-1}^{(j)}}(m). \tag{4.18}$$

Exploiting a suitably chosen proposal q, the predicted MC approximation for particles $j = 1, \ldots, J_{k-1}$ can be obtained according to,

$$\tilde{m}_k^{(j)} \sim q(m|m_{k-1}^{(j)}, \mathcal{Z}_k) \tag{4.19}$$

According to the PHD-FBRM update of equation 4.11, following the measurement $\mathcal{Z}_k$ at time k, the posterior map intensity may then be approximated by,

$$v_k(m|X_k) \approx \sum_{j=1}^{J_k} \omega_k^{(j)} \delta_{m_k^{(j)}}(m) \tag{4.20}$$

where, based on equation 4.11, the weights are initially calculated via,

$$\tilde{\omega}_k^{(j)} = \left[1 - P_D(\tilde{m}_k^{(j)}|X_k) + \sum_{z \in \mathcal{Z}_k} \frac{\varsigma_z(\tilde{m}_k^{(j)}, X_k)}{c_k(z) + \sum_{i=1}^{J_k} \varsigma_z(\tilde{m}_k^{(i)}, X_k)\omega_{k-1}^{(i)}}\right] \omega_{k-1}^{(j)} \tag{4.21}$$

with,

$$\varsigma_z(\tilde{m}_k^{(j)}, X_k) = P_D(\tilde{m}_k^{(j)}|X_k)g(z|\tilde{m}_k^{(j)}, X_k). \tag{4.22}$$

Resampling then results in the required posterior intensity parameters, $\{m_k^{(j)}, \omega_k^{(j)}\}_{j=1}^{J_k}$. Note that contrary to standard SMC implementations, the sum of the weights $\omega_k^{(j)}$ used to calculate the updated intensity function of equation 4.20 is not unity. In this case, the sum $\sum_{j=1}^{J_k} \omega_k^{(j)} = \hat{\mathfrak{m}}_k$, the estimated dimensionality of the feature map.

4.5.1.1 State Initialisation and Estimation

The static map FBRM implementation requires the map intensity to be *a priori* initialised. This is analogous in principle to grid mapping algorithms, in which the occupancy state is typically initialised at a non-informative value of 0.5. For the FBRM case, a non-informative initialisation condition is an intensity with uniform spatial distribution over the entire mapping state space. The integral of the intensity would give the expected number of features within the map. This can be achieved through an expected feature density, however, in practice, the results are not overly sensitive to this initialisation parameter. At any time k, the posterior map estimate can be extracted by segmenting the posterior particle approximation into $\hat{\mathfrak{m}}_k$ clusters (perhaps with a K-means clustering algorithm) and using the cluster centroid as the estimate of the feature location.

4.5.1.2 Particle Representation Considerations

Unlike particle filter implementations for robot localisation filters [37], maintaining a fixed number of particles for a robotic mapping filter may result in depletion issues where there is an insufficient number of particles to adequately represent the spatial uncertainty of all estimates features. Therefore the map is typically resampled with respect to ρ, the number of particles per feature, to result in $J_k = \rho\hat{\mathfrak{m}}_k$. Furthermore, a static particle transition (equation 4.19) may deteriorate the accuracy of the filter since the measurements would not be bound to the initial particle lattice (such as is the case for GBRM filters). A Brownian transition of very small covariance can therefore improve estimation accuracy.

In practise, the static-map FBRM filter can be prone to large dimensionality estimation errors, especially in the case of features with non-unity detection probability. In such cases, the following pseudo-static implementation is better suited, as the transition equation directly accounts for predicted dimensionality changes in the map, for instance as a result of features only being detectable well within the sensor field of view.

4.5.1.3 Static Map, SMC PHD FBRM Pseudo Code

This section details the pseudo-code of the proposed filter. Table 4.1 outlines in the initialisation algorithm while Table 4.2 describes a simple prediction algorithm assuming a static map. Table 4.3 shows how to update the map given the latest measurement and vehicle pose. Tables 4.4 and 4.5 detail the resampling and estimation processes of the filter.

Table 4.1 SMC-PHD-FBRM-Initialise

Algorithm SMC-PHD-FBRM-Initialise$\big(v_0(m|X_0)\big)$
// For all map particles
1. for i = 1 to J_0 do
// Initialise strategy (e.g. grid lattice)
2. $\quad m_0^{(i)} = [x\ y]$
// Estimate of feature density
3. $\quad \omega_0^{(i)} = \alpha$
4. end for
// The initialised static map PHD
5. $v_0(m|X_0) = \{\omega_0^{(i)}, m_0^{(i)}\}_{j=1}^{J_0}$
6. return$\big(v_0(m|X_0)\big)$

Table 4.2 SMC-PHD-FBRM-Predict

Algorithm SMC-PHD-FBRM-Predict$\big(v_{k-1}(m|X_{k-1})\big)$
// PHD Prediction
1. for i = 1 to J_{k-1} do
// Static map assumption with perturbation
2. $\quad \tilde{m}_k^{(i)} = m_{k-1}^{(i)} + \delta(m)$
3. end for
4. $J_{k|k-1} = J_k$
// The predicted map PHD
5. $v_{k|k-1}(m|X_k) = \{\omega_{k-1}^{(j)}, \tilde{m}_k^{(j)}\}_{j=1}^{J_{k|k-1}}$
6. return$\big(v_{k|k-1}(m|X_{k-1})\big)$

Table 4.3 SMC-PHD-FBRM-Update

Algorithm SMC-PHD-FBRM-Update$\big(\mathcal{Z}_k, X_k, v_{k|k-1}(m|X_{k-1})\big)$
// To obtain $v_k(m|X_k)$
1. for j = 1 to $\mathfrak{z}_k$ do
2. $\quad \varsigma_k^{(j)} = \sum_{i=1}^{J_{k|k-1}} \mathcal{N}\big(z; H_k\tilde{m}_k^{(i)}, R\big)\omega_{k-1}^{(i)}$
3. end
// evaluate (4.21)
4. for i = 1 to $J_{k|k-1}$ do
5. $$\tilde{\omega}_k^{(i)} = \big(1 - P_D(\tilde{m}|X_k) + \varpi\big)\omega_{k-1}^{(i)} + \sum_{j}^{\mathfrak{z}_k} \frac{\mathcal{N}\big(z^j; H_k\tilde{m}_k^{(i)}, R\big)\omega_{k-1}^{(i)}}{c_k(z) + \varsigma_k^{(j)}}$$
6. end for
7. $v_k(\tilde{m}|X_k) = \{\tilde{\omega}_k^{(j)}, \tilde{m}_k^{(j)}\}_{j=1}^{J_{k|k-1}}$
8. return$\big(v_k(\tilde{m}|X_k)\big)$

Table 4.4 SMC-PHD-FBRM-Resample

Algorithm SMC-PHD-FBRM-Resample$\big(v_k(\tilde{m}|X_{k-1})\big)$
// Resample the updated PHD
1. $\hat{\mathfrak{m}}_k = \sum_{j=1}^{J_{k|k-1}} \tilde{\omega}_k^{(j)}$
// Normalise weights for resampling
2. Resample $\{\frac{\tilde{\omega}_k^{(j)}}{\hat{\mathfrak{m}}_k}, \tilde{m}_k^{(j)}\}_{j=1}^{J_{k|k-1}}$
3. Rescale resampled weights by $\hat{\mathfrak{m}}_k$ to get $\{\omega_{k-1}^{(j)}, m_k^{(j)}\}_{j=1}^{J_k}$
4. $v_k(m|X_k) = \{\omega_k^{(j)}, m_k^{(j)}\}_{j=1}^{J_k}$
5. return$\big(v_k(m|X_k)\big)$

Table 4.5 SMC-PHD-FBRM-Estimate

Algorithm SMC-PHD-FBRM-Estimate$\big(v_k(m|X_k)\big)$
// Initialise the map estimate
1. $\hat{\mathcal{M}}_k = \emptyset$
2. Cluster $\{m_k^{(j)}\}_{j=1}^{J_k}$ into $\hat{\mathfrak{m}}_k$ clusters, $\epsilon^{(i)}$
2. for i = 1 to $\hat{\mathfrak{m}}_k$ do
// concatenate estimate
4. $\quad \hat{\mathcal{M}}_k = [\hat{\mathcal{M}}_k \text{ centroid}(\epsilon^{(i)})]$
6. end for
7. return $\big(\hat{\mathcal{M}}_k\big)$

4.5.2 The Pseudo-static Map: A GM PHD-FBRM Implementation

This section presents a Gaussian mixture implementation of the pseudo-static map PHD-FBRM recursion. Recall from section 4.4.2, the pseudo-static map is a map which is modelled as static in feature location, but monotonically increasing in size as new features are observed. As with the SMC implementation described in section 4.4.1, the Gaussian components of the mixture represent the multitude of locations of features in the map, while their masses represent the number of features in that given region. In this case, let the prior map intensity, v_{k-1}, be a Gaussian mixture of the form,

$$v_{k-1}(m|X_{k-1}) = \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}\big(m; \mu_{k-1}^{(j)}, P_{k-1}^{(j)}\big) \tag{4.23}$$

which is a mixture of J_{k-1} Gaussians, with $w_{k-1}^{(j)}$, $\mu_{k-1}^{(j)}$ and $P_{k-1}^{(j)}$ being their corresponding predicted weights, means and covariances respectively. Note that, in contrast to the previous implementation of Section 4.5.1, the map state is now time dependent. Let the new feature intensity in equation 4.17, $b_k(m|\mathcal{Z}_{k-1}, X_{k-1})$ at time k, also be a Gaussian mixture of the form

$$b_k(m|\mathcal{Z}_{k-1}, X_{k-1}) = \sum_{j=1}^{J_{b,k}} w_{b,k}^{(j)} \mathcal{N}\big(m; \mu_{b,k}^{(j)}, P_{b,k}^{(j)}\big) \tag{4.24}$$

where, $J_{b,k}$ defines the number of Gaussians in the new feature intensity at time k and $w_{b,k}^{(j)}$, $\mu_{b,k}^{(j)}$ and $P_{b,k}^{(j)}$ determine the shape of the new feature GM proposal density according to a chosen strategy. This is analogous to the proposal distribution in the particle filter and provides an initial estimate of the new features entering the map. More details on the chosen strategy are provided in Section 4.5.2.1. The predicted intensity $v_{k|k-1}$ is therefore also a Gaussian mixture,

$$v_{k|k-1}(m|X_k) = \sum_{j=1}^{J_{k|k-1}} w_{k|k-1}^{(j)} \mathcal{N}\big(m; \mu_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\big) \tag{4.25}$$

which consists of $J_{k|k-1} = J_{k-1} + J_{b,k}$ Gaussians representing the union of the prior map intensity, $v_{k-1}(m|X_k)$, and the proposed new feature intensity according to equation 4.17 where,

$$\left.\begin{array}{l} w_{k|k-1}^{(j)} = w_{k-1}^{(j)} \\ \mu_{k|k-1}^{(j)} = \mu_{k-1}^{(j)} \\ P_{k|k-1}^{(j)} = P_{k-1}^{(j)} \end{array}\right\} \text{for } j \in \{1, \ldots, J_{k-1}\} \text{ (previously observed features)}$$

$$\left.\begin{array}{l} w_{k|k-1}^{(j)} = w_{b,k}^{(j-J_{k-1})} \\ \mu_{k|k-1}^{(j)} = \mu_{b,k}^{(j-J_{k-1})} \\ P_{k|k-1}^{(j)} = P_{b,k}^{(j-J_{k-1})} \end{array}\right\} \text{for } j \in \{(J_{k-1}+1), \ldots, J_{k|k-1}\} \text{ (newly observed features).}$$

Since the measurement likelihood is also of Gaussian form, it can be seen from equation 4.11, that the posterior intensity, v_k is then also a Gaussian mixture given by,

$$v_k(m|X_k) = v_{k|k-1}(m|X_k)\left[1 - P_D(m|X_k) + \sum_{z \in \mathcal{Z}_k} \sum_{j=1}^{J_{k|k-1}} v_{G,k}^{(j)}(z, m|X_k)\right]. \tag{4.26}$$

From the general PHD filter update equation 4.11, the components of the above equation are given by,

$$v_{G,k}^{(j)}(z, m|x_k) = w_k^{(j)}(z|x_k)\mathcal{N}(m; \mu_{k|k}^{(j)}, P_{k|k}^{(j)}) \tag{4.27}$$

$$w_k^{(j)}(z|X_k) = \frac{P_D(m|X_k) w_{k|k-1}^{(j)} q^{(j)}(z, X_k)}{c_k(z) + \sum_{i=1}^{J_{k|k-1}} P_D(m|X_k) w_{k|k-1}^{(i)} q^{(i)}(z, X_k)} \tag{4.28}$$

where, $q^{(i)}(z, X_k) = \mathcal{N}\big(z; H_k \mu_{k|k-1}^{(i)}, S_k^{(i)}\big)$. The components $\mu_{k|k}^{(j)}$ and $P_{k|k}^{(j)}$ can be obtained from the standard EKF update equations,

$$S_k^{(i)} = R_k + \nabla H_k P_{k|k-1}^{(i)} \nabla H_k^T \tag{4.29}$$

$$K_k^{(i)} = P_{k|k-1}^{(i)} \nabla H_k^T [S_k^{(i)}]^{-1} \tag{4.30}$$

$$\mu_{k|k}^{(i)} = \mu_{k|k-1}^{(i)} + K_k^{(i)}(z - H_k(\mu_{k|k-1}^{(i)})) \tag{4.31}$$

$$P_{k|k}^{(i)} = [I - K_k^{(i)} \nabla H_k] P_{k|k-1}^{(i)} \tag{4.32}$$

with ∇H_k being the Jacobian of the measurement equation with respect to the landmarks estimated location. As stated previously, the clutter RFS, $\mathcal{C}_k$, is assumed Poisson distributed [18], [14] in number and uniformly spaced over the mapping region. Therefore the clutter intensity is given by,

$$c_k(z) = \lambda_c V U(z) \tag{4.33}$$

where λ_c is the clutter density per scan, V is the volume of the surveillance region and and $U(z)$ denotes a uniform distribution on the measurement space.

4.5.2.1 The FBRM New Feature Proposal Strategy

While any arbitrary strategy can be used for the new feature intensity $b_k(m|\mathcal{Z}_{k-1}, X_{k-1})$, an intuitive strategy closely related to previous vector-based implementations is used in this work. As seen in equation 4.24, given a GM intensity representation, the mean, covariance and weight of each Gaussian must be chosen. The GM component means and covariances determine the spatial distribution on the likely location of new features entering the map, with the sum of the weights, $\sum_{j=1}^{J_{b,k}} w_{b,k}^{(j)}$ then providing an estimate of the expected number of new features to appear at time k.

The new feature intensity at time k, $b_k(m|\mathcal{Z}_{k-1}, X_{k-1})$, is adaptively determined by the previous measurements $\mathcal{Z}_{k-1}$ and the previous vehicle state X_{k-1}. The components of the GM of equation 4.24 are then determined according to

$$w_{b,k}^{(j)} = 0.01, \quad \mu_{b,k}^{(j)} = h^{-1}(z_{k-1}^{j}, X_{k-1}),$$
$$P_{b,k}^{(j)} = h^{'}(\mu^{(j)}, X_{k-1})\mathrm{R}[h^{'}(\mu^{(j)}, X_{k-1})]^T$$

where h^{-1} is the inverse measurement equation, R is the measurement noise covariance and $h^{'}(\mu^{(j)}, X_{k-1})$ is the Jacobian of the measurement model function with respect to the Gaussian state, j. Therefore, the implementation initially considers *all* detections at time $k-1$ to be potential new features at time k. This allows for features of low detection probability, which perhaps only become detectable at close ranges, to be reliably estimated.

4.5.2.2 Gaussian Management and State Estimation

Note from (4.26) that every measurement is combined with every Gaussian component, resulting in an explosive growth in the number of Gaussians in the posterior intensity. Therefore, as with previous GM implementations [38], pruning and merging operations are required. Gaussians which are determined sufficiently close (through a Mahalanobis distance threshold) are merged into a singular Gaussian as in [33]. Recall from equation 3.18 that the integral of the intensity function over a region on the space of features, S, equals the number of features present in that region. Therefore, in some instances (particularly for closely lying features and/or high measurement noise), more than one feature can be represented by a single Gaussian component. A multi-level threshold

is thus used to estimate the dimensionality according to ϵ_L, with L being the number of features modelled by a single Gaussian component. State estimation occurs by extracting the means and covariances of the Gaussian components whose associated weights exceed the given thresholds.

4.5.2.3 Pseudo-static Map, GMM PHD FBRM Pseudo Code

This section details the pseudo-code of the GMM-PHD-FBRM filter. Table 4.6 outlines the generation of the feature birth density whilst Table 4.7 details the construction of the predicted GMM PHD. Table 4.8 presents its update given the current measurement, Table 4.9 describes the Gaussian Management algorithm and finally, Table 4.10 shows the process used to extract feature map estimates at each time.

Table 4.6 GMM-PHD-FBRM-Birth

Algorithm GMM-PHD-FBRM-Birth$\left(\mathcal{Z}_{k-1}, X_{k-1}\right)$
// Implementation of section 4.5.2.1 (Any strategy is valid)
// For each measurement
1. for i = 1 to $\mathfrak{z}_{k-1}$ do
// initialise the mean
2. $\mu_{b,k}^{(i)} = h^{-1}(z_{k-1}^{(i)}, X_{k-1})$
// initialise the covariance
3. $P_{b,k}^{(i)} = h^{'}(m_{b,k}^{(i)}, X_{k-1})$
// initialise the weight
4. $\omega_{b,k}^{(i)} = \alpha$
5. end for
// Set the number of birth components
6. $J_{b,k} = \mathfrak{z}_{k-1}$
// Construct birth PHD
7. $b_k(m|\mathcal{Z}_{k-1}, X_{k-1}) = \{\mu_{b,k}^{(i)}, P_{b,k}^{(i)}, \omega_{b,k}^{(i)}\}_{i=1}^{J_{b,k}}$
8. return $\left(b_k(m|\mathcal{Z}_{k-1}, X_{k-1}) \right)$

4.6 Algorithm Performance

This section compares the proposed finite-set-based framework and GM implementation to popular solutions from the literature. Regardless of the choice of vehicle state representation, the vast majority of feature-based implementations adopt an EKF framework coupled with data association and map management methods for propagating the feature map estimate [17], [14],

Table 4.7 GMM-PHD-FBRM-Predict

Algorithm GMM-PHD-FBRM-Predict$\big(\mathcal{Z}_{k-1}, X_{k-1}, v_{k-1}(m|X_{k-1})\big)$
// To obtain equation (4.25)
// PHD Prediction: existing components
1. for i = 1 to J_{k-1} do
// Static map assumption
2. $\mu_{k|k-1}^{(i)} = \mu_{k-1}^{(i)}, P_{k|k-1}^{(i)} = P_{k-1}^{(i)}, \omega_{k|k-1}^{(i)} = \omega_{k-1}^{(i)}$
3. end for
// Generate map birth components
4. GMM-PHD-FBRM-Birth$(\mathcal{Z}_{k-1}, X_{k-1})$
// PHD Prediction: new components
5. for i = 1 to $J_{b,k}$ do
// Components from birth proposal $b_k(m|\mathcal{Z}_{k-1}, X_k)$
6. $\mu_{k|k-1}^{(J_{k-1}+i)} = \mu_{b,k}^{(i)}, P_{k|k-1}^{(J_{k-1}+i)} = P_{b,k}^{(i)}, \omega_{k|k-1}^{(J_{k-1}+i)} = \omega_{b,k}^{(i)}$
7. end for
// Total number of Gaussian components in $v_{k|k-1}(m|X_{k-1})$
8. $J_{k|k-1} = J_{k-1} + J_{b,k}$
// The predicted map PHD
9. $v_{k|k-1}(m|X_k) = \{\mu_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}, \omega_{k|k-1}^{(i)}\}_{i=1}^{J_{k|k-1}}$
10. return$\big(v_{k|k-1}(m|X_k)\big)$

[1], [15], [15], [18]. As emphasised throughout this chapter, in their estimation of the feature map, estimates of the feature *number* and location are generated through independent filters. Under such methods, it is not clear whether Bayes optimality can be achieved, i.e. how is the Bayes risk defined and minimised. A simple nearest-neighbour EKF implementation is first adopted, coupled with the 'log-odds' map management method. The map-management approach 'propagates' the map size by assigning an intuitive log odds score to each associated or unassociated-associated feature. The probability of a feature existing can then be readily recovered from the log-odds representation. For this analysis, a 95% χ^2 association confidence window is chosen. This is identical to the map estimation method of FastSLAM (for a single particle) [17], however in the analysis of this Chapter, the vehicle trajectory is assumed *known* and is not estimated. A joint compatibility branch and bound (JCBB) [39] data association approach, is also adopted, however given that the true location of the vehicle is assumed known, there is lack of measurement error correlation which JCBB exploits, thus limiting its usefulness for mapping-only trials. As with the proposed framework, map size estimates are extracted by comparison with a predefined threshold, ϵ. Tentative feature lists are maintained for unconfirmed features, which are then either added as confirmed features (with an increase in the map size estimate), or

Table 4.8 GMM-PHD-FBRM-Update

Algorithm GMM-PHD-FBRM-Update$\left(\mathcal{Z}_k, X_k, v_{k|k-1}(m|X_{k-1})\right)$
// Initialise number of Gaussian components
1. L = 0
// Missed detections and Update Terms
2. for i = 1 to $J_{k|k-1}$ do
// Increment component counter
3. L = L+1
// miss-detected component update of (4.26)
4. $\mu_k^{(L)} = \mu_{k|k-1}^{(i)}, P_k^{(L)} = P_{k|k-1}^{(i)}$
// miss-detected weight update of (4.26)
5. $\omega_k^{(L)} = (1 - P_D)\omega_{k|k-1}^{(i)}$
// measurement prediction
6. $z_{k|k-1}^{(i)} = h(\mu_{k|k-1}^{(i)}, X_k)$
// Calculate Jacobian
7. $H = h^{'}(\mu_{k|k-1}^{(i)}, X_k)$
// Kalman Gain of (4.30)
8. $K_k^{(i)} = P_{k|k-1}^{(i)}[H]^T[S_k^{(i)}]^{-1}$
// Innovation Covariance of (4.29)
9. $S_k^{(i)} = HP_{k|k-1}^{(i)}[H]^T + R$
// Updated Covariance of (4.32)
10. $P_{U,k}^{(i)} = [I - K_k^{(i)}H]P_{k|k-1}^{(i)}$
11. end for
// For each measurement
12. for i = 1 to $\mathfrak{z}_k$ do
// For each map PHD component
13. for j = 1 to $J_{k|k-1}$ do
// Updated component mean of (4.31)
15. $\mu_k^{(L+j)} = \mu_{k|k-1}^{(j)} + K_k^{(j)}(z_k^{(i)} - z_{k|k-1}^{(j)})$
// Updated GMM component covariance
16. $P_k^{(L+j)} = P_{U,k}^{(j)}$
// Numerator of (4.27)
17. $\tau^{(j)} = P_D\omega_{k|k-1}^{(j)}|2\pi S_k^{(j)}|^{-0.5}$
$\times \exp\left((z_k^{(i)} - z_{k|k-1}^{(j)})[S_k^{(j)}]^{-1}(z_k^{(i)} - z_{k|k-1}^{(j)})\right)$
18. end for

Table 4.8 (*Continued)*

// For each map PHD component
19. for j = 1 to $J_{k|k-1}$ do
// denominator of (4.27)
20. $\omega_k^{(L+j)} = \tau^{(j)}/(c(z) + \sum_{l=1}^{J_{k|k-1}} \tau^{(l)})$
21. end for
22. L = L+$J_{k|k-1}$
23. end for
// Number of components in updated GMM
24. J_k = L
// The updated map PHD
25. $v_k(m|X_k) = \{\mu_k^{(i)}, P_k^{(i)}, \omega_k^{(i)}\}_{i=1}^{J_k}$
26. return$\big(X_k, v_k(m|X_k)\big)$

deleted. The purpose of the trials is not to indicate superiority over existing approaches, but to confirm the 'sensible' operation of this fundamentally new approach to propagating and evaluating feature-based map estimates. These results form the basis for future extensions to the SLAM problem, and to applications in cluttered environments, where the true qualities of the RFS framework will be demonstrated.

Simulated trials are carried out based on the feature map shown previously in Figure 2.5. For simplicity of presentation, the environment shall be restricted to comprise only point features, however the framework can readily be extended to other less structured environments with the use of robust feature extraction techniques. For each trial, all filters receive identical measurement sequences. The existence threshold is set at $\epsilon_N = N - 0.4$. Therefore, for previous approaches, $\epsilon_1 = 0.6$, since only one feature can be represented per Gaussian. For the proposed PHD solution, the Gaussian mass indicates the number of features, thus multi-feature thresholds, $\epsilon_2 = 1.6$, $\epsilon_3 = 2.6$, $\epsilon_4 = 3.6$ etc. are used.

As outlined in [17], the independent map management algorithms incorporated into vector-based recursions also require score values to be intuitively set. To this end, an associated feature receives a score of $+0.5$, while an unassociated feature (within the sensor field of view), receives a score of -0.2. A simple existence counting rule is therefore established. While other methods of estimating the map dimensionality exist in the literature [1], as emphasised throughout this book, the independence of such approaches from the Bayesian update, compromises estimation optimally. The following FBRM error metric results are obtained from equation 4.6, with parameter $c = 5$, while the effects of adjusting this parameter are discussed in Section 4.6.5. Results of applying the RFS map estimation framework in a real, outdoor environment will be shown in Section 4.6.7.

Table 4.9 GMM-PHD-FBRM-Prune

Algorithm GMM-PHD-FBRM-Prune$\big(v_k(m|X_k), D_{\min}, T_{\min}, J_{\max}\big)$
// Initialise number of Gaussian components
1. L = 0
// Remove components with weight below $T_{\min}$
2. I = $\{i = 1, \ldots, J_k | \omega_k^{(i)} > T_{\min}\}$ // Gaussian merging
3. do while $I \neq \emptyset$
// Increment component counter
4. L = L+1
// Get index of max weight component
5. j = $\arg\max_{i \in I} \omega_k^{(i)}$
// Cluster those within distance $D_{\min}$
6. K = $\{i \in I | (\mu_k^{(i)} - \mu_k^{(j)})^T [P_k^{(i)}]^{-1} (\mu_k^{(i)} - \mu_k^{(j)}) \leq D_{\min}\}$
// Combine component weights
7. $\tilde{\omega}_k^{(L)} = \sum_{i \in K} \omega_k^{(i)}$
// Weighted average mean
8. $\tilde{\mu}_k^{(L)} = \frac{1}{\tilde{\omega}_k^{(L)}} \sum_{i \in K} \omega_k^{(i)} \mu_k^{(i)}$
// Combined covariance
9. $\tilde{P}_k^{(L)} = \frac{1}{\tilde{\omega}_k^{(L)}} \sum_{i \in K} \omega_k^{(i)} \big(P_k^{(i)} + (\tilde{\mu}_k^{(L)} - \mu_k^{(i)})(\tilde{\mu}_k^{(L)} - \mu_k^{(i)})^T\big)$
// Remove K from I and repeat
10. I = I - K
11. end for
12. $J_k = L$
// If max component number exceeded
12. if $J_k \geq J_{\max}$
13. replace $\{\tilde{\omega}_k^{(i)}, \tilde{\mu}_k^{(i)}, \tilde{P}_k^{(i)}\}_{i=1}^{J_k}$ with those of the $J_{\max}$ largest weights.
11. end for
// The pruned map PHD
13. $v_k(m|X_k) = \{\tilde{\omega}_k^{(i)}, \tilde{\mu}_k^{(i)}, \tilde{P}_k^{(i)}\}_{i=1}^{J_k}$
14. return $\big(v_k(m|X_k)\big)$

4.6.1 FBRM Error vs. Measurement Noise

Increases in measurement noise (range/bearing), have the subsequent effect of increasing data association ambiguity in vector based methods, and hence the difficulty of the FBRM problem. To demonstrate the performance of the proposed method, trials are carried out in which the measurements are subjected to increasing amounts of noise. The measurement noise covariance for each trial is set at R= $\gamma[(0.25m)^2\ 0; 0\ (0.5^o)^2]$, where $\gamma \in [1, \ldots, 100]$.

Table 4.10 GMM-PHD-FBRM-Estimate

Algorithm GMM-PHD-FBRM-Estimate$\left(v_k(m|X_k), T_{\text{feature}}\right)$
// Initialise the map estimate
1. $\hat{\mathcal{M}}_k = \emptyset$
2. for i = 1 to J_k do
3. if $\omega_k^{(i)} > T_{\text{feature}}$
// concatenate estimate
4. $\hat{\mathcal{M}}_k = [\hat{\mathcal{M}}_k \ \mu_k^{(i)}]$
5. end if
6. end for
7. return $\left(\hat{\mathcal{M}}_k\right)$

Figure 4.1 shows a comparison of the final FBRM error of the posterior feature map estimate for each filter at differing noise inflation values, γ.

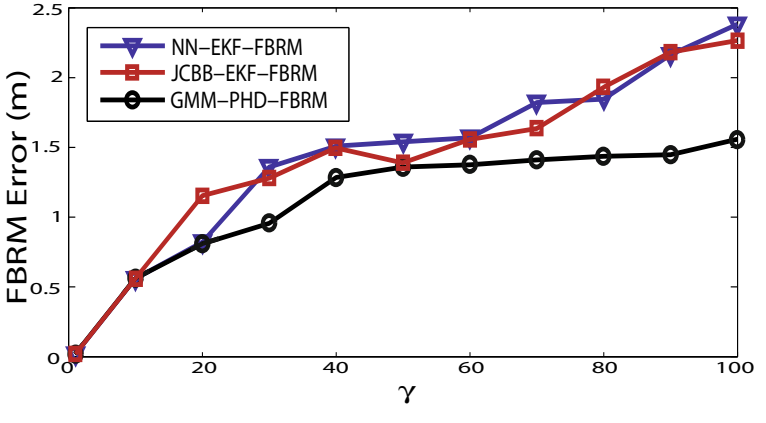


Fig. 4.1 Comparison of FB mapping error vs. measurement noise for the proposed filters and classical vector EKF solutions.

The result demonstrates the reasonable performance of the new framework for estimating the feature-based map. Note that for $\gamma \leq 20$, all filters return comparable map estimates, indicating little data association ambiguity. As γ increases however, association errors are introduced into the NN and JCBB solutions resulting in the continual deterioration of the map estimation accuracy. Note that the lack of measurement error correlation (since these feature

map estimation trials assume a known vehicle trajectory), reduces the effectiveness of JCBB. While the error in the GMM-PHD approach also increases, it appears robust to an increasing γ.

4.6.2 FBRM Error vs. Clutter Rate

Despite widespread acknowledgement that clutter is a prominent component of measurement uncertainty, a review of the literature indicates that algorithm performance versus increasing clutter rates is not often reported [39], [14], [18]. In this chapter, the clutter density is defined as the density of clutter measurements within the sensor field of view. Furthermore, clutter measurements are uniformly distributed in polar space, i.e. the sensor measurement space. Setting the measurement noise multiplier at a fixed value of $\gamma = 20$, which from Figure 4.1 is seen to be the point just prior to significant deviation in filter performances, the map estimation error for various clutter densities is analysed.

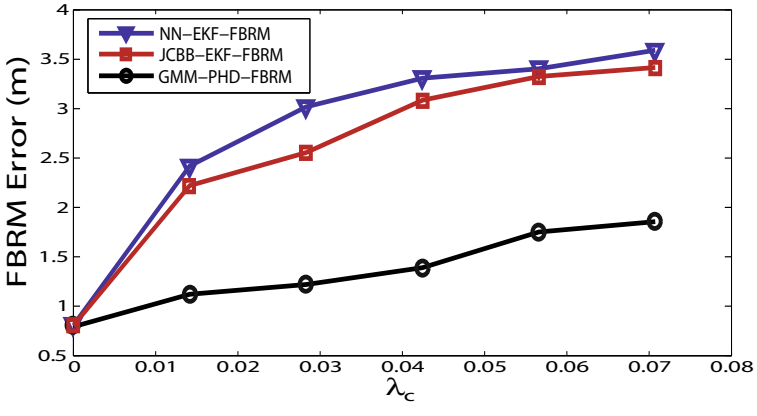


Fig. 4.2 Feature mapping error vs. clutter density for vector based NN-EKF and JCBB-EKF approaches and the proposed PHD framework. The proposed approach is seen to perform well in high clutter.

Figure 4.2 plots the map estimation error for clutter densities ranging from $\lambda_c = 0$ to $\lambda_c = 0.0707m^{-2}$, which correspond to a mean number of (Poisson distributed) clutter measurements of 0 to 50 per 360^o scan within a maximum range of $15m$. Given the commonly adopted map management methods, it is unspecified how scoring regimes should be altered for a given clutter rate. Furthermore, feature existence and spatial estimates cannot be jointly propagated in current frameworks as outlined previously in Chapter 3 (Section 3.2.1). The proposed framework however, directly incorporates clutter probabilities into the filter recursion of equation 4.11, thus no parameter

adjustment/tuning is required. Although the error for each approach monotonically increases with clutter rate, it is evident that the proposed framework is robust and produces meaningful map estimates. Note again that the effectiveness of JCBB in these trials is limited by the lack of measurement error correlation. Comparison with JCBB is thus dropped for the remainder of the analysis.

4.6.3 FBRM Error vs. Dynamic Object Density

This section demonstrates algorithm robustness in the presence of dynamic objects, given its static feature assumption in the formulation. Dynamic objects can corrupt the static map estimation process and, under this framework, are considered to be disturbances, which should not be declared as features. The measurement noise multiplier is again fixed at $\gamma = 20$, with the clutter density set at $\lambda_c = 0.014m^{-2}$ (10 clutter measurements per scan). Dynamic features are simulated to be uniformly distributed and evolve in time according to a Brownian motion model, $X_k = X_{k-1} + \omega_k$ with $\omega_k \sim \mathcal{N}(0, Q)$. The detection probability of a dynamic feature is set equal to that of a static feature ($P_D = 0.95$). Taking a map area to be $20m \times 20m$ (as in Figure 2.5), the density of dynamic features is increased and the effects on the mapping accuracy examined. Figure 4.3 plots the posterior map estimation error at various densities of moving features. The results verify the merits of the proposed approach in that mapping accuracy does not drastically deteriorate in the presence of dynamic objects.

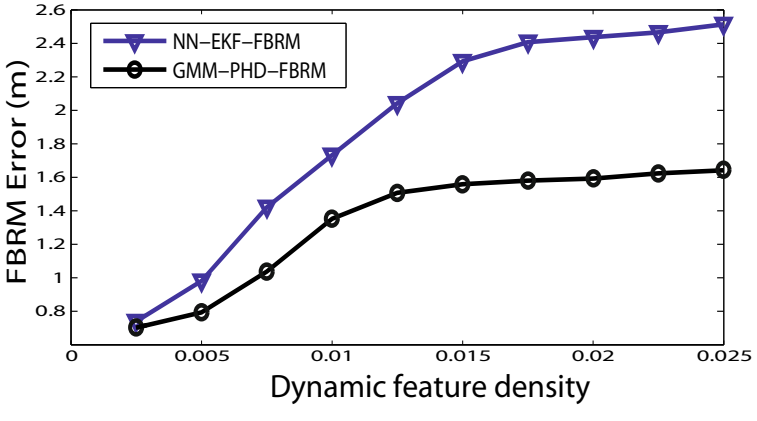


Fig. 4.3 Comparison of the map estimation error in the presence of increasing densities of moving features.

4.6.4 FBRM Error vs. Detection Probability

The influence of feature detection probability (a joint function of the signal detection probability and the feature extraction algorithm) is investigated in this section. Measurement noise and clutter rates are set as in the dynamic object trial of Section 4.6.3, while the feature detection probability is varied from 0.6 to 1. As is the case for the clutter trial of Section 4.6.2, most map-management methods do not have a feature detection probability parameter [14], [18]. However, as shown in Figure 4.4 in some instances the results can in fact be superior, depending on the sequence of measurements supplied to the filter. As is evident from the update equation 4.11, given a single missed detection of feature j at time k, the updated weight becomes $(1-P_D)w_k^{(j)}$. Given two successive missed detections, the weight becomes $(1-P_D)(1-P_D)w_k^{(j)}$, which typically would be below the map dimensionality estimation threshold, ϵ. Consequently, given successive missed detections prior to exiting the sensor field of view, features may not be correctly declared resulting in an increased mapping error. This problem becomes more evident as P_D decreases. This is in contrast to scoring regimes which depend on the total *number* of detections/missed detection of a given feature, as opposed to the *order* of the detection sequence. Enhancements to the proposed filter to improve this aspect are currently under investigation.

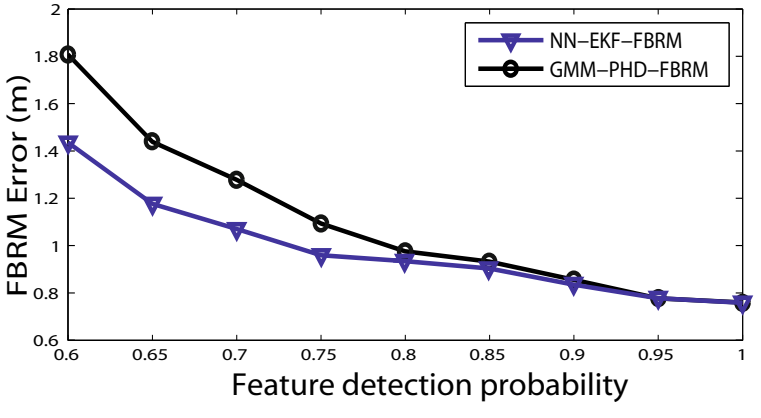


Fig. 4.4 Posterior map estimation error at increasing feature detection probabilities.

4.6.5 FBRM Error Metric Analysis

Figure 4.5 shows some simple generated input data, with the resulting posterior feature map estimates depicted in Figure 4.6. A simple example is used to demonstrate the use of the map estimation metric. Visual inspection reveals that, for the same set of measurements, a single false feature is declared with the NN-EKF approach coupled with increased feature localisation error when compared with the proposed method. For a quantitative comparison of the map estimate, as introduced in Section 4.3, the c parameter in equation 4.6 gives the relative weighting of feature number and location estimation error. Further insight into the effects of a given choice of c is shown in Figure 4.7. As is evident in the figure an increasing c parameter results in an

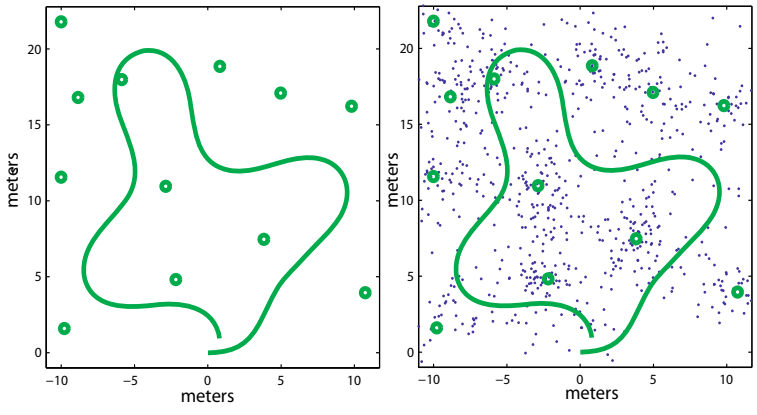


Fig. 4.5 A sample FBRM trial illustrating the ground truth (left) and raw measurement data (right).

increasing overall error for the NN-EKF estimate. This is due to the contribution of the single false feature, which correspondingly has no effect on the error reported by GMM-PHD posterior estimate, since in this particular trial, it has correctly estimated the true number of features. The c parameter also determines the maximum distance at which an estimate is classified as a poorly localised feature estimate, as opposed to a false feature declaration, and should be chosen based on the maximum allowable estimated feature location error in a given application.

For a given feature estimate-ground truth assignment, the value of p in equation 4.6 influences the contribution of the localisation estimation error. The visually evident improved feature location estimates of Figure 4.6 (right hand figure) are evident in Figure 4.7 by a lower total error reported for a given choice of p. To isolate the feature localisation estimation aspect, comparisons are also shown in which the false features from the NN-EKF estimate, was both ignored and included.

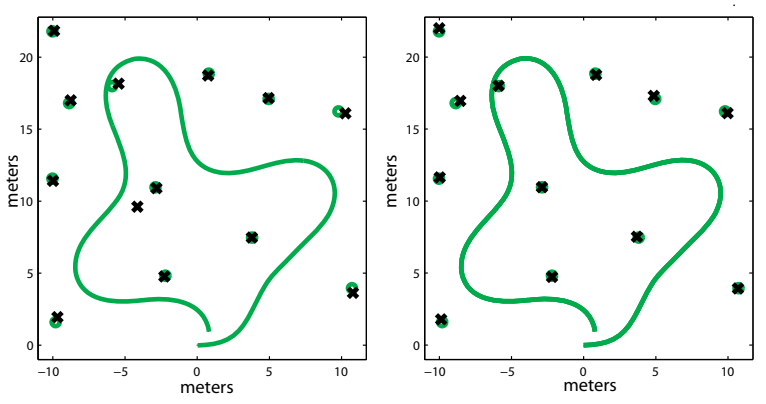


Fig. 4.6 Posterior FB map estimates from the classical NN-EKF-FBRM filter (left) and the proposed GMM-PHD-FBRM filter (right). Visual inspection indicates an improved map estimate from the proposed method, since all features are correctly declared (without false alarm) at 'closer' distances to the ground truth.

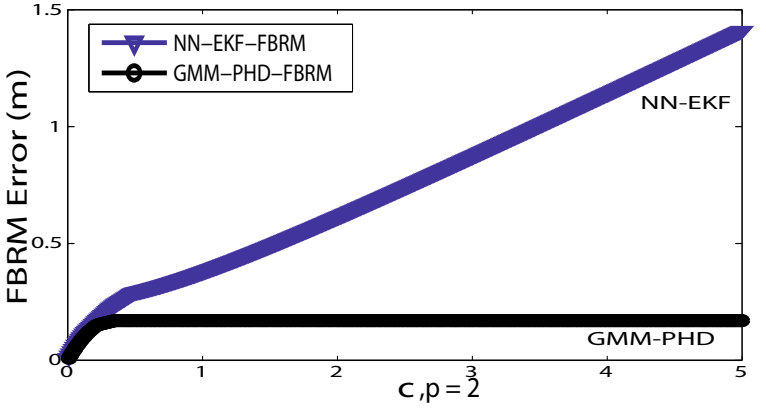


Fig. 4.7 FBRM error vs. c parameter, for a given value of p. The c parameter primarily determines the contribution of dimensionality errors to the total reported error.

4.6.6 Computational Complexity Analysis

At time k, given $\mathfrak{z}_k$ measurements and $\mathfrak{m}_k$ map states, the computational complexity of a naive implementation of a NN-EKF FBRM solution is $\mathcal{O}(\mathfrak{z}_k \mathfrak{m}_k)$, due to conditional feature / measurement independencies given an assumed known vehicle trajectory and evaluation of the measurement-map state associations. According to update equation 4.11, the complexity of the proposed solution is also $\mathcal{O}(\mathfrak{z}_k \mathfrak{m}_k)$. Absolute computational load is compared

through side by side C++ implementations of varying map dimensionality, on Intel(R) duo-core 1.73GHz processors with 2GB RAM. Figure 4.8 reports the average measurement update execution time, obtained through averaging 1000 Monte Carlo updates, for an increasing map dimensionality. The figure clearly illustrates the expected linear increase in computation load for both approaches, with the proposed method requiring more processing time than a naive NN-EKF implementation, primarily due to the pruning and merging operations required for GMM implementations. Despite its increased load, real-time implementation is possible.

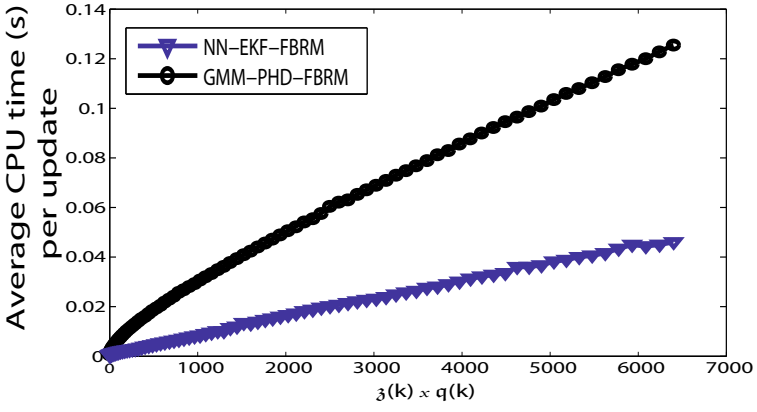


Fig. 4.8 Comparison of the computational complexity, demonstrating the linear increase in load with map dimensionality.

4.6.7 Outdoor Experiment

This section presents a practical implementation of the proposed finite-set-based framework for ground-based autonomous vehicles to demonstrate its applicability to real-world scenarios. The testing environment is a section of the Nanyang Technological University (NTU) university campus which contains numerous point features (trees, lamp posts, fire hydrants) which comprise the point-feature map. Figure 4.9 gives an overview of the testing area, with the inset vehicle-centric perspective showing the typical point features present. As with any practical implementation of an estimation algorithm, ground-truth is essential for error evaluation. For an FBRM implementation, both the true number and locations of all point features must be determined. This was achieved as best as possible, through observation of the synchronised video stream and successive manual scan matching of all the corresponding

laser data. The manually extracted centroid of all laser point features identified on the video stream then provided the ground truth locations[6]. The goal of an FBRM algorithm is therefore to estimate the number and location of all the point features within the mapped region (over a path of approximately 300m).

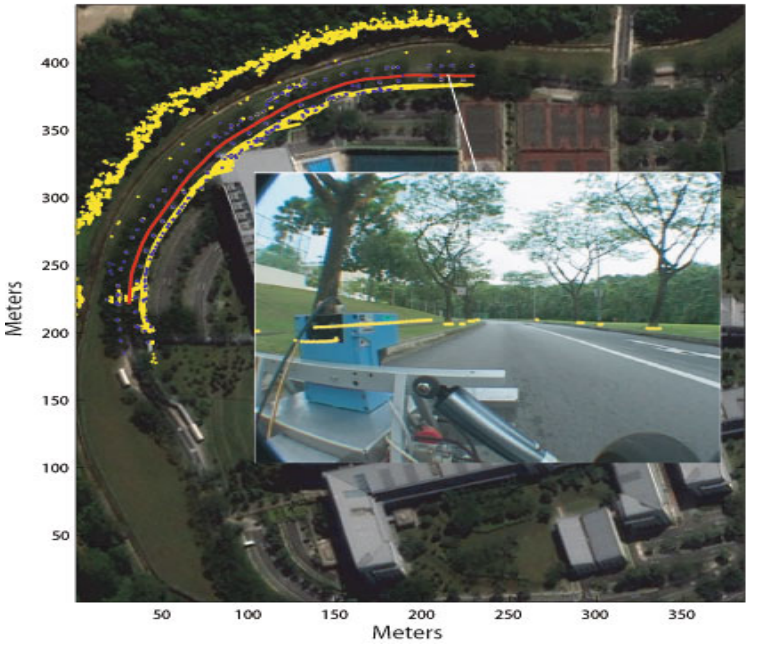


Fig. 4.9 An overview of the testing ground within the NTU university campus. The path (red) and raw laser measurements (yellow) are superimposed on a portion of a satellite image of the campus. The inset of a vehicle-centric view shows the environment mainly comprising point features such as trees and lamp posts. Laser data is projected into the image plane for verification.

For clarity of presentation, the preceding simulated trials examined only the final posterior map estimates. However, taking the true map at time k, $\mathcal{M}_k$, to be the subset of the entire map which has entered the sensor field of view, the quality of the map estimate can be examined over the course of the trial. Feature number estimates are compared in Figure 4.10 at each time step with the true number also provided. It can be seen that feature number estimation error is introduced into the vector based approach as the feature initialisation and termination routines generally rely on accurate data association decisions. These can be prone to failure in environments of fluctuating detection probabilities and frequent spurious measurements.

[6] Note this does not account for the potential presence of any sensor bias.

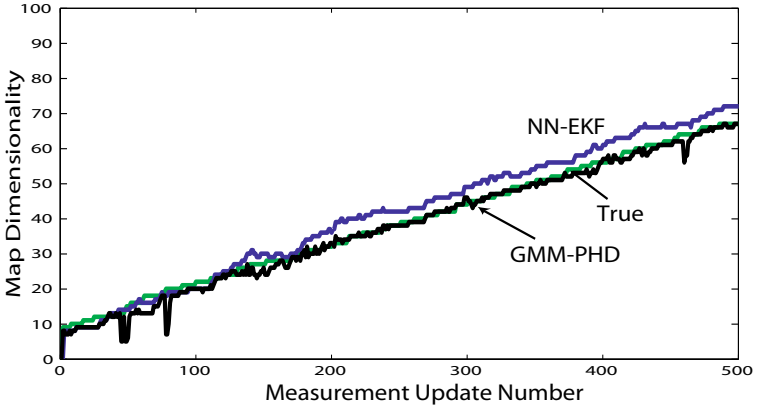


Fig. 4.10 The posterior feature map dimensionality estimate following each measurement update compared to the ground truth.

A graphical comparison of the final posterior map estimates is depicted in Figure 4.11, highlighting some mapping errors from both filters. It is evident that the proposed approach reports fewer false features, but also missed some features of poor detectability and/or feature extraction reliability. Features of low detection probability, may be less consistently identified in the proposed method as outlined previously in Section 4.6.4. Figure 4.12 plots the FBRM joint estimation error after each update. The ideal error is the mapping error (as a function of the entire map), given that each feature that enters the sensor field of view is instantaneously detected and assigned an error-free localisation estimate. The proposed method closely follows the ideal error with temporary glitches as some dimensionality estimation errors occur (as seen in Figure 4.10). The final posterior estimation error is also seen to be significantly less than that of the NN-EKF approach.

It is remarked that this example serves to demonstrate that a first order approximation to the proposed framework generates sensible results to the feature map estimation problem. Better approximation and more efficient implementation requires further investigation as well as comparisons with numerous sophisticated data association and/or map management techniques. Such investigations will be left until Chapters 5 and 6 in which RFS SLAM estimation is demonstrated.

4.7 Summary

This chapter proposed a random set theoretic framework for feature based (FB) map estimation, in the context of autonomous navigation problems. Estimating a FB map encompasses estimating the location of an unknown

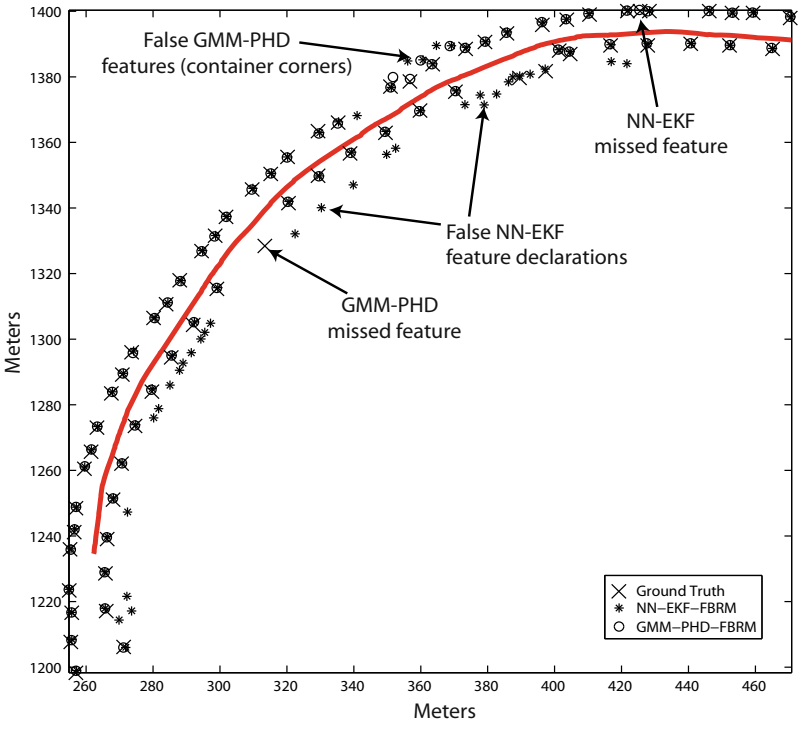


Fig. 4.11 Final posterior feature map estimate comparison. The proposed method demonstrates improved feature mapping capabilities, however some false features are evident where container corners, were misinterpreted by the clustering algorithm as being point features.

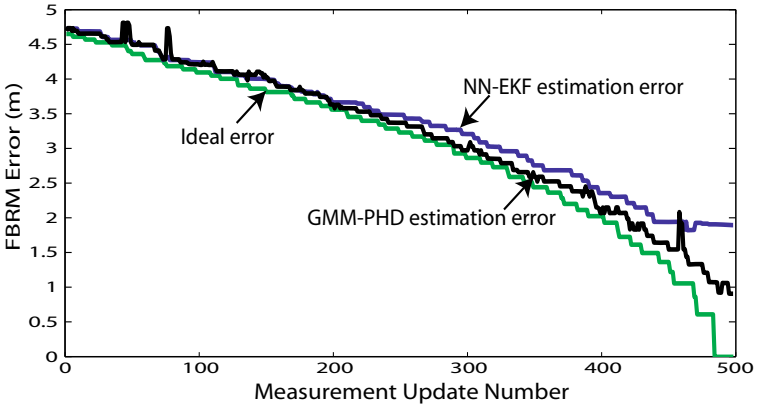


Fig. 4.12 FBRM estimation error at each measurement update in comparison to the total error given perfect detection and estimation once a feature enters the sensor's field of view.

number of features, however most current solutions do not jointly estimate the number of features and their locations. This is necessary as the classical vector-based approach cannot jointly encapsulate spatial and existence uncertainty. By adopting a random finite set map and measurement, the uncertainty in the map size and feature locations can be jointly propagated and estimated. This new approach therefore generalises the notion of Bayes optimality to the realistic situation of an *a priori* unknown number of features. Furthermore, the new framework allows for a mathematically consistent error metric to be readily adopted for the joint evaluation of FB map estimation error.

To demonstrate the applicability of the proposed framework to solve real FB autonomous problems, it was shown how the first order approximation, the PHD filter, could be implemented. Through simulated and real experimental trials, a proof-of-concept was presented. The proposed filter alleviates the need for data association and map dimensionality estimation filters, as the proposed theory incorporates these sources of uncertainty into the Bayesian recursion. Given its non-reliance on data association, the proposed approach may be more suited to applications with high clutter and/or a highly manoeuvring vehicle.

4.8 Bibliographical Remarks

Whilst most RM algorithms are occupancy-grid based, feature-based RM algorithms are typically adopted from feature-based SLAM algorithms, which have been modified to incorporate a known vehicle location. That is, examination of the posterior map estimate from a feature-based SLAM algorithm where the vehicle location uncertainty is zero. This has the effect of removing the cross-correlations between landmarks in the map and the vehicle state, a property which is also exploited in Rao-Blackwellised FastSLAM implementations [17].

The most common types of exteroceptive sensors deployed for FBRM applications are 2D range-bearing measuring devices. Such sensors' measurements are subject to often assumed Gaussian range and bearing measurement noise, which are typically used to update the time predicted map state through an EKF filtering framework [18] [15]. Due to the non-linearity of the measurement equation, Taylor approximations of the 'extended' filter are required.

In general, the FB autonomous framework is closely related to the multi-sensor, multi-target filtering problem, where the objective is to jointly estimate the time-varying number of targets and their states from sensor measurements in the presence of data association uncertainty, detection uncertainty, clutter and noise. The first systematic treatment of this problem using random set theory was conceived by Mahler in 1994 [40], which later developed into finite set statistics (FISST). Moreover, this treatment was

developed as part of a unified framework for data fusion using random set theory. [41] provides an excellent overview of FISST.

Mahler refined FISST to what he called generalised FISST and published this along with the derivation of the Probability Hypothesis Density (PHD) filter in 2003 [2]. Additionally, the relationship between FISST set derivatives and probability density (as Radon Nikodym derivatives of probability measures) for random finite sets was established in [4]. Further, generic sequential Monte Carlo (SMC) implementations of the multi-object Bayes filter and PHD filter, with convergence analysis was also provided. Numerous independent SMC solutions were also proposed [4], [42], [43], [44] and applied to a wide range of practical problems including feature point tracking in image sequences [45], tracking acoustic sources from Time Difference Of Arrival (TDOA) measurements [46], and tracking using bi-static radar data [47]. The reader is referred to [48] for a more complete survey of applications.

As indicated in Chapter 3, the FISST Bayes multi-object filter is generally intractable. Therefore, in 2000, Mahler proposed to approximate the multi-object Bayes recursion by propagating the Probability Hypothesis Density (PHD) of the posterior multi-object state [2], [3].

In 2005 a closed-form solution to the PHD recursion for the linear Gaussian multi-target model was published together with the Gaussian Mixture (GM) PHD filter for linear and mildly non-linear multi-target models [33]. While more restrictive than SMC approaches, Gaussian mixture implementations are much more efficient. Moreover, they obviate the need for clustering - an expensive step in the SMC implementation. Convergence results for the GM-PHD filter were established in [49]. Further extensions were introduced in 2006 through the cardinalised PHD (CPHD) recursion - a generalisation of the PHD recursion that jointly propagates the posterior PHD and the posterior distribution of the number of targets [50], [51]. A detailed treatment can be found in Mahler's recent book [3].

Since FB map estimation involves the estimation of a set of features using noisy, and cluttered measurements, it is evident that the mathematical foundation of multi-object tracking is related to the FBRM framework and thus provides motivation to re-examine the rationale and map estimation optimality of existing approaches. The following section therefore examines the underlying estimation theory fundamental to the problem of estimating the FB map.

Chapter 5
An RFS 'Brute Force' Formulation for Bayesian SLAM

5.1 Introduction

The feature-based (FB) SLAM scenario is a vehicle moving through an environment represented by an unknown number of features. The classical problem definition is one of "*a state estimation problem involving a variable number of dimensions*" [28]. The SLAM problem requires a robot to navigate in an unknown environment and use its suite of on board sensors to both construct a map and localise itself within that map without the use of any *a priori* information. Often, in the planar navigation context, a vehicle is assumed to acquire measurements of its surrounding environment using on board range-bearing measuring sensors. This requires joint estimates of the three dimensional robot pose (Cartesian x and y coordinates, as well as the heading angle θ), the number of features in the map as well as their two dimensional Euclidean coordinates. For a real world application, this should be performed incrementally as the robot manoeuvres about the environment. As the robot motion introduces error, coupled with a feature sensing error, both localisation and mapping must be performed simultaneously [8]. As mentioned in Chapter 2, for any given sensor, an FB decision is subject to detection and data association uncertainty, spurious measurements and measurement noise, as well as bias.

The majority of proposed algorithms, stemming from the seminal work of [8], adopt an augmented state containing both the vehicle pose estimate and the estimate of the map. It is important to note however, that the example discussed in [8] consisted of a map containing features of unity detection probability, assumed the measurement-feature association was known, and that the sensor reported no spurious measurements. With these strict assumptions, the Kalman based SLAM estimate is indeed Bayes optimal. This work was incorporated into multiple Kalman-based solutions to the SLAM problem [52].

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 79–96.
springerlink.com

The presence of detection uncertainty and spurious measurements have also been long acknowledged in the SLAM community, and subsequently feature initialisation and termination algorithms are frequently incorporated into the vector-based SLAM algorithm [52] (also shown in figure 1.1). This chapter again emphasises that these are required due to the inability of a vector representation to incorporate uncertainty in the *number* of dimensions, and highlights that such methods (which are independent of the filter recursion) compromise filter performance. As shown in this chapter, this can result in filter divergence and large mapping error, especially in scenarios of high clutter and large data association ambiguity. Through the re-formulation of the classical SLAM problem, and by explicitly incorporating the problem of a variable number of dimensions into the filter recursion, increased robustness in noisy scenarios will be demonstrated.

A finite set-valued measurement allows for the inclusion of spurious measurements directly into the measurement equation, as introduced in equation 2.10, which is then the union of the set-state dependant measurements (as is the case in the classical Bayesian SLAM formulations) and the set of spurious measurements. A finite set-valued map state can be constructed from the set union of the existing features and the new features which may appear in the map due to the motion of the robot, as introduced in equation 3.21.

This chapter subsequently casts the SLAM problem into a random set theoretic filtering problem that incorporates the joint estimation of the vehicle pose, feature number and corresponding feature locations. The term "Brute force" is used to describe the concept presented here, since each estimated feature is augmented with a hypothesised vehicle trajectory. While this is theoretically sound, and simple to implement, its computational burden becomes obvious when one considers that a single PHD is propagated, which requires many Gaussian functions to represent a single feature, since that feature can be augmented with many hypothesised trajectories. Although computationally intractable in any realistic environment with significant numbers of features, its implementation is included here to demonstrate a viable, and theoretically simple, RFS based SLAM solution. A more elegant, and computationally tractable solution, based on Rao-Blackwellisation, will be the subject of Chapter 6.

This chapter is organised as follows. A recap of the RFS Bayesian, SLAM formulation is given in Section 5.2, based on the RFS map transition function introduced in Chapter 3 and the RFS measurement model, introduced in Chapter 2. Section 5.3 introduces an augmented joint vehicle-map RFS to incorporate vehicle location uncertainty. The PHD of the augmented state recursion is then presented, and the PHD-SLAM filter is introduced. Using Gaussian noise assumptions, an extended-Kalman Gaussian Mixture (GM) implementation is developed in Section 5.4. This implementation accounts for the non-linearity in the measurement equation and jointly estimates the feature number in the map, their corresponding states and the vehicle pose. Importantly, this can be achieved without the need for explicit data

association decisions and/or feature management algorithms. Simulated mapping and pose estimation results are shown in Section 5.6 where the proposed GM-PHD SLAM filter is tested on simulated data which contains a large number of spurious measurements. Results show the efficacy of the proposed framework for solving the Bayesian SLAM problem. Real RFS based SLAM results in both outdoor and coastal environments (at sea) will be given for the formulation presented in chapter 6.

5.2 RFS Formulation of the Bayesian SLAM Problem

The starting point for the RFS, SLAM formulation is a recap of the map transition equation for RFSs, introduced in Chapter 2 (equation 2.10) and the RFS measurement model introduced in Chapter 3 (equation 3.21).

To incorporate the fact that new features enter the map, $\mathcal{M}_k$, with time, let the map state $\mathcal{M}_k$ be an RFS which evolves in time according to,

$$\mathcal{M}_k = \mathcal{M}_{k-1} \cup \mathcal{B}_k \tag{5.1}$$

comprising the set union of the previous RFS multi-feature map, $\mathcal{M}_{k-1}$ and the RFS of the new features at time k, $\mathcal{B}_k$. These sets are assumed mutually independent.

As in equation 2.10 (repeated below for convenience), to contend with the realistic situation of missed detections and clutter, the measurement is also modelled as an RFS. Given the vehicle state, X_k, and the map $\mathcal{M}_k$, the measurement consists of a set union,

$$\mathcal{Z}_k = \bigcup_{m \in \mathcal{M}_k} \mathcal{D}_k(m, X_k) \cup \mathcal{C}_k(X_k) \tag{5.2}$$

where $\mathcal{D}_k(m, X_k)$ is the RFS of the measurement generated by a feature at m and $\mathcal{C}_k(X_k)$ is the RFS of the spurious measurements at time k.

For each feature, $m \in \mathcal{M}_k$, and $z \in \mathcal{Z}_k$,

$$\mathcal{D}_k(m, X_k) = \{z\} \tag{5.3}$$

with probability density $P_D(m, X_k)g(z|m, X_k)$ and $\mathcal{D}_k(X_k, m) = \emptyset$ with probability $1 - P_D(m, X_k)$, where $P_D(m^{\mathfrak{m}_k}, X_k)$ is the probability of the sensor detecting the $\mathfrak{m}_k^{th}$ feature from pose X_k. Using Finite Set Statistics [3], the probability density that the sensor produces the measurement set $\mathcal{Z}_k$ given the vehicle state X_k and map $\mathcal{M}_k$ at time k is then given by [2]:

$$g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k) = \sum_{\mathcal{W} \subseteq \mathcal{Z}_k} g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k) g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W}) \tag{5.4}$$

with $g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k)$ denoting the density of the RFS of observations, $\mathcal{D}_k(m, X_k)$, generated from the features in the observed map $\mathcal{M}_k$ given the state of the vehicle, and $g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W})$ denoting the density of the clutter RFS, $\mathcal{C}_k$. $g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k)$ describes the likelihood of receiving a measurement from the elements of the set-valued map which incorporates detection uncertainty and measurement noises. $g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W})$ models the spurious measurement rate of the sensor and is typically *a priori* assigned [18] [14]. Expanding the multi-target RFS Bayes recursion of [3] to include the vehicle state, the optimal Bayesian SLAM filter then jointly propagates the set of features and the vehicle location according to,

$$p_k(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0) = \frac{g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k)p_{k|k-1}(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0)}{\int\int g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k)p_{k|k-1}(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0)dX_k\mu(d\mathcal{M}_k)} \quad (5.5)$$

where,

$$p_{k|k-1}(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0) = \int f_X(X_k|X_{k-1}, U_{k-1})p_{k-1}(X_{0:k-1}, \mathcal{M}_{k-1}|\mathcal{Z}_{0:k-1}, U_{0:k-2}, X_0)dX_{k-1} \quad (5.6)$$

and μ is a reference measure on the space of features. As noted in Chapter 2, in a direct implementation of the vector-based Bayesian SLAM recursion of equation 2.15, computational complexities and multiple integrals generally lead to intractable solutions. Therefore the PHD approximation introduced in Chapter 3 is used.

5.3 The 'Brute Force' PHD SLAM Filter

The RFS formulation of the general Bayesian SLAM problem was described in the previous section. This section modifies the formulation to admit a compact PHD solution. The key to the approach is the introduction of a new RFS state, $\mathcal{Y}$, which comprises the unordered set of $\mathfrak{n}$ elements, ζ, such that at time k,

$$\mathcal{Y}_k = \{\zeta_k^1, \zeta_k^2, \cdots, \zeta_k^{\mathfrak{n}_k}\} \quad (5.7)$$

where each ζ_k comprises a map state, m_k, conditioned on a vehicle trajectory, $X_{0:k}$. $\mathfrak{n}_k$ will be defined below. Conditioning each feature state, m, on the history of vehicle poses $X_{0:k}$, introduces a conditional independence between feature measurements allowing the joint states, ζ_k to be independently propagated through the PHD SLAM framework [17]. Following the introduction of the PHD SLAM filter in Section 3.4, a "Brute force" PHD SLAM filter can then be derived if, in equations 3.23 and 3.24,

$$\Gamma_k \longrightarrow \zeta_k. \quad (5.8)$$

Each element of $\mathcal{Y}_k$, evolves in time according to the transition $f(\zeta_k|\zeta_{k-1}, U_{k-1})$ and, if the feature in ζ_k is detected by the sensor at the conditioning pose X_k, a measurement z is generated with likelihood $P_D(\zeta_k)g_k(z|\zeta_k)$. The precise reason why the conditioning of the feature state m, with each of the N hypothesised vehicle trajectories to form the state ζ_k, is permitted under the RFS framework is due to Campbell's theorem, as explained in Appendix A. Therefore, let the vehicle state be sampled by N particles, to produce $N \times \mathfrak{m}_k = \mathfrak{n}_k$ augmented states, ζ_k. Given a set of augmented features, ζ_k, joint estimates of the number of features, their locations, as well as the vehicle state, can then be obtained. Hence this SLAM implementation estimates a potentially extremely large, single PHD (intensity function) containing representations of each feature, conditioned on each vehicle trajectory. Referring to the GM PHDs of figures 3.2 and 3.3, in the "Brute force" SLAM case, multiple Gaussians are necessary to represent each single feature, since each must be conditioned on every one of the N hypothesised trajectories. In contrast to the FBRM case of Chapter 4, computationally, a copy of each hypothesised vehicle trajectory (rather than the single known vehicle pose) is necessary to condition each feature. The PHD-SLAM recursion can then be formulated in terms of the state elements ζ_k. The prediction of the state intensity $v_{k|k-1}(\zeta_k)$ is then given by

$$\begin{aligned} v_{k|k-1}(\zeta_k) &= \int f(\zeta_k|\zeta_{k-1}, U_{k-1}) v_{k-1}(\zeta_{k-1}) d\zeta_{k-1} + b_k(\zeta_k|X_k) \\ &= \int f(\zeta_k|X_{k-1}, m_k, U_{k-1}) v_{k-1}(X_{k-1}, m_k) dX_{k-1} + b_k(\zeta_k|X_k) \end{aligned} \tag{5.9}$$

where $f(\zeta_k|\zeta_{k-1}, U_{k-1})$ incorporates both the assumed vehicle and feature predicted transition functions. As before, $b_k(\zeta_k|X_k)$ models the new features entering the vehicles FoV. The corrector equation for the combined state, SLAM intensity function is

$$\begin{aligned} v_k(\zeta_k) = v_{k|k-1}(\zeta_k) \Bigg[& 1 - P_D(\zeta_k) \\ & + \sum_{z \in \mathcal{Z}_k} \frac{P_D(\zeta_k) g_k(z|\zeta_k)}{c_k(z|X_k) + \int P_D(\xi) g_k(z|\xi) v_{k|k-1}(\xi) d\xi} \Bigg] \end{aligned} \tag{5.10}$$

where again at time k,

$$\begin{aligned} b_k(\zeta_k|X_k) &= \text{intensity of the new feature RFS } \mathcal{B}_k, \\ g_k(z|\zeta_k) &= \text{likelihood of } z, \text{ given the joint state } \zeta_k, \\ P_D(\zeta_k) &= \text{probability of detection of the feature in} \\ &\quad \zeta_k, \text{ given the pose in } \zeta_k, \\ c_k(z|X_k) &= \text{intensity of the clutter RFS } \mathcal{C}_k(X_k). \end{aligned}$$

In [33], Gaussian noise assumptions were used to obtain closed form solutions for the target tracking PHD filter. Similarly for the PHD-SLAM filter, GM techniques can be applied to solve the PHD-SLAM joint intensity recursion of equation 5.11. It is also possible to use a particle-based approach [4], however, for mildly non-linear problems the Gaussian mixture approach is much more efficient. The following section thus presents a GM implementation of the PHD-SLAM filter, while a particle based version is left until Chapter 6.

5.4 Gaussian Mixture (GM) PHD-SLAM

Let the joint intensity, $v_{k-1}(\zeta_{k-1})$, at time $k-1$ be a Gaussian mixture of the form,

$$v_{k-1}(\zeta_{k-1}) = \sum_{i=1}^{N \times J_{k-1}} w_{k-1}^{(i)} \mathcal{N}\big(\zeta; \mu_{k-1}^{(i)}, P_{k-1}^{(i)}\big) \tag{5.11}$$

composed of $N \times J_{k-1}$ Gaussians, with $w_{k-1}^{(i)}$, $\mu_{k-1}^{(i)}$ and $P_{k-1}^{(i)}$ being their corresponding weights, means and covariances respectively. Note that the weight, $w_{k-1}^{(i)}$ is a weight on *both* a particular feature state, m, and a particular vehicle pose $X_{k-1}^{(n)}$, i.e. on the joint state $\zeta_{k-1}^{(i)}$.

Since the map is assumed static, the joint state transition density is

$$f_X(X_k^{(n)} | X_{k-1}^{(n)}, U_{k-1}) \delta(m_k - m_{k-1}) \tag{5.12}$$

where $X_{k-1}^{(n)}$ is one of N vehicle pose particles at time $k-1$ and $\delta(m_k - m_{k-1})$ is a Dirac delta function to mathematically incorporate the fact that the map must remain static in this case. Let the new feature intensity at time k also be a Gaussian mixture,

$$b_k = \sum_{i=1}^{N \times J_{b,k}} w_{b,k}^{(i)} \mathcal{N}\big(\zeta; \mu_{b,k}^{(i)}, P_{b,k}^{(i)}\big) \tag{5.13}$$

where $w_{b,k}^{(i)}$, $\mu_{b,k}^{(i)}$ and $P_{b,k}^{(i)}$ determine the shape of the new feature GM proposal density according to a chosen strategy. This is analogous to the proposal distribution in the particle filter [17] and provides an initial estimate of the new features entering the map (see Section 5.4.1). Again, each new feature density component is conditioned on each predicted vehicle pose particle, $X_k^{(n)}$ to form the $N \times J_{b,k}$ components of the GM new feature intensity. That is, for each hypothesised vehicle trajectory, a set of $J_{b,k}$ Gaussian components are initialised. Recalling that each Gaussian models the feature and vehicle trajectory, $J_{b,k}$ copies of a given trajectory are distributed to $J_{b,k}$ features. This is in contrast to classical Rao-Blackwellised approaches, as presented in the following chapter, where only a single copy of each trajectory is required.

However, in the PHD-SLAM filter, this conditioning allows for each joint state (feature and trajectory) to be independently estimated via the PHD framework. Therefore, the predicted intensity, $v_{k|k-1}(\zeta_k)$ is also a Gaussian mixture

$$v_{k|k-1}(\zeta_k) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} \mathcal{N}\big(\zeta; \mu_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}\big) \tag{5.14}$$

where, $J_{k|k-1} = N(J_{b,k} + J_{k-1})$ and,

$$\left.\begin{array}{l} w_{k|k-1}^{(i)} = w_{k-1}^{(i)} \\ \mu_{k|k-1}^{(i)} = \mu_{k|k-1}^{(i)} \\ P_{k|k-1}^{(i)} = P_{k-1}^{(i)} \end{array}\right\} \text{for i} \in \{1, \ldots, N{\times}J_{k-1}\} \text{ (previously observed features)}$$

$$\left.\begin{array}{l} w_{k|k-1}^{(i)} = w_{b,k}^{(i)} \\ \mu_{k|k-1}^{(i)} = \mu_{b,k}^{(i)} \\ P_{k|k-1}^{(i)} = P_{b,k}^{(i)} \end{array}\right\} \text{for i} \in \{N{\times}J_{k-1}+1, \ldots, N{\times}J_{b,k}\} \text{ (newly observed features).}$$

Assuming a Gaussian measurement likelihood, $g(z|\zeta_k)$, analysis of equation 5.11 shows that the joint posterior intensity, $v_k(\zeta_k)$, is consequently also a Gaussian mixture,

$$v_k(\zeta_k) = v_{k|k-1}(\zeta_k)\left[1 - P_D(\zeta_k) + \sum_{z \in \mathcal{Z}_k} \sum_{i=1}^{J_{k|k-1}} v_{G,k}^{(i)}(z|\zeta_k)\right] \tag{5.15}$$

where,

$$v_{G,k}^{(i)}(z|\zeta_k) = w_k^{(i)} \mathcal{N}(\zeta; \mu_{k|k}^{(i)}, P_{k|k}^{(i)}) \tag{5.16}$$

$$w_k^{(i)} = \frac{P_D(\zeta_k) w_{k|k-1}^{(i)} q^{(i)}(z, \zeta_k)}{c_k(z) + \sum_{j=1}^{J_{k|k-1}} P_D(\zeta_k) w_{k|k-1}^{(j)} q^{(j)}(z, \zeta_k)} \tag{5.17}$$

with, $q^{(i)}(z, \zeta_k) = \mathcal{N}\big(z;\ H_k \mu_{k|k-1}^{(i)},\ S_k^{(i)}\big)$. The component distributions are represented by their first and second moments obtained from the standard EKF update equations,

$$S_k^{(i)} = R_k + \nabla H_k P_{k|k-1}^{(i)} \nabla H_k^T \tag{5.18}$$

$$K_k^{(i)} = P_{k|k-1}^{(i)} \nabla H_k^T [S_k^{(i)}]^{-1} \tag{5.19}$$

$$\mu_{k|k}^{(i)} = \mu_{k|k-1}^{(i)} + K_k^{(i)}(z - H_k \mu_{k|k-1}^{(i)}) \tag{5.20}$$

$$P_{k|k}^{(i)} = [I - K_k^{(i)} \nabla H_k] P_{k|k-1}^{(i)} \tag{5.21}$$

with ∇H_k being the Jacobian of the measurement equation with respect to the features estimated location. As stated previously, the clutter RFS, $\mathcal{C}_k$, is assumed Poisson distributed [18], [14] in number and uniformly spaced over the sensor surveillance region. Therefore the clutter intensity is,

$$c_k(z) = \lambda_c V \mathcal{U}(z) \tag{5.22}$$

where λ_c is the average number of clutter returns, V is the volume of the sensor's surveillance region and $\mathcal{U}(z)$ denotes a uniform distribution over range and bearing. As stated in Section 5.1, the Gaussian number growth of this formulation becomes very large and hence Gaussian pruning and merging methods are used as in [33].

5.4.1 The SLAM New Feature Proposal Strategy

The new feature proposal density, equation 5.13, is similar to the proposal function used in particle filters, and is used to give some *a priori* information to the filter about where features are likely to appear in the map. In SLAM, with no *a priori* information, b_k, may be uniformly distributed in a non-informative manner about the space of features (analogous to the prior map used in occupancy grid algorithms). However, in this work the feature birth proposal at time k is chosen to be the set of measurements at time k–1, $\mathcal{Z}_{k-1}$. The sum $\sum_{i=1}^{N \times J_{b,k}} w_{b,k}^{(i)}$ then gives an estimate of the expected number of new features to appear at time k. The components of the Gaussian mixture used to form b_k are determined in exactly the same way as for the FBRM case, described in Section 4.5.2.1.

5.5 Brute Force SLAM Pseudo-code

This section details the pseudo-code of the proposed PHD-SLAM algorithm. Table 5.1 presents the birth proposal algorithm which accommodates new features entering the map as well as aiding particle diversity since each new potential feature is seeded with a corresponding potential vehicle pose. Table 5.2 outlines in the prediction module while Tables 5.3 and 5.3 describe the update module. Vehicle pose and map estimation are achieved via the process detailed in Table 5.4.

Table 5.1 PHD-SLAM-Birth

Algorithm PHD-SLAM-Birth$\big(\mathcal{Z}_{k-1}, v_{k-1}(\zeta|\mathcal{Z}_{k-1})\big)$
// Generate eqn.(5.13). Note: Any arbitrary strategy is valid.
// For each measurement
1. for i = 1 to $\mathfrak{z}_{k\text{-}1}$ do
// For each particle
2. for j = 1 to J_{k-1} do
// initialise the feature mean
3. $\nu_{b,k} = h^{-1}(z_{k-1}^{(i)}, X_{k-1}^{(j)})$
// initialise the concatenated state mean
4. $\mu_{b,k}^{((i-1)\times J_{k-1}+j)} = [X_{k-1}^{(j)}\ \nu_{b,k}]$
// initialise the covariance
5. $P_{b,k}^{((i-1)\times J_{k-1}+j)} = h^{'}(\nu_{b,k}, X_{k-1}^{(j)})$
// initialise the weight
6. $\omega_{b,k}^{((i-1)\times J_{k-1}+j)} = \alpha$
7. end for
8. end for
// Set the number of birth components
9. $J_{b,k} = \mathfrak{z}_{k\text{-}1} \times J_{k-1}$
// Construct birth PHD
10. $b_k(\zeta|\mathcal{Z}_{k-1}) = \{\mu_{b,k}^{(i)}, P_{b,k}^{(i)}, \omega_{b,k}^{(i)}\}_{i=1}^{J_{b,k}}$
11. return $\big(\ b_k(\zeta|\mathcal{Z}_{k-1})\ \big)$

5.6 Algorithm Performance

This section analyses the performance of the proposed GM-PHD SLAM filter in a simulated environment, and compares it to a FastSLAM implementation using maximum likelihood data association decisions and Log-Odds feature management [17]. The vehicle is assumed to be travelling at $3ms^{-1}$ while subject to velocity and steering input noises of $1ms^{-1}$ and 5^o respectively. Only 10 particle samples are used for both filters and both filters receive the same noisy input samples and sensor measurements. Two simulated comparisons are performed in an 'Easy' and 'Difficult scenario'. For the 'easy' scenario, the clutter parameter, $\lambda_c = 0$ m^{-2}, feature detection probability is 0.95, and the measurement noises are $0.25m$ in range and 0.5^o in bearing. For the 'difficult' scenario, $\lambda_c = 10$ m^{-2} (i.e. 10 false alarms occur for every square metre of area within the sensors FoV), feature detection probability is again 0.95 and the measurements noises are set at $12.5m$ in range and 25^o in bearing. The effect of the artificially large measurement noises are to give the appearance of closely spaced features, hampering data association decisions and feature map building.

Table 5.2 PHD-SLAM-Predict

Algorithm PHD-SLAM-Predict $\left(v_{k-1}(\zeta|\mathcal{Z}_{k-1}), U_{k-1}\right)$
// Generate eqn.(5.14)
// Generate map birth components
1. GMM-PHD-FBRM-Birth$(\mathcal{Z}_{k-1}, v_{k-1}(\zeta|\mathcal{Z}_{k-1}))$
// append PHD with birth components
2. for i = 1 to $J_{b,k}$ do
3. $\quad \omega_{k-1}^{(J_{k-1}+i)} = \omega_{b,k}^{(i)}$
4. $\quad \mu_{k-1}^{(J_{k-1}+i)} = \mu_{b,k}^{(i)}$
5. $\quad P_{k-1}^{(J_{k-1}+i)} = P_{b,k}^{(i)}$
6. end for
// increment component counter
7. $J_{k|k-1} = J_{k-1} + J_{b,k}$
8. for i = 1 to $J_{k|k-1}$
// sample a pose from the vehicle model
9. $\quad X_{k|k-1}^{(i)} \sim f_X(X_k^{(i)}|X_{k-1}^{(i)}, U_{k-1})$
// static map assumption
10. $\quad \nu_{k|k-1}^{(i)} = \nu_{k-1}^{(i)}$
11. $\quad \mu_{k|k-1}^{(i)} = [X_{k|k-1}^{(i)} \ \nu_{k|k-1}^{(i)}]$
12. $\quad P_{k|k-1}^{(i)} = P_{k-1}^{(i)}$
13. $\quad \omega_{k|k-1}^{(i)} = \omega_{k-1}^{(i)}$
14. end for
// The predicted SLAM PHD
15. $v_{k|k-1}(\zeta|\mathcal{Z}_{k-1}) = \{\mu_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}, \omega_{k|k-1}^{(i)}\}_{i=1}^{J_{k|k-1}}$
16. return$\left(v_{k|k-1}(\zeta|\mathcal{Z}_{k-1})\right)$

Figure 5.1 shows the estimated vehicle trajectory and corresponding feature map from both filters, in the case of the 'Easy scenario'.

Both results compare well with ground truth (green). This result verifies the accuracy of the proposed PHD-SLAM filter, in its ability to jointly estimate the vehicle trajectory, the number of features, and their corresponding location, without the need for external data association and feature map management methods, as are required by FastSLAM (and other vector-based solutions).

The missed feature declaration highlights an issue of the proposed method with respect to $P_D(\zeta_k)$. In the presented implementation, this is simply a binary function which has an assumed value of 0.95 if the feature is predicted to be within the sensor FoV, and 0 if it is not. Vehicle and feature estimation uncertainty may result in a feature erroneously being hypothesised of being within the FoV, or vice-versa. If the proposed filter then receives a

Table 5.3 PHD-SLAM-Update

Algorithm PHD-SLAM-Update$\left(\mathcal{Z}_k, v_{k|k-1}(\zeta|\mathcal{Z}_{k-1})\right)$

// Initialise number of Gaussian components

1. L = 0

// Missed detections and Update Terms

2. for i = 1 to $J_{k|k-1}$ do

// Increment component counter

3. L = L+1

// Updated feature equals predicted feature

4. $\nu_k = \nu_{k|k-1}^{(i)}, P_k^{(L)} = P_{k|k-1}^{(i)}$

// Updated joint mean

5. $\mu_k^{(L)} = [\nu_k \; X_{k|k-1}^{(i)}]$

// weight decreased

6. $\omega_k^{(L)} = (1 - P_D)\omega_{k|k-1}^{(i)}$

// measurement prediction

7. $z_{k|k-1}^{(i)} = h(\nu_{k|k-1}^{(i)}, X_{k|k-1}^{(i)})$

3// Calculate Jacobian

8. $H = h^{'}(\nu_{k|k-1}^{(i)}, X_{k|k-1}^{(i)})$

// Innovation Covariance of (5.18)

9. $S_k^{(i)} = HP_{k|k-1}^{(i)}[H]^T + R$

// Kalman Gain of (5.19)

10. $K_k^{(i)} = P_{k|k-1}^{(i)}[H]^T[S_k^{(i)}]^{-1}$

// Updated Covariance of (5.21)

11. $P_{U,k}^{(i)} = [I - K_k^{(i)}H]P_{k|k-1}^{(i)}$

12. end for

// For each measurement

13. for i = 1 to $\mathfrak{z}_k$ do

// For each component

14. for j = 1 to $J_{k|k-1}$ do

// Updated map component mean

15. $\nu_k = \nu_{k|k-1}^{(j)} + K_k^{(j)}(z_k^{(i)} - z_{k|k-1}^{(j)})$

// Updated joint mean

16. $\mu_k^{(L+j)} = [\nu_k \; X_{k|k-1}^{(i)}]$

// Updated GMM component covariance

17. $P_k^{(L+j)} = P_{U,k}^{(j)}$

// Numerator of (5.16)

18. $\tau^{(j)} = P_D\omega_{k|k-1}^{(j)}|2\pi S_k^{(j)}|^{-0.5}$
$\times \exp\left((z_k^{(i)} - z_{k|k-1}^{(j)})[S_k^{(j)}]^{-1}(z_k^{(i)} - z_{k|k-1}^{(j)})\right)$

19. end for

Table 5.3 (*Continued*)

// For each map PHD component
20. for j = 1 to $J_{k|k-1}$ do
// denominator of (5.16)
21. $\omega_k^{(L+j)} = \tau^{(j)}/(c(z) + \sum_{l=1}^{J_{k|k-1}} \tau^{(l)})$
22. end for
23. L = L+$J_{k|k-1}$
24. end for
// Number of components in updated GMM
25. J_k = L
// The updated map PHD
26. $v_k(\zeta|\mathcal{Z}_k) = \{\mu_k^{(i)}, P_k^{(i)}, \omega_k^{(i)}\}_{i=1}^{J_k}$
27. return$\big(v_k(\zeta|\mathcal{Z}_k)\big)$

Table 5.4 PHD-SLAM-Estimate

Algorithm PHD-SLAM-Estimate$\big(v_k(\zeta|\mathcal{Z}_k), T_{\text{feature}}\big)$
// Initialise the map estimate
1. $\hat{\mathcal{M}}_k = \emptyset$
2. $\Omega_k = 0$
3. for i = 1 to J_k do
4. $\Omega_k = \Omega_k + \omega_k^{(i)}$
5. if $\omega_k^{(i)} > T_{\text{feature}}$
// concatenate estimate
6. $\hat{\mathcal{M}}_k = [\hat{\mathcal{M}}_k \ \nu_k^{(i)}]$
7. end if
8. end for
// expected pose
9. $\hat{X}_k = \dfrac{1}{\Omega_k} \sum_{i=1}^{J_k} \omega_k^{(i)} X_k^{(i)}$
10. return $\big(\hat{X}_k, \hat{\mathcal{M}}_k\big)$

measurement contrary to the prediction, the resulting feature weight may be detrimentally reduced, and a missed feature declaration may occur. The uncertainty in the estimated sensor FoV is not considered in this implementation.

The raw measurements as well as the final posterior joint estimate of both filters for the 'hard' scenario are presented respectively in figures 5.2 and 5.3. As can be seen, the FastSLAM filter shows divergence of the estimated vehicle trajectory from its true value, as well as many falsely estimated map

features. The proposed filter however, displays dramatically reduced feature-based mapping error in the face of large data association uncertainty and large quantities of spurious measurements, reporting only a single false feature and a single missed feature over the entire run. This is expected, as the clutter rate is integrated directly into the filter recursion in an optimal manner and feature management is performed jointly with feature and vehicle location estimation. The key to this vast difference in performance in cluttered environments, is evident in figure 5.4. The figure shows the estimated number of features in the map over time, for both the discussed filters, as well as the number of false measurements at each time instant. The proposed filter accurately tracks the true number of features over time, whereas the FastSLAM filter deviates drastically in the face of the challenging spurious measurements and data association ambiguities. Estimating the true number of state dimensions influences the accuracy of the overall SLAM filter. The estimated vehicle trajectory also displays less error than that of the FastSLAM approach. Similarly to FastSLAM, increased trajectory estimation accuracy may be possible by increasing the number of pose samples. Figure 5.5 compares the estimated vehicle heading over the course of the test, highlighting the increased accuracy of the proposed filter.

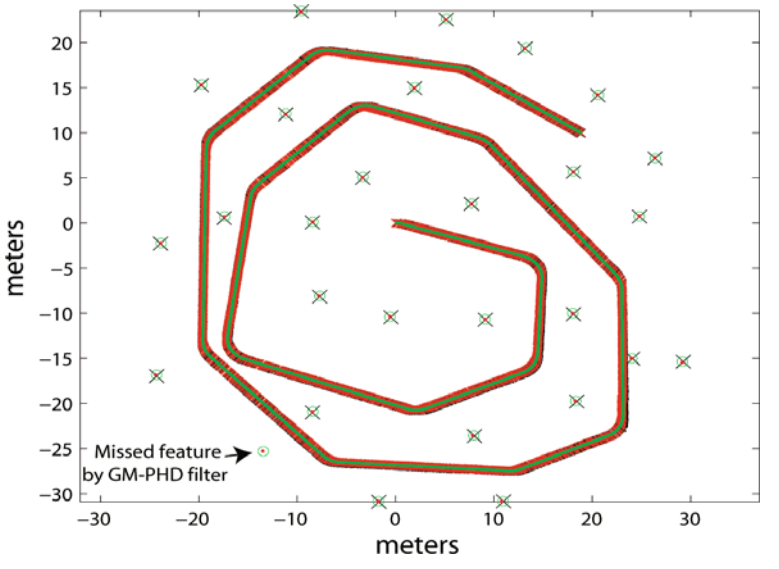


Fig. 5.1 Comparative results for the proposed GM-PHD SLAM filter (black) and that of FastSLAM (red), compared to ground truth (green).

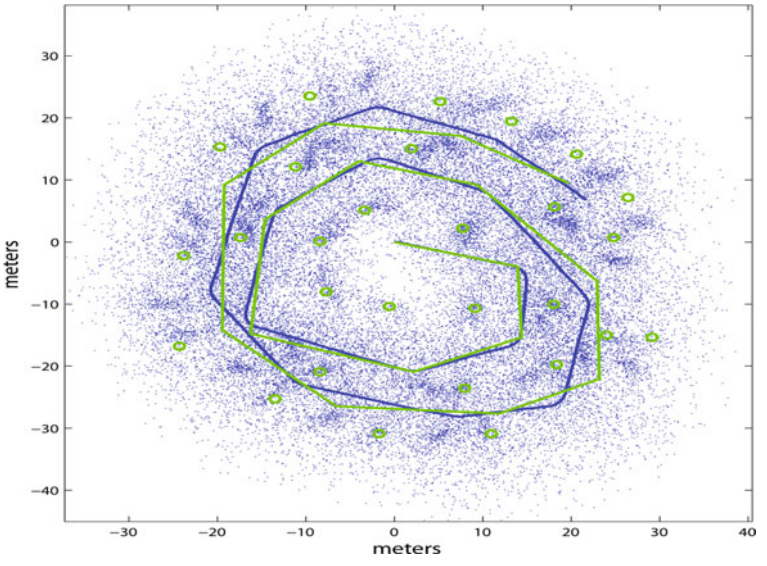


Fig. 5.2 The predicted vehicle trajectory (blue) along with the raw sensor measurements for the 'hard' scenario, at a clutter density of $0.03m^{-2}$. Also superimposed are the ground truth trajectory and feature map (black crosses).

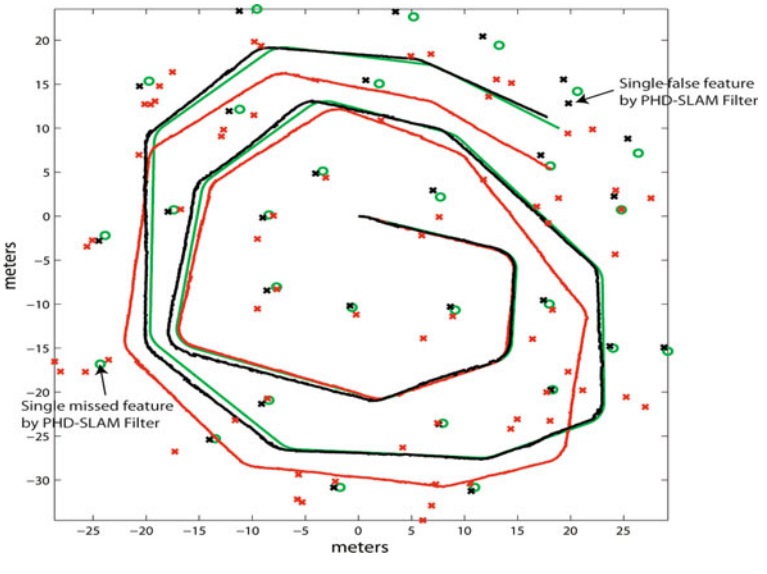


Fig. 5.3 The estimated trajectories of the GM-PHD SLAM filter (black) and that of FastSLAM (red). Estimated feature locations (crosses) are also shown with the true features (green circles)

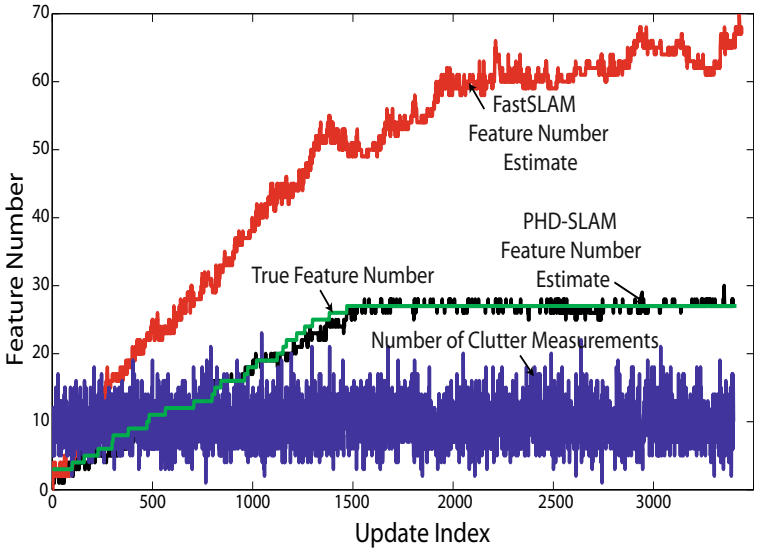


Fig. 5.4 The estimated feature number for the proposed (black) and FastSLAM (red) filters. The green curve shows the ideal estimate, which grows as more features enter the FoV of the sensor, during vehicle motion. The blue curve shows the randomly generated number of clutter measurements.

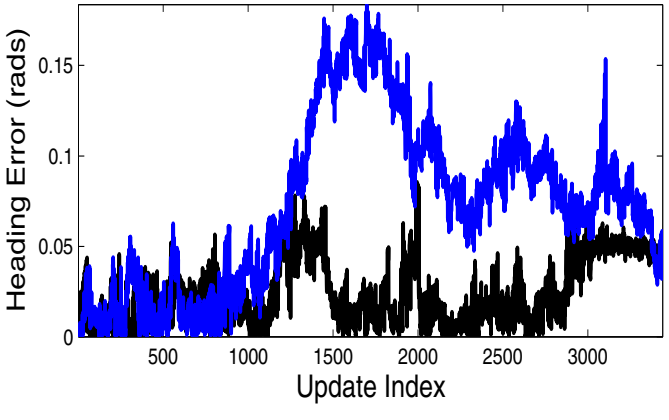


Fig. 5.5 The error in vehicle heading estimate for the proposed (black) and FastSLAM (blue) filters.

5.6.1 A Note on Computational Complexity

As is evident from the update of equation 5.15, the proposed algorithm scales linearly with $\mathcal{O}(NJ_{k|k-1}\mathfrak{Z}_k)$, which equals that of a naive FastSLAM implementation. Future work will address reducing this to a Log order complexity of the number predicted map states $J_{k|k-1}$. The presented results illustrate the effectiveness of the new finite-set based SLAM framework, and the proposed GM-PHD implementation, when compared to vector-based solutions which fail to jointly consider the entire system uncertainty.

5.7 Summary

This chapter outlined a 'Brute force' formulation of the Bayesian SLAM problem, using random set theory. The set theoretic approach allows for detection uncertainty, spurious measurements as well as data association uncertainty to be incorporated directly into the filter recursion. This is in contrast to vector-based SLAM which requires additional algorithms and pre/post processing to solve the data association problem prior to filter update, and to extract estimates of the number of features present in the map. These are necessary as such sources of uncertainty are not considered in the classical vector-based measurement model and subsequent filter recursion. Previous Bayesian SLAM solutions also lack a concept of Bayesian optimality as the variable dimensionality problem is not jointly considered.

Propagating the first order statistic of the random set (the probability hypothesis density) is a common method of reducing the computational requirements of implementing the set-valued Bayesian recursion. By augmenting the feature state with a history of vehicle poses, conditional independencies between the features and the vehicle state are introduced. The joint vehicle feature RFS was shown to maintain the necessary Poisson assumptions for application of the tracking based PHD recursion for the PHD-SLAM problem. A Gaussian mixture implementation of the PHD-SLAM filter was outlined assuming a Gaussian system with non-linear measurement and process models. The proposed finite-set filter was compared to a FastSLAM implementation with explicit (per particle) data association decisions and feature management methods. Results show the proposed filter performing similarly to FastSLAM in an 'easy' scenario, and considerably outperforming it in a 'hard' scenario.

5.8 Bibliographical Remarks

As explained in Chapter 2, it is critical for data association algorithms not to use falsely declared targets in their hypothesis decision making process. In [17], This is a well known fact in the SLAM community, which has been addressed by various researchers. For example, M. Montemerlo *et. al.* use methods from occupancy-grid based robotic mapping [12] to estimate the number of landmarks in the map state. A probabilistic evidence for landmark existence is derived from the measurement and propagated through a Log-Odds discrete Bayes recursion. This discrete recursion is independent of the main RM state estimation filter and is effectively a post-processing of the map state estimate at each time step. The output of the maximum likelihood data association decision is incorporated into the existence update. If the landmark is not associated (and therefore assumed undetected) the existence posterior is reduced, thus inherently assuming that the probability of detection of all the landmarks is unity. Based on a predefined-defined log-odds threshold, landmarks are then either added or removed from the map state. The number of landmarks in the map state at any given time then gives an estimate of the number of landmarks in the map, as *every* component of the map state is assumed to be a valid landmark. However, since in reality, landmarks have non-unity detection probabilities, a missed-detection does not always indicate the non-presence of the landmark.

Another method of landmark management was introduced by D. Makarsov in [14] and used in [1] which outlined the so-called 'Geometric feature track quality' measure of landmark existence. This measure is inversely proportional to the innovation between a predicted landmark and the measurements and is therefore only updated when measurements are made (and associated). It does not consider the frequent sensor errors in terms of detection uncertainty and spurious measurements. Other techniques [18] simply use the number of successive associations over a fixed set of measurement frames which requires both low clutter rates and correct association hypotheses. Again, such methods are effectively post/pre-processing techniques which are independent of the state estimating filter recursion.

Sequential Monte Carlo (SMC) solutions to Bayesian SLAM also gained popularity [28] through the use of Rao-Blackwellised particle filters. FastSLAM [17] displayed impressive results by sampling over the vehicle trajectory and applying independent Kalman filters to estimate the location of the hypothesised map features. By conditioning the map estimates on the history of vehicle poses, a conditional measurement independence is invoked which allows the correlations introduced in [8] to be discarded.

A Gaussian mixture solution to the Bayesian SLAM problem was also described in [53] which approximated both the transition and measurement densities as Gaussian mixtures and propagated the joint state through a Bayes recursion.

Extensions to of the PHD filter to SLAM were first given in [54] [55], upon which this chapter is based.

Chapter 6
Rao-Blackwellised RFS Bayesian SLAM

6.1 Introduction

This chapter proposes an alternative Bayesian framework for feature-based SLAM, again in the general case of uncertain feature number and data association. As in Chapter 5, a first order solution, coined the probability hypothesis density (PHD) SLAM filter, is used, which jointly propagates the posterior PHD of the map and the posterior distribution of the vehicle trajectory. In this chapter however, a Rao-Blackwellised (RB) implementation of the PHD-SLAM filter is proposed based on the GM PHD filter for the map and a particle filter for the vehicle trajectory, with initial results presented in [56] and further refinements in [57].

A tractable PHD approximation to the SLAM problem is derived, which propagates the posterior PHDs of *multiple trajectory-conditioned maps* and the posterior distribution of the trajectory of the vehicle. Furthermore, this approach to SLAM admits the concept of an 'expected' map via the PHD construct, which is not available in previous SLAM approaches.

The chapter is organised as follows. Section 6.2 discusses the factorised RFS SLAM recursion, in which the posterior density of the map, conditioned on the trajectory, and the trajectory itself can be propagated jointly. The RFS framework is then applied to this factorised solution, where it is demonstrated that subtle differences, regarding the use of sets, make a direct, naive implementation of FastSLAM to the RFS problem inappropriate. In particular, the likelihood of the measurement, conditioned on the trajectory, which is necessary for the calculation of the particle weights, cannot be approximated under an EKF framework, as in FastSLAM [58]. Solutions, which give a closed form solution to this problem, are presented in this section. Section 6.3 outlines a Rao-Blackwellised implementation of the PHD-SLAM filter. The necessary steps to implement the PHD filter for the estimation of the map and the vehicle trajectory are given, along with pseudo code. Section 6.4 presents and discusses the Rao-Blackwellised RFS SLAM performance.

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 97–126.
springerlink.com

Demonstrations of simulated examples are given, due to the simplicity of generating both trajectory and map ground truth values, necessary for true performance evaluation assessment. This is followed by an implementation with real, short range, millimetre wave (MMW) radar data, and a mobile robot platform, in a car park environment. The advantages of these sensors over other devices such as laser range finders is discussed in [16]. Data sets from the radar are recorded, along with odometry and single axis gyro rate data, from a moving vehicle. Comparisons are made with classical, vector based, EKF SLAM which utilises the Joint Compatibility Branch and Bound (JCBB) [39] data association method and FastSLAM with Multiple Hypothesis (MH) data association. Further comparative results, in a much larger scenario, where accurate SLAM performance in the presence of high clutter is essential, are demonstrated at sea, in a coastal environment, using an "Autonomous kayak" [59] as the vehicle and a commercially available X-Band radar. The performance improvement, in the presence of clutter is clearly demonstrated. Comparisons and discussions of the computational complexity of the algorithms is also given.

6.2 The Rao-Blackwellised (RB) PHD-SLAM Filter

Since the full RFS-SLAM Bayes filter of equations 2.14 and 2.15 is numerically intractable, it is again necessary to look for tractable but principled approximations. This section derives a recursion that jointly propagates the posterior PHD of the map and the posterior density of the vehicle trajectory. Analogous to FastSLAM, the RFS-SLAM recursion can be factorised as shown in Section 6.2.1. Section 6.2.2 discusses the PHD estimator in the context of this factorised recursion, Section 6.2.3 addresses the PHD representation of the map component only while Section 6.2.4 extends this algorithm to perform SLAM.

6.2.1 The Factorised RFS-SLAM Recursion

Using standard conditional probability, the joint posterior RFS-SLAM density of equation 2.15 can be decomposed as,

$$p_k(\mathcal{M}_k, X_{1:k}|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0) = \\ p_k(X_{1:k}|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0)p_k(\mathcal{M}_k|\mathcal{Z}_{0:k}, X_{0:k}). \quad (6.1)$$

Thus, the recursion for the joint RFS map-trajectory posterior density according to equation 2.15 is equivalent to jointly propagating the posterior density of the map conditioned on the trajectory and the posterior density of the trajectory. In this section, as before, for compactness,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) = p_{k|k-1}(\mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) \tag{6.2}$$

$$p_k(\mathcal{M}_k|X_{0:k}) = p_k(\mathcal{M}_k|\mathcal{Z}_{0:k}, X_{0:k}) \tag{6.3}$$

$$p_k(X_{1:k}) = p_k(X_{1:k}|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0) \tag{6.4}$$

and it follows that,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) = \int f_{\mathcal{M}}(\mathcal{M}_k|\mathcal{M}_{k-1}, X_k) p_{k-1}(\mathcal{M}_{k-1}|X_{0:k-1}) \delta\mathcal{M}_{k-1} \tag{6.5}$$

$$p_k(\mathcal{M}_k|X_{0:k}) = \frac{g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k) p_{k|k-1}(\mathcal{M}_k|X_{0:k})}{g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})} \tag{6.6}$$

$$p_k(X_{1:k}) = g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) \frac{f_X(X_k|X_{k-1}, U_{k-1}) p_{k-1}(X_{1:k-1})}{g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1})}. \tag{6.7}$$

Apart from adopting RFS likelihoods for the measurement and map, the recursion defined by equations 6.5, 6.6 and 6.7 is similar to that in FastSLAM [58], [60]. However, the use of RFS likelihoods has important consequences in the evaluation of equation 6.7, to be seen later in Section 6.2.4. In FastSLAM, it should be noted that the map recursion of equation 6.6 is further decomposed into the product of K independent densities for each of the K features assumed to exist in the map. Furthermore, FastSLAM is conditioned on the inherently unknown data association assignments. In contrast, RFS-SLAM is not conditioned on any data association hypotheses to determine the number of features to estimate and the recursion of equation 6.6 is that of a RFS feature map. Consequently, the propagation of the map involves probability densities of random finite sets and marginalisation over the map involves set integrals. Similar to FastSLAM, the effect of the trajectory conditioning on RFS-SLAM is to render each feature estimate conditionally independent and thus the map correlations, critical to EKF-SLAM [1], are not required.

6.2.2 The PHD in RFS-SLAM

Recall from Section 3.3.1, that an optimal estimator for a random vector is the conditional expectation. Many state-of-the-art SLAM algorithms adopt Sequential Monte Carlo (SMC) methods. It is well known that SMC techniques are more amenable to expectation operations than maximisation operations, since if p is approximated by a set of weighted samples $\{\eta^{(i)}, X^{(i)}\}_{i=1}^{N}$, then [61], [62],

$$\sum_{i=1}^{N} \eta^{(i)} X^{(i)} \rightarrow \mathbb{E}[X] \tag{6.8}$$

as $N \to \infty$. However, in FastSLAM [58], it is common to choose the trajectory sample with the highest weight as the estimate of the vehicle path, and its corresponding map, as the estimate of the map. It is easier to establish convergence in SMC implementations if we use the expected path and expected map. However, it is not clear how the expected map is interpreted.

The PHD construct allows an alternative notion of expectation for maps, thereby fully exploiting the advantage of an SMC approximation. The PHD, v, is a function defined on the feature space satisfying equation 3.18. Recall from Section 3.3.4.1, that the value of the PHD at a point gives the expected number of features at that point while the integral over any given region gives the expected number of features in that region. A salient property of the PHD construct in map estimation is that the posterior PHD of the map is indeed the expectation of the trajectory-conditioned PHDs. More concisely,

$$v_k(m) = \mathbb{E}\left[v_k(m|X_{1:k})\right], \tag{6.9}$$

where the expectation is taken over the vehicle trajectory $X_{1:k}$. This result follows from the standard properties of the PHD (intensity function) of an RFS, see for example classical texts such as [31], [32]. Thus the PHD construct provides a natural framework to average feature map estimates, while incorporating both unknown associations and different feature numbers. This differs dramatically from vector based map averaging methods which require feature identification tracking and rule-based combinations [63]. In contrast, map averaging for grid-based maps is straight forward due to both known grid alignments and number of cells. While the practical merits of an expected feature map estimate for SLAM using a single sensor may be unclear at this time, related operations such as 'feature map addition' may be of use in sensor fusion and multi-robot SLAM applications. Furthermore, the PHD construct admits a Bayes optimal estimator for the map, as discussed previously in Section 3.3.1.

6.2.3 PHD Mapping

This section details the trajectory-conditioned PHD mapping recursion of equation 6.6, as was first proposed in [64]. The predicted and posterior RFS maps are approximated by Poisson RFSs with PHDs $v_{k|k-1}(m|X_{0:k})$ and $v_k(m|X_{0:k})$ respectively,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod_m v_{k|k-1}(m|X_{0:k})}{\exp\left(\int v_{k|k-1}(m|X_{0:k})dm\right)} \tag{6.10}$$

$$p_k(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod\limits_{m\in\mathcal{M}_k} v_k(m|X_{0:k})}{\exp\left(\int v_k(m|X_{0:k})dm\right)}. \tag{6.11}$$

In essence, this approximation assumes that features are IID and the number of features is Poisson distributed. This PHD approximation has been proven to be powerful and effective in multi-target tracking [3]. Poisson approximations for the number of new features have also been adopted in certain SLAM solutions [14]. In light of the above advantages of representing an RFS with sequential Monte Carlo methods, the PHD filter for the SLAM problem can be implemented in Rao-Blackwellised form. Again, referring to the PHD predictor – corrector of equations 3.23 and 3.24, substituting

$$\Gamma_k \longrightarrow m|X_{0:k} \tag{6.12}$$

the PHD predictor equation then becomes

$$v_{k|k-1}(m|X_{0:k}) = v_{k-1}(m|X_{0:k-1}) + b(m|X_k) \tag{6.13}$$

where $b(m|X_k)$ is the PHD of the new feature RFS, $\mathcal{B}(X_k)$, initially introduced in Section 3.4.

The corresponding Rao-Blackwellised, PHD corrector equation is then

$$v_k(m|X_{0:k}) = v_{k|k-1}(m|X_{0:k})\Bigg[1 - P_D(m|X_k) + \sum_{z\in\mathcal{Z}_k} \frac{P_D(m|X_k)g_k(z|m,X_k)}{c_k(z|X_k) + \int P_D(\xi|X_k)g_k(z|\xi,X_k)v_{k|k-1}(\xi|X_{0:k})d\xi}\Bigg] \tag{6.14}$$

where

$$\begin{aligned} P_D(m|X_k) &= \text{the probability of detecting a feature at } m, \text{ from vehicle pose } X_k. \\ c_k(z|X_k) &= \text{PHD of the clutter RFS } \mathcal{C}_k(X_k) \text{ (in equation 2.10) at time } k \text{ and,} \\ g_k(z|m,X_k) &= \text{the measurement model of the sensor at time } k. \end{aligned} \tag{6.15}$$

In contrast to the "Brute force" SLAM approach of chapter 5, the RB PHD SLAM filter estimates multiple, conditionally independent PHDs (intensity functions). Each independent map PHD, is conditioned on each of the N hypothesised vehicle trajectories (particles). Referring again to the GM example representations of PHDs in figures 3.2 and 3.3, in any particular map PHD, each Gaussian representing a/some possible feature(s) is conditioned on one of the N hypothesised vehicle trajectories. N such conditionally independent PHDs, one per hypothesised trajectory, are then propagated.

6.2.4 PHD-SLAM

This section extends the trajectory-conditioned PHD mapping recursion to the SLAM problem. With the hindsight of FastSLAM [58], the most obvious extension of PHD mapping [64] to SLAM is to exploit the factorisation equations 6.5, 6.6 and 6.7, e.g. PHD for mapping and particle filtering for localisation. This technique requires the computation of the posterior density of the vehicle trajectory in equation 6.7, in particular the term $g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})$, which requires set integration,

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) = \int p(\mathcal{Z}_k, \mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_{0:k})\delta\mathcal{M}_k. \tag{6.16}$$

This set integral is numerically intractable and a naive approach is to directly apply the EKF approximation proposed for FastSLAM [65]. However, an EKF approximation cannot be used since the likelihood equation 6.16, defined on the space of finite-sets, and its FastSLAM counterpart, defined on a Euclidean space, are two fundamentally different quantities and it is not known how they are even related. Therefore, in this case, it is fundamentally incorrect to use the EKF approximation in [58] as it will not result in a valid density, and thus its product with equation 6.6 cannot give the joint posterior of the map and pose. An EKF approximation requires a hypothesised data association assignment. Since there is no concept of data association in the RFS-SLAM framework (there is no fixed ordering of features or measurements in the state), alternative methods of evaluation of equation 6.16 are required.

Fortunately, by rearranging equation 6.6, it can be seen that $g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})$ is merely the normalising constant,

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) = \frac{g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k)p_{k|k-1}(\mathcal{M}_k|X_{0:k})}{p_k(\mathcal{M}_k|X_{0:k})}. \tag{6.17}$$

Note in the above, that the LHS does not contain the variable $\mathcal{M}_k$, while the RHS has $\mathcal{M}_k$ in both the denominator and numerator. In essence, $\mathcal{M}_k$ in equation 6.17 is a dummy variable, and thus equation 6.17 holds for any arbitrary choice of $\mathcal{M}_k$. This allows the substitution of any choice of $\mathcal{M}_k$ to evaluate $g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})$. This is an important result, which allows for the likelihood of the measurement conditioned on the trajectory (but not the map), to be calculated in closed-form, as opposed to using the EKF approximations in [58]. The following considers two simple choices: (derivations can be seen in Appendix B.)

6.2.4.1 The Empty Strategy

It was mentioned in Section 3.3.4.4, that if the RFS $\mathcal{M}_k$ is Poisson distributed in its number, and the points within $\mathcal{M}_k$ are IID distributed, then

the probability density of $\mathcal{M}_k$ can be recovered exactly from the PHD intensity function according to equation 3.19. Similarly the predicted and posterior RFS maps can be approximated by Poisson RFSs with PHDs $v_{k|k-1}(m|X_{0:k})$ and $v_k(m|X_{0:k})$ respectively,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod_{m\in\mathcal{M}_k} v_{k|k-1}(m|X_{0:k})}{\exp\left(\int v_{k|k-1}(m|X_{0:k})dm\right)} \tag{6.18}$$

$$p_k(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod_{m\in\mathcal{M}_k} v_k(m|X_{0:k})}{\exp\left(\int v_k(m|X_{0:k})dm\right)}. \tag{6.19}$$

Setting $\mathcal{M}_k = \emptyset$, and using the Poisson RFS approximations, equation 6.18 and equation 6.19, as well as the RFS measurement likelihood, equation 5.4 shown in Section 5.2, it follows that (see Appendix B)

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) \approx \kappa_k^{\mathcal{Z}_k} \exp\left(\hat{\mathfrak{m}}_k - \hat{\mathfrak{m}}_{k|k-1} - \int c_k(z|X_k)dz\right), \tag{6.20}$$

where, $\kappa_k^{\mathcal{Z}_k} = \prod_{z\in\mathcal{Z}_k} c_k(z|X_k)$ with, $c_k(z|X_k)$ being the PHD of the measurement clutter RFS $\mathcal{C}_k(X_k)$. In addition, $\hat{\mathfrak{m}}_k = \int v_k(m|X_{0:k})dm$ and $\hat{\mathfrak{m}}_{k|k-1} = \int v_{k|k-1}(m|X_{0:k})dm$. Equation 6.20 gives the closed form likelihood of the measurement, conditioned on the trajectory, and not on the map.

6.2.4.2 The Single Feature Strategy

In a similar manner, to evaluate the normalising constant for the case of $\mathcal{M}_k = \{\bar{m}\}$, again using equations 6.18, 6.19, and 5.4,

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) \approx \frac{1}{\Gamma}\left[\left((1 - P_D(\bar{m}|X_k))\kappa_k^{\mathcal{Z}_k} + P_D(\bar{m}|X_k)\sum_{z\in\mathcal{Z}_k} \kappa_k^{\mathcal{Z}_k-\{z\}} g_k(z|\bar{m}, X_k)\right) v_{k|k-1}(\bar{m}|X_{0:k})\right] \tag{6.21}$$

with,

$$\Gamma = \exp\left(\hat{\mathfrak{m}}_{k|k-1} - \hat{\mathfrak{m}}_k + \int c_k(z)dz\right) v_k(\bar{m}|X_{0:k}). \tag{6.22}$$

For this choice of $\mathcal{M}_k$, $\bar{m}$ can be, for instance, the feature with the least uncertainty or the maximum measurement likelihood. It is possible to choose $\mathcal{M}_k$ with multiple features, however this will increase the computational burden. Due to the presence of the measurement likelihood term $g_k(z|\bar{m}, X_k)$, it is

expected that, in general, the single feature map update will outperform that of the empty map update. Note that in equation 6.17, every choice of $\mathcal{M}_k$ would give the same result, however equations 6.20 and 6.21 use different *approximations* of $p_k(\mathcal{M}_k|X_{0:k})$, yielding slightly different results. In principle, any map strategy can be used including more features, however the computation required to evaluate the trajectory conditioned measurement likelihood would also increase. The following section presents a Rao-Blackwellised implementation of the proposed PHD-SLAM filter.

6.3 Rao-Blackwellised Implementation of the PHD-SLAM Filter

Following the description of the PHD-SLAM filter in the previous section, a Rao-Blackwellised (RB) implementation is detailed in this section. In essence, a particle filter is used to propagate the vehicle trajectory (equation 6.7), and a Gaussian mixture (GM) PHD filter is used to propagate each trajectory-conditioned map PHD (equation 6.6). As such, let the PHD-SLAM density at time $k-1$ be represented by a set of N particles,

$$\left\{\eta_{k-1}^{(i)}, X_{0:k-1}^{(i)}, v_{k-1}^{(i)}(m|X_{0:k-1}^{(i)})\right\}_{i=1}^{N},$$

where $X_{0:k-1}^{(i)} = [X_0, X_1^{(i)}, X_2^{(i)}, \ldots, X_{k-1}^{(i)}]$ is the i^{th} hypothesised vehicle trajectory and $v_{k-1}^{(i)}(m|X_{0:k-1}^{(i)})$ is its map PHD. The filter then proceeds to approximate the posterior density by a new set of weighted particles,

$$\left\{\eta_{k}^{(i)}, X_{0:k}^{(i)}, v_{k}^{(i)}(m|X_{0:k}^{(i)})\right\}_{i=1}^{N},$$

as follows:

6.3.1 PHD Mapping – Implementation

Let the prior map PHD for the i^{th} particle, $v_{k-1}^{(i)}(m|X_{k-1}^{(i)})$, be a Gaussian mixture of the form,

$$v_{k-1}^{(i)}(m|X_{k-1}^{(i)}) = \sum_{j=1}^{J_{k-1}^{(i)}} \omega_{k-1}^{(i,j)} \mathcal{N}\big(m; \mu_{k-1}^{(i,j)}, P_{k-1}^{(i,j)}\big) \tag{6.23}$$

which is a mixture of $J_{k-1}^{(i)}$ Gaussians, with $\omega_{k-1}^{(i,j)}$, $\mu_{k-1}^{(i,j)}$ and $P_{k-1}^{(i,j)}$ being the corresponding predicted weights, means and covariances respectively for the j^{th} Gaussian component of the map PHD of the i^{th} trajectory. Let the new feature intensity for the particle, $b(m|\mathcal{Z}_{k-1}, X_k^{(i)})$, from the sampled pose, $X_k^{(i)}$ at time k also be a Gaussian mixture of the form

$$b(m|\mathcal{Z}_{k-1}, X_k^{(i)}) = \sum_{j=1}^{J_{b,k}^{(i)}} \omega_{b,k}^{(i,j)} \mathcal{N}\big(m; \mu_{b,k}^{(i,j)}, P_{b,k}^{(i,j)}\big) \tag{6.24}$$

where, $J_{b,k}^{(i)}$ defines the number of Gaussians in the new feature intensity at time k and $\omega_{b,k}^{(i,j)}$, $\mu_{b,k}^{(i,j)}$ and $P_{b,k}^{(i,j)}$ are the corresponding components. This is analogous to the proposal distribution in the particle filter and provides an initial estimate of the new features entering the map.

The predicted intensity is therefore also a Gaussian mixture,

$$v_{k|k-1}^{(i)}(m|X_k^{(i)}) = \sum_{j=1}^{J_{k|k-1}^{(i)}} \omega_{k|k-1}^{(i,j)} \mathcal{N}\big(m; \mu_{k|k-1}^{(i,j)}, P_{k|k-1}^{(i,j)}\big) \tag{6.25}$$

which consists of $J_{k|k-1}^{(i)} = J_{k-1}^{(i)} + J_{b,k}^{(i)}$ Gaussians representing the union of the prior map intensity, $v_{k-1}(m|X_{k-1}^{(i)})$, and the proposed new feature intensity according to equation 6.13. Since the measurement likelihood is also of Gaussian form, it follows from equation 6.14 that the posterior map PHD, $v_k^{(i)}(m|X_k^{(i)})$ is then also a Gaussian mixture given by

$$v_k^{(i)}(m|X_k^{(i)}) = v_{k|k-1}^{(i)}(m|X_k^{(i)}) \left[1 - P_D(m|X_k^{(i)}) + \sum_{z \in \mathcal{Z}_k} \sum_{j=1}^{J_{k|k-1}^{(i)}} v_{G,k}^{(i,j)}(z, m|X_k^{(i)})\right]. \tag{6.26}$$

The components of the above equation are given by,

$$v_{G,k}^{(i,j)}(z, m|X_k^{(i)}) = \psi_k^{(i,j)}(z|X_k^{(i)}) \mathcal{N}(m; \mu_{k|k}^{(i,j)}, P_{k|k}^{(i,j)}) \tag{6.27}$$

$$\psi_k^{(j)}(z|X_k^{(i)}) = \frac{P_D(m|X_k^{(i)}) \omega_{k|k-1}^{(i,j)} \mathcal{N}\big(z; H_k \mu_{k|k-1}^{(i,j)}, S_k^{(i,j)}\big)}{c(z) + \sum_{\ell=1}^{J_{k|k-1}^{(i)}} P_D(m|X_k^{(i)}) \omega_{k|k-1}^{(i,\ell)} \mathcal{N}\big(z; H_k \mu_{k|k-1}^{(i,\ell)}, S_k^{(i,\ell)}\big)} \tag{6.28}$$

The terms $\mu_{k|k}$, $P_{k|k}$ and S_k can be obtained using any standard filtering technique such as EKF or UKF [66]. In this chapter, the EKF updates are adopted.

The clutter RFS, C_k, is assumed Poisson distributed [14] in number and uniformly spaced over the mapping region. Therefore the clutter intensity is given by, $c(z) = \lambda_c \mathcal{U}(z)$, where λ_c is the average number of clutter measurements and $\mathcal{U}(z)$ denotes a uniform distribution on the measurement space. As with other feature-based GM implementations [38], pruning and merging operations are required to curb the explosive growth in the number of Gaussian components of the posterior map PHD. These operations are carried out as in [33].

6.3.2 The Vehicle Trajectory – Implementation

The proposed filter adopts a particle approximation of the posterior vehicle trajectory, $p_k(X_{1:k})$, which is sampled/re-sampled as follows:

Step 1: *Sampling Step*

- For $i = 1, ..., N$, sample $\widetilde{X}_k^{(i)} \sim f_X(\widetilde{X}_k^{(i)} | X_{k-1}^{(i)}, U_{k-1})$ and set

$$\widetilde{\eta}_k^{(i)} = \frac{g_k(\mathcal{Z}_k | \mathcal{Z}_{0:k-1}, \widetilde{X}_{0:k}^{(i)}) f_X(\widetilde{X}_k^{(i)} | X_{k-1}^{(i)}, U_{k-1})}{f_X(\widetilde{X}_k^{(i)} | X_{k-1}^{(i)}, U_{k-1})} \eta_{k-1}^{(i)}. \tag{6.29}$$

- Normalise weights: $\sum_{i=1}^{N} \widetilde{\eta}_k^{(i)} = 1$.

Step 2: *Resampling Step*

- Resample $\left\{ \widetilde{\eta}_k^{(i)}, \widetilde{X}_{0:k}^{(i)} \right\}_{i=1}^{N}$ to get $\left\{ \eta_k^{(i)}, X_{0:k}^{(i)} \right\}_{i=1}^{N}$.

Since the vehicle transition density is chosen as the proposal density as with FastSLAM 1.0 [58],

$$\widetilde{\eta}_k^{(i)} = g_k(\mathcal{Z}_k | \mathcal{Z}_{0:k-1}, \widetilde{X}_{0:k}^{(i)}) \eta_{k-1}^{(i)} \tag{6.30}$$

which can be evaluated in closed form according to $\mathcal{M}_k$ being the empty map (equation 6.20) or $\mathcal{M}_k$ being a single feature map (equation 6.21), where

$$\widehat{\mathfrak{m}}_{k|k-1}^{(i)} = \sum_{j=1}^{J_{k|k-1}^{(i)}} \omega_{k|k-1}^{(i,j)} \text{ and } \widehat{\mathfrak{m}}_k^{(i)} = \sum_{j=1}^{J_k^{(i)}} \omega_k^{(i,j)}. \tag{6.31}$$

Note that the incorporation of the measurement conditioned proposal of FastSLAM 2.0 can also be accommodated in this framework. This direction of research focuses on more efficient SMC approximations and is an avenue for further studies.

6.3.3 SLAM State Estimation and Pseudo-code

As mentioned in the introduction, in contrast to previous SLAM algorithms, the PHD map representation allows for a natural ability to average feature maps. That is, independent map estimates from N independent trajectory particles can be readily averaged into an expected estimate, even with map estimates of different size and without having to resolve the intra-map feature associations. Consequently, in the case of the RB-PHD-SLAM filter, both the expected vehicle trajectory and feature map can be determined as follows:

Given the posterior set of weighted particles and corresponding map PHDs,

$$\left\{\eta_k^{(i)}, X_{0:k}^{(i)}, v_k^{(i)}(m|X_{0:k}^{(i)})\right\}_{i=1}^{N},$$

and $\bar{\eta} = \sum_{i=1}^{N} \eta_k^{(i)}$ then,

$$\widehat{X}_{0:k} = \frac{1}{\bar{\eta}} \sum_{i=1}^{N} \eta_k^{(i)} X_{0:k}^{(i)}. \tag{6.32}$$

As demonstrated previously in Section 6.2.4, the posterior PHD of the map is the expectation of the trajectory-conditioned PHDs and thus

$$v_k(m|X_{0:k}) = \frac{1}{\bar{\eta}} \sum_{i=1}^{N} \eta_k^{(i)} v_k^{(i)}(m|X_{0:k}^{(i)}). \tag{6.33}$$

If $\hat{\mathfrak{m}}_k = \int v_k(m|X_{0:k})dm$, is the mass of the posterior map PHD, the expected map estimate can then be extracted by choosing the $\hat{\mathfrak{m}}_k$ highest local maxima. The pseudo-code for the RB-PHD-SLAM filter are provided in tables 6.1, 6.2, while that of appropriate estimators is provided in Tables 6.3 and 6.4, which continues as Table 6.4. The following section presents the results and analysis of the proposed filter, with comparisons to standard SLAM algorithms.

6.4 Results and Analysis

This section presents the results and analysis of the proposed approach using both simulated and real experimental datasets. Initial comparisons are made with the FastSLAM [58] algorithm with maximum likelihood data association, using a mutual exclusion constraint and a 95% χ^2 confidence gate. These comparisons are demonstrated with a simulated single loop vehicle trajectory carrying a simulated range – bearing measuring sensor and a real, land based vehicle using a millimetre wave (MMW) radar for feature extraction. To further demonstrate the abilities of the RB-RFS-SLAM approach,

Table 6.1 RB-PHD-SLAM-Predict

Algorithm RB-PHD-SLAM-Predict
$\left(\{\eta_{k-1}^{(i)}, X_{0:k-1}^{(i)}, v_{k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\}_{i=1}^{N}, \mathcal{Z}_{k-1}, U_{k-1}\right)$
// Construct (6.25)
1. for i = 1 to N do
// Sample pose
2. $\tilde{X}_k^{(i)} \sim f_X(\tilde{X}_k^{(i)}|X_{k-1}^{(i)}, U_{k-1})$
// Predict map
3. GMM-PHD-FBRM-Predict $\left(\mathcal{Z}_{k-1}, X_{k-1}^{(i)}, v_{k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\right.$
4. end for
5. return $\left(\{\eta_{k-1}^{(i)}, \tilde{X}_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\}_{i=1}^{N}\right)$

Table 6.2 RB-PHD-SLAM-Update

Algorithm RB-PHD-SLAM-Update
$\left(\{\eta_{k-1}^{(i)}, \tilde{X}_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\}_{i=1}^{N}, \mathcal{Z}_k\right)$
1. for i = 1 to N do
// Update map PHD
2. GMM-PHD-FBRM-Update $\left(\mathcal{Z}_k, \tilde{X}_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\right)$
// Predicted PHD mass
3. $\hat{\mathfrak{m}}_{k|k-1} = \sum_{j=1}^{J_{k|k-1}^{(i)}} \omega_{k|k-1}^{(i,j)}$
// Updated PHD mass
4. $\hat{\mathfrak{m}}_k = \sum_{j=1}^{J_k^{(i)}} \omega_k^{(i,j)}$
5. if(Empty Strategy TRUE) do
// Updated trajectory weight of (6.20)
6. $\tilde{\eta}_k^{(i)} = \left(c(z)^{|\mathcal{Z}_k|} \exp^{(\hat{\mathfrak{m}}_k - \hat{\mathfrak{m}}_{k|k-1} - \lambda_c)}\right)\eta_{k-1}^{(i)}$
7. end if
8. if(Single Feature Strategy TRUE) do
// Select a given map state
9. $\mathrm{j} = \{i = 1, \ldots, J_k^{(i)} | m^{(i,j)} = \bar{m}\}$
10. $a = (1 - P_D)c(z)^{|\mathcal{Z}_k|} + P_D\omega_{k|k-1}^{(i,j)} \times \sum_{z \in \mathcal{Z}_k} (c(z)^{|\mathcal{Z}_k|-1})\mathcal{N}(z; z_{k|k-1}^{(i,j)}, S_k^{(i,j)})$
11. $b = \exp^{(\hat{\mathfrak{m}}_{k|k-1} - \hat{\mathfrak{m}}_k + \lambda_c)}\, \omega_k^{(i,j)}$
// Update trajectory weight of (6.21)
12. $\tilde{\eta}_k^{(i)} = \frac{a}{b}\tilde{\eta}_{k|k-1}^{(i)}$
13. end if
14. end for
15. return $\left(\{\tilde{\eta}_k^{(i)}, \tilde{X}_k^{(i)}, v_k^{(i)}(m|\mathcal{Z}_k, X_k)\}_{i=1}^{N}\right)$

Table 6.3 RB-PHD-SLAM-MAPestimate

Algorithm RB-PHD-SLAM-MAPestimate
$\left(\{\eta_{k-1}^{(i)}, X_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1})\}_{i=1}^{N}, \mathcal{Z}_k, T_{\text{feature}}\right)$
// Initialise the map estimate
1. $\hat{\mathcal{M}}_k = \emptyset$
2. I = {1,. . . ,N}
// Get index of max weight component
3. j = $\arg\max_{i \in I} \eta_k^{(i)}$
// Estimated Trajectory
4. $\hat{X}_{0:k} = X_{0:k}^{(j)}$
// Estimate Map from corresponding map PHD
5. for i = 1 to $J_k^{(j)}$ do
6. if $\omega_k^{(j,i)} > T_{\text{feature}}$
// concatenate estimate
7. $\hat{\mathcal{M}}_k = [\hat{\mathcal{M}}_k \ \mu_k^{(j,i)}]$
8. end if
9. end for
// RB-PHD-SLAM MAP Estimate
10. return $\left(\hat{X}_{0:k}, \hat{\mathcal{M}}_k\right)$

further, somewhat more complicated, experiments are carried out in which the benchmark algorithms used are the classical FastSLAM [58] but with with Multiple Hypothesis Data association [67] and the Joint Compatibility Branch and Bound (JCBB) EKF [39]. In this second set of experiments, in the simulation, multiple vehicle loop trajectories are executed and for the real experiment, a much larger scenario, where accurate SLAM performance in the presence of high clutter is essential, is demonstrated at sea, in a coastal environment, using an "Autonomous kayak" [59] as the vehicle and a commercially available X-Band radar.

In all experiments, the 'single feature map' trajectory weighting of equation 6.21 is adopted for the proposed RB-PHD-SLAM filter. An implementation using the 'empty map update' of equation 6.20 was presented in [56]. While any feature can theoretically be selected to generate the trajectory weighting, in this implementation, that which generates the maximum likelihood amongst all measurements is chosen. A comprehensive study as to the best suited feature selection strategies is left to future work.

Current SLAM filters deal with clutter through 'feature management' routines, such as the landmark's quality [1] or a binary Bayes filter [58]. These operations are typically independent of the joint SLAM filter update, whereas the proposed approach unifies feature management, data association

Table 6.4 RB-PHD-SLAM-EAPestimate

Algorithm RB-PHD-SLAM-EAPestimate

$$\left(\{\eta_{k-1}^{(i)}, X_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1})\}_{i=1}^{N}, \mathcal{Z}_k, T_{\text{feature}}, D_{\min}\right)$$

1. $\Omega_k = 0$
2. for i = 1 to N do
3. $\quad \Omega_k = \Omega_k + \eta_k^{(i)}$
4. end for

// expected trajectory

5. $\hat{X}_{0:k} = \dfrac{1}{\Omega_k} \displaystyle\sum_{i=1}^{N} \eta_k^{(i)} X_{0:k}^{(i)}$

// Initialise number of Gaussian components

6. l = 0
7. for i = 1 to N
8. $\quad$ for j = 1 to $J_k^{(i)}$
9. $\quad\quad$ l = l+1
10. $\quad\quad \bar{\omega}_k^{(l)} = \eta_k^{(i)} \omega_k^{(i,j)}$
11. $\quad\quad \bar{\mu}_k^{(l)} = \mu_k^{(i,j)}$
12. $\quad\quad \bar{P}_k^{(l)} = P_k^{(i,j)}$
13. $\quad$ end for
14. end for
15. $\mathcal{R} = \{1, \ldots, l\}$

// Initialise number of merged Gaussian components

16. L = 0

// Gaussian merging

17. do while $\mathcal{R} \neq \emptyset$

// Increment component counter

18. $\quad$ L = L+1

// Get index of max weight component

19. $\quad$ j = $\arg\max_{r \in \mathcal{R}} \bar{\omega}_k^{(r)}$

// Cluster those within distance $D_{\min}$

20. $\quad$ K = $\{r \in \mathcal{R} | (\bar{\mu}_k^{(r)} - \bar{\mu}_k^{(j)})^T [\bar{P}_k^{(r)}]^{-1} (\bar{\mu}_k^{(r)} - \bar{\mu}_k^{(j)}) \leq D_{\min}\}$

// Combine component weights

21. $\quad \tilde{\omega}_k^{(L)} = \displaystyle\sum_{i \in K} \bar{\omega}_k^{(i)}$

// Weighted average mean

22. $\quad \tilde{\mu}_k^{(L)} = \dfrac{1}{\tilde{\omega}_k^{(L)}} \displaystyle\sum_{i \in K} \bar{\omega}_k^{(i)} \bar{\mu}_k^{(i)}$

Table 6.4 (*Continued*)

// Combined covariance
// Remove K from $\mathcal{R}$ and repeat
23. $\tilde{P}_k^{(L)} = \frac{1}{\tilde{\omega}_k^{(L)}} \sum_{i \in K} \bar{\omega}_k^{(i)} \left(\bar{P}_k^{(i)} + (\tilde{\mu}_k^{(L)} - \bar{\mu}_k^{(i)})(\tilde{\mu}_k^{(L)} - \bar{\mu}_k^{(i)})^T \right)$
24. $\mathcal{R} = \mathcal{R} - K$
25. end while
26. for i = 1 to L do
27. if $\tilde{\omega}_k^{(i)} > T_{\text{feature}}$
// concatenate estimate
28. $\hat{\mathcal{M}}_k = [\hat{\mathcal{M}}_k \ \tilde{\mu}_k^{(i)}]$
29. end if
30. end for
// RB-PHD-SLAM EAP Estimate
31. return $\left(\hat{X}_{0:k}, \hat{\mathcal{M}}_k \right)$

and state filtering into a single Bayesian update. While these methods have been used successfully, they generally discard the sensor's detection (P_D) and false alarm (P_{FA}) probabilities and thus can be erroneous when subject to large clutter rates and/or measurement noise. Since the proposed approach assumes knowledge of these probabilities, as seen in equation 6.14, a modified feature management routine coined the 'feature existence filter' (see Appendix C), which incorporates both P_D and P_{FA}, is used with the benchmark algorithms in an attempt to be 'fairer' to them in the comparisons.

To quantify the map estimation error, a metric must be adopted which jointly evaluates the error in the feature location *and* number estimates. Current methods typically examine the location estimates of a selected number of features and obtain their Mean Squared Error (MSE) using ground truth [1]. As such, vector-based error metrics are applied to feature maps and the error in the estimated *number* of features is neglected. While there are several metrics for finite-set-valued estimation error, that of [23] has been demonstrated to be most suitable [64], [56]. Therefore, the set map error metric described in Chapter 4 (equation 4.6) is therefore once again used to gauge the mapping performance in terms of estimated and actual feature number, as well spatial error. In the following sections, this metric along with the root mean squared error (RMSE) and graphical comparisons are used to demonstrate the merits of the RB-PHD-SLAM filter.

6.4.1 Simulated Datasets

Comparisons of RB-RFS-SLAM with standard vector based SLAM algorithms are firstly presented in the form of simulated trials due to the ease of generating known ground truth (trajectories and maps) for estimation error evaluation.

6.4.1.1 Simulation: Single Loop Trajectory

The filter parameters used for the single loop trajectory simulated trial are shown in Table 6.5. The measurement noise was inflated to hinder data association in the vector-based filter. For both filters, both the maximum a posterior (MAP) and expected a posterior (EAP) trajectories are reported. For FastSLAM, the map of the highest weighted trajectory estimate is used as the map estimate, while for the proposed filter, both the map of the highest weighted trajectory and the EAP map estimate are determined for comparison. 50 Monte Carlo (MC) trials were carried out.

Table 6.5 Filter parameters used for the single loop trajectory trial.

Filter Parameter	Value
Velocity input standard deviation (std)	1.5 m/s
Steering input std.	9.5^o
Range measurement std.	1.75m
Bearing measurement std.	3.5^o
Probability of Detection P_D	0.95
Clutter rate λ_c	5m^{-2}
Sensor maximum range	10m
Sensor Field-of-View (FoV)	360^o
Feature existence threshold	0.5

Figure 6.1 shows a sample of the raw data used in the trials, with the green circles depicting the true feature locations. A quantitative evaluation of the estimation results is provided through the RMSE, along with standard deviations, of the trajectory estimate as shown in figure 6.2. Without knowledge of P_D and P_{FA}, the benchmark approach can be made to appear highly erroneous due to poor feature management. Incorporating this information can improve the result, however the feature management is still effectively a post-filter update processing method. The RB-PHD-SLAM algorithm is significantly more robust due to the RFS feature map representation and Bayesian recursion which jointly performs feature management and state estimation.

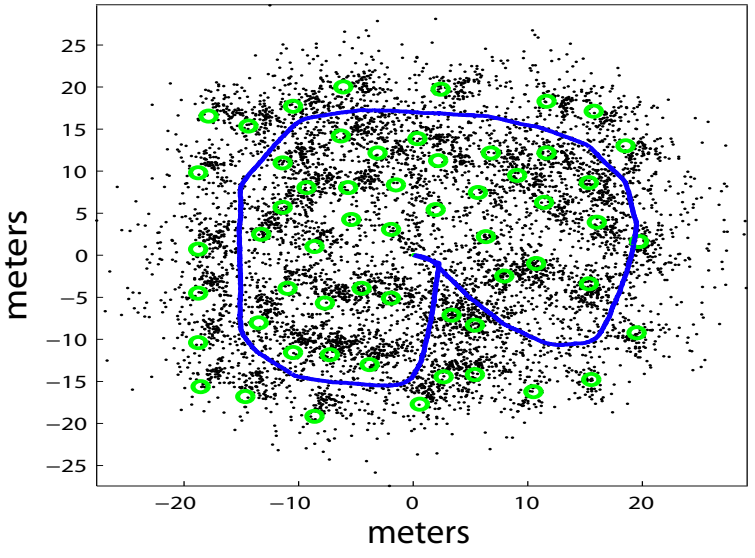


Fig. 6.1 The simulated environment showing point features (green circles). A sample measurement history (black points) plotted from a sample noisy trajectory (blue line) is also shown.

In terms of the map estimation, figure 6.3 depicts both the true and estimated number of features as the robot explores the map, with the proposed approach closely tracking the true number. Note that since this trial is a simulation, the true number of features which have entered the vehicle's FoV during its entire trajectory, can be calculated exactly. Since this result does not examine the locations of the estimated features, the set metric of equation 4.6 is used to compare map estimates, as shown in figure 6.4. The figure shows the 'ideal' mapping error (i.e. every feature is instantly estimated by its true coordinates when it enters the sensor FoV), which converges to zero once all features in the map have been scanned. The mean and std of the map estimates for both the benchmark and proposed approach are plotted, with that of the RB-PHD-SLAM filter reporting less map estimation error. A qualitative depiction of the posterior estimates from both approaches is provided in figures 6.5 and 6.6, demonstrating the usefulness of the RFS approach and the associated RB-PHD-SLAM filter. In both figures, the green line and circles represent the ground truth vehicle trajectory and feature locations respectively. The black crosses represent the estimated map. In the case of FastSLAM, this is derived with respect to the MAP FastSLAM trajectory estimate (the particle (trajectory) with the final maximum weight). In each figure, the blue line indicates the MAP trajectory estimate, which corresponds to the particle with the maximum weight, at each time and the red line corresponds to the expected trajectory estimate, which is the weighted

average of all particles at each time (see Tables 6.3 and 6.4). Given that the filter jointly incorporates data association and feature number uncertainty into its Bayesian recursion, it is more robust to large sensing error, as it does not rely on hard measurement-feature assignment decisions. Furthermore, it jointly estimates the number of features and their locations, alleviating the need for popular feature management methods [1], [58].

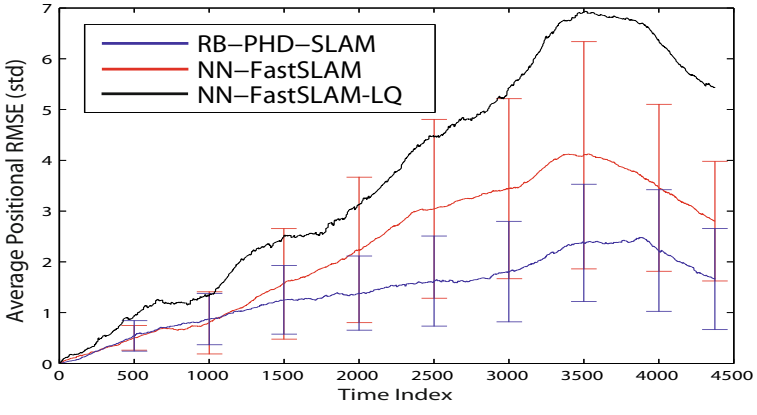


Fig. 6.2 The mean and standard deviation of the expected trajectory estimates of RB-PHD-SLAM vs. that of FastSLAM over 50 MC runs. LQ refers to an implementation with the 'landmark quality' method of [1].

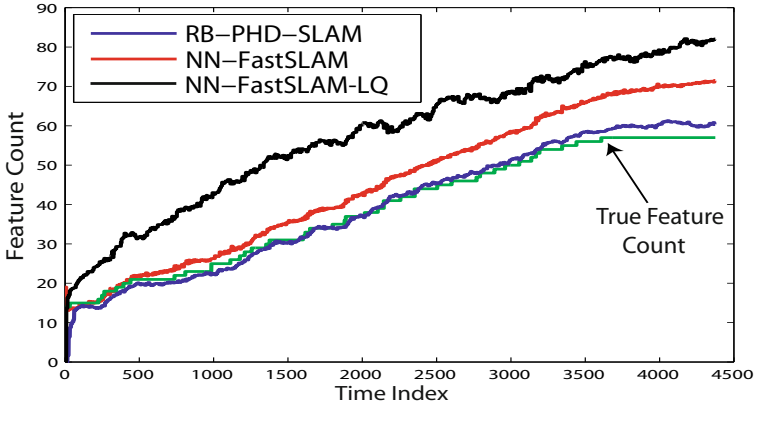


Fig. 6.3 The average estimated number of features in the map vs. ground truth for each approach. The feature number estimate of RB-PHD-SLAM can be seen to closely track that of the ground truth. Clearly there is no distinction between correctly estimated feature and false feature in this result.

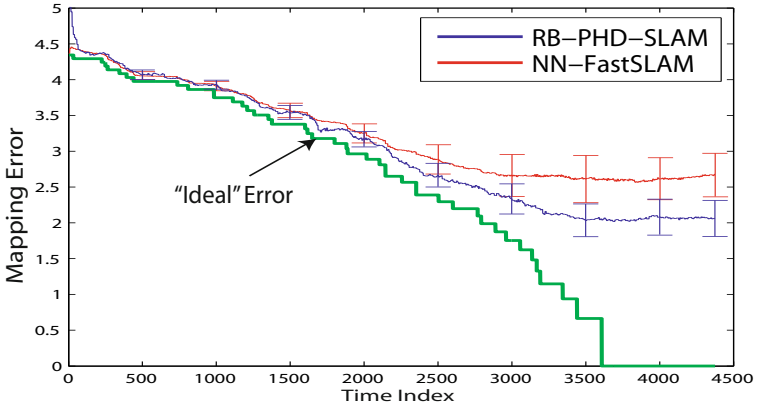


Fig. 6.4 A comparative plot of the mean and standard deviation of the map estimation error vs. time. The error incorporates that of the number of features, shown in figure 6.3 as well as their positional estimates. Note that the 'ideal' error converges to zero, an important property for SLAM filters and map estimation comparisons.

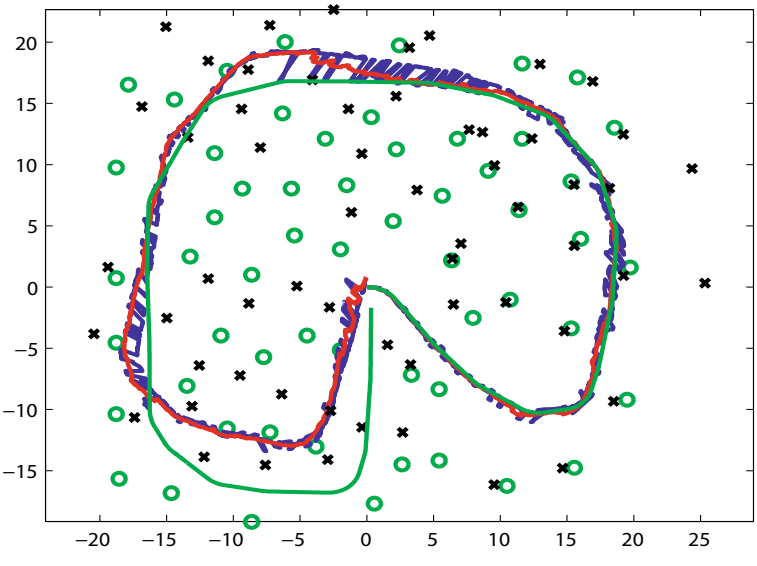


Fig. 6.5 A sample posterior estimate from FastSLAM showing error in the estimated trajectory and feature map. The green circles and line show the ground truth feature locations and path respectively. The black crosses show the FastSLAM estimated map (feature locations). The blue line shows the MAP trajectory estimate and the red line shows the expected trajectory estimate.

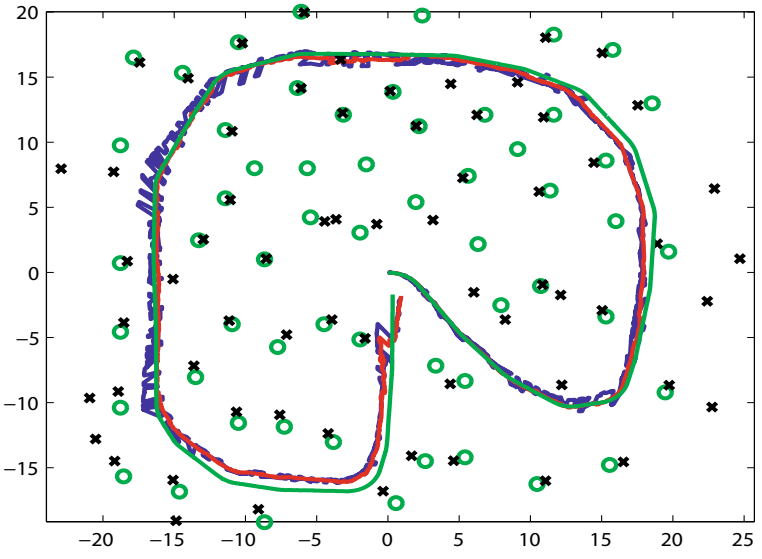


Fig. 6.6 The posterior estimate given the same inputs / measurements as those used in figure 6.5. Again, the green circles and line show the ground truth feature locations and path respectively. The black crosses show the RB-PHD-SLAM estimated map (feature locations). The RB-PHD-SLAM filter demonstrates its robustness and accuracy given high clutter and data association ambiguity.

6.4.1.2 Simulation: Multiple Loop Trajectories

The parameters for the more complex, multiple loop trajectory, simulated trials are shown in table 6.6. A 95% validation gate is used throughout. For each

Table 6.6 Filter parameters used for the single loop trajectory trial.

Filter Parameter	Value
Velocity input standard deviation (std)	2.0 m/s
Steering input std.	5.0^o
Range measurement std.	1.00m
Bearing measurement std.	2.0^o
Probability of Detection P_D	0.95
Clutter rate λ_c	20m^{-2}
Sensor maximum range	10m
Sensor Field-of-View (FoV)	360^o
Feature existence threshold	0.5

SLAM filter, 50 Monte Carlo (MC) trials were carried out in which all methods received identical sequences of control inputs and measurements. The RB based filters used 50 trajectory particles each, while for MH-FastSLAM a maximum limit of 2000 particles (number of hypotheses considered prior to resampling) was used.

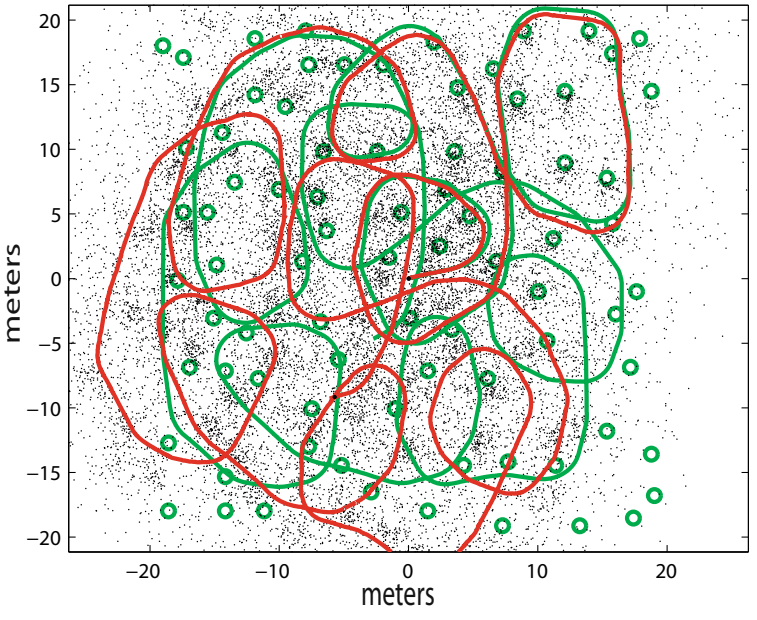


Fig. 6.7 The simulated environment showing point features (green circles) and true vehicle trajectory (green line). A sample measurement history plotted from a sample noisy trajectory (red line) is also shown (black points).

Figure 6.7 shows a sample of the raw input data used in the trials, superimposed onto the ground truth feature map and path. A comparison of the average trajectory estimation errors for all three filters is then presented in Figure 6.8. In terms of the estimated trajectory, the first order RB-PHD-SLAM algorithm can be seen to outperform the vector based FastSLAM with robust data association, but does not quite achieve the estimation accuracy of JCBB-EKFSLAM. This is primarily due to the fact that JCBB is very conservative in its choice of measurement-feature associations (jointly compatible constraint) resulting in very few false association pairs influencing the trajectory estimation. However, later results in Figures 6.9, 6.10 and 6.12 highlight the drawbacks of achieving such impressive localisation results.

In terms of the map estimation component of each SLAM algorithm, Figure 6.9 depicts both the true and estimated number of features as the vehicle explores the map, with the proposed RB-PHD-SLAM approach seen to closely

track the true number of features in the explored map. Erroneous associations for the MH-FastSLAM approach result in excessive feature declarations, while the conservative (only including those which are jointly compatible) association decisions of JCBB-EKF SLAM reduces the number of correct associations. Since vector based feature management routines are typically dependant on the data association decisions, this dramatically influences the map estimation error.

Incorporating the estimation error in both the number and location of features in the map, Figure 6.10 plots the map error distance of equation 4.6 for each approach. Note that in order to generate this result, the vector based maps of FastSLAM and JCBB-EKFSLAM are temporarily 'assumed' to be sets. The proposed method can be seen to report the least mapping error due it is robust ability to jointly incorporate uncertainty in feature locations and numbers, while erroneous feature estimates contribute to increased mapping error for the vector based approaches. A qualitative depiction of the posterior estimates is provided in Figure 6.11, demonstrating the usefulness of the RFS-SLAM framework and the proposed RB-PHD-SLAM filter.

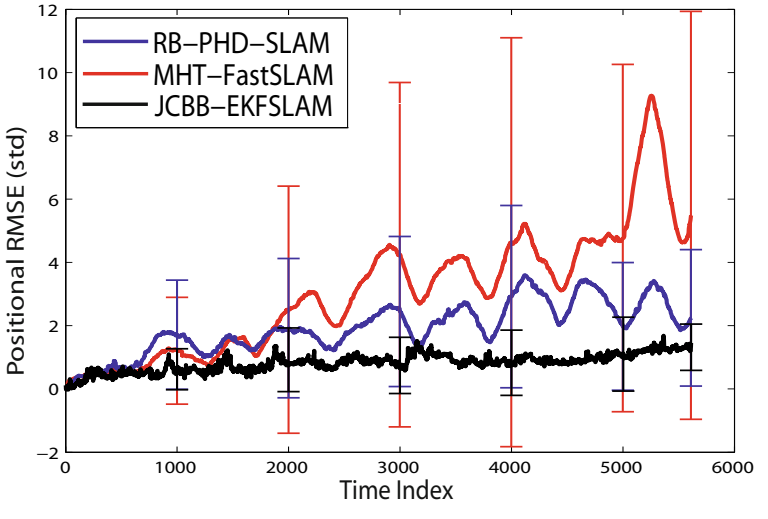


Fig. 6.8 The mean and standard deviation of the trajectory estimates from each filter over 50 MC runs versus time.

6.4.2 A Note on Computational Complexity

As can be observed from the implementation of Section 6.3, the computational complexity of RB-PHD-SLAM is, $\mathcal{O}(\mathfrak{m}_{k|k}N)$ i.e. linear in the number of features (in the FoV), linear in the number of measurements and linear in

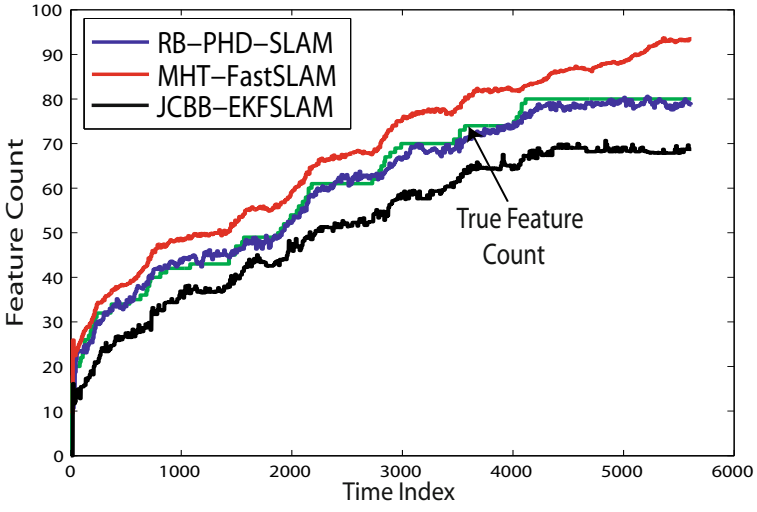


Fig. 6.9 The average estimated number of features in the map for each filter versus time, compared to the ground truth number of features in the explored map $\mathcal{M}_k$. The feature number estimate of RB-PHD-SLAM can be seen to closely track that of the ground truth.

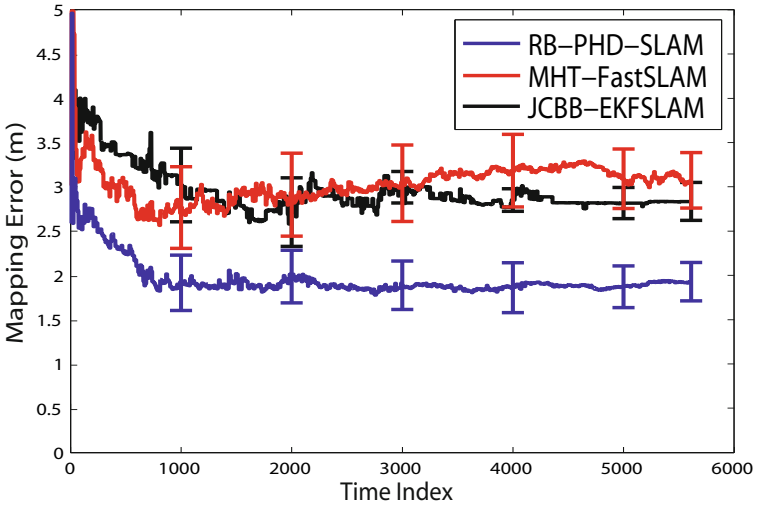


Fig. 6.10 A comparative plot of the mean and standard deviation of the map estimation error for each filter vs. time. At any given time, for the ideal case, the mapping error from equation 4.6 wrt. the explored map is zero.

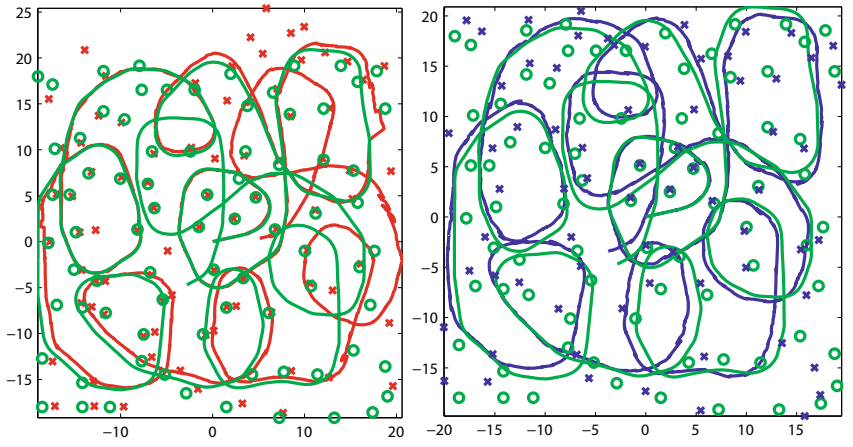


Fig. 6.11 Comparisons of the posterior SLAM estimates from MH-FastSLAM (left, red) and the proposed RB-PHD-SLAM (right, blue). The ground truth trajectory and map are represented by the green line and circles respectively. The RB-PHD-SLAM filter demonstrates its robustness and accuracy given high clutter and data association ambiguity.

the number of trajectory particles. The linear complexity of each particle in the mapping filter was verified previously in Figure 4.8.

For a single thread implementation, Figure 6.12 shows that the computational time is comparable with that of the MH-FastSLAM algorithm, both of which are less expensive than JCBB-EKF SLAM as its hypothesis tree grows in the presence of high clutter. Note that due to the Rao-Blackwellised structure of RB-PHD-SLAM, binary tree based enhancements, such as those applied to traditional FastSLAM [58], can be readily developed to further reduce the complexity to $\mathcal{O}\left(\mathfrak{z}_k N \log(\mathfrak{m}_k)\right)$. Furthermore, in contrast to data association based methods, the proposed approach admits numerous other computational enhancements, since the map PHD update can be segmented, executed in parallel and subsequently fused for state estimation. This is in contrast to DA based approaches which are scalable.

6.4.3 Outdoor Experiments

6.4.3.1 Land Based SLAM with MMW Radar

This section discusses the performance of the proposed framework, using a millimetre wave radar SLAM dataset in a university car park environment. Millimetre wave radar offers numerous advantages over standard laser-based systems, returning a power vs. range spectrum. This allows for customised detection algorithms to be developed, however it can be prone to high clutter

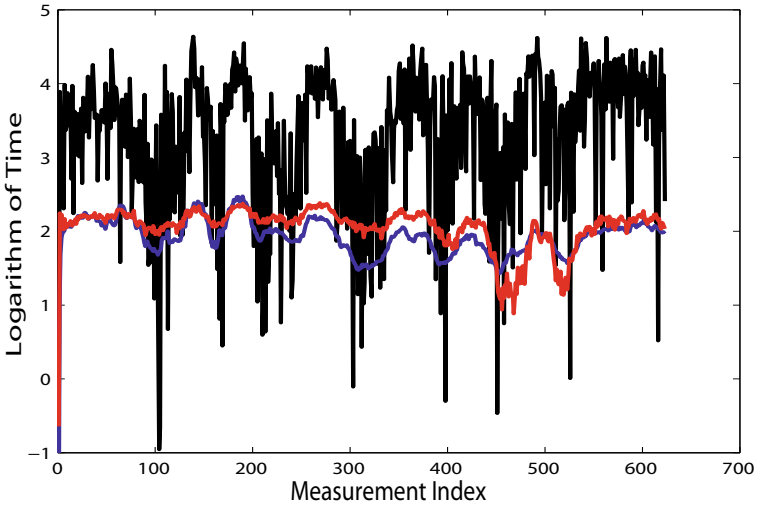


Fig. 6.12 A comparison of the computation time per measurement update for RB-PHD-SLAM (blue), MH-FastSLAM (red) and JCBB-EKFSLAM (black).

rates [16]. The extracted radar point clusters, plotted relative to the odometry only pose estimates of the vehicle, as well as the odometry pose estimates themselves are depicted in figure 6.13 (left). The information displayed in this figure can be thought of as the information input to the SLAM algorithms, which must be processed to yield the best estimated trajectory and map. Given the tree coverage and surrounding buildings in the area, GPS is generally not available. Ground truth was thus obtained by manually matching successive scans from a laser range finder which was also mounted on the vehicle, with graphical verification also provided in figure 6.13 (right). The vehicle was driven at approximately 1.5m/s around 3 loops, with a control input frequency of 10Hz and a radar measurement frequency of 2.5Hz. The car park environment is comprised of buildings, bushes, trees, fire hydrants, curbs, medians, a car etc.

Given the small-sized loop, the maximum range of the radar was set at 15m and both FastSLAM, with maximum likelihood data association, and RB-PHD-SLAM were executed on the dataset. Figure 6.14 depicts the posterior estimated trajectory and map using the FastSLAM algorithm (left) and that from RB-PHD-SLAM (right), given the same control input samples. Given the noisy measurements from the radar sensor, the merits of the proposed approach are demonstrated. It should be noted that, as is the case with any experimental dataset, the ground truth feature map is extremely difficult to obtain, making it challenging to evaluate the feature map estimation error.

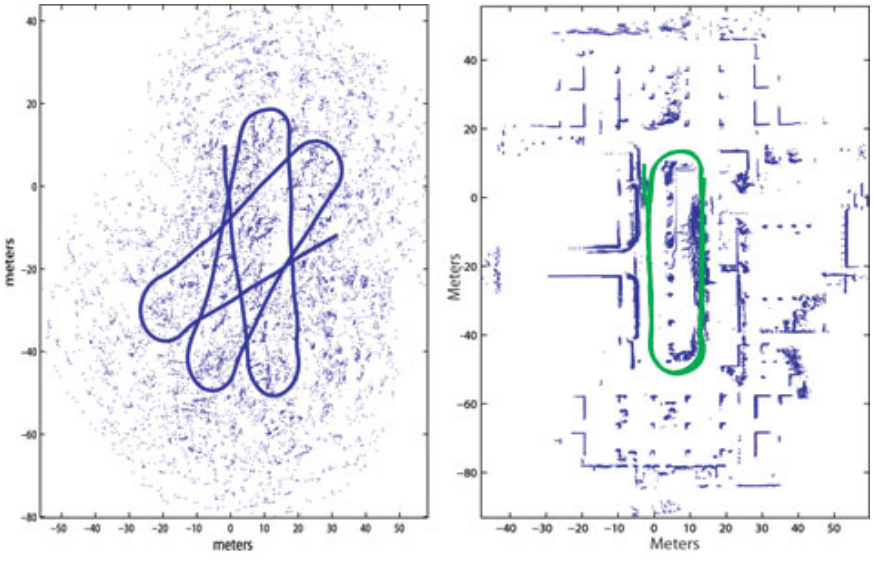


Fig. 6.13 Left: Odometery and extracted clusters from the radar data, representing the raw inputs to the SLAM algorithms. Right: The ground truth trajectory (green) obtained by matching laser data due to a lack of GPS data.

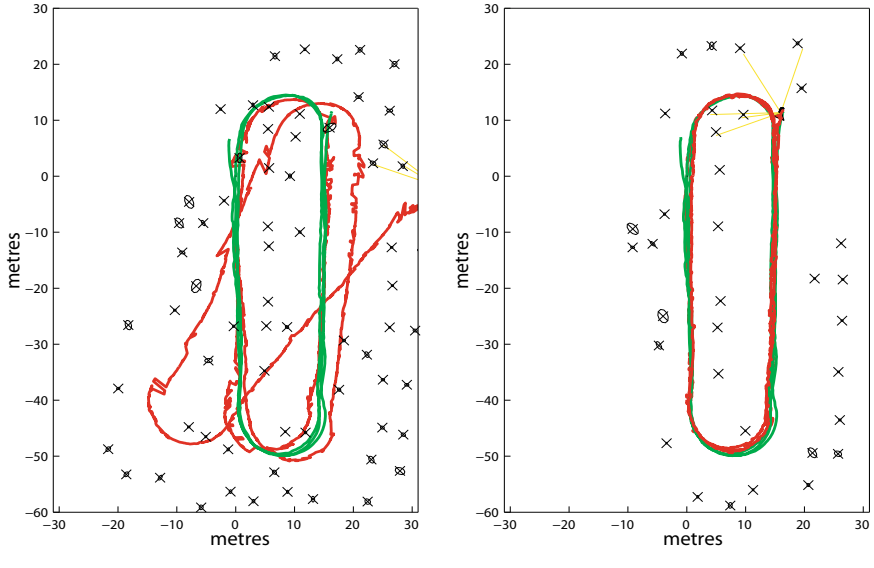


Fig. 6.14 Left: The posterior estimate from FastSLAM using the radar-based car park dataset. Right: The posterior estimate from RB-PHD-SLAM using the same dataset. The proposed integrated Bayesian framework for SLAM, incorporating DA and feature management enhances the robustness of the SLAM algorithm given noisy measurements.

6.4.3.2 Sea Based SLAM with X-Band Radar

This section discusses the filter's performance in a surface-based marine environment, using an X-band radar mounted on a powerboat. In order to maximise the detection of all sea surface point features (comprising boats, buoys, etc.), a low detection threshold is required, which subsequently increases the clutter rate. GPS data is available for measuring the ground truth trajectory, while sea charts and data from surrounding vessels' Automatic Identification Systems provide the feature map ground truth. The test site is off the Southern coast of Singapore, as shown in Figure 6.15, where the boat was driven in a loop trajectory of 13Km. Adaptive thresholding methods were applied to extract relative point measurements from the radar data [59]. The maximum range of the radar, logging at 0.5Hz, was limited to 1Km. While heading measurements were available via a low grade on-board single axis gyroscope, due to the lack of Doppler velocity logs, the speed was estimated at 8 knots (4.1 m/s).

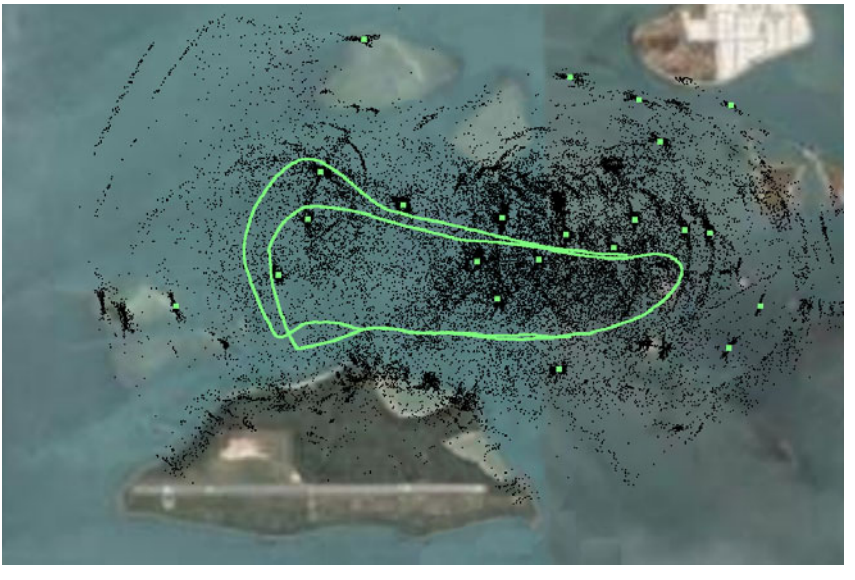


Fig. 6.15 Overview of the test site ($1^o13'$ N,$103^o43'$ E), showing the GPS trajectory (green line) and GPS coordinates-ordinates (green dots) of the point feature map. The point feature measurement history is also provided (black dots).

Figure 6.16 compares the posterior SLAM estimates from MH-FastSLAM and RB-PHD-SLAM, with Figure 6.17 comparing the estimated map sizes. The proposed approach can be seen to generate more accurate localisation and feature number estimates, however it can also be seen that some feature

estimates are misplaced in comparison to the ground truth feature map. The framework is still demonstrated to be useful for high clutter feature-based SLAM applications.

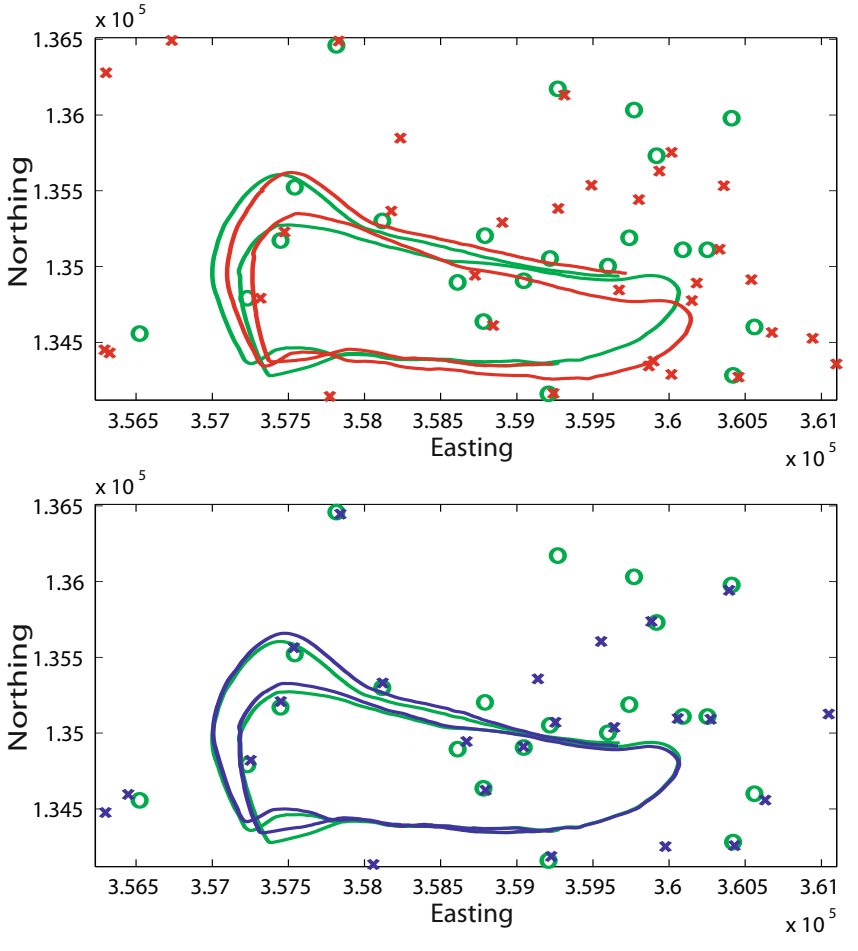


Fig. 6.16 Top: The posterior SLAM estimate (red) from MH-FastSLAM and Bottom: The posterior SLAM estimate (blue) from RB-PHD-SLAM, in comparison to the ground truth (green).

6.5 Summary

This chapter presented a tractable solution for the feature-based SLAM problem. The finite set representation of the map admits the notion of an expected

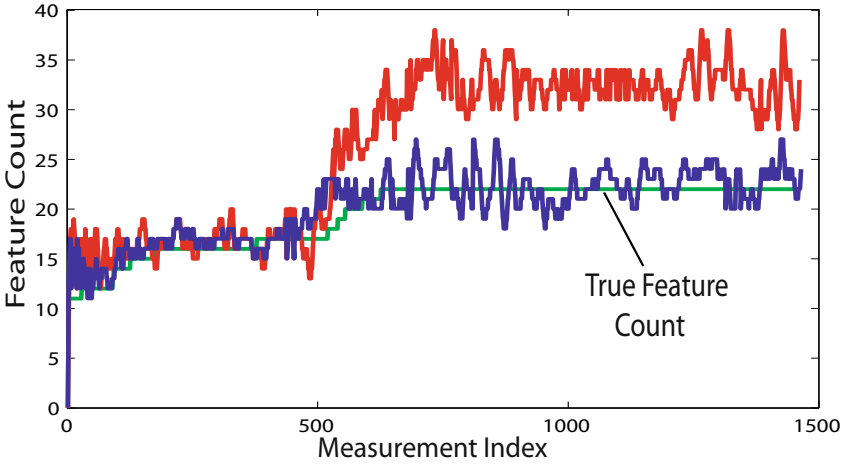


Fig. 6.17 Comparison of the number of estimated features for each approach. The noisy estimates are likely due to deviations from the Poisson clutter assumption in places.

map in the form of a PHD or intensity function. A Rao-Blackwellised implementation of the filter was proposed, in which the PHD of the map was propagated using a Gaussian mixture PHD filter, and a particle filter propagated the vehicle trajectory density. A closed form solution for the trajectory weighting was also presented, alleviating the need for approximation, which is commonly used.

Analysis was carried out, both in a simulated environment through Monte Carlo trials and an outdoor SLAM experimental dataset based on a millimetre wave radar sensor. Results demonstrated the robustness of the proposed filter, particularly in the presence of large data association uncertainty and clutter, illustrating the merits of adopting an RFS approach to SLAM.

In terms of its computational complexity, the Rao-Blackwellised SLAM filter was shown to be linear in the number of estimated features, measurements and trajectory particles. It should be noted that computational enhancements are possible, in terms of parallelisable operations, which are not possible with vector based approaches requiring data association.

6.6 Bibliographical Remarks

The RFS approach to SLAM was first suggested in [54] with preliminary studies using 'brute force' implementations shown in Chapter 5. The approach modelled the joint vehicle trajectory and map as a single RFS, and recursively propagated its first order moment.

Initial results of a Rao-Blackwellised (RB) implementation of the PHD-SLAM filter, were presented in [56]. This chapter extends [56], to present a more rigorous analysis of the RFS approach to SLAM, an improved version of the PHD-SLAM filter as well as real and simulated experimental results, and is an extended version of [57]. The merits of the RFS approach are demonstrated, particularly in situations of high clutter and data association ambiguity.

A factorised approach to SLAM was established in the, now well known, FastSLAM concept [58]. However, this chapter has shown that the same factorisation method applied to vectors in FastSLAM, cannot be applied to sets, since it results in invalid densities in the feature space. Therefore one of the main contributions of this chapter is a technique which allows such a factorisation to be applied to sets in a principled manner.

Chapter 7
Extensions with RFSs in SLAM

7.1 Introduction

This book demonstrates that the inherent uncertainty of feature maps and feature map measurements can be naturally encapsulated by random finite set models, and subsequently in Chapter 5 proposed the multi-feature RFS-SLAM framework and recursion of equations 5.5 and 5.6. The SLAM solutions presented thus far focussed on the joint propagation of the the first-order statistical moment or expectation of the RFS map, i.e. its Probability Hypothesis Density, v_k, and the vehicle trajectory. Recall from Chapter 3 that the integral of the PHD, which operates on a feature state space, gives the expected number of features in the map, at its maxima represent regions in Euclidean map space where features are most likely to exist.

As will be demonstrated in this chapter, the proposed RFS-SLAM framework admits numerous alternative approximations and implementations. While the PHD-SLAM approach propagates the PHD of the map (encompassing the *expected* number of features) and the vehicle trajectory, it is also possible to append the propagated map PHD with the *distribution* of the number of features, as opposed to just its mean. This can dramatically reduce the variance in the estimated number of features in the map when compared with the PHD-only approach and subsequently improve the robustness and mapping accuracy of PHD-SLAM.

In contrast to the moment approximation methods of the RFS-SLAM recursion based on the PHD, a direct approximation of the multi-feature RFS-SLAM recursion can also be constructed and propagated. Indeed, by adopting a multi-Bernoulli representation of the RFS map, each feature's individual probability of existence and probability density, as well as the vehicle trajectory can be propagated. By modelling each feature's existence probability, this approach is conceptually analogous to existing vector-based SLAM approaches which also attempt to incorporate individual feature existence estimates.

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 127–136.
springerlink.com

7.2 Alternative RFS Map Representations

This section presents alternative RFS models for a feature map. The PHD-SLAM and RB-PHD-SLAM approaches presented thus far, approximate the prior and predicted RFS map densities by Poisson RFSs. Recall from section 3.3.4.4, that the Poisson RFS can be completely characterised by its PHD,

$$p_k(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod_{m\in\mathcal{M}_k} v_k(m|X_{0:k})}{\exp(\int v_k(m|X_{0:k})dm)}. \tag{7.1}$$

Since the mass of the PHD ($\int v_k(m|X_{0:k})dm$) is a real number, the cardinality of a Poisson RFS map is captured by the single parameter of the denominator term, $\exp(\int v_k(m|X_{0:k})dm)$, that is, the distribution of the size of the map is also approximated by a Poisson distribution. Since the mean and variance of a Poisson distribution are the same, large variations are likely in the estimated number of features for high density maps and/or sensors with large FoVs.

7.2.1 The Independent and Identically Distributed (IID) Cluster RFS

A more general approximation of the map RFS is an IID cluster process,

$$p_k(\mathcal{M}_k|X_{0:k}) \approx \frac{\mathfrak{m}!\rho_k(\mathfrak{m}) \prod_{m\in\mathcal{M}_k} v_k(m|X_{0:k})}{\left(\int v_k(m|X_{0:k})dm\right)^{\mathfrak{m}}} \tag{7.2}$$

where, as before $\mathfrak{m}$ is the cardinality of the map $\mathcal{M}_k$ and now $\rho_k(\mathfrak{m})$ is the distribution of the size of the feature map. The approximation generalises the Poisson RFS representation to allow for any arbitrary cardinality distribution, constrained only by the property $\sum_{n=0}^{\infty} n\rho_k(n) = \int v_k(m|X_{0:k})dm$. It can be seen that by replacing $\rho_k(\mathfrak{m})$ with a Poisson distribution, (7.2) reduces to (7.1). Thus the spatial randomness of the features is still encapsulated by a Poisson RFS, but the distribution of the number of features is not restricted to a Poisson distribution. The subsequent SLAM formulation based on this map approximation is presented in section 7.3.1.

7.2.2 The Multi-Bernoulli RFS

Recall from Chapter 4, that an individual feature measurement was modelled as a Bernoulli RFS, that is, (1) the measurement is not received, with a probability equal to that of a missed detection $(1 - P_D)$ (2) the measurement

is received with a probability equal to the detection probability of that feature and when it is received, has a measurement likelihood $P_D g_k(z|m, X_k)$. Equivalently, an individual map feature can be represented by a Bernoulli RFS which,

1. does not exist with a probability equal to that of its non-existence probability, $(1 - P_E)$,
2. exists with a probability equal to its existence probability P_E and when it does exist, has a spatial likelihood $p_k(m|\mathcal{Z}_k, X_{0:k})$.

Generalising to a feature map comprising $\mathfrak{m}_k$ features, the multi-Bernoulli RFS can be written as

$$p_k(\mathcal{M}_k = \emptyset) = \prod_{j=1}^{\mathfrak{m}_k} \left(1 - P_E^{(j)}\right) \tag{7.3}$$

and in the case of any or all n features existing,

$$p_k(\mathcal{M}_k \approx \{m^1, ..., m^n\}) = p_k(\emptyset) \sum_{1 \leq i_1 \neq \cdots \neq i_n \leq \mathfrak{m}_k} \prod_{j=1}^{n} \frac{P_E^{(i_j)} p_k^{(i_j)}(m^j|\mathcal{Z}_k, X_{0:k})}{1 - P_E^{(i_j)}}. \tag{7.4}$$

Note that this map representation is an approximation of the multi-feature map posterior density as opposed to a moment-based approximation of other approaches. The subsequent SLAM formulation based on this map approximation is presented in section 7.3.2. Furthermore, while the moment approximation methods encapsulate the mean (for the Poisson map model) and distribution (for the IDD cluster model) of the number of features in the map, the Multi-Bernoulli representation is analogous to popular vector based SLAM approaches [58], [1], [15] in that each feature has its own probability of existence. However, by using this RFS representation, the mathematical ambiguity highlighted previously in section 2.5 is overcome as the density for each feature jointly incorporates the existence and non-existence probabilities and still integrates to unity implying it is a valid density function.

7.3 Extended RB-RFS-SLAM Formulations

This section details the derivations of the appropriate equations necessary for Rao-Blackwellised SLAM implementations based on the extended RFS map models of the previous section.

7.3.1 RB Cardinalised PHD-SLAM

In the case of the IID cluster RFS map representation of section 7.2.1, the so-called Cardinalised PHD (CPHD) filter [51] can be readily modified to propagate the map PHD *and cardinality distribution* forward in time as measurements arrive. If a set of N weighted particles comprising,

$$\left\{\text{Particle Weight, Trajectory Sample, Map PHD, Map Cardinality Dist.}\right\}_{i=1}^{N}$$

i.e.

$$\left\{\eta_{k-1}^{(i)},\ \ X_{0:k-1}^{(i)},\ \ v_{k-1}^{(i)}(m|X_{0:k-1}^{(i)}),\ \ \rho_{k-1}^{(i)}(n|X_{0:k-1}^{(i)})\right\}_{i=1}^{N} \tag{7.5}$$

is available at time $k-1$, then given a vehicle input, U_{k-1} and measurement, $\mathcal{Z}_k$, the RB-CPHD-SLAM filter generates an updated set of particles,

$$\left\{\eta_{k}^{(i)},\ \ X_{0:k}^{(i)},\ \ v_{k}^{(i)}(m|X_{0:k}^{(i)}),\ \ \rho_{k}^{(i)}(n|X_{0:k}^{(i)})\right\}_{i=1}^{N} \tag{7.6}$$

whereby both the PHD (intensity function) v *and* the cardinality distribution ρ are updated through predictor – corrector equations via the following processes.

7.3.1.1 Map Update

If the predicted vehicle pose, $\tilde{X}_k^{(i)}$, is sampled from the vehicle transition density $f_X(\tilde{X}_k|X_{k-1},U_{k-1})$ and, as introduced previously in Chapter 4, if the prior, $v_{k-1}^{(i)}(m|X_{k-1}^{(i)})$ and birth, $b_k^{(i)}(m|\tilde{X}_k^{(i)})$, PHDs are available for each particle, then for each trajectory sample, the predicted map PHD is,

$$v_{k|k-1}^{(i)}(m|\tilde{X}_k^{(i)}) = v_{k-1}^{(i)}(m|X_{k-1}^{(i)}) + b_k^{(i)}(m|\tilde{X}_k^{(i)}). \tag{7.7}$$

In addition, if the prior, $\rho_{k-1}^{(i)}(n|X_{k-1}^{(i)})$ and birth cardinality distributions $\rho_{b,k}^{(i)}(n|\tilde{X}_k^{(i)})$ are available, then the predicted map cardinality distribution must also be evaluated. Since robotic feature mapping has been formulated as a finite-set estimation problem in this book, which was solved by modifying the PHD framework in previous chapters, a similar concept can be adopted such that the CPHD filtering framework [68], [51], can be modified to solve the feature mapping problem in the case of an IID cluster RFS map. Thus, the predicted map cardinality distribution can be obtained from,

$$\rho^{(i)}_{k|k-1}(n|\tilde{X}^{(i)}_k) = \sum_{j=0}^{n}\left(\rho^{(i)}_{b,k}(n-j|\tilde{X}^{(i)}_k)\sum_{l=j}^{\infty} C^l_j\Big(\int v^{(i)}_{k-1}(m|X^{(i)}_{k-1})dm\Big)^{l-j}\rho^{(i)}_{k-1}(l|X^{(i)}_{k-1})\right) \tag{7.8}$$

where C^l_i denotes the binomial coefficient.

Similarly, given the RFS measurement at time k, $\mathcal{Z}_k$, the updated map PHD is then given by,

$$\begin{aligned} v^{(i)}_k(m|\tilde{X}^{(i)}_k) = {} & \frac{\sum_{n=0}^{\infty} \Gamma^1_k\big[v^{(i)}_{k|k-1}(m|\tilde{X}^{(i)}_k),\mathcal{Z}_k\big](n)\rho^{(i)}_{k|k-1}(n|\tilde{X}^{(i)}_k)}{\sum_{n=0}^{\infty} \Gamma^0_k\big[v^{(i)}_{k|k-1}(m|\tilde{X}^{(i)}_k),\mathcal{Z}_k\big](n)\rho^{(i)}_{k|k-1}(n|\tilde{X}^{(i)}_k)} \\ & \times(1-P_D)v^{(i)}_{k|k-1}(m|\tilde{X}^{(i)}_k) \\ & +\Bigg(\sum_{z\in\mathcal{Z}_k}\frac{\sum_{n=0}^{\infty} \Gamma^1_k\big[v^{(i)}_{k|k-1}(m|\tilde{X}^{(i)}_k),\mathcal{Z}_k/\{z\}\big](n)\rho^{(i)}_{k|k-1}(n|\tilde{X}^{(i)}_k)}{\sum_{n=0}^{\infty} \Gamma^0_k\big[v^{(i)}_{k|k-1}(m|\tilde{X}^{(i)}_k),\mathcal{Z}_k\big](n)\rho^{(i)}_{k|k-1}(n|\tilde{X}^{(i)}_k)} \\ & \times\psi_{z,k}(m)v^{(i)}_{k|k-1}(m|\tilde{X}^{(i)}_k)\Bigg) \end{aligned} \tag{7.9}$$

where,

$$\psi_{z,k}(m) = \frac{\int c_k(z|\tilde{X}^{(i)}_k)dz}{c_k(z|\tilde{X}^{(i)}_k)} \times P_D(m|\tilde{X}^{(i)}_k)g_k(z|m,\tilde{X}^{(i)}_k) \tag{7.10}$$

and,

$$\begin{aligned} & \Gamma^q_k\big[v_{k|k-1}(m|\tilde{X}_k),\mathcal{Z}_k\big](n) = \\ & \sum_{j=0}^{\min(\mathfrak{z}_k,n)}(\mathfrak{z}_k-j)!\rho_{\kappa,k}(\mathfrak{z}_k-j)|\tilde{X}_k)P^n_{j+q}\frac{\big(\int(1-P_D)v_{k|k-1}(m|\tilde{X}_k)dm\big)^{n-(j+q)}}{\big(\int v_{k|k-1}(m|\tilde{X}_k)dm\big)^{n}} \\ & \qquad\qquad \times e_j(\Xi(v_{k|k-1},\mathcal{Z}_k)). \end{aligned} \tag{7.11}$$

Here, $\mathfrak{z}_k$ represents the number of features observed at time k, P^n_{j+q} denotes the permutation coefficient, $e_j(\cdot)$ denotes the elementary symmetric function of order j and,

$$\Xi(v_{k|k-1},\mathcal{Z}_k) = \{\int v_{k|k-1}(m|\tilde{X}_k)\psi_{z,k}(m)dm : z\in\mathcal{Z}_k\}. \tag{7.12}$$

The updated map cardinality is obtained from,

$$\rho_k^{(i)}(n|\tilde{X}_k^{(i)}) = \frac{\Gamma_k^0\left[v_{k|k-1}^{(i)}(m|\tilde{X}_k^{(i)}), \mathcal{Z}_k\right](n)\rho_{k|k-1}^{(i)}(n|\tilde{X}_k^{(i)})}{\sum_{n=0}^{\infty}\Gamma_k^0\left[v_{k|k-1}^{(i)}(m|\tilde{X}_k^{(i)}), \mathcal{Z}_k\right](n)\rho_{k|k-1}^{(i)}(n|\tilde{X}_k^{(i)})} \tag{7.13}$$

7.3.1.2 Trajectory Update

If, as before, the predicted vehicle pose, $\tilde{X}_k^{(i)}$, is sampled from the vehicle transition density, $f_X(\tilde{X}_k|X_{k-1}, U_{k-1})$, then the updated weight is calculated according to,

A. The Empty Strategy

$$\tilde{\eta}_k^{(i)} = \frac{\mathfrak{z}_k!\rho_k(\mathfrak{z}_k|\tilde{X}_k^{(i)})\kappa_k^{\mathcal{Z}_k}}{\left(\int c_k(z|\tilde{X}_k^{(i)})dz\right)^{\mathfrak{z}_k}} \times \frac{\rho_{k|k-1}(0|\tilde{X}_{0:k}^{(i)})}{\rho_k(0|\tilde{X}_{0:k}^{(i)})}\eta_{k-1}^{(i)} \tag{7.14}$$

where, as before, $\kappa_k^{\mathcal{Z}_k} = \prod_{z\in\mathcal{Z}_k} c_k(z|X_k)$ and c_k is the PHD of the RFS clutter measurement. Since the map strategy is empty, there are assumed no features present for the update and thus the cardinality distribution is written as, $\rho_k(0|\cdot)$.

B. The Single Feature Strategy

$$\begin{aligned}\tilde{\eta}_k^{(i)} = &\frac{\left(1 - P_D(\bar{m}|\tilde{X}_k^{(i)})\right)\kappa_k^{\mathcal{Z}_k} + P_D(\bar{m}|\tilde{X}_k^{(i)})\sum_{z\in\mathcal{Z}_k}\kappa_k^{\mathcal{Z}_k - z}g_k(z|\bar{m}, \tilde{X}_k^{(i)})}{\exp\left(\int c_k(z|\tilde{X}_k^{(i)})dz\right)} \\ &\times \frac{\rho_{k|k-1}^{(i)}(1|\tilde{X}_{0:k}^{(i)})v_{k|k-1}^{(i)}(\bar{m}|\tilde{X}_{0:k}^{(i)})\int v_k^{(i)}(m|\tilde{X}_{0:k}^{(i)})dm}{\rho_k^{(i)}(1|\tilde{X}_{0:k}^{(i)})v_k^{(i)}(\bar{m}|\tilde{X}_{0:k}^{(i)})\int v_{k|k-1}^{(i)}(m|\tilde{X}_{0:k}^{(i)})dm}\eta_{k-1}^{(i)}\end{aligned} \tag{7.15}$$

where, as before, $\bar{m}$, is a feature selected according to a given strategy (i.e. least uncertainty, highest measurement likelihood etc.). As before, since the strategy is a that of a single feature, the cardinality distribution is written as, $\rho_k(1|\cdot)$.

7.3.1.3 Estimator

As with the RB-PHD-SLAM filter, a MAP estimate of the posterior trajectory density can be taken to estimate the vehicle pose at each time. To estimate the map, EAP or MAP estimates of the number of features in the

map can be extracted from the chosen particle's posterior cardinality distribution. Conditioned on this estimate, as with the RB-PHD-SLAM filter, the feature estimates can be chosen from the peaks of the posterior map PHD. Given that the CPHD filter propagates the *distribution* of the number of features as opposed to just its mean, it is anticipated that the map, and subsequently the trajectory, estimates from RB-CPHD-SLAM would be remarkably improved in comparison to RB-PHD-SLAM. This is an avenue for further research.

7.3.2 RB Multi-target Multi-Bernoulli (MeMBer) SLAM

In the case of the Multi-Bernoulli RFS map representation of section 7.2.2, the so-called Cardinalised Multi-target Multi-Bernoulli (CMeMBer) filter [69] can be readily modified to propagate the *map density* forward in time as measurements arrive. If a set of weighted particles,

$$\left\{\eta_{k-1}^{(i)},\ X_{0:k-1}^{(i)},\ \{\big(P_{E,k-1}^{(i,j)},\ p_{k-1}^{(j)}(m|X_{0:k-1}^{(i)})\big)\}_{j=1}^{J_{k-1}^{(i)}}\right\}_{i=1}^{N} \tag{7.16}$$

comprising the particle weight, trajectory sample, the existence probability and spatial density of each feature in each trajectory's map, is available at time $k-1$, then given a vehicle input, U_{k-1} and measurement, $\mathcal{Z}_k$, the RB-MeMBer-SLAM filter generates an updated set of particles,

$$\left\{\eta_{k}^{(i)},\ X_{0:k}^{(i)},\ \{\big(P_{E,k}^{(i,j)},\ p_{k}^{(j)}(m|X_{0:k}^{(i)})\big)\}_{j=1}^{J_{k}^{(i)}}\right\}_{i=1}^{N} \tag{7.17}$$

via the following processes.

7.3.2.1 Map Update

If the predicted vehicle pose, $\tilde{X}_k^{(i)}$, is sampled from the vehicle transition density, $f_X(\tilde{X}_k|X_{k-1},U_{k-1})$, and the parameters of the prior, $\pi_{k-1} = \{\big(P_{E,k-1}^{(i,j)},p_{k-1}^{(j)}(m|X_{0:k-1}^{(i)})\big)\}_{j=1}^{J_{k-1}^{(i)}}$ and birth, $\{\big(P_{b,E,k}^{(i,j)},p_{b,k}^{(j)}(m|\tilde{X}_k^{(i)})\big)\}_{j=1}^{J_{b,k}^{(i)}}$, multi-Bernoulli RFSs are available for each particle, then for each trajectory sample, the parameters of the predicted map are obtained from,

$$\pi_{k|k-1}^{(i)} = \pi_{k-1}^{(i)} \bigcup \{\big(P_{b,E,k}^{(i,j)},p_{b,k}^{(j)}(m|\tilde{X}_k^{(i)})\big)\}_{j=1}^{J_{b,k}^{(i)}}. \tag{7.18}$$

Again, since the feature mapping problem is cast into a finite-set estimation framework, related filters in the tracking community [3], [69], can be modified to perform the map update. Therefore, given the RFS measurement at time k, $\mathcal{Z}_k$, the updated map for each sample trajectory particle, $\tilde{X}_k^{(i)}$, is then given by,

$$\pi_k^{(i)} = \{(P_{E,k}^{(i,j)}, p_k^{(j)}(m|\tilde{X}_{0:k}^{(i)}))\}_{j=1}^{J_k^{(i)}} \tag{7.19}$$

where, $J_k^{(i)} = J_{k-1}^{(i)} + J_{b,k}^{(i)} + \mathfrak{z}_k$. If $J_{k|k-1}^{(i)} = J_{k-1}^{(i)} + J_{b,k}^{(i)}$, then for $j = 1, \ldots, J_{k|k-1}^{(i)}$,

$$P_{E,k}^{(i,j)} = P_{E,k-1}^{(i,j)} \frac{1 - \int P_D(m|\tilde{X}_k^{(i)}) p_{k-1}^{(j)}(m|\tilde{X}_k^{(i)}) dm}{1 - P_{E,k-1}^{(i,j)} \int P_D(m|\tilde{X}_k^{(i)}) p_{k-1}^{(j)}(m|\tilde{X}_k^{(i)}) dm} \tag{7.20}$$

$$p_k^{(j)}(m|\tilde{X}_{0:k}^{(i)}) = \frac{(1 - P_D(m|\tilde{X}_k^{(i)})) p_{k-1}^{(j)}(m|X_{0:k-1}^{(i)})}{1 - \int P_D(m|\tilde{X}_k^{(i)}) p_{k-1}^{(j)}(m|X_{0:k-1}^{(i)}) dm} \tag{7.21}$$

and for $j = J_{k|k-1}^{(i)} + 1, \ldots, J_k^{(i)}$,

$$P_{E,k}^{(i,j)} = \frac{\sum_{l=1}^{J_{k|k-1}^{(i)}} \frac{P_{E,k-1}^{(i,l)}(1 - P_{E,k-1}^{(i,l)}) \int P_D(m|\tilde{X}_k^{(i)}) p_{k-1}^{(l)}(m|\tilde{X}_{0:k}^{(i)}) g_k(z|m, \tilde{X}_k^{(i)}) dm}{\left(1 - P_{E,k-1}^{(i,l)} \int P_D(m|\tilde{X}_k^{(i)}) p_{k-1}^{(l)}(m|\tilde{X}_{0:k}^{(i)}) dm\right)^2}}{\kappa_k + \sum_{l=1}^{J_{k|k-1}^{(i)}} \frac{P_{E,k-1}^{(i,l)} \int P_D(m|\tilde{X}_k^{(i)}) p_{k-1}^{(l)}(m|\tilde{X}_{0:k}^{(i)}) g_k(z|m, \tilde{X}_k^{(i)}) dm}{1 - P_{E,k-1}^{(i,l)} \int P_D(m|\tilde{X}_k^{(i)}) p_{k-1}^{(l)}(m|\tilde{X}_{0:k}^{(i)}) dm}} \tag{7.22}$$

$$p_k^{(j)}(m|\tilde{X}_{0:k}^{(i)}) = \frac{\sum_{l=1}^{J_{k|k-1}^{(i)}} \frac{P_{E,k-1}^{(i,l)}}{(1 - P_{E,k-1}^{(i,l)})} P_D(m|\tilde{X}_k^{(i)}) g_k(z|m, \tilde{X}_k^{(i)}) p_{k-1}^{(l)}(m|\tilde{X}_{0:k}^{(i)})}{\sum_{l=1}^{J_{k|k-1}^{(i)}} \frac{P_{E,k-1}^{(i,l)}}{(1 - P_{E,k-1}^{(i,l)})} \int P_D(m|\tilde{X}_k^{(i)}) g_k(z|m, \tilde{X}_k^{(i)}) p_{k-1}^{(l)}(m|\tilde{X}_{0:k}^{(i)}) dm} \tag{7.23}$$

7.3.2.2 Trajectory Update

If the predicted vehicle pose, $\tilde{X}_k^{(i)}$, is sampled from the vehicle transition density, $f_X(\tilde{X}_k|X_{k-1}, U_{k-1})$, then the updated weight is calculated according to,

A. The Empty Strategy

$$\eta_k^{(i)} = \frac{\kappa_k^{\mathcal{Z}_k}}{\exp\left(\int c_k(z|\tilde{X}_k^{(i)})dz\right)} \frac{\prod_{j=1}^{\mathfrak{m}_{k|k-1}^{(i)}}\left(1-P_{E,k-1}^{(i,j)}\right)}{\prod_{j=1}^{\mathfrak{m}_k^{(i)}}\left(1-P_{E,k}^{(i,j)}\right)}\eta_{k-1}^{(i)}. \tag{7.24}$$

B. The Single Feature Strategy

$$\eta_k^{(i)} = \frac{\left(1-P_D(\bar{m}|\tilde{X}_k^{(i)})\right)\kappa_k^{\mathcal{Z}_k} + P_D(\bar{m}|\tilde{X}_k^{(i)})\sum_{z\in\mathcal{Z}_k}\kappa_k^{\mathcal{Z}_k-z}g_k(z|\bar{m},\tilde{X}_k^{(i)})}{\exp\left(\int c_k(z|\tilde{X}_k^{(i)})dz\right)}$$
$$\times\frac{1}{\Gamma}\prod_{j=1}^{\mathfrak{m}_{k|k-1}^{(i)}}\left(1-P_{E,k-1}^{(i,j)}\right)\times\frac{1}{\Gamma}\sum_{j=1}^{\mathfrak{m}_{k|k-1}^{(i)}}\frac{P_{E,k-1}^{(i,j)}p_{k|k-1}^{(j)}(\bar{m}|\tilde{X}_{0:k}^{(i)})}{1-P_{E,k-1}^{(i,j)}}\eta_{k-1}^{(i)}$$

where,

$$\Gamma = \prod_{j=1}^{\mathfrak{m}_k^{(i)}}\left(1-P_{E,k}^{(i,j)}\right)\sum_{j=1}^{\mathfrak{m}_k^{(i)}}\frac{P_{E,k}^{(i,j)}p_k^{(j)}(\bar{m}|\tilde{X}_{0:k}^{(i)})}{1-P_{E,k}^{(i,j)}} \tag{7.25}$$

7.3.2.3 Estimators

As with the RB-PHD-SLAM filter, an MAP estimate of the posterior trajectory density can be used to estimate the vehicle pose at each time. Given that the posterior existence probability of each feature is available, feature map estimate can be obtained from the means or modes of the posterior densities of each hypothesised map state with an existence probability greater than a given threshold. Using a Bernoulli RFS to represent each feature allows for the joint encapsulation of its existence probability and location in a single PDF, in contrast to existing SLAM methods [1], [58], [18]. It is expected that RB-MeMBer-SLAM would perform well in the presence of highly non-linear process and/or measurement models.

7.4 Summary

This chapter has detailed enhancements to the PHD-SLAM framework proposed in earlier chapters by exploiting alternative RFS map approximations and developing suitable filters which can be readily implemented via RB

techniques. Firstly, the RB-CPHD-SLAM filter was formulated, which relaxes the Poisson distribution assumption on the number of features in the map, and propagates the map cardinality distribution as opposed to just its mean. Indeed, since the Poisson RFS is a special case of the IID cluster RFS (with a Poisson cardinality distribution), the RB-PHD-SLAM filter can be regarded as a special case of the RB-CPHD-SLAM filter. The formulation also included the weighting functions necessary for a RB implementation as well as suitable methods of state estimation.

The RB-MeMBer-SLAM filter adopted a Multi-Bernoulli RFS map representation, which jointly encapsulates both the existence probability and the spatial density of each feature. In essence, it is conceptually equivalent to how the vast majority of existing vector-based SLAM approaches process uncertainty in the size of the feature map. However, since an RFS model is adopted, the tools of RFS filtering techniques are required to develop a suitable filter. The RB-MeMBer-SLAM filter directly approximates the joint multi-feature vehicle trajectory posterior as opposed to the moment approximation methods based on the PHD.

Appendix A
Concatenation of the Feature State m with Hypothesised Vehicle Trajectories – Campbell's Theorem

If $\mathbb{L}$ denotes the space of features and $\mathbb{K}$ denotes the space of vehicle states, Campbell's theorem [70] implies that the intensity of the point process on $\mathbb{L} \times \mathbb{K}$ formed by the Cartesian product of a point process on $\mathbb{L}$, with intensity $\tilde{v}$, and a point process on the mark space (a vehicle pose particle) $\mathbb{K}$, is

$$v(X_k, m) = p(X_k|m)\tilde{v}(m), \tag{A.1}$$

where $p(X_k|m)$ is the mark distribution given a point m of the original point process on $\mathbb{L}$. Moreover, if the point process on $\mathbb{L}$ (the set of features) is Poisson, then the product point process on $\mathbb{L} \times \mathbb{K}$ is also Poisson [70]. As the RFS of the joint vehicle and map state is therefore Poisson, the derivation established in [2] can be incorporated into this work to include the joint vehicle-feature, augmented state.

Appendix B
Derivation of $g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})$ for the RB-PHD-SLAM Filter

Recall equation 6.6,

$$p_k(\mathcal{M}_k|X_{0:k}) = \frac{g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k)p_{k|k-1}(\mathcal{M}_k|X_{0:k})}{g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})}$$

and the Poisson RFS approximations,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod_{m\in\mathcal{M}_k} v_{k|k-1}(m|X_{0:k})}{\exp\left(\int v_{k|k-1}(m|X_{0:k})dm\right)},$$

$$p_k(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod_{m\in\mathcal{M}_k} v_k(m|X_{0:k})}{\exp\left(\int v_k(m|X_{0:k})dm\right)}.$$

B.0.1 The Empty Strategy

Rearranging, and assigning $\mathcal{M}_k = \emptyset$ gives,

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) = g_k(\mathcal{Z}_k|\emptyset, X_k) \times \frac{\prod_{m\in\mathcal{M}_k} v_{k|k-1}(m|X_{0:k})}{\prod_{m\in\mathcal{M}_k} v_k(m|X_{0:k})} \times \frac{\exp\left(\int v_k(m|X_{0:k})dm\right)}{\exp\left(\int v_{k|k-1}(m|X_{0:k})dm\right)}$$

Since, $\mathcal{M}_k = \emptyset$, the empty set measurement likelihood is that of the clutter RFS (Poisson),

$$g_k(\mathcal{Z}_k|\emptyset, X_k) = \frac{\prod_{z\in\mathcal{Z}_k} c_k(z|X_k)}{\exp(\int c_k(z|X_k)dz)}.$$

The PHDs $v_{k|k-1}$ and v_k are empty, implying their product is 1, $\hat{\mathfrak{m}}_{k|k-1} = \int v_{k|k-1}(m|X_{0:k})dm$ and $\hat{\mathfrak{m}}_k = \int v_k(m|X_{0:k})dm$, giving,

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) = \prod_{z\in\mathcal{Z}_k} c_k(z|X_k)\exp\left(\hat{\mathfrak{m}}_k - \hat{\mathfrak{m}}_{k|k-1} - \int c_k(z|X_k)dz\right).$$

Note that while for the empty map choice, the likelihood $g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})$ does not contain a measurement likelihood term $g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k)$, the history of measurements and trajectories are incorporated into the predicted and updated intensity terms, whose integrals appear as the terms $\hat{\mathfrak{m}}_{k|k-1}$ and $\hat{\mathfrak{m}}_k$ respectively.

B.0.2 The Single Feature Strategy

Assigning $\mathcal{M}_k = \{\bar{m}\}$, with $\bar{m}$ being chosen according to a given strategy,

$$\begin{aligned} g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) &= g_k(\mathcal{Z}_k|\bar{m}, X_k) \times \frac{\prod_{m\in\mathcal{M}_k} v_{k|k-1}(m|X_{0:k})}{\prod_{m\in\mathcal{M}_k} v_k(m|X_{0:k})} \times \frac{\exp\left(\hat{\mathfrak{m}}_k\right)}{\exp\left(\hat{\mathfrak{m}}_{k|k-1}\right)} \\ &= g_k(\mathcal{Z}_k|\bar{m}, X_k) \times \frac{v_{k|k-1}(\bar{m}|X_{0:k})}{v_k(\bar{m}|X_{0:k})} \times \exp\left(\hat{\mathfrak{m}}_k - \hat{\mathfrak{m}}_{k|k-1}\right) \end{aligned}$$

If $\mathcal{M}_k = \{\bar{m}\}$, thus from (5.4),

$$\begin{aligned} g_k(\mathcal{Z}_k|\bar{m}, X_k) = &\frac{(1 - P_D(\bar{m}|X_k)) \prod_{z\in\mathcal{Z}_k} c_k(z|X_k)}{\exp(\int c_k(z|X_k)dz)} + \\ &P_D(\bar{m}|X_k)\frac{\sum_{z\in\mathcal{Z}_k}\left(\prod_{z\in\mathcal{Z}_k-\bar{z}} c_k(z|X_k)\right) g(z|\bar{m}, X_k)}{\exp(\int c_k(z|X_k)dz)} \end{aligned}$$

then,

$$\begin{aligned} g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) = &\left(\frac{(1 - P_D(\bar{m}|X_k)) \prod_{z\in\mathcal{Z}_k} c_k(z|X_k)}{\exp(\int c_k(z|X_k)dz)} + \right. \\ &\left. P_D(\bar{m}|X_k)\frac{\sum_{z\in\mathcal{Z}_k}\left(\prod_{z\in\mathcal{Z}_k-\bar{z}} c_k(z|X_k)\right) g(z|\bar{m}, X_k)}{\exp(\int c_k(z|X_k)dz)}\right) \frac{v_{k|k-1}(\bar{m}|X_k)}{v_k(\bar{m}|X_k)\exp(\hat{\mathfrak{m}}_{k|k-1} - \hat{\mathfrak{m}}_k)} \end{aligned}$$

Appendix C
FastSLAM Feature Management

This appendix details the feature management routine developed for FastSLAM in a cluttered environment, providing it with knowledge of the detection and false alarm probabilities for a fair comparison with the RFS approach. As with standard approaches [1], tentative new features are declared for unassociated measurements. The 'existence probability' of each feature, $P_{E,k}^{(j)}$, given a 95% confidence gate and prior existence probability of $P_{E,k-1}^{(j)}$, then evolves through a binary Bayes filter according to

---

Step 1: (Obtain association details within FoV)

$$\bar{J} = \{j \in M_k | m^j \in \text{FoV } \& \ m^j \text{not associated.}\}$$

$$J = \{j \in M_k | m^j \in \text{FoV } \& \ m^j \text{associated.}\}$$

Step 2: (Calculate hit, miss and association probabilities)

$$_{FA} = \lambda_c / (R_{MAX}^2 \times \pi)$$

$$P_{miss}^{(\bar{J})} = (1 - P_D) \times P_{E,k-1}^{(\bar{J})} + P_D \times 0.05 \times P_{E,k-1}^{(\bar{J})}$$

$$P_{hit}^{(J)} = P_D \times P_{assoc}^{(J)} P_{E,k-1}^{(J)}$$

$$P_{assoc}^{(J)} = \frac{1}{2\pi} |S_k|^{-1/2} \exp(-0.5 \upsilon S_k^{-1} \upsilon^T)$$

Step 3: (Update Existence probabilities)

$$P_{E,k}^{(\bar{J})} = \frac{P_{miss}^{(\bar{J})}}{P_{miss}^{(\bar{J})} + (1 - P_{FA})(1 - P_{E,k-1}^{(\bar{J})})}$$

$$P_{E,k}^{(J)} = \frac{P_{hit}^{(J)}}{P_{hit}^{(J)} + P_{FA}(1 - P_{E,k-1}^{(J)})}$$

This ad-hoc but effective routine enhances the robustness of standard SLAM feature management as shown in figures 6.2 and 6.3 when exposed to

high clutter rates. Thus both the benchmark and proposed approach receive the same information for each filter loop. However, one of the fundamental merits of the proposed RFS framework is that feature management (and data association) are jointly incorporated into a single SLAM Bayesian update.

References

1. Dissanayake, G., Newman, P., Durrant-Whyte, H.F., Clark, S., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotic and Automation 17(3), 229–241 (2001)
2. Mahler, R.: Multi-target bayes filtering via first-order multi-target moments. IEEE Transactions on AES 4(39), 1152–1178 (2003)
3. Mahler, R.: Statistical Multisource Multitarget Information Fusion. Artech House, Boston (2007)
4. Vo, B.N., Singh, S., Doucet, A.: Sequential monte carlo methods for multi-target filtering with random finite sets. IEEE Transactions on Aerospace and Electronic Systems 41(4), 1224–1245 (2005)
5. Haworth, C.D., Saint-Pern, Y., Clark, D., Trucco, E., Petillot, Y.R.: Detection and tracking of multiple metallic objects in millimetre-wave images. Int. J. Comput. Vision 71(2), 183–196 (2007)
6. Tobias, M., Lanterman, A.: A probability hypothesis density based multi-target tracking with multiple bistatic range and doppler observations. IEE Radar Sonar and Navigation 152(3), 195–205 (2005)
7. Moravec, H., Elfes, A.E.: High resolution maps from wide angle sonar. In: Proceedings of the 1985 IEEE International Conference on Robotics and Automation, pp. 116–121 (March 1985)
8. Smith, R., Self, M., Cheeseman, P.: A stochastic map for uncertain spatial relationships. In: The Fourth International Symposium of Robotics Research, pp. 467–474 (1987)
9. Kuipers, B., Byun, Y.T.: A robust qualitative method for spatial learning in unknown environments. In: Proc. 8th National Conference on Artificial Intelligence, AAAI 1988. AAAI Press/The MIT Press (1988)
10. Gutmann, J.S., Konolige, K.: Incremental mapping of large cyclic environments. In: The Conference on Intelligent Robots and Applications (CIRA), Monterey, CA (1999)
11. Konolige, K.: Improved occupancy grids for map building. Auton. Robots 4(4), 351–367 (1997)
12. Thrun, S.: Learning occupancy grids with forward models. Autonomous Robots 15(2), 111–127 (2003)
13. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Transactions on Robotics 23(1), 34–45 (2007)
14. Makarsov, D., Durrant-Whyte, H.F.: Mobile vehicle navigation in unknown environments: a multiple hypothesis approach. In: IEE Proceedings of Contr. Theory Applict., vol. 142 (July 1995)

15. Guivant, J., Nebot, E., Baiker, S.: Autonomous navigation and map building using laser range sensors in outdoor applications. Journal of Robotic Systems 17(10), 565–583 (2000)
16. Mullane, J., Adams, M., Wijesoma, W.S.: Robotic mapping using measurement likelihood filtering. International Journal of Robotics Research 2(28), 172–190 (2009)
17. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: 18th Int. Joint Conf. on Artificial Intelligence, pp. 1151–1156. Morgan-Kaufmann Publishers, San Francisco (2003)
18. Wijesoma, W.S., Perera, L.D.L., Adams, M.D.: Toward multidimensional assignment data association in robot localization and mapping. IEEE Transactions on Robotics 22(2), 350–365 (2006)
19. Martin, M.C., Moravec, H.P.: Robot evidence grids. Technical Report CMU-RI-TR-96-06, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 (March 1996)
20. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
21. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. Autonomous Robot Vehicles, 167–193 (1990)
22. Hoffman, J.R., Mahler, R.: Multi-target miss distance via optimal assignment. IEEE Transactions on Systems, Man and Cybernetics 34(3), 327–336 (2004)
23. Schuhmacher, D., Vo, B.T., Vo, B.N.: A consistent metric for performance evaluation of multi-object filters. IEEE Transactions on Signal Processing 86(8), 3447–3457 (2008)
24. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. Computer 22(6), 46–57 (1989)
25. Thrun, S.: Particle filter in robotics. In: Uncertainty in AI, UAI (2002)
26. Bibby, C., Reid, I.: Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association. In: Robotics Science and Systems (RSS-III), Atlanta, Georgia, USA (June 2007)
27. Vo, B.T.: Random finite sets in multi-object filtering. PhD thesis, University of Western Australia (2008)
28. Thrun, S.: Particle filters in robotics. In: 17th Annual Conference on Uncertainty in AI (2002)
29. Mahler, R.: Implementation and application of PHD/CPHD filters. In: Int'l Conf. on Information Fusion, Seattle, WA (2009)
30. Erdinc, O., Willett, P., Bar-Shalom, Y.: The bin-occupancy filter and its connection to the phd filters. IEEE Transactions on Signal Processing 57(11), 4232–4246 (2009)
31. Stoyan, D., Kendall, W.S., Mecke, J.: Stochastic Geometry and Its Applications, 2nd edn. John Wiley and Sons, Inc., New York (1995)
32. Daley, D.J., Vere-Jones, D.: An Introduction to the Theory of Point Processes. Springer, NY (2002)
33. Vo, B.N., Ma, W.K.: The Gaussian mixture probability hypothesis density filter. IEEE Transactions on Signal Processing 54(11), 4091–4104 (2006)
34. Vo, B.T., Mahler, R., Vo, B.N.: CPHD filtering with unknown clutter rate and detection profile. IEEE Transactions on Signal Processing (to appear, 2011)
35. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(9), 850–863 (1993)

36. Durrant-Whyte, H.F., Bailey, T.: Simultaneous localization and mapping: Part I. IEEE Robotics and Automation Magazine 13(2), 99–110 (2006)
37. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: Fastslam: A factored solution to the simultaneous localization and mapping problem. In: Eighteenth national conference on Artificial intelligence, pp. 593–598. American Association for Artificial Intelligence, Menlo Park (2002)
38. Durrant-Whyte, H.F., Majumder, S., de Battista, M., Scheding, S.: A bayesian algorithm for simultaneous localisation and map building. In: The Tenth International Symposium of Robotics Research (ISRR), Victoria, Australia (2001)
39. Niera, J., Tardos, J.D.: Data association in stochastic mapping using the joint compatibility test. IEEE Transactions on Robotics and Automation 17(6), 890–897 (2001)
40. Mahler, R.: Global integrated data fusion. In: Proc. 7th Nat. Symp. on Sensor Fusion, vol. 1, pp. 187–199 (1994)
41. Mahler, R.: An introduction to multisource-multitarget statistics and applications. Lockheed Martin Technical Monograph (March 2000)
42. Sidenbladh, H.: Multi-target particle filtering for the probability hypothesis density. In: Proc. Int'l Conf. on Information Fusion, Cairns, Australia, pp. 800–806 (2003)
43. Zajic, T., Ravichandran, R., Mahler, R., Mehra, R., Noviskey, M.: Joint tracking and identification with robustness against unmodeled targets. In: Proc. SPIE Signal Processing, Sensor Fusion and Target Recognition XII, vol. 5096, pp. 279–290 (2003)
44. Sidenbladh, H., Wirkander, S.-L.: Tracking random sets of vehicles in terrain. In: Proc. 2003 IEEE Workshop on Multi-Object Tracking, WI, USA (June 2003)
45. Ikoma, N., Uchino, T., Maeda, H.: Tracking of feature points in image sequence by smc implementation of the phd filter. In: Proc. SICE Annual Conference, vol. 2, pp. 1696–1701 (2004)
46. Vo, B.-N., Singh, S., Ma, W.-K.: Tracking multiple speakers with random sets. In: Proc. Int. Conf. Acoustics, Speech and Signal Processing, Montreal, Canada, vol. 2 (2004)
47. Tobias, M., Lanterman, A.: A probability hypothesis density-based multi-target tracking with multiple bistatic range and doppler observations. IEE Radar Sonar and Navigation 152(3), 195–205 (2003)
48. Mahler, R.: A survey of phd filter and cphd filter implementations. In: Proc. SPIE Defense & Security Symposium of Signal Processing, Sensor Fusion and Target Recognition XII (April 2007)
49. Clark, D.E., Vo, B.-N.: Convergence analysis of the gaussian mixture probability hypothesis density filter. IEEE Transactions on Signal Processing 55(4), 1024–1212 (2007)
50. Mahler, R.: Phd filters of higher order in target number. IEEE Transactions on Aerospace and Electronic Systems 43(4), 1523–1543 (2007)
51. Vo, B.T., Vo, B.N., Cantoni, A.: Analytic implementations of the cardinalized probability hypothesis density filter. IEEE Transactions on Signal Processing 55(7), 3553–3567 (2007)
52. Durrant-Whyte, H.F., Bailey, T.: Simultaneous localization and mapping: Part I. IEE Robotics and Automation Magazine 13(2), 99–110 (2006)
53. Majumder, S.: Sensor Fusion and Feature Based Navigation for Subsea Robots. PhD thesis, The University of Sydney (August 2001)

54. Mullane, J., Vo, B.N., Adams, M., Wijesoma, W.S.: A random set formulation for bayesian slam. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, France (September 2008)
55. Mullane, J.: Autonomous Navigation: On Issues Concerning Measurement Uncertainty. PhD thesis, Dept. of EEE, Nanyang Technological University, Singapore (2009)
56. Mullane, J., Vo, B.N., Adams, M.D.: Rao-blackwellised PHD SLAM. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Alaska, USA (May 2010)
57. Mullane, J., Vo, B.N., Adams, M.D., Vo, B.T.: A random-finite-set approach to Bayesian SLAM. IEEE Transactions on Robotics 27(2), 268–282 (2011)
58. Montemerlo, M., Thrun, S., Siciliano, B.: FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics. Springer, Heidelberg (2007)
59. Mullane, J.S., Keller, S., Rao, A., Adams, M.D., Yeo, A., Hover, F.S., Patrikalakis, N.M.: X-band radar based slam in Singapore off-shore environment. In: Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision (ICARCV 2010), Singapore (December 2010)
60. Murphy, R.: Bayesian map learning in dynamic environments. In: Proc. Conf. Neural Inf. Process. Syst., Colorado, pp. 1015–1021 (1999)
61. Del Moral, P., Jacod, J.: The Monte-Carlo Method for filtering with discrete time observations. Central Limit Theorems. The Fields Institute Communications American Mathematical Society (2002)
62. Crisan, D., Doucet, A.: A survey of convergence results on particle filtering methods for practitioners 50(3), 736–746 (2002)
63. Brooks, A., Bailey, T.: HybridSLAM: Combining FastSLAM and EKF-SLAM for reliable mapping. In: The Eighth International Workshop on the Algorithmic Foundations of Robotics, Mexico (December 2008)
64. Mullane, J., Vo, B.N., Adams, M., Wijesoma, W.S.: A random set approach to slam. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Visual Mapping and Navigation in Outdoor Environments, Japan (May 2009)
65. Kalyan, B., Lee, K.W., Wijesoma, W.S.: FISST-SLAM: Finite set statistical approach to simultaneous localization and mapping. International Journal of Robotics Research 29(10), 1251–1262 (2010)
66. Julier, S.J., Uhlmann, J.K.: A new extension of the kalman filter to nonlinear systems. In: Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, Florida, pp. 35–45 (1997)
67. Nieto, J., Guivant, J., Nebot, E., Thrun, S.: Real time data association for FastSLAM. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 412–418 (September 2003)
68. Mahler, R.: PHD filters of higher order in target number. IEEE Transactions on AES 43(4), 1523–1543 (2009)
69. Vo, B.T., Vo, B.N., Cantoni, A.: The cardinality balanced multi-target multibernoulli and its implementations. IEEE Transactions on Signal Processing 57(2), 409–423 (2009)
70. Kingman, J.F.C.: Poisson Processes. Oxford Studies in Probability. Oxford University Press, USA (1993)

Springer Tracts in Advanced Robotics

Edited by B. Siciliano, O. Khatib and F. Groen

Further volumes of this series can be found on our homepage: springer.com

Vol. 72: Mullane, J.; Vo, B.-N.; Adams, M.; Vo, B.-T.
Random Finite Sets for Robot Mapping and SLAM
146 p. 2011 [978-3-642-21389-2]

Vol. 71: XXX

Vol. 70: Pradalier, C.; Siegwart, R.; Hirzinger, G. (Eds.)
Robotics Research
752 p. 2011 [978-3-642-19456-6]

Vol. 69: Rocon, E.; Pons, J.L.
Exoskeletons in Rehabilitation Robotics
138 p. 2010 [978-3-642-17658-6]

Vol. 68: Hsu, D.; Isler, V.; Latombe, J.-C.; Ming C. Lin (Eds.)
Algorithmic Foundations of Robotics IX
424 p. 2010 [978-3-642-17451-3]

Vol. 67: Schütz, D.; Wahl, F.M. (Eds.)
Robotic Systems for Handling and Assembly
460 p. 2010 [978-3-642-16784-3]

Vol. 66: Kaneko, M.; Nakamura, Y. (Eds.)
Robotics Research
450 p. 2010 [978-3-642-14742-5]

Vol. 65: Ribas, D.; Ridao, P.; Neira, J.
Underwater SLAM for Structured Environments Using an Imaging Sonar
142 p. 2010 [978-3-642-14039-6]

Vol. 64: Vasquez Govea, A.D.
Incremental Learning for Motion Prediction of Pedestrians and Vehicles
153 p. 2010 [978-3-642-13641-2]

Vol. 63: Vanderborght, B.;
Dynamic Stabilisation of the Biped Lucy Powered by Actuators with Controllable Stiffness
281 p. 2010 [978-3-642-13416-6]

Vol. 62: Howard, A.; Iagnemma, K.; Kelly, A. (Eds.):
Field and Service Robotics
511 p. 2010 [978-3-642-13407-4]

Vol. 61: Mozos, Ó.M.
Semantic Labeling of Places with Mobile Robots
134 p. 2010 [978-3-642-11209-6]

Vol. 60: Zhu, W.-H.
Virtual Decomposition Control – Toward Hyper Degrees of Freedom Robots
443 p. 2010 [978-3-642-10723-8]

Vol. 59: Otake, M.
Electroactive Polymer Gel Robots – Modelling and Control of Artificial Muscles
238 p. 2010 [978-3-540-23955-0]

Vol. 58: Kröger, T.
On-Line Trajectory Generation in Robotic Systems – Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events
230 p. 2010 [978-3-642-05174-6]

Vol. 57: Chirikjian, G.S.; Choset, H.; Morales, M., Murphey, T. (Eds.)
Algorithmic Foundations of Robotics VIII – Selected Contributions of the Eighth International Workshop on the Algorithmic Foundations of Robotics
680 p. 2010 [978-3-642-00311-0]

Vol. 56: Buehler, M.; Iagnemma, K.; Singh S. (Eds.)
The DARPA Urban Challenge – Autonomous Vehicles in City Traffic
625 p. 2009 [978-3-642-03990-4]

Vol. 55: Stachniss, C.
Robotic Mapping and Exploration
196 p. 2009 [978-3-642-01096-5]

Vol. 54: Khatib, O.; Kumar, V.; Pappas, G.J. (Eds.)
Experimental Robotics: The Eleventh International Symposium
579 p. 2009 [978-3-642-00195-6]

Vol. 53: Duindam, V.; Stramigioli, S.
Modeling and Control for Efficient Bipedal Walking Robots
211 p. 2009 [978-3-540-89917-4]

Vol. 52: Nüchter, A.
3D Robotic Mapping
201 p. 2009 [978-3-540-89883-2]

Vol. 51: Song, D.
Sharing a Vision
186 p. 2009 [978-3-540-88064-6]

Vol. 50: Alterovitz, R.; Goldberg, K.
Motion Planning in Medicine: Optimization and Simulation Algorithms for Image-Guided Procedures
153 p. 2008 [978-3-540-69257-7]

Vol. 49: Ott, C.
Cartesian Impedance Control of Redundant and Flexible-Joint Robots
190 p. 2008 [978-3-540-69253-9]

Vol. 48: Wolter, D.
Spatial Representation and Reasoning for Robot Mapping
185 p. 2008 [978-3-540-69011-5]

Vol. 47: Akella, S.; Amato, N.; Huang, W.; Mishra, B.; (Eds.)
Algorithmic Foundation of Robotics VII
524 p. 2008 [978-3-540-68404-6]

Vol. 46: Bessière, P.; Laugier, C.; Siegwart R. (Eds.)
Probabilistic Reasoning and Decision Making in Sensory-Motor Systems
375 p. 2008 [978-3-540-79006-8]

Vol. 45: Bicchi, A.; Buss, M.; Ernst, M.O.; Peer A. (Eds.)
The Sense of Touch and Its Rendering
281 p. 2008 [978-3-540-79034-1]

Vol. 44: Bruyninckx, H.; Přeučil, L.; Kulich, M. (Eds.)
European Robotics Symposium 2008
356 p. 2008 [978-3-540-78315-2]

Vol. 43: Lamon, P.
3D-Position Tracking and Control for All-Terrain Robots
105 p. 2008 [978-3-540-78286-5]

Vol. 42: Laugier, C.; Siegwart, R. (Eds.)
Field and Service Robotics
597 p. 2008 [978-3-540-75403-9]

Vol. 41: Milford, M.J.
Robot Navigation from Nature
194 p. 2008 [978-3-540-77519-5]

Vol. 40: Birglen, L.; Laliberté, T.; Gosselin, C.
Underactuated Robotic Hands
241 p. 2008 [978-3-540-77458-7]

Vol. 39: Khatib, O.; Kumar, V.; Rus, D. (Eds.)
Experimental Robotics
563 p. 2008 [978-3-540-77456-3]

Vol. 38: Jefferies, M.E.; Yeap, W.-K. (Eds.)
Robotics and Cognitive Approaches to Spatial Mapping
328 p. 2008 [978-3-540-75386-5]

Vol. 37: Ollero, A.; Maza, I. (Eds.)
Multiple Heterogeneous Unmanned Aerial Vehicles
233 p. 2007 [978-3-540-73957-9]

Vol. 36: Buehler, M.; Iagnemma, K.; Singh, S. (Eds.)
The 2005 DARPA Grand Challenge – The Great Robot Race
520 p. 2007 [978-3-540-73428-4]

Vol. 35: Laugier, C.; Chatila, R. (Eds.)
Autonomous Navigation in Dynamic Environments
169 p. 2007 [978-3-540-73421-5]

Vol. 34: Wisse, M.; van der Linde, R.Q.
Delft Pneumatic Bipeds
136 p. 2007 [978-3-540-72807-8]

Vol. 33: Kong, X.; Gosselin, C.
Type Synthesis of Parallel Mechanisms
272 p. 2007 [978-3-540-71989-2]

Vol. 32: Milutinović, D.; Lima, P.
Cells and Robots: Modeling and Control of Large-Size Agent Populations
130 p. 2007 [978-3-540-71981-6]