ulm university universität ulm

Multi-Object Tracking Using Random Finite Sets

Stephan Reuter

Band 11

Schriftenreihe des Instituts für Mess-, Regel- und Mikrotechnik

Stephan Reuter

Multi-Object Tracking Using Random Finite Sets

Schriftenreihe des
Instituts für Mess-, Regel- und Mikrotechnik
Universität Ulm

Herausgeber:
Prof. Dr.-Ing. Klaus Dietmayer

Band 11

Stephan Reuter

Multi-Object Tracking
Using Random Finite Sets

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Dissertation, Universität Ulm,
Fakultät für Ingenieurwissenschaften und Informatik, 2014

Impressum

Universität Ulm
Institut für Mess-, Regel- und Mikrotechnik
Prof. Dr.-Ing. Klaus Dietmayer
Albert-Einstein-Allee 41
89081 Ulm
http://www.uni-ulm.de/mrm

Eine Übersicht über alle Bände der Schriftenreihe finden Sie unter
http://www.uni-ulm.de/mrmschriften.

Institut für Mess-, Regel- und Mikrotechnik der Universität Ulm 2014
Print on Demand

ISBN 978-3-941543-12-6 e-ISBN 978-3-941543-13-3

ulm university universität
uulm

Multi-Object Tracking Using Random Finite Sets

DISSERTATION

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS
(Dr.-Ing.)

der Fakultät für Ingenieurwissenschaften
und Informatik der Universität Ulm

von

Stephan Reuter
aus Crailsheim

Gutachter:	Prof. Dr.-Ing. Klaus Dietmayer
	Priv.-Doz. Dr. Wolfgang Koch
Amtierende Dekanin:	Prof. Dr. Tina Seufert

Ulm, 23.06.2014

Acknowledgements

I would like to thank my supervisor Prof. Dr.-Ing. Klaus Dietmayer for his exceptional support during the last years, the stimulating discussions and the excellent working environment. In particular, I am thankful for the scientific freedom and the support of the research stay in Australia.

Further, I would like to acknowledge Priv.-Doz. Dr. Wolfgang Koch for numerous helpful discussions and for being the second examiner of my thesis.

I would also like to thank Associate Prof. Ba-Tuong Vo and Prof. Ba-Ngu Vo for providing the opportunity to visit them for several weeks in Australia. It was a great pleasure to collaborate with such keen and friendly guys. Thanks again for all the discussions, the invaluable feedback, and your hospitality.

My gratitude also goes to my colleagues at the Institute of Measurement, Control, and Microtechnology for the pleasant working atmosphere, interesting discussions, and several frequent events like the ”Kaffee-Ecke”, the Friday morning ”Nusszopf”, and the traditional Friday afternoon meeting which really enriched the daily work. Additionally, I would like to thank Hendrik Deusch, Daniel Meissner, Marcus Konrad, Dr. Michael Munz, Dominik Nuss, Florian Seeliger, Alexander Scheel, Jürgen Wiest and Benjamin Wilking for invaluable discussions as well as the helpful and crucial suggestions. I would also like to acknowledge the brilliant work of Oliver Betz, Franz Degenhard, Thomas Löffler, and Martin Nieß who constructed the experimental platform for the SFB/TRR 62 and integrated all required sensors in the experimental vehicle. Further, I would like to thank Dr. Michael Buchholz and Jürgen Remmlinger for the LaTeX template and Claudia Fricke for handling all administrative matters.

My dissertation was done within the Transregional Collaborative Research Center SFB/TRR 62 ”Companion-Technology for Cognitive Technical Systems” supported by the German Research Foundation (DFG). I would especially like to thank Prof. Dr. Susanne Biundo-Stephan and Dr. Bernd Schattenberg who facilitated the research stay at the Curtin University in Perth, Australia.

Special thanks go to my parents for all their support during the last years and for motivating me to follow my own way. Last but not least, I would like to thank Nadine for all her love and the encouragement during the last years.

Abstract

The aim of multi-object tracking is the estimation of the number of objects and their individual states using a sequence of measurements. While state of the art algorithms use object individual single-object trackers, the multi-object Bayes filter models the multi-object state as well as the measurement process using random finite sets which naturally represent the uncertainty in the number of objects as well as in the state of the objects. Hence, a realization of a random finite set valued random variable represents the complete environment and facilitates the incorporation of object interactions. In the update of the multi-object Bayes filter, the multi-object likelihood function averages over all possible track to measurement associations which avoids error-prone association decisions. During the last decade, several tractable approximations of the multi-object Bayes filter have been proposed based on the statistical moments or parameterization. However, the approximations are prone to unstable cardinality estimates, the influence of missed detections on well-separated objects ("spooky effect"), or a biased cardinality estimate and do not allow for the incorporation of object interactions any more due to the required approximations.

In this thesis, the first real-time capable sequential Monte Carlo implementation of the multi-object Bayes filter and its application to real-world sensor data are presented. The proposed implementation of the multi-object Bayes filter is based on an approximation of the multi-object likelihood function which significantly reduces the computational complexity. Further, several methods to incorporate object interactions in the prediction step of the multi-object Bayes filter are proposed. Additionally, a novel multi-object tracking algorithm, the labeled multi-Bernoulli filter, is proposed in this thesis. The approximation of the multi-object posterior density using labeled multi-Bernoulli random finite sets results in an accurate and real-time capable tracking algorithm. The labeled multi-Bernoulli filter facilitates an implementation using Gaussian mixtures and is capable to track a significantly larger number of objects than the sequential Monte Carlo implementation of the multi-object Bayes filter.

The proposed tracking algorithms are evaluated using simulated data as well as real-world sensor data. The performance of the algorithms is compared to other approximations of the multi-object Bayes filter like the cardinalized probability hypothesis density filter and the cardinality balanced multi-target multi-Bernoulli filter. Additionally, the labeled multi-Bernoulli filter is compared to the joint integrated probabilistic data association filter in the context of vehicle environment perception.

Kurzfassung

Das Ziel der Multi-Objekt-Verfolgung ist die simultane Schätzung der Objektanzahl sowie des Zustands der einzelnen Objekte unter Nutzung einer Sequenz von Messungen. Während nach dem Stand der Technik häufig ein eigenes Filter für jedes Objekt verwendet wird, modelliert das Multi-Objekt-Bayesfilter den Multi-Objekt-Zustand sowie den Messprozess des Sensors mit Hilfe von Random Finite Sets, welche die Unsicherheit über die Objektanzahl und die Objektzustände auf natürliche Weise repräsentieren. Folglich stellt eine Random Finite Set Variable alle detektierten Objekte in einem festgelegten Bereich dar und ermöglicht die Modellierung von Abhängigkeiten zwischen einzelnen Objekten. Im Innovationsschritt des Multi-Objekt-Bayesfilters mittelt die Multi-Objekt-Likelihoodfunktion über alle möglichen Assoziationen zwischen existierenden Tracks und empfangenen Messungen, wodurch eine fehleranfällige Assoziationsentscheidung vermieden wird. Im letzten Jahrzehnt wurden mehrere Approximationen des Multi-Objekt-Bayesfilters präsentiert, welche unter Nutzung der statistischen Momente beziehungsweise durch eine Parametrisierung der Multi-Objekt-Verteilung die Berechnungskomplexität signifikant verringern. Die Approximationen führen jedoch zu ungewünschten Nebeneffekten wie beispielsweise einer stark schwankenden Objektanzahl, einem Einfluss von Fehldetektionen auf weit entfernte Objekte (”spooky”-Effekt) sowie einem Offset bei der Objektanzahl. Außerdem führen die Approximationen dazu, dass eine Modellierung von Abhängigkeiten zwischen den Objekten nicht mehr möglich ist.

Im Rahmen dieser Arbeit wird die erste echtzeitfähige sequentielle Monte-Carlo Implementation des Multi-Objekt-Bayesfilters präsentiert. Die Implementierung basiert hierbei auf einer Approximation der Multi-Objekt-Likelihoodfunktion, welche die Berechnungskomplexität stark reduziert. Des Weiteren werden mehrere Methoden zur Modellierung von Abhängigkeiten zwischen den Objekten vorgestellt, die in den Prädiktionsschritt des Multi-Objekt-Bayesfilters integriert werden können. Außerdem wird ein neuartiger Multi-Objekt-Trackingalgorithmus, das Labeled Multi-Bernoulli Filter, präsentiert. Die Approximation der a posteriori Multi-Objekt Wahrscheinlichkeitsdichtefunktion durch ein Labeled Multi-Bernoulli Random Finite Set ermöglicht einerseits eine hohe Schätzgüte und gewährleistet andererseits eine echtzeitfähige Implementierung des Filters. Das Labeled Multi-Bernoulli Filter ermöglicht eine Implementierung mittels Gauß-Mixturen und ist in der Lage eine deutlich höhere Anzahl an Objekten zu verfolgen als das sequentielle Monte-Carlo Multi-Objekt-Bayesfilter.

Die entwickelten Trackingalgorithmen werden abschließend anhand simulierter sowie realer Sensordaten evaluiert. Die Ergebnisse der Algorithmen werden mit existierenden Approximation des Multi-Objekt Bayes-Filters wie beispielsweise dem Cardinalized Probability Hypothesis Density Filter und dem Cardinality Balanced Multi-Target Multi-Bernoulli Filter verglichen. Darüber hinaus wird das Labeled Multi-Bernoulli Filter noch mit dem Joint Integrated Probabilitic Data Association Filter im Kontext der Fahrzeugumfelderfassung verglichen.

Contents

Chapter 1

Introduction

The recursive estimation of an object's state using a series of uncertain measurements is crucial for applications like the navigation and control of vehicles or aircraft. Since the 1960's, the Kalman filter [Kal60] is widely used for recursive state estimation since it enables a closed-form calculation of the estimation problem. While the original derivation of the Kalman filter is based on a least-squares approach, an alternative derivation from a Bayesian point of view using Gaussian distributions is proposed in [HL64]. Thus, the Kalman filter is a Bayes optimal state estimator if all distributions are Gaussian and the corresponding measurement models and process models are linear.

Within the Apollo program of the National Aeronautics and Space Administration, the Kalman filter was used to estimate the trajectory of the spacecraft. In the following decades, the Kalman filter was extensively used in military applications like air surveillance. In contrast to navigation and control, the military applications required the simultaneous state estimation for a time-varying number of objects since new objects may appear and already existing objects may disappear at any time. Hence, a multi-object tracking algorithm is required which jointly estimates the number of objects and the state of each individual object using uncertain sensor measurements. The sensor delivers a set of measurements comprising object detections and false alarms. Additionally, some of the objects may not be detected. In order to handle the arising ambiguity in the track to measurement association, a variety of data association approaches have been proposed, e.g. probabilistic data association (PDA) [BT75] and multi-hypothesis tracking (MHT) [Rei79].

Starting in the 1990's, a rigorous extension of the Bayes filter to multi-object tracking applications was developed by Mahler [GMN97; Mah07a] which is based on the finite set statistics. A state of the multi-object Bayes filter is given by a random finite set (RFS), which represents the complete environment. Since the multi-object likelihood function averages over all possible associations, the multi-object Bayes filter produces an evolving estimate of the multi-object probability density and requires additional post-processing to extract track estimates from the distribution. During the last decade, a number of tractable approximations of the multi-object Bayes filter have been proposed: the probability hypothesis density (PHD) filter [Mah03], the cardinalized probability hypothesis density (CPHD) filter [Mah07b], and the cardinality balanced multi-target multi-Bernoulli (CB-MeMBer) filter [Mah07a]. While the PHD and the CPHD filter propagate the first moment of the multi-object density (and the cardinality distribution) over time, the CB-MeMBer filter approximates the multi-object posterior by the parameters of a multi-Bernoulli distribution.

Although the representation of the multi-object state using random finite sets facilitates an incorporation of object interactions in the multi-object Bayes filter, the commonly used PHD, CPHD, and CB-MeMBer implementations assume that all objects move independently of each other. However, the possible state transitions are restricted by physical constraints in scenarios with closely spaced objects. Within this thesis, the incorporation of object interactions into the first real-time capable sequential Monte-Carlo (SMC) implementation of the full multi-object Bayes filter is proposed. The modeling of the interactions is based on hard-core point processes [Mat60] as well as the social force model [HM95]. The hard-core point processes rely on the concept of removing points of a process (or objects from a set) until each point has the required minimum distance to all other points. In contrast, the social force model keeps all points of the process and ensures physically possible states by adapting the motion of an object to the environmental constraints.

Recently, Vo and Vo introduced the class of labeled random finite sets which facilitates an analytical implementation of the multi-object Bayes filter using δ-generalized labeled multi-Bernoulli (δ-GLMB) distributions [VV13b]. The δ-GLMB filter significantly outperforms the CPHD and the CB-MeMBer filters at the cost of a higher computational complexity. A drawback of the δ-GLMB filter is the exponentially increasing amount of components required to represent the multi-object state adequately. This thesis proposes the labeled multi-Bernoulli (LMB) filter and its implementation using Gaussian mixture (GM) and SMC methods. Since the LMB filter approximates the posterior δ-GLMB distribution by an LMB RFS with matching intensity function, the LMB filter can be considered as a computationally efficient approximation of the δ-GLMB filter. The LMB filter may also be interpreted as a generalized version of the CB-MeMBer filter.

The proposed real-time capable SMC implementation of the full multi-object Bayes filter is applied to a pedestrian tracking scenario using a static sensor setup. The filter additionally incorporates knowledge about currently observable areas using a state dependent detection probability and models interactions between the pedestrians within the prediction step.

Especially in the context of advanced driver assistance systems and autonomous driving, a continuous and reliable perception and tracking of all objects in the environment is required for applications like adaptive cruise control and emergency brake assists. Further, situation analysis as well as decision-making require the evolving multi-object state estimate. Thus, the Gaussian mixture implementation of the LMB filter is applied to vehicle environment perception and its performance is compared to the joint integrated probabilistic data association (JIPDA) implementation proposed in [Mah09b; Mun11]. Several typical scenarios on highways and rural roads are investigated to illustrate possible performance gains of the LMB filter.

This thesis is organized as follows: First, the single-object Bayes filter as well as its implementation using the Kalman filter and sequential Monte-Carlo (SMC) methods are introduced in Chapter 2. Additionally, standard multi-object tracking approaches using object-individual single-object trackers are reviewed. Chapter 3 presents the basics of the finite set statistics and introduces the multi-object Bayes filter as well as the required multi-object motion and measurement models. Further, the computationally tractable approximations of the multi-object Bayes filter are outlined and discussed with respect to computational complexity and accuracy. A real-time sequential Monte-Carlo (SMC) implementation of the multi-object Bayes filter is proposed in Chapter 4 which achieves a significant reduction of the computational complexity by approximating the multi-object likelihood function. Additionally, two approaches to model object interactions within the multi-object Bayes filter are proposed which are based on hard-core point processes and social forces. In Chapter 5, a novel multi-object tracking algorithm, the labeled multi-Bernoulli (LMB) filter, is proposed and its implementation using Gaussian mixture (GM) and sequential Monte-Carlo (SMC) methods is explained in detail. Finally, Chapter 6 presents the evaluation of the SMC multi-object Bayes filter and the LMB filter using simulated sensor data as well as real-world sensor data.

Chapter 2

Fundamentals of Object Tracking

The aim of object tracking is the estimation of the dynamic object state based on uncertain measurements. The estimation process additionally uses sensor characteristics as well as motion patterns of the objects. In Section 2.1, the recursive Bayes filter which is central to the object tracking algorithms considered in this thesis is introduced. Afterwards, standard implementations of the Bayes filter using Gaussian distributions or sequential Monte-Carlo (SMC) methods are summarized in Sections 2.2 and 2.3. Section 2.4 outlines a variety of bottom-up approaches for the application of the single-object Bayes filter to multi-object tracking. Finally, consistency checks for single-object tracking algorithms are presented in Section 2.5.

2.1 Bayes Filter

In tracking applications, the state of an object is commonly described by a state vector x. Thus, the aim of object tracking is the estimation of the posterior probability density function (PDF)

$$p_{k+1}(x_{k+1}) \triangleq p_{k+1}(x_{k+1}|z_{1:k+1}) \tag{2.1}$$

of the state x_{k+1} of an object using the set of all measurements $z_{1:k+1} = \{z_1, ..., z_{k+1}\}$ for the discrete sampling times up to time $k+1$.

The motion of an object between two successive time steps k and $k+1$ is commonly described using a Markov transition

$$x_{k+1|k} = f(x_k, v_k) \tag{2.2}$$

where v_k denotes an additive noise which is required to handle the uncertainties of the assumed motion model. An alternative representation of the object's motion is given by the Markov transition density

$$f_{k+1|k}(x_{k+1|k}|x_k, z_{1:k}). \tag{2.3}$$

Using the Markov property which states that all information about the past is represented by the current state x_k, the Markov transition density simplifies to

$$f_{k+1|k}(x_{k+1|k}|x_k). \tag{2.4}$$

The prediction of the prior PDF $p_k(x) = p_k(x|z_{1:k})$ to time $k+1$ is obtained using the Chapman-Kolmogorov equation

$$p_{k+1|k}(x_{k+1}) = \int f_{k+1|k}(x_{k+1}|x_k)p_k(x_k)dx_k, \tag{2.5}$$

where $p_{k+1|k}(x_{k+1}) \triangleq p_{k+1|k}(x_{k+1}|z_{1:k})$.

The measurement update of the Bayes filter is based on the measurement equation

$$z_{k+1|k} = h_{k+1}(x_{k+1}, w_{k+1}) \tag{2.6}$$

which transforms the predicted state x_{k+1} into the measurement space taking the measurement noise w_{k+1} into account. The measurement equation facilitates an innovation of the object's state in the measurement space using the likelihood function

$$g(z_{k+1}|x_{k+1}) \tag{2.7}$$

which is obtained using the measurement equation (2.6). The update of the Bayes filter is consequently given by

$$p_{k+1}(x_{k+1}) = \frac{g(z_{k+1}|x_{k+1}) \cdot p_{k+1|k}(x_{k+1})}{\int g(z_{k+1}|x_{k+1}) \cdot p_{k+1|k}(x_{k+1})dx_{k+1}}. \tag{2.8}$$

Hence, the prediction and update equations of the Bayes filter facilitate the recursive estimation of an object's state if the object's initial distribution $p_0(x_0)$ is available.

Since the calculation of predicted values is based on prior values and the calculation of posterior values depends only on the predicted values, the time indices k and $k+1$ corresponding to prior and posterior values are dropped in the following for reasons of clarity. However, the time indices $k+1|k$ of predicted values are abbreviated by "+" since the indication of predicted values is required to avoid ambiguities.

2.2 Kalman Filter

The Kalman filter [Kal60] facilitates a closed-form implementation of the Bayes filter in case of Gaussian distributed signals and probability densities as well as linear process and measurement models. Since a Gaussian distribution is completely characterized by the first and second statistical moment (i.e. the mean value $\hat{x}$ and the covariance $\underline{P}$), it is sufficient to propagate the moments over time instead of the entire PDFs. While the Kalman filter is derived using the least squares approach in [Kal60], Ho and Lee [HL64] derived the identical equations for prediction and update in a top-down Bayesian approach using probability density functions. Thus, the Kalman filter is a Bayes-optimal single-object state estimator if the requirements of the derivation are fulfilled.

In the Kalman filter, the initial estimate of the object state is assumed to be given by a Gaussian distribution

$$\mathcal{N}\left(x; \hat{x}, \underline{P}\right) \triangleq \frac{1}{\sqrt{\det(2\pi\underline{P})}} \exp\left(-\frac{1}{2}(x-\hat{x})^{\mathrm{T}}\underline{P}^{-1}(x-\hat{x})\right) \tag{2.9}$$

with mean value $\hat{x}$ and covariance $\underline{P}$. In case of linear motion and measurement models, the prediction equation (2.2) and the measurement equation (2.6) simplify to

$$x_+ = \underline{F}x + v \tag{2.10}$$

$$z_+ = \underline{H}x_+ + w \tag{2.11}$$

where $\underline{F}$ is the process or state transition matrix and $\underline{H}$ is the measurement matrix. Further, v denotes the process noise and w the measurement noise which are assumed to be uncorrelated in the following. If the process noise v follows a zero-mean Gaussian distribution, the state transition of a Gaussian distribution (2.9) yields the predicted object state and estimation error covariance

$$\hat{x}_+ = \underline{F}\hat{x}, \tag{2.12}$$

$$\underline{P}_+ = \underline{F}\underline{P}\underline{F}^{\mathrm{T}} + \underline{Q}, \tag{2.13}$$

where $\underline{Q}$ is the covariance matrix of the process noise.

In general, a sensor is not able to measure all components of the predicted state $\hat{x}_+$. In case of a zero-mean Gaussian measurement noise w and a linear measurement model, the predicted state $\hat{x}_+$ with covariance $\underline{P}_+$ induces a Gaussian distributed predicted measurement

$$z_+ = \underline{H}\hat{x}_+ \tag{2.14}$$

with according covariance

$$\underline{R}_+ = \underline{H}\underline{P}_+\underline{H}^{\mathrm{T}}. \tag{2.15}$$

Since the Kalman filter requires all signals to follow a Gaussian distribution, the measurement uncertainty of an obtained measurement z is represented by the covariance matrix $\underline{R}$. In the measurement update step of the Kalman filter, the residual

$$\gamma = z - z_+ \tag{2.16}$$

of the measurement z and predicted measurement z_+ is required for the innovation of the object state. The corresponding innovation covariance of the residual is given by

$$\underline{S} = \underline{H}\underline{P}_+\underline{H}^{\mathrm{T}} + \underline{R}. \tag{2.17}$$

The filter gain

$$\underline{K} = \underline{P}_+\underline{H}^{\mathrm{T}}\underline{S}^{-1} \tag{2.18}$$

of the Kalman filter models the dependency of the posterior object state

$$\hat{x} = \hat{x}_+ + \underline{K}\left(z - z_+\right) = \hat{x}_+ + \underline{K}\gamma \tag{2.19}$$

on the uncertainties of the process and measurement model. On the one hand, a high measurement noise $\underline{R}$ in combination with a small predicted estimation error covariance $\underline{P}_+$ implies a small value of $\underline{K}$ and the posterior state is dominated by the predicted state $\hat{x}_+$. On the other hand, the posterior state is dominated by the current measurement z for a small measurement noise $\underline{R}$ and a high estimation error covariance $\underline{P}_+$. Finally, the posterior estimation error covariance is given by

$$\begin{aligned}\underline{P} &= \underline{P}_+ - \underline{K}\underline{S}\underline{K}^{\mathrm{T}} \\ &= (\underline{I} - \underline{K}\underline{H})\,\underline{P}_+\,(\underline{I} - \underline{K}\underline{H})^{\mathrm{T}} + \underline{K}\underline{R}\underline{K}^{\mathrm{T}},\end{aligned} \tag{2.20}$$

where both forms are mathematically equivalent but the numerical stability of the second version is higher due to the quadratic appearance of the term $(\underline{I} - \underline{K}\underline{H})$ (see [BP99, pp. 158ff.]). For notational convenience, the short form $\underline{P}_+ - \underline{K}\underline{S}\underline{K}^{\mathrm{T}}$ is used in the following.

2.2.1 Constant Gain Kalman Filter

Due to the required matrix inversion, the computationally expensive part of the Kalman filter is the update of the estimation error covariance matrix. The constant gain Kalman filters significantly reduce the computational costs by assuming the Kalman gain $\underline{K}$ in (2.18) to be constant. Due to the constant gain, the calculation of the estimation error covariance $\underline{P}$ is not required any more and it is sufficient to propagate the mean value $\hat{x}$ over time. Thus, constant gain filters can be considered as approximations of the Bayes filter using only the first statistical moment of the PDF $p(x)$.

Since the Kalman gain converges for constant process and measurement noise, the assumption of a constant gain is suitable for such tracking applications. However, the constant gain filter is not optimal during object initialization due to a mismatching estimation error covariance. Two well-known constant gain Kalman filters are the α-β and α-β-γ [HM04a; Kal84] trackers for the constant velocity (CV) and the constant acceleration (CA) motion models [BP99; BWT11].

2.2.2 Kalman Filter for Non-Linear Systems

The derivation of the Kalman filter requires linear motion and measurement models. However, non-linear motion models like the constant turn rate and constant velocity (CTRV) model as well as non-linear measurement models for sensors delivering range and angle measurements are required in a huge number of tracking applications. The extended Kalman filter (EKF) [BF88; BP99; SSM62] and the unscented Kalman filter (UKF) [JU04; JUD00; JUD95] facilitate the application of the Kalman filter in case of slightly non-linear motion or measurement models.

The EKF uses a Taylor series approximation to linearize the non-linear functions f and h given by (2.2) and (2.6) which allows for keeping the structure of the Kalman filter by using the linearized system and measurement matrices. In most applications, the Taylor series is truncated after the linear term (first-order EKF). A truncation after the quadratic term leads to the second-order EKF. In the first-order EKF, the linear system matrix is given by the Jacobian matrix

$$\underline{F}^J = \frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \tag{2.21}$$

which is evaluated for the current state estimate $\hat{x}$ and replaces $\underline{F}$ in the standard equations of the Kalman filter. Analog, the linearization of the measurement equation yields the measurement matrix

$$\underline{\mathrm{H}}^J = \frac{\partial h}{\partial x} = \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x_1} & \cdots & \frac{\partial h_m}{\partial x_n} \end{pmatrix} \tag{2.22}$$

which is evaluated at the predicted state $\hat{x}_+$. By replacing the system matrix $\underline{\mathrm{F}}$ and the measurement matrix $\underline{\mathrm{H}}$ with the linearized matrices $\underline{\mathrm{F}}^J$ and $\underline{\mathrm{H}}^J$, the EKF facilitates the application of the Kalman filter equations in case of slightly non-linear tracking applications.

While the EKF linearizes the non-linear process or measurement models, the aim of the UKF is to approximate the probability density function itself. The Gaussian distribution is approximated using sigma points $\mathcal{X}$. The sigma points are selected on a level curve of the Gaussian distribution and the number of points depends on the dimension of the Gaussian distribution. Each of the sigma points is transformed using the non-linear process or measurement equations. The resulting sigma points are subsequently used to obtain the transformed state representation. For additional details about the UKF, refer to [BWT11; JU04].

2.3 Particle Filter

The Kalman filter requires the probability density to be of a parametric form which is completely characterized by its first and second statistical moment. Consequently, the Kalman filter is not able to handle arbitrary distributions. Further, the EKF and UKF approximations are only suitable for applications with moderately non-linear process and measurement models. In contrast, a particle filter [AMGC02; DFG01; Gor97; GSS93; IB98; RAG04] facilitates a non-parametric implementation of the Bayes filter which approximates arbitrary probability density functions by a finite number of samples. Consequently, a particle filter is able to represent arbitrary distributions including multi-modal distributions. The particle filter is also known as the sequential Monte-Carlo (SMC) method.

In a particle filter, the PDF $p(x|z_{1:k})$ is approximated by ν particles

$$x^{(1)},\ \ldots, x^{(\nu)} \sim p(x|z_{1:k}), \tag{2.23}$$

where each particle $x^{(i)}$ is a realization of the random variable x. A normalized weight $w^{(i)} > 0$ is associated with each particle such that $\sum_{i=1}^{\nu} w^{(i)} = 1$. Consequently, the approximation of the PDF $p(x|z_{1:k})$ using ν particles is given by the weighted sum of the particles:

$$p(x|z_{1:k}) \approx \sum_{i=1}^{\nu} w^{(i)} \cdot \delta_{x^{(i)}}(x), \tag{2.24}$$

where the Kronecker delta function is defined by

$$\delta_{x^{(i)}}(x) = \begin{cases} 1 & \text{if } \ x = x^{(i)} \\ 0 & \textit{otherwise}. \end{cases} \tag{2.25}$$

Using a huge number of particles $\nu \to \infty$, an exact representation of a PDF is feasible.

The prediction of the particles to the time of the next measurement is realized by sampling the predicted particles from the transition density f_+:

$$x_+^{(i)} \sim f_+(\cdot|x^{(i)}, z_{1:k}). \tag{2.26}$$

The weights of the particles are not modified during the prediction step.

In the update step of the particle filter, the likelihood function $g(z|x_+^{(i)})$ has to be evaluated for each of the ν particles. The weight of the particles is updated by

$$w^{(i)} \triangleq \frac{g(z|x_+^{(i)})w_+^{(i)}}{\sum_{e=1}^{\nu} g(z|x_+^{(e)})w_+^{(e)}}, \tag{2.27}$$

where the denominator ensures that the updated weights sum up to one again. The state of the updated particles is equivalent to the state of the predicted particles:

$$x^{(i)} \triangleq x_+^{(i)}. \tag{2.28}$$

Finally, the estimate of the object's a posteriori state is obtained by the weighted sum of the particles:

$$x = \sum_{i=1}^{\nu} w^{(i)} x^{(i)}. \tag{2.29}$$

A drawback of the particle approximation is the degeneration of the particles, i.e. the weights concentrate on a very small number of particles. The degeneration is due to the fact, that the variance of the weights increases during the update step and never decreases. In order to avoid degeneration, particles with very small weights are eliminated while particles with a high weight are drawn several times during the so called resampling process. In the literature, several resampling strategies including multinomial, systematic, and residual resampling [DC05; DFG01] are available. The strategies differ in the computational complexity as well as in the approximation accuracy.

2.3.1 Importance Sampling

The particle filter requires to draw samples from arbitrary distributions. While uniform and Gaussian distributions allow for an efficient generation of the samples, it is in general not possible to draw samples directly from arbitrary PDFs $p(x)$. In this case, the particles are drawn from a proposal density function $q(x)$ which is similar to $p(x)$:

$$p(x) > 0 \Rightarrow q(x) > 0 \;\forall\; x \in \mathbb{R}^n. \tag{2.30}$$

A common choice for the proposal density $q(x)$ are uniform or Gaussian distributions with according parameters. However, the drawn samples do not approximate the PDF $p(x)$ but the proposal $q(x)$. Hence, the drawn particles have to be weighted using the importance weights

$$\tilde{w}(x^{(i)}) = \frac{p(x^{(i)})}{q(x^{(i)})}. \tag{2.31}$$

Finally, the obtained importance weights have to be normalized to obtain the according weight of each particle:

$$w(x^{(i)}) = \frac{\tilde{w}(x^{(i)})}{\sum_{j=1}^{\nu} \tilde{w}(x^{(j)})}. \tag{2.32}$$

Within the particle filter equation of the previous subsection, the importance sampling approach is applied for sampling the predicted particles if it is not possible to draw the particles directly from the distribution.

2.4 Multi-Object Tracking

The filters presented in the previous sections implicitly assume that only one track and one measurement are present. However, most applications require the simultaneous tracking of several objects and the measurement to track assignment is usually ambiguous due to inaccurate measurements, missed detections, and false alarms. Thus, several approaches to enable multi-object tracking using a bank of Kalman or particle filters have been proposed during the last decades.

The scheme in Figure 2.1 illustrates the commonly used sequential processing chain of these multi-instance trackers. First, a detection or segmentation algorithm is applied

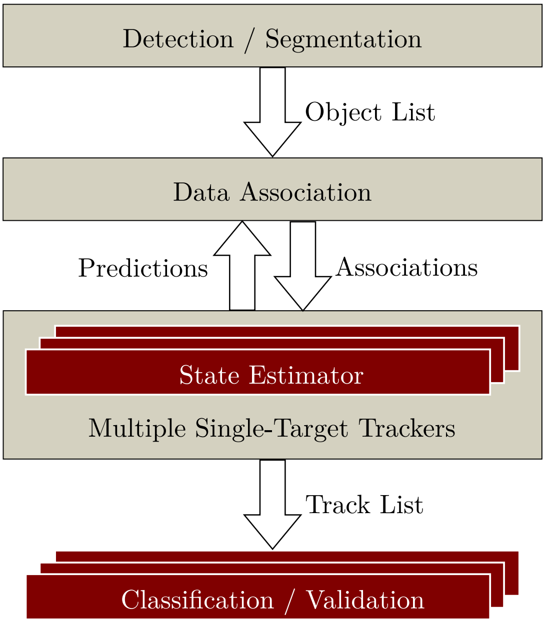


Figure 2.1: Multi-object tracking using a bank of single-object trackers [Mah09b].

to the raw data of the sensor to obtain the object list. An example for a detection algorithm is the application of a cascaded classifier [VJ01] on a raw image to detect faces, cars or other kinds of objects. Segmentation algorithms are applied to the data of sensors which deliver more than one measurement (e.g. a point cloud) for a single object to enable the use of the so-called point target assumption. Due to missed detections and/or false alarms, the number of measurements delivered by the detection or segmentation process is usually not equal to the number of tracked objects. Thus, a data association algorithm is required which assigns the received measurements to the currently available object individual single-object trackers. The determined associations are used to update the state estimate of each tracker. Finally, the list of tracks is passed to the classification and validation stage which decides if an object actually exists and determines the class of the object.

Obviously, the applied heuristics between the four steps of the scheme in Figure 2.1 lead to a loss of information. However, the most critical part in multi-object tracking applications is the data association step since the measurement to track assignments tend to be ambiguous. Additionally, an erroneous association decision at time k is irreversible which typically leads to track losses in complex situations.

2.4.1 Gating

In order to simplify the data association step, measurement validation procedures [BWT11, pp. 108ff.] are commonly used to restrict the number of track to measurement assignments. In Kalman filter applications, the quadratic form of the Mahalanobis distance (MHD) between a predicted measurement z_+ and a received measurement z is given by

$$d^2_{\mathrm{MHD}}(z, z_+) \triangleq (z - z_+)^{\mathrm{T}} \underline{\mathrm{S}}^{-1} (z - z_+) = \gamma^{\mathrm{T}} \underline{\mathrm{S}}^{-1} \gamma, \tag{2.33}$$

where $\underline{\mathrm{S}}$ is the according innovation covariance introduced in (2.17). The quadratic form of the MHD follows a χ^2 distribution with $\dim(z)$ degrees of freedom. The measurements within the gate of track i are obtained using

$$\mathrm{Z}^{(i)}(\vartheta) = \left\{ z : d^2_{\mathrm{MHD}} \left(z, z_+^{(i)} \right) < \vartheta \right\}, \tag{2.34}$$

where ϑ is the gating threshold. Using the gate probability p_G, which denotes the probability that the true measurement is located within the gating region, the gating threshold ϑ is obtained from the cumulative distribution function of the χ^2 distribution with $\dim(z)$ degrees of freedom. It is common to use gate probabilities corresponding to the σ intervals of the Gaussian distribution, e.g. $p_G = 0.9973$ for the 3σ gate. In case of small gates $\sqrt{\vartheta} \ll 3$, the probability that the true measurement is located within the gate is significantly smaller than one. Hence, the gating probability p_G has to be considered in the filter updates in addition to the detection probability.

2.4.2 Nearest Neighbor Approaches

The nearest neighbor (NN) algorithm simply associates the measurement with the smallest euclidean or Mahalanobis distance to each object. If there are several measurements within the gate of an object, the association is ambiguous and the probability of assigning an incorrect measurement is considerably high. Since erroneous associations are irreversible, the performance of the NN algorithm rapidly decreases in such situations. Additionally, the NN algorithm does not ensure the commonly used assumption that a measurement is generated by at most one object since it permits the association of a measurement to more than one object.

The global nearest neighbor (GNN) approach [BP99] ensures the above assumption by finding the best association hypothesis using all tracks and measurements. An association hypothesis for m measurements and n tracks is defined by the mapping

$$\theta : \{1, \ldots, n, \mathrm{NO}\} \rightarrow \{0, 1, \ldots, m\} \tag{2.35}$$

where "0" denotes the missed detection and NO represents the possibility of a new born object. Within an association hypothesis θ, a measurement z_j is required to be uniquely assigned to a track $x^{(i)}$, i.e. $\theta(i) = \theta(j) > 0$ if and only if $i = j$. Consequently, an association hypothesis does not necessarily assign the measurement with the smallest Mahalanobis distance to an object.

By representing all track to measurement associations using an assignment matrix A where each entry a_{ij} represents an association $\theta(i) = j$, finding the best association hypothesis corresponds to solving the assignment problem which was originally investigated in the context of assigning a number of jobs to the available workers. Hence, optimal assignment algorithms like the Hungarian method [Kuh55] or the Munkres algorithm [BL71; Mun57] are used to obtain the best association hypothesis θ. Alternative approaches to solve the assignment problem are the Jonker-Volgenant [JV87] and the auction [Ber88; Ber90] algorithm.

2.4.3 Probabilistic Data Association

The performance of the NN approaches significantly decreases in scenarios with a high number of false alarms due to the hard association decisions in ambiguous situations. The premise of the probabilistic data association (PDA) [BF88; BT75] is the calculation of a weighted state update using all possible track to measurement associations and a subsequent approximation of the posterior PDF using a single Gaussian distribution. Due to the weighted state update, the PDA avoids the hard and possibly erroneous association decisions of the NN approaches at the cost of an increased estimation error covariance [BWT11, p. 201].

In order to perform the weighted state update, the PDA determines the association probabilities $\beta^{(i,j)}$ for each track i and all measurements $j = 1, \ldots, m$ of the measurement set $\mathrm{Z}_{k+1} = \{z_1, \ldots, z_m\}$. Consequently, the posterior PDF of object $x^{(i)}$ is given by the weighted sum of the $m + 1$ association hypotheses:

$$p(x^{(i)}|z_1, \ldots, z_m) = \sum_{j=0}^{m} \beta^{(i,j)} p(x^{(i)}|z_j), \tag{2.36}$$

where $j = 0$ represents the missed detection and

$$\sum_{j=0}^{m} \beta^{(i,j)} = 1. \tag{2.37}$$

The PDF $p(x^{(i)}|z_j)$ represents the posterior state of object $x^{(i)}$ for the association of measurement z_j. In case of a missed detection, the posterior state of the object corresponds to the predicted state of the object. Obviously, the posterior PDF (2.36) does not follow a Gaussian distribution any more, even if all of the distributions $p(x^{(i)}|z_j)$ are Gaussians. Hence, an approximation of (2.36) is required to facilitate the application of the Kalman filter equations for the next measurement update. First, the posterior state is calculated for each association hypothesis:

$$\hat{x}^{(i,j)} = \hat{x}_{+}^{(i)} + \underline{\mathrm{K}}^{(i,j)}(z_j - z_{+}^{(i)}). \tag{2.38}$$

In case of a missed detection, the posterior state corresponds to the predicted state:

$$\hat{x}^{(i,0)} = \hat{x}_{+}^{(i)} \tag{2.39}$$

Finally, the posterior state estimate for object $x^{(i)}$ is obtained by the weighted mean over all possible association hypotheses:

$$\hat{x}^{(i)} = \sum_{j=0}^{m} \beta^{(i,j)} \hat{x}^{(i,j)}. \tag{2.40}$$

The according estimation error covariance is given by

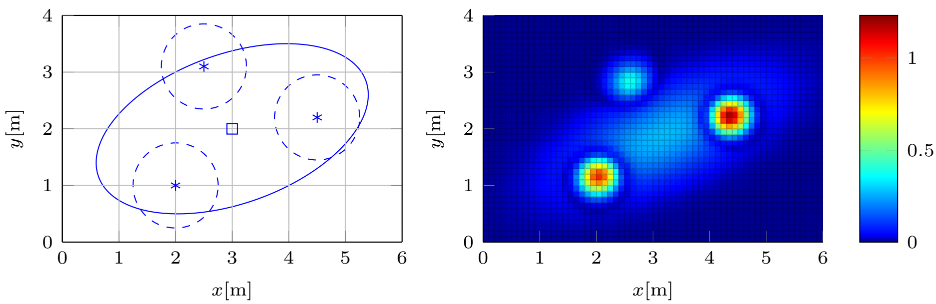
$$\underline{\mathrm{P}}^{(i)} = \sum_{j=0}^{m} \beta^{(i,j)} \left[\underline{\mathrm{P}}_{+}^{(i)} - \underline{\mathrm{K}}^{(i,j)} \underline{\mathrm{S}}^{(i,j)} \left[\underline{\mathrm{K}}^{(i,j)} \right]^{\mathrm{T}} + (\hat{x}^{(i,j)} - \hat{x}^{(i)})(\hat{x}^{(i,j)} - \hat{x}^{(i)})^{\mathrm{T}} \right] \tag{2.41}$$

where the last summand represents the uncertainty due to the approximation by a single Gaussian distribution. The contributing weight of an association hypothesis is proportional to the spatial likelihood $g(z_j|x_{+}^{(i)})$ of measurement j for track i. Explicit equations for the calculation of the probabilities $\beta^{(i,j)}$ are given for example in [BF88; BWT11].

The approximation of the posterior (2.36) using a single Gaussian distribution obviously implies an approximation error. Consider the example depicted by Figure 2.2a for illustration of the PDA approximation error where three measurements (blue stars) are obtained within the 3σ gate of a track (blue square). Additionally, the PDA takes into account the possibility of a missed detection. Figure 2.2b illustrates the absolute error

$$\Delta(x) = |p_{\text{PDA}}(x) - p(x^{(i)}|z_1, \dots, z_m)| \tag{2.42}$$

for the approximation of the multi-modal posterior density $p(x^{(i)}|z_1, \dots, z_m)$ by a single Gaussian distribution with posterior density $p_{\text{PDA}}(x) = \mathcal{N}(x; \hat{x}^{(i)}, \underline{\text{P}}^{(i)})$. Obviously, a significant approximation error is obtained at the locations of the three measurements. However, the approximation error is negligible in scenarios with at most one measurement and a high detection probability.



(a) Predicted measurement of a track (square) and three measurements (stars) with according 3σ bounds.

(b) Approximation error $\Delta(x)$: absolute difference between exact posterior PDF and the approximation using a single Gaussian distribution.

Figure 2.2: PDA example.

Similar to NN approaches, the performance of PDA is decreasing in situations where a measurement is located within the gate of several objects. To improve the tracking performance in such situations, the joint probabilistic data association (JPDA) algorithm [BWT11; FBS83] performs a weighted state update using all possible association hypotheses θ. An intuitive representation of all association hypotheses is a hypotheses tree. Figure 2.3 depicts the hypotheses tree for a scenario with two established tracks and two measurements where $\theta(i) = j$ represents the association of object i with measurement j. Obviously, the number of association hypotheses is combinatorial which results in an exponential complexity. Hence, the evaluation of all hypotheses is only feasible for a small number of tracks and measurements.

An application of JPDA to scenarios with a higher number of tracks and measurements is possible using Murty's algorithm [Mur68] which facilitates the determination of the k-best association hypotheses without evaluating all possible association hypotheses.

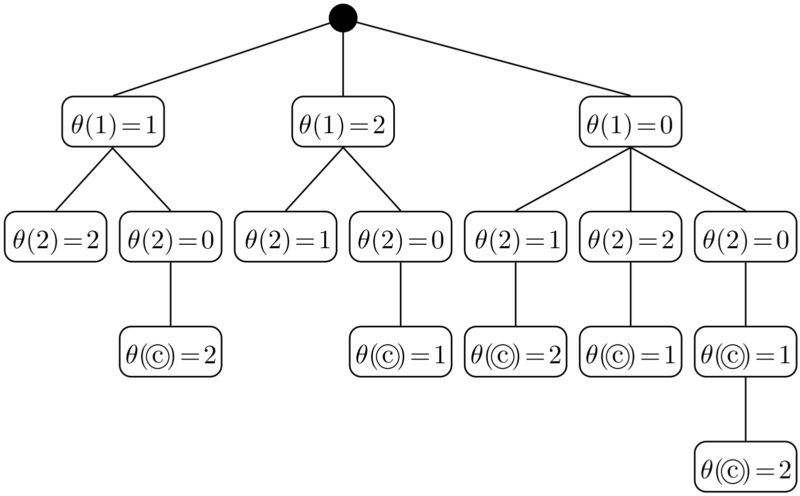


Figure 2.3: PDA hypotheses tree for two objects and two measurements: a node $\theta(i) = j$ denotes the association of object i to measurement j, where $j = 0$ represents the missed detection. For simplicity, the nodes for new born tracks and the clutter source are combined and are represented by the symbol $i =$ ©.

Murty's algorithm starts with finding the best association hypothesis using one of the optimal assignment algorithms. Afterwards, the obtained association hypothesis is used to find the next best hypotheses by removing one of the optimal associations $\theta(i) = j$ from the possible associations. A detailed example of Murty's algorithm is given in [BP99, pp. 346ff.]. Alternative solutions include the reduction of the computational complexity using the structure of the hypotheses tree with its recurring subtrees [Mas04] as well as an agglomerative grouping procedure [Mun11] which partitions the set of tracks and the set of measurements into groups. The agglomerative grouping facilitates a real-time implementation since a maximum computation time for each group is ensured using an adaptive gating threshold.

The integrated probabilistic data association (IPDA) [MES94] is an extension of the PDA which additionally incorporates track initiation and deletion based on an additional Markov chain for the object existence. In [ME04], the joint integrated probabilistic data association (JIPDA) filter is proposed which uses the joint association probabilities in the filter update and the additional existence estimation. The linear multi-target integrated probabilistic data association (LM-IPDA) [ML08] is an approximation of the JIPDA algorithm which considers nearby objects as scatterers in order to achieve a multi-object tracking algorithm with linear complexity. Due to the additional estimation of the existence probability, these approaches allow for a probabilistic track initiation and determination. The existence probability of a track represents the certainty that a track with a given association history represents an existing object under the assumed parameters of the measurement model. Further, the existence probabilities provide useful information for the decision process of higher-level systems since a more accurate risk analysis is possible. Thus, higher-level systems typically specify application dependent existence probability thresholds. A low threshold implies a fast track confirmation in combination with a possibly high number of false positive tracks while a high threshold significantly reduces the number of false positive tracks at the cost of a slower track confirmation.

2.4.4 Multi-Hypothesis Tracking

The multi-hypothesis tracking (MHT) [Rei79] algorithm significantly differs from the PDA based tracking algorithms. While the PDA based algorithms combine the multiple hypotheses of the measurement update step before the next update, the MHT approach keeps all hypotheses and expects that the subsequent measurements resolve the association uncertainty. Consider again the example depicted in Figure 2.2a for the illustration of MHT. Assume that there is only a single prior hypothesis H_1 which involves a single track $x^{(i)}$. In this case, a total number of four posterior hypotheses is possible if the birth of additional tracks is neglected. The first posterior hypothesis represents the missed detection of track $x^{(i)}$, i.e. all measurements are assumed to be clutter measurements. Each of the remaining three hypotheses associates one of the measurements to the existing track and expects the two other measurements to be originated by the clutter source. Obviously, the number of hypotheses increases exponentially in ambiguous situations. Hence, the computational complexity is reduced using pruning and merging techniques as well as grouping strategies. In [CH96], Cox and Hingorani present an efficient MHT implementation using Murty's algorithm [Mur68] which obtains the k best hypotheses without evaluating all possible hypotheses.

In addition to the hypothesis-oriented MHT approach in [Rei79], a track-oriented MHT implementation is possible [BP99]. While the hypothesis-oriented MHT propagates the hypotheses over time, the track-oriented MHT only propagates the obtained tracks over time and reconstructs the hypotheses using compatible tracks before the next measurement update. A set of tracks is considered to be compatible if each measurement contributes to at most one of the tracks.

2.4.5 Discussion

The preceding subsections introduced several commonly used multi-object tracking algorithms. In scenarios with unambiguous data association, high detection probability, and low false alarm rate, all methods achieve similar tracking results. However, the tracking performance significantly differs in challenging scenarios. In [BWT11, pp. 400ff.], the differences between the tracking results of a NN, a PDA, and a JPDA filter are illustrated using an example with two crossing objects. Using a NN algorithm, both tracks tend to follow the measurements of one of the objects after crossing which results in a huge estimation error for one of the objects. In contrast, the PDA algorithm uses the measurements of both objects within the filter update in ambiguous situations. Since both measurements are approximately equally likely during crossing, the PDA yields a compromise estimate between the ground truth positions of the two objects. The JPDA algorithm outperforms the NN and the PDA algorithm is such scenarios since it constructs global association hypotheses which ensure that each measurement is assigned to at most one track. During the crossing, the JPDA returns an increased estimation error covariances due to the association uncertainty. After the filter resolved the ambiguous situation, the estimation error covariances of the tracks decrease again and the JPDA algorithm delivers accurate track estimates.

In [BP99], the performance of the algorithms is evaluated with respect to the intensity of false alarms. Compared to a GNN implementation, the PDA delivers comparable results for approximately three times the false alarm density while MHT provides similar results for a false alarm densities of factor 10 to 100.

2.5 Consistency of State Estimators

In case of estimating static parameters, the consistency of an estimator implies that the estimate converges to the true value with an increasing number of measurements. Due to the time-variant state of the objects in tracking applications, an identical definition of consistency is not possible. In the following, the consistency criteria for state estimators proposed in [BF88, pp. 70ff.] are summarized.

In applications with an available ground truth state x, the state estimate $\hat{x}$ and the estimation error covariance $\underline{\mathrm{P}}$ are utilized to require the properties

$$E\{\widetilde{x}\} \overset{!}{=} 0 \tag{2.43}$$

$$E\{\widetilde{x}\widetilde{x}^{\mathrm{T}}\} \overset{!}{=} \underline{\mathrm{P}} \tag{2.44}$$

for the expectation values E of the residual $\widetilde{x} = x - \hat{x}$. While (2.43) ensures an estimator without bias, (2.44) requires the estimated covariance $\underline{\mathrm{P}}$ to match the mean square error. Obviously, (2.44) implicitly contains (2.43). The validity of (2.44) is examined using the normalized estimation error squared (NEES)

$$\epsilon = \widetilde{x}^{\mathrm{T}}\underline{\mathrm{P}}^{-1}\widetilde{x}. \tag{2.45}$$

In case of a linear Gaussian system, a tracking filter is consistent if the mean value $\bar{\epsilon}$ over N independent simulation runs is within the confidence interval of a χ^2 distribution with $N \cdot \dim(\mathrm{x})$ degrees of freedom (interval boundaries have to be divided by N). Using only a single run, approximately six out of 100 NEES values may be located outside of the 95% confidence interval in case of a two dimensional state.

Typically, an object's ground truth state is only available in case of simulated sensor data. Consequently, the NEES does not facilitate the evaluation of a filter's consistency in online applications. The normalized innovation squared (NIS) is designed to detect the filter divergence in online applications which coincides with the detection of unexpected measurements. In case of linear Gaussian motion and measurement models, the NIS is given by the Mahalanobis distance

$$\epsilon(z) = \gamma^{\mathrm{T}}\underline{\mathrm{S}}^{-1}\gamma, \tag{2.46}$$

of the measurement residual $\gamma = z - z_+$ with respect to the innovation covariance $\underline{\mathrm{S}}$ given by (2.17). Obviously, the NIS also follows a χ^2 distribution but the degrees of freedom are given by the dimension $\dim(\mathrm{z})$ of the measurement space. While the NEES is tested using the ensemble average value over N independent simulation runs, the evaluation of the NIS in online applications is realized using the time average value which presumes ergodicity.

In addition to consistency, the absolute estimation error of a state estimator is of great interest which is typically described using the root mean square (RMS) error

$$\mathrm{RMS}(\widetilde{x}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\|x - \hat{x}\|_2)^2} \tag{2.47}$$

averaged over N Monte-Carlo runs, where

$$\|x - \hat{x}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2} \tag{2.48}$$

is the euclidean distance between true and estimated state. Note, a meaningful RMS error is only obtained if all of the considered state components have the same units. Thus, it is reasonable to calculate separate RMS errors for position, velocity, and so forth.

Chapter 3

Random Finite Set Statistics

The multi-object tracking algorithms introduced in the previous chapter enable multi-object tracking by using a bank of object individual single-object trackers. Consequently, an explicit track to measurement association is required in these algorithms which tends to be ambiguous in scenarios with closely spaced objects or high false alarm rates. Additionally, a tracker of one object does not have any information about the existence of other nearby objects and the track initialization and termination is often based on heuristics.

A generalization of the single-object Bayes filter to multi-object tracking applications is feasible by modeling the states as well as the observations using random finite sets (RFSs) which are a well-known concept in point process theory [DV88; Mat75; SKM95]. Finite set statistics (FISST) [GMN97; Mah03; Mah07a] facilitates an intuitive application of the random finite set theory to multi-object tracking applications by casting the problem into the familiar framework of Bayesian statistics. Consequently, the resulting multi-object Bayes filter is a rigorous extension of the standard Bayes filter to multi-object tracking. The tutorial papers "Statistics 101" [Mah04] and "Statistics 102"[Mah13b] are suitable for the first steps into the FISST. For a more detailed mathematical foundation, refer to [Mah07a]. Details on the connection between FISST and measure theoretic probability are established in [Vo08; VSD05].

The intention of this chapter is to summarize the main concepts of FISST and to introduce the multi-object Bayes filter. Additionally, the recent approach of labeled RFSs [VV13b] is outlined and several computationally tractable approximations of the multi-object Bayes filter are discussed with respect to their computational complexity and performance.

3.1 Random Finite Sets

In single-object tracking, random vectors are conveniently used to represent the state of the objects. In the context of multi-object tracking, a drawback of random vectors is the missing representation of the uncertain number of objects. An RFS

$$\mathrm{X} = \{x^{(1)}, \ldots, x^{(n)}\} \tag{3.1}$$

consists of n unordered points with random object states $x^{(1)}, \ldots, x^{(n)}$ where $n \geq 0$ is a random number. Consequently, an RFS X naturally represents the uncertainty about the number of objects in a multi-object state and uses random vectors to represent the state of the individual objects. Furthermore, the RFS

$$\mathrm{Z} = \{z^{(1)}, \ldots, z^{(m)}\} \tag{3.2}$$

is conveniently used to describe the measurement process which returns a random number of measurements whose values $z^{(1)}, \ldots, z^{(m)}$ are also random. Here, the randomness in the number of received measurements is due to the possibility of false alarms and missed detections. The RFS representation of the multi-object state and the measurement process are fundamental for the derivation of the multi-object Bayes filter which is an extension the Bayes filter (see Section 2.1) to multi-object tracking applications.

3.1.1 Notation and Abbreviations

In order to distinguish RFSs from random vectors, multi-object states are denoted by capital letters (e.g. X) while single-object states are denoted by small letters (e.g. x). To simplify notations, identical symbols are used for an RFS and its realization in the following. Further, spaces are represented by blackboard bold letters. Hence, all state vectors x are elements of the state space $\mathbb{X}$ and all measurements z are elements of the measurement space $\mathbb{Z}$. The finite subsets of a space $\mathbb{X}$ are denoted by $\mathcal{F}(\mathbb{X})$ and all possible subsets comprising exactly n elements are represented by $\mathcal{F}_n(\mathbb{X})$.

For notational simplicity, a set $\mathrm{X} = \{x\}$ comprising only a single state vector is conveniently abbreviated by x. Further, the inner product of two continuous functions $f(x)$ and $g(x)$ is abbreviated using

$$\langle f, g \rangle \triangleq \int f(x)g(x)dx. \tag{3.3}$$

In case of discrete sequences, the integral on the right hand side of (3.3) reduces to a sum. Additionally, the multi-object exponential notation [VV13b]

$$h^{\mathrm{X}} \triangleq \prod_{x \in \mathrm{X}} h(x) \tag{3.4}$$

is used to shorten the notation for real-valued functions $h(x)$ (e.g. probability densities) which have to be evaluated for all state vectors of an RFS X. By definition, $h^{\mathrm{X}} = 1$ in case of $\mathrm{X} = \emptyset$.

The generalized Kronecker delta function

$$\delta_{\mathrm{Y}}(\mathrm{X}) \triangleq \begin{cases} 1, & \text{if } \mathrm{X} = \mathrm{Y} \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

is extensively used in this thesis and facilitates the application of the Kronecker delta function to arbitrary arguments like integers, vectors, and sets. Additionally, the indicator or inclusion function

$$1_{\mathrm{Y}}(\mathrm{X}) \triangleq \begin{cases} 1, & \text{if } \mathrm{X} \subseteq \mathrm{Y} \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

is used to determine if a set X is a subset of Y.

3.2 Multi-Object Probability Distributions

The spatial uncertainty of a random vector x is commonly represented by a probability density function $p(x)$. Analogously, a multi-object probability density function $\pi(\mathrm{X})$ facilitates the representation of the uncertainty about the multi-object state X which incorporates the uncertainty about the number of objects and their individual states. The multi-object probability density depends on the number of objects represented by X and is given by

$$\pi(\mathrm{X}) = \begin{cases} \pi(\emptyset) & \text{if } \mathrm{X} = \emptyset \\ \pi(\{x^{(1)}\}) & \text{if } \mathrm{X} = \{x^{(1)}\} \\ \pi(\{x^{(1)}, x^{(2)}\}) & \text{if } \mathrm{X} = \{x^{(1)}, x^{(2)}\} \\ \vdots & \vdots \end{cases} \tag{3.7}$$

In [Mah07a, pp. 349ff.], several illustrative examples for multi-object probability densities are revealed. One of the examples is a single twinkling star in the night sky which is only visible for an observer with probability r and has a spatial distribution

$p(x)$. Consequently, the multi-object probability density for this example is given by

$$\pi(\mathrm{X}) = \begin{cases} 1-r & \text{if } \mathrm{X} = \emptyset \\ r \cdot p(x) & \text{if } \mathrm{X} = \{x\} \\ 0 & \textit{otherwise}, \end{cases} \tag{3.8}$$

where the assumption of a single star implies a probability of $\pi(\mathrm{X}) = 0$ for all sets with cardinality $|\mathrm{X}| \geq 2$.

Due to the dependency of (3.7) on the number of objects, the evaluation of the integral over a multi-object probability density function requires the utilization of a set integral [Mah07a, pp. 360ff.]

$$\int \pi(X)\delta X = \sum_{i=0}^{\infty} \frac{1}{i!} \int\limits_{\mathbb{X}^i} \pi(\{x^{(1)}, ..., x^{(i)}\}) dx^{(1)} \cdots dx^{(n)}, \tag{3.9}$$

where $\pi(\{x^{(1)}, ..., x^{(i)}\}) = 0$ if $|\{x^{(1)}, ..., x^{(i)}\}| \neq i$. Obviously, the evaluation of the set integral (3.9) for all cardinalities $0, \ldots, \infty$ is not computationally tractable. However, for multi-object probability densities like the one given by (3.8) the computation of the set integral is possible since only a subset of all possible cardinalities has to be evaluated. Observe, that the evaluation of the set integral in this example still requires the computation of multi-dimensional integrals which may not be solved analytically in general.

The cardinality distribution of a random finite set X, which denotes the probability that X contains exactly n vectors, is given by

$$\rho(n) = \Pr(|\mathrm{X}| = n) = \frac{1}{n!} \int \pi\left(\left\{x^{(1)}, \ldots, x^{(n)}\right\}\right) dx^{(1)} \cdots dx^{(n)}. \tag{3.10}$$

Consequently, the cardinality distribution provides an estimate for the number of objects represented by X using e.g. the maximum of ρ or a weighted mean.

3.2.1 Independent Identically Distributed Cluster Random Finite Set

The multi-object probability density of an independent identically distributed (i.i.d.) cluster RFS with $\mathrm{X} = \{x^{(1)}, \cdots, x^{(n)}\}$ is given by

$$\pi(\mathrm{X}) = n! \cdot \rho(n) \cdot p(x^{(1)}) \cdots p(x^{(n)}) \tag{3.11}$$

where $\rho(n)$ is the probability distribution of the cardinality for $n \geq 0$ and $p(x^{(i)})$ denotes the probability density function representing the spatial distribution of the ith object.

3.2.2 Poisson Random Finite Set

A Poisson RFS is directly obtained from the i.i.d. cluster process by replacing the general cardinality distribution $\rho(n)$ with the Poisson distribution

$$\rho(n) = e^{-\lambda} \cdot \frac{\lambda^n}{n!}, \tag{3.12}$$

where λ is the expected number of objects. Consequently, the multi-object Poisson RFS is given by

$$\pi(\mathrm{X}) = e^{-\lambda} \cdot \lambda^n \cdot p(x^{(1)}) \cdots p(x^{(n)}). \tag{3.13}$$

In multi-object tracking applications, a Poisson RFS with intensity density

$$\kappa(z) = \lambda_c \cdot c(z) \tag{3.14}$$

is commonly used to model the false alarm process with a mean number of λ_c false alarms. In most applications, the probability density $c(z)$ is assumed to be a uniform distribution over the complete measurement space due to the absence of precise information.

3.2.3 Multi-Bernoulli Random Finite Set

An intuitive way to represent the uncertainty about the existence of an object is the usage of a Bernoulli RFS X. A Bernoulli RFS X is empty with probability $1 - r$ and is a singleton with probability r. The probability r is commonly referred to as the existence probability of the singleton representing an object with spatial distribution $p(x)$ on the space $\mathbb{X}$. The cardinality distribution of a Bernoulli RFS X is given by a Bernoulli distribution with parameter r and its probability density is given by [Mah07a, pp. 368]

$$\pi(\mathrm{X}) = \begin{cases} 1 - r, & \text{if } \mathrm{X} = \emptyset, \\ r \cdot p(x), & \text{if } \mathrm{X} = \{x\}. \end{cases} \tag{3.15}$$

A multi-Bernoulli RFS X is the extension of the Bernoulli RFS to the multi-object case. Under the assumption that the Bernoulli RFSs $\mathrm{X}^{(i)}$ of the objects are independent of each other, the multi-Bernoulli RFS for M objects is given by $X = \bigcup_{i=1}^{M} X^{(i)}$. Thus, the parameter set $\{(r^{(i)}, p^{(i)})\}_{i=1}^{M}$ completely defines a multi-Bernoulli RFS. The probability density function for $\mathrm{X} = \emptyset$ corresponds to the probability that none of the M objects exists:

$$\pi(\emptyset) = \prod_{j=1}^{M}(1 - r^{(j)}). \tag{3.16}$$

In order to obtain the probability density function for $\mathrm{X} = \{x^{(1)}, \ldots, x^{(n)}\}$, it is necessary to sum over all n possible permutations of the vectors $x^{(i)}$. Then, the multi-object probability density is given by [Mah07a]

$$\pi(\{x^{(1)}, ..., x^{(n)}\}) = \prod_{j=1}^{M}\left(1 - r^{(j)}\right) \sum_{1 \leq i_1 \neq \ldots \neq i_n \leq M} \prod_{j=1}^{n} \frac{r^{(i_j)} p^{(i_j)}(x^{(j)})}{1 - r^{(i_j)}}. \tag{3.17}$$

In the following, the probability density of a multi-Bernoulli RFS is abbreviated using $\pi = \{(r^{(i)}, p^{(i)})\}_{i=1}^{M}$. The cardinality distribution of a multi-Bernoulli RFS X is obtained by neglecting the spatial distribution in (3.17):

$$\rho(n) = \prod_{j=1}^{M}\left(1 - r^{(j)}\right) \sum_{1 \leq i_1 \neq \ldots \neq i_n \leq M} \prod_{j=1}^{n} \frac{r^{(i_j)}}{1 - r^{(i_j)}}. \tag{3.18}$$

According to [Mah07a, pp. 584-585], the probability hypothesis density (PHD) of a multi-Bernoulli RFS is given by

$$v(x) = \sum_{j=1}^{M} r^{(j)} p^{(j)}(x). \tag{3.19}$$

Consequently, the mean cardinality of a multi-Bernoulli RFS is

$$\hat{N} = \sum_{j=1}^{M} r^{(j)}. \tag{3.20}$$

In multi-object tracking applications, the multi-Bernoulli RFS is commonly used to model the object detection process. Additionally, the cardinality balanced multi-target multi-Bernoulli filter (see Section 3.7.3) approximates the multi-object posterior density using a multi-Bernoulli RFS and provides a tractable multi-object tracking algorithm which propagates only the parameters of the multi-Bernoulli RFS over time.

3.3 Labeled Multi-Object Probability Distributions

The multi-object probability distributions presented in the previous subsection are well suited for filtering multi-object states. In order to use the RFS based multi-object probability distributions for multi-object tracking, a subsequent track extraction as well as a track to track association for successive time steps is required.

In order to simplify the estimation of the objects' trajectories, the state $x \in \mathbb{X}$ of an object is augmented by a label $\ell \in \mathbb{L}$. The discrete label space $\mathbb{L}$ is given by $\mathbb{L} = \{\alpha_i : i \in \mathbb{N}\}$ where $\mathbb{N}$ represents the set of positive integers and $\alpha_i = \alpha_j$ implies $i = j$. For notational convenience, the mapping α is assumed to be the identity mapping $\alpha_i = i$ in the following.

Augmenting an unlabeled state $x \in \mathbb{X}$ with a label $\ell \in \mathbb{L}$ delivers the labeled state $\boldsymbol{x} = (x, \ell)$. Thus, a labeled multi-object state $\mathbf{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\}$ is a finite set on the space $\mathbb{X} \times \mathbb{L}$. However, this does not ensure distinct labels within the labeled multi-object state $\mathbf{X}$ since the same label may be assigned to several objects. The class of labeled RFSs [VV13b] ensures distinct labels within a multi-object state $\mathbf{X}$, i.e. a realization of the labeled multi-object state $\mathbf{X}$ may not contain two or more objects with identical label.

The labeled RFSs are based on an indicator function which invalidates labeled multi-object states whose labels are not distinct. The indicator function requires the set of track labels of a labeled multi-object state $\mathbf{X}$ which is given by

$$\mathcal{L}(\mathbf{X}) = \{\mathcal{L}(\boldsymbol{x}) : \boldsymbol{x} \in \mathbf{X}\}, \tag{3.21}$$

where $\mathcal{L}(\boldsymbol{x}) = \ell$ is the projection of the labeled state $\boldsymbol{x}$ (defined on the space $\mathbb{X} \times \mathbb{L}$) to the label space $\mathbb{L}$. Obviously, the track labels for a realization $\mathbf{X}$ of a labeled RFS are distinct if the number of track labels corresponds to the cardinality of multi-object state, i.e. $|\mathcal{L}(\mathbf{X})| = |\mathbf{X}|$. Thus, the distinct label indicator [VV13b]

$$\Delta(\mathbf{X}) = \delta_{|\mathbf{X}|}(|\mathcal{L}(\mathbf{X})|) \tag{3.22}$$

facilitates the discrimination between valid and invalid labeled multi-object states.

The unlabeled version of a labeled RFS is defined as the projection from the labeled state space $\mathbb{X} \times \mathbb{L}$ into the unlabeled state space $\mathbb{X}$ using marginalization which yields [VV13b]

$$\pi(\{x^{(1)}, \dots, x^{(n)}\}) = \sum_{(\ell^{(1)}, \dots, \ell^{(n)}) \in \mathbb{L}^n} \boldsymbol{\pi}(\{(x^{(1)}, \ell^{(1)}), \dots, (x^{(n)}, \ell^{(n)})\}).$$

An additional property of the class of labeled RFSs is the equivalence of the cardinality distribution of a labeled RFS and its unlabeled version.

The following subsections introduce the labeled RFS distributions used in this thesis. For further details about the class of labeled RFSs, refer to [VV13b].

3.3.1 Labeled Multi-Bernoulli Random Finite Set

A labeled multi-Bernoulli (LMB) RFS [VV13b] is the labeled version of a multi-Bernoulli RFS, where the component indices are interpreted as track labels. Consequently, an LMB RFS is completely described by the parameter set

$$\boldsymbol{\pi}(\mathbf{X}) = \{(r^{(\ell)}, p^{(\ell)})\}_{\ell \in \mathbb{L}}. \tag{3.23}$$

Following [VV13b], an LMB RFS is obtained by the sampling procedure given in Algorithm 3.1. First, the labeled multi-object state is initialized by $\mathbf{X} = \emptyset$. Afterwards, a sample of the uniform distribution $\mathcal{U}(0,1)$ is drawn for each track $\ell \in \mathbb{L}$ to determine whether the track ℓ exists or not. If the drawn random number is smaller than the existence probability of the Bernoulli component, the track is existing and is added to the multi-object state $\mathbf{X}$. Thus, the probability that track ℓ is included in the multi-object state $\mathbf{X}$ is proportional to its existence probability.

Algorithm 3.1 Sampling of an LMB RFS

$\mathbf{X} = \emptyset$;
for $\ell \in \mathbb{L}$ **do**
  sample u $\sim \mathcal{U}(0,1)$;
  if $u \leq r^{(\ell)}$ **then**
    sample $x \sim p^{(\ell)}(\cdot)$;
    $\mathbf{X} = \mathbf{X} \cup \{(x, \ell)\}$;
  end if
end for

The multi-object probability density of an LMB RFS on $\mathbb{X} \times \mathbb{L}$ with state space $\mathbb{X}$ and label space $\mathbb{L}$ is given by

$$\boldsymbol{\pi}\left(\left\{(x^{(1)}, \ell_1), ..., (x^{(n)}, \ell_n)\right\}\right) = \delta_n(|\{\ell_1, ..., \ell_n\}|) \prod_{i \in \mathbb{L}} \left(1 - r^{(i)}\right) \prod_{\ell=1}^{n} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)} p^{(\ell)}(x)}{1 - r^{(\ell)}}. \tag{3.24}$$

An LMB RFS is not a multi-Bernoulli RFS on $\mathbb{X}\times\mathbb{L}$, however its unlabeled version is a multi-Bernoulli RFS on $\mathbb{X}$. A more compact representation of an LMB density is obtained using the multi-object exponential notation:

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) w(\mathcal{L}(\mathbf{X})) p^{\mathbf{X}}, \tag{3.25}$$

where

$$w(L) = \prod_{i\in\mathbb{L}} \left(1 - r^{(i)}\right) \prod_{\ell\in L} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)}}{1 - r^{(\ell)}}, \tag{3.26}$$

$$p(x,\ell) = p^{(\ell)}(x). \tag{3.27}$$

The PHD and the mean cardinality of the unlabeled version of an LMB RFS are given by (3.19) and (3.20).

3.3.2 Generalized Labeled Multi-Bernoulli RFS

In an LMB RFS, all Bernoulli components are assumed to be statistically independent and the weight (3.26) of each realization $\mathbf{X}$ depends on the existence probabilities $r^{(\ell)}$ of the individual tracks $\ell \in \mathcal{L}(\mathbf{X})$. Consequently, the cardinality distribution of an LMB RFS follows the one of a multi-Bernoulli distribution, which is unimodal[1] [Sam65; Wan93].

The generalized labeled multi-Bernoulli (GLMB) RFS [VV13b] is a generalization of the LMB RFS which facilitates arbitrary weights and cardinality distributions. For a state space $\mathbb{X}$ and a label space $\mathbb{L}$, a GLMB RFS is given by

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{c\in\mathbb{C}} w^{(c)}(\mathcal{L}(\mathbf{X})) \left[p^{(c)}\right]^{\mathbf{X}}, \tag{3.28}$$

where the discrete index set $\mathbb{C}$ enables multiple realizations for a set of track labels $L = \mathcal{L}(\mathbf{X})$. Obviously, the weights $w^{(c)}(L)$ have to be normalized and the spatial distributions $p^{(c)}$ have to be probability density functions, i.e.

$$\sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} w^{(c)}(L) = 1, \tag{3.29}$$

$$\int p^{(c)}(x,\ell)\, dx = 1. \tag{3.30}$$

[1]Due to the discretization of the cardinality distribution, it is possible that two cardinalities n and $n+1$ are equally likely.

Comparing (3.28) with the alternative formulation (3.25) of an LMB RFS, it is obvious that an LMB RFS is a special case of the GLMB RFS which comprises only a single component c for each realization $\mathbf{X}$ and whose weights follow a multi-Bernoulli distribution (3.26) with parameters $r^{(\ell)}$. As mentioned in the previous subsection, an LMB RFS comprises statistically independent tracks. Since the existence of one track affects the association probabilities for other tracks, the measurement updated tracks are not statistically independent any more. Thus, an LMB RFS may not exactly represent the multi-object posterior. In contrast, the GLMB RFS facilitates multiple components (or hypotheses) for a set of track labels L and naturally represents the data association uncertainty of the filter update by generating a hypothesis c for each possible track to measurement association. Hence, a GLMB RFS allows for the exact representation of the statistically dependent tracks after the measurement update. The approximation error due to an LMB representation of the multi-object posterior will be examined in Section 5.2.3.

3.3.3 δ-Generalized Labeled Multi-Bernoulli RFS

The δ-generalized labeled multi-Bernoulli (δ-GLMB) RFS is an equivalent representation of a GLMB RFS which is used in [VV13b] to develop the δ-GLMB multi-object tracking filter. Within a δ-GLMB RFS, the parameters of the GLMB RFS are substituted by

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

Here, Ξ denotes an arbitrary discrete space and the density of a δ-GLMB RFS follows

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(I,\xi)}\right]^{\mathbf{X}}. \tag{3.31}$$

Obviously, (3.28) and (3.31) are equivalent in the general case. However, the δ-GLMB RFS facilitates a significant reduction of the computational complexity, if the discrete space Ξ represents the history of track label to measurement associations (see Section 5.1). In this case, an association history ξ implicitly contains the set of track labels I and allows to abbreviate $p^{(I,\xi)}$ by $p^{(\xi)}$ within the δ-GLMB equations. Consequently, the required number of spatial distributions within the δ-GLMB representation is only $|\Xi|$ while the GLMB representation requires $|\mathcal{F}(\mathbb{L}) \times \Xi|$, i.e. the δ-GLMB representation prevents the duplicate calculation of identical spatial distributions.

Since the δ-GLMB RFS is substantial for the derivation of the LMB filter in Chapter 5, some additional properties of the distribution are required. Following [VV13b], the cardinality distribution of a δ-GLMB RFS is obtained by

$$\rho(n) = \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum_{L\in\mathcal{F}_n(\mathbb{L})} w^{(I,\xi)}\delta_I(L) = \sum_{(I,\xi)\in\mathcal{F}_n(\mathbb{L})\times\Xi} w^{(I,\xi)}. \tag{3.32}$$

Hence, the probability of cardinality n is given by summing up the weights of all hypotheses (I,ξ) with $|I| = n$. The δ-GLMB representation further allows to extract information about a track with label ℓ using the PHD

$$\begin{aligned} v(x) &= \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum_{\ell\in\mathbb{L}} p^{(I,\xi)}(x,\ell) \sum_{L\subseteq\mathbb{L}} 1_L(\ell) w^{(I,\xi)}\delta_I(L) \\ &= \sum_{\ell\in\mathbb{L}} \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} 1_I(\ell) p^{(I,\xi)}(x,\ell) \end{aligned} \tag{3.33}$$

of its unlabeled version. Since the PHD comprises the sum over all labels $\ell \in \mathbb{L}$, the PHD of an individual track with label ℓ is given by

$$v^{(\ell)}(x) = \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} 1_I(\ell) p^{(I,\xi)}(x,\ell). \tag{3.34}$$

Here, the inclusion function $1_I(\ell)$ ensures that only hypotheses which include label ℓ are used. Since (3.34) represents the PHD of a single track with label ℓ, the integral over the PHD corresponds to the existence probability of a track with label ℓ:

$$\begin{aligned} r^{(\ell)} &= \int \Bigg(\sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} 1_I(\ell) p^{(I,\xi)}(x,\ell) \Bigg) dx \\ &= \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} 1_I(\ell) \int p^{(I,\xi)}(x,\ell) dx \\ &= \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} 1_I(\ell). \end{aligned} \tag{3.35}$$

The δ-GLMB representation of an LMB RFS with state space $\mathbb{X}$, finite label space $\mathbb{L}$, and parameter set $\boldsymbol{\pi} = \{r^{(\ell)}, p^{(\ell)}\}_{\ell\in\mathbb{L}}$ is given by

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{I\in\mathcal{F}(\mathbb{L})} w(I)\delta_I(\mathcal{L}(\mathbf{X})) p^{\mathbf{X}} \tag{3.36}$$

$$= \Delta(\mathbf{X}) w(\mathcal{L}(\mathbf{X})) p^{\mathbf{X}}, \tag{3.37}$$

where the weights $w(L)$ and spatial distributions $p(x,\ell)$ are given by (3.26) and (3.27). Since an LMB RFS comprises a union of independent Bernoulli tracks, an LMB RFS may not comprise several hypotheses for a given set of track labels. This corresponds to setting the discrete space $\Xi = \emptyset$ in the δ-GLMB density (3.31). Observe that (3.37) uses the property that all summands in (3.36) are zero except for $I = \mathcal{L}(\mathbf{X})$. Moreover, (3.37) is equivalent to the LMB density (3.25).

3.4 Multi-Object Likelihood Functions

The single-object likelihood function $g(z|x_+)$, used in the innovation step of the Bayes filter, only evaluates the distance between the predicted measurement and the received measurement with respect to the measurement uncertainty. In contrast, the multi-object likelihood function $g(\mathrm{Z}|\mathrm{X}_+)$ has to represent the entire measurement process, i.e., it has to incorporate the field of view (FOV) of the sensor, the detection probability, the false alarm rate, and even drop-outs of the data transmission (e.g. due to limited bandwidth). An explicit handling of the FOV is not necessary, since a state dependent detection probability usually incorporates the FOV information. Further, data transmission drop-outs are neglected in the following.

Figure 3.1 illustrates the standard multi-object measurement model defined in [Mah07a], which is based on the following assumptions: The sensor observes the objects in the scene with a single-object spatial likelihood function $g(z|x_+)$ and each measurement is generated by at most one object. Additionally, an existing object generates a measurement with the state dependent detection probability $p_D(x_+)$ and is not detected with probability $1 - p_D(x_+)$. The false alarm process is assumed to follow a Poisson distribution with an expected number of λ_c clutter measurements which are distributed according to a spatial distribution $c(z)$. In most applications, $c(z)$ is assumed to be uniformly distributed over the sensor's field of view. The false alarm process is assumed to be statistically independent of the object detection process and all of the measurements are assumed to be conditionally independent of the objects' state.

Due to missed detections, false alarms, and the spatial uncertainty of target generated measurements, the track to measurement association is often ambiguous. Since there is no a priori knowledge about the correct track to measurement association available, the best assumption is to assume all association hypotheses to be equally likely. Consequently, the multi-object likelihood functions in [Mah07a] average over all association hypotheses $\theta : \{1,\dots,n\} \to \{0,1,\dots,m\}$ for n objects and m measurements where '0' is appended to the set of measurements to cover missed detections. Using the assumption that a measurement is generated by at most one object, the following

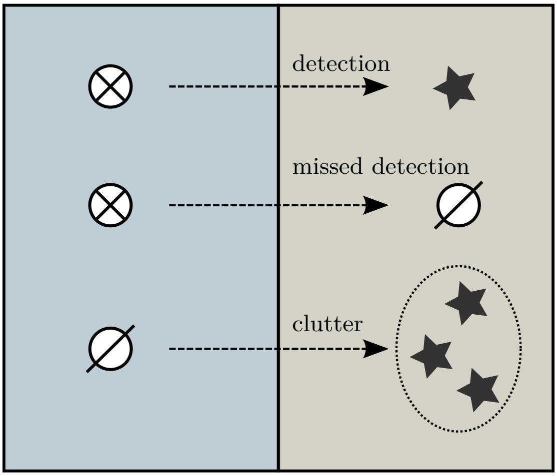


Figure 3.1: Possible associations within the multi-object likelihood function: measurements are either generated by an object or by the clutter source. Additionally, the likelihood is required to represent the possibility of a missed detection.

property has to be fulfilled: $\theta(i) = \theta(j) > 0$ if and only if $i = j$. The property ensures that a measurement $z_{\theta(i)}$ is uniquely assigned to track i. For $\theta(i) = 0$, no measurement is assigned to object i, i.e. object i is not detected.

Using the track to measurement associations θ, the standard multi-object likelihood function [Mah07a]

$$g(\mathrm{Z}|\mathrm{X}_+) = \pi_C(\mathrm{Z})\pi(\emptyset|\mathrm{X}_+) \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_D\big(x_+^{(i)}\big) \cdot g\big(z_{\theta(i)}|x_+^{(i)}\big)}{\big(1 - p_D\big(x_+^{(i)}\big)\big) \cdot \lambda_c c\big(z_{\theta(i)}\big)} \tag{3.38}$$

covers both missed detections and Poisson distributed false alarms. Here, $p_D(\cdot)$ is the state and sensor dependent detection probability, $g(\cdot|\cdot)$ is the single-object likelihood function, λ_c denotes the expected number of false alarms, and $c(\cdot)$ is the spatial distribution of the false alarms. Further, the probability that none of the objects is detected is abbreviated by

$$\pi(\emptyset|\mathrm{X}_+) = \prod_{i=1}^{n} \big(1 - p_D\big(x_+^{(i)}\big)\big), \tag{3.39}$$

and the probability that all measurements $z \in \mathrm{Z}$ are false alarms is given by

$$\pi_C(\mathrm{Z}) = e^{-\lambda_c} \prod_{z \in \mathrm{Z}} \lambda_c c(z). \tag{3.40}$$

The multi-object likelihood function averages over all association hypotheses and each association hypothesis is represented by an additive term in the multi-object likelihood (3.38). In case of a state independent detection probability $p_D(x_+) = p_D$ and equally likely association hypotheses, the multi-object likelihood function (3.38) differs from the likelihood of an MHT approach (see Section 2.4.4) just by a constant factor [Mah07a, p. 422–424].

Further multi-object likelihood functions for special cases like state dependent false alarms, extended objects, or unresolved objects are discussed in [Mah07a]. For the purpose of this thesis, (3.38) is sufficient.

3.5 Multi-Object Markov Densities

In single-object tracking, it is sufficient to predict an object's state x to the time of the next measurement using a suitable Markov density $f_+(x_+|x)$. Multi-object motion models are far more complex since the motion of the objects is only one part of the model. In addition to the object individual state transitions, a multi-object motion model has to represent object appearance and disappearance. Figure 3.2 illustrates the possible state transitions covered by the multi-object motion model. In general, the multi-object motion model may additionally incorporate object spawning [Mah07a], i.e. a new object is originated by an already existing object. However, spawning is not relevant for the applications considered within this work and is neglected in the following.

The multi-object Markov density mathematically resembles the multi-object likelihood function, since appearing and disappearing objects in the motion model correspond to false alarms and missed detections in the likelihood function. The standard multi-object motion model comprises the following assumptions: All objects move independent of each other according to a single-object Markov transition density $f_+(x_+|x)$, each object survives with a state dependent survival probability $p_S(x)$, and new objects appear according to a Poisson distributed birth density $\pi_B(\mathrm{X})$. Further, the birth density is assumed to be statistically independent of persisting objects. The multi-object Markov density for the state transitions depicted by Figure 3.2 is

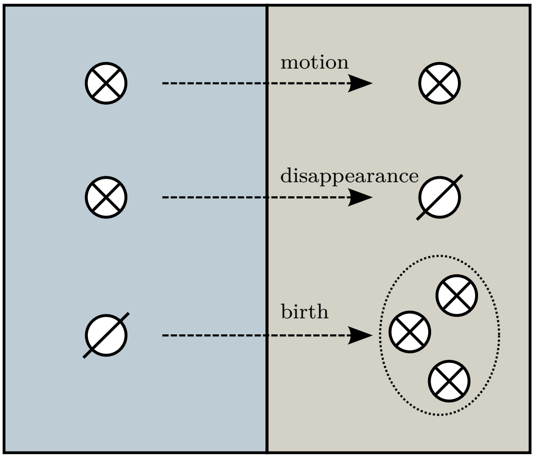


Figure 3.2: Illustration of the multi-object motion model: objects move according to their motion model or disappear during the prediction. Further, new objects appear based on a birth model.

represented by [Mah07a]

$$f_+(\mathrm{X}_+|\mathrm{X}) = \pi_B(\mathrm{X}_+)\pi_+(\emptyset|\mathrm{X}) \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_S(x^{(i)}) \cdot f_+(x_+^{(\theta(i))}|x^{(i)})}{(1 - p_S(x^{(i)})) \cdot \lambda_B p_B(x_+^{(\theta(i))})}, \tag{3.41}$$

where $p_S(\cdot)$ is the state dependent survival probability and λ_B is the expected number of new born objects distributed according to the probability density $p_B(\cdot)$. Similar to (3.38), the sum in (3.41) goes over all possible associations $\theta : \{1, \ldots, n'\} \rightarrow \{0, 1, \ldots, n\}$ where an association $\theta(i) = 0$ corresponds to the disappearance of object i while an association $\theta(i) > 0$ implies that object i survives. The probability that all objects within the predicted RFS are new born objects is given by

$$\pi_B(\mathrm{X}_+) = e^{-\lambda_B} \prod_{i=1}^{n} \lambda_B p_B(x_+^{(i)}) \tag{3.42}$$

and

$$\pi_+(\emptyset|\mathrm{X}) = \prod_{i=1}^{n'} \Big(1 - p_S(x^{(i)})\Big) \tag{3.43}$$

denotes the probability that none of the previously existing objects survives. For all associations $\theta(i) > 0$, the product under the sum in (3.41) cancels the contribution of the corresponding state vector $x^{(i)}$ to (3.42) and (3.43). For additional multi-object motion models covering extended targets, unresolved targets, and coordinated multi-object motion refer to [Mah07a].

3.6 Multi-Object Bayes Filter

The random finite set statistics allow for a rigorous extension of the single-object Bayes filter to multi-object filtering. Since the random variable in the multi-object Bayes filter is a random finite set, the obtained estimate naturally comprises the uncertain number of objects as well as the state uncertainty of the individual objects.

Using the Markov assumption, the entire information about the multi-object state at a time k is captured by the multi-object posterior density $\pi(\mathrm{X})$. The prediction of the multi-object posterior to the time of the next measurement $k+1$ is obtained using the Chapman-Kolmogorov equation and the multi-object Markov density (3.41) [Mah07a]:

$$\pi_+(X_+) = \int f(X_+|X)\pi(X)\delta X. \tag{3.44}$$

Obviously, (3.44) has the same form as the prediction of the single-object Bayes filter (2.5). However, the integral in (3.44) is a set integral of the form (3.9) and the random variables are random finite sets instead of random vectors.

The multi-object posterior density at $k+1$ is obtained using the multi-object Bayes filter update [Mah07a]

$$\pi(X|Z) = \frac{g(Z|X)\pi(X)}{\int g(Z|X)\pi(X)\delta X} \tag{3.45}$$

which uses the multi-target likelihood function $g(Z|X)$ (see Section 3.4) to update the predicted multi-object density. Again, the only differences to the single-object Bayes filter update (2.8) are the set integral in the denominator and the RFS random variables instead of the random vectors.

Similar to the single-object Bayes filter, the multi-object Bayes filter facilitates an implementation using SMC methods. Due to the computational complexity of the multi-object likelihood function (3.38), the SMC implementation of the multi-object Bayes filter has not been applied to real sensor data so far. Using an approximation of the multi-object likelihood function, a real-time capable implementation of the filter is presented in Chapter 4. Additionally, the proposed implementation incorporates the modeling of interactions between objects in the environment. Using the standard multi-object motion model as well as the standard multi-object likelihood, the δ-GLMB filter [VV13b] facilitates an analytical implementation of the multi-object Bayes filter using SMC and Gaussian mixture (GM) methods. A detailed description of the δ-GLMB filter is given in Section 5.1. Subsequently, the results of the δ-GLMB filter are used in Chapter 5 to derive the labeled multi-Bernoulli filter which provides a tractable and accurate approximation of the δ-GLMB filter.

3.7 Approximations of the Multi-Object Bayes Filter

During the last decade, several tractable approximations of the multi-object Bayes filter have been proposed. Inspired by the constant gain and the Kalman filter, which approximate the single-object Bayes filter using the moments of the probability density function, the probability hypothesis density (PHD) filter and the cardinalized probability hypothesis density (CPHD) filter propagate the first order moment of the multi-object probability density function $\pi(\mathrm{X}|\mathrm{Z}_{1:k})$ over time. The CPHD filter is a partial second moment approximation since it additionally propagates the full cardinality distribution. Another approximation is the cardinality balanced multi-target multi-Bernoulli (CB-MeMBer) filter which propagates the parameters of a multi-Bernoulli RFS over time instead of the moments of the multi-object probability density. The main differences of the approximations are additionally summarized in Appendix C.

3.7.1 Probability Hypothesis Density Filter

The probability hypothesis density (PHD) filter [Mah03] is an approximation of the multi-object Bayes filter which propagates only the first moment of a multi-object probability density $\pi(\mathrm{X})$ over time. In point process theory (PPT), the first-order multi-object moment $v(x)$ of $\pi(\mathrm{X})$ is conveniently called the intensity density [DV03]. However, in the context of multi-object filtering the intensity density is commonly referred to as the probability hypothesis density. The PHD $v(x)$ is obviously not a probability distribution, since the integral over the PHD represents the expected number of objects

$$\hat{N} = \int v(x)dx. \tag{3.46}$$

Within the derivation of the PHD filter prediction, the following assumptions are necessary:

- the motion of each object follows a single-object Markov transition density $f_+(x_+|x)$
- an existing object with state x survives from time step k to $k+1$ with probability $p_S(x)$
- new objects appear at time $k+1$ according to the birth intensity $b(x)$.

The derivation of the PHD filter prediction in [Mah03] additionally facilitates spawning of new objects based on existing objects. Since spawning is unlikely to occur in the scenarios investigated in this work, it is not considered in the following. The prediction of the prior PHD yields

$$v_+(x) = \int p_S(\xi) f_+(x|\xi) v(\xi) d\xi + b(x), \tag{3.47}$$

where the first term represents the motion of the surviving objects and $b(x)$ represents the new born objects.

The corrector step of the PHD filter is based on the standard multi-object measurement model introduced in Section 3.4. Thus,

- the likelihood that an object with state x generates a measurement z is given by the spatial likelihood $g(z|x)$.
- an object with state x is detected by the sensor with detection probability $p_D(x)$.
- the false alarm process follows a Poisson distribution and creates an average number of λ_c false alarms whose spatial distribution is given by the probability density $c(z)$.

In order to derive closed-form equations for the corrector, it is necessary to assume that the predicted PHD approximately follows a multi-object Poisson distribution (see Section 3.2.2). Using these assumptions, the posterior PHD [Mah03]

$$v(x) = \big(1 - p_D(x)\big) v_+(x) + \sum_{z \in \mathrm{Z}} \frac{p_D(x) g(z|x) v_+(x)}{\lambda_c c(z) + \int p_D(\xi) g(z|\xi) v_+(\xi) d\xi}, \tag{3.48}$$

comprises the possibility of a missed detection (represented by the first term) and the update of the predicted intensity $v_+(x)$ using each of the $|\mathrm{Z}|$ measurements (represented by the sum over all $z \in \mathrm{Z}$).

Due to the approximation of the full multi-object posterior by its first moment, the prediction and update equations of the PHD filter do not incorporate set integrals any more which significantly reduces the computational complexity. Similar to the single-object Bayes filter, the PHD filter facilitates an implementation using SMC methods [Mah07a; Sid03; VSD05] where the posterior intensity is approximated by

$$v(x) = \sum_{i=1}^{\nu} w^{(i)} \delta_{x^{(i)}}(x). \tag{3.49}$$

In contrast to SMC implementations of the single-object Bayes filter, the particle weights $w^{(i)}$ in the SMC implementation of the PHD filter are not normalized since the integral over the PHD corresponds to the estimated number of objects.

Based on the work on Gaussian sum filters in [AS72; SA71], Vo *et al.* proposed a Gaussian mixture implementation of the PHD filter, the Gaussian mixture probability hypothesis density (GM-PHD) filter [VM06], which approximates the posterior intensity by

$$v(x) = \sum_{i=1}^{J} w^{(i)} \mathcal{N}\left(x; \hat{x}^{(i)}, \underline{\mathrm{P}}^{(i)}\right), \tag{3.50}$$

where $\hat{x}^{(i)}$ is the mean value of the Gaussian distribution and $\underline{\mathrm{P}}^{(i)}$ is the according covariance matrix. In contrast to the Gaussian sum filters, the weights $w^{(i)}$ are not normalized, since the integral over the intensity function again represents the estimated number of targets. Similar to the Kalman filter, the GM-PHD filter assumes linear Gaussian motion and measurement models to ensure that the predicted and posterior distribution are again a Gaussian mixture. Additionally, the prediction step requires the birth density to be a Gaussian mixture and assumes a state independent survival probability:

$$p_S(x) = p_S.$$

Further, the detection probability is assumed to be state independent:

$$p_D(x) = p_D.$$

The assumptions of state independent survival and detection probabilities are due to the appearance of both $p_D(x)$ and $1 - p_D(x)$ in the corrector equation. Thus, modeling the detection probability as a Gaussian distribution incorporates a non-Gaussian distribution for $1 - p_D(x)$ [Mah07a, p. 625]. However, a closed-form solution for the GM-PHD filter may also be obtained if the state dependent survival and detection probabilities are modeled by exponential mixtures [RGV+13; VM06]. Since the GM-PHD filter is based on the standard equations of the Kalman filter, an application to slightly non-linear problems using the EKF or UKF introduced in 2.2.2 is straightforward. Further details about the derivation as well as pseudo-code for the implementation of the GM-PHD filter can be found in [VM06].

A weakness of the PHD filter is the unstable estimate of the number of objects [EWB05] which is due to the approximation of the multi-object distribution by its first moment. Erdinc *et al.* [EWB05] illustrate the small memory for the number of objects using a one target scenario where missed detections are present. By using a high detection probability, the estimated number of objects drops rapidly if a single missed detection occurs. Further, the estimated number of objects jumps from approximately zero to one, if the object is detected again after several missed detections. In [MSRD06], Mählisch *et al.* present a heuristic approach to achieve stable cardinality estimates where a constant gain filter is used to smooth the total weight of the PHD after each measurement update.

Despite the unstable cardinality estimate, the PHD filter has been applied to a wide range of applications. A principal reason for its popularity is the considerably lower computational complexity of the PHD filter compared to filters like JIPDA or MHT. Since the PHD filter does not require explicit track to measurement associations, the computational complexity is only of order $\mathcal{O}(mn)$, where m is the number of measurements and n corresponds to the number of objects. Applications of the PHD filter include but are not limited to uncalibrated aerial videos [PPP+09], airborne radar [JDSW12], ultra wideband radar [JZT+13], passive radar [PE13], sonar images [CRPB07], and video images [EAGG13]. Further, the PHD filter has been applied to extended object tracking [GLO10; GLO11; LGO11; Mah09a], RFS based simultaneous localization and mapping (SLAM) [MVAV11a; MVAV11b], and box-particle filtering [SGM+12].

In [RCV10; RCVV12], Ristic *et al.* introduced an adaptive birth intensity for the PHD filter which significantly reduces the required number of particles by adapting the proposal density for new born particles using the current set of measurements. A similar approach is also proposed in [HL10] for the GM-PHD filter. In [VPT06], Vo *et al.* propose a multiple model probability hypothesis density (MM-PHD) filter which utilizes several motion models to represent the motion of maneuvering objects. The approach in [VPT06] is consistent with the approach in [Mah12]. The MM-PHD filter is applied to a time-lapse cell microscopy imaging system in [RGV+13] where two linear motion models (random walk and CA) are used for the cells. In [MRD13; MRWD13], the MM-PHD filter is applied to environment perception at an urban intersection using a network of laser range finders utilizing a linear CV model and a non-linear CTRV model [SRW08]. While the CTRV model is well-suited for vehicles, trucks, and bikes, the CV model is appropriate for pedestrians and unknown object classes.

3.7.2 Cardinalized Probability Hypothesis Density Filter

In single-object tracking, a Kalman filter significantly outperforms constant-gain Kalman filters in most applications since a constant-gain filter only propagates the first moment of the probability density function while the Kalman filter additionally propagates the second-order moment. Thus, a filter which additionally propagates the second-order moment of the multi-object probability density function is supposed to show superior performance compared to the PHD filter and to solve the issue with unstable cardinality estimates. In [Mah01; Mah07a] the development of a second-order approximation of the multi-object Bayes filter is discussed. Since these filters are expected to be computationally intractable, Mahler proposed the cardinalized probability hypothesis density (CPHD) filter in [Mah07b]. The CPHD filter is a partial second-order multi-object moment filter, which propagates the entire cardinality distribution $\rho(n)$ of $\pi(\mathrm{X})$ in addition to the PHD $v(x)$. The CPHD filter is outlined in the following, for further details refer to [Mah07a; Mah07b; VVC07].

Since the CPHD filter is closely related to the PHD filter, the required assumptions for the multi-object motion model and the observations are similar. In the prediction step, the CPHD filter uses the same models for object motion, persistence, and newborn objects as the PHD filter. Additionally, the CPHD filter requires the a priori multi-object distribution to follow an i.i.d. cluster process.

For a given posterior intensity $v(x)$, the predicted intensity [VVC07]

$$v_+(x) = \int p_S(\xi) f_+(x|\xi) v(\xi) d\xi + b(x) \tag{3.51}$$

of the CPHD filter is equivalent to the predicted intensity of the PHD filter in (3.47) since spawning is neglected. Additionally, the predicted cardinality distribution is given by [VVC07]

$$\rho_+(n) = \sum_{j=0}^{n} \rho_B(n-j) \sum_{l=j}^{\infty} C_j^l \frac{\langle p_S, v\rangle^j \langle 1 - p_S, v\rangle^{l-j}}{\langle 1, v\rangle^l} \rho(l), \tag{3.52}$$

where ρ_B is the cardinality distribution of target birth, $C_j^l = \frac{l!}{j!(l-j)!}$ is the binomial coefficient, $p_S(x)$ is the survival probability, and $\langle \cdot, \cdot \rangle$ denotes the inner product defined by (3.3).

Equivalent to the PHD filter, the corrector step of the CPHD filter assumes a measurement model comprising a single-object spatial likelihood $g(z|x)$ and a state dependent detection probability $p_D(x)$. In contrast to the PHD filter, the false alarm process is modeled by the more general i.i.d. cluster process. Further, the predicted multi-object probability distribution has approximately to be an i.i.d. cluster process to obtain a closed-form solution. The posterior cardinality distribution and the posterior intensity for a set of measurements Z are given by [VVC07]

$$\rho(n) = \frac{\Psi^0\left[v_+, \mathrm{Z}\right](n)\cdot\rho_+(n)}{\left\langle\Psi^0\left[v_+, \mathrm{Z}\right], \rho_+\right\rangle}, \tag{3.53}$$

$$\begin{aligned} v(x) = &\frac{\left\langle\Psi^1\left[v_+, \mathrm{Z}\right], \rho_+\right\rangle}{\left\langle\Psi^0\left[v_+, \mathrm{Z}\right], \rho_+\right\rangle}\cdot(1-p_D(x))\cdot v_+(x) \\ &+\sum_{z\in\mathrm{Z}}\frac{\left\langle\Psi^1\left[v_+, \mathrm{Z}\backslash\{z\}\right], \rho_+\right\rangle}{\left\langle\Psi^0\left[v_+, \mathrm{Z}\right], \rho_+\right\rangle}\cdot\psi_z(x)v_+(x), \end{aligned} \tag{3.54}$$

where

$$\Psi^u\left[v, \mathrm{Z}\right](n) = \sum_{j=0}^{\min(|\mathrm{Z}|,n)}(|\mathrm{Z}|-j)!\rho_C(|\mathrm{Z}|-j)P^n_{j+u}\cdot\frac{\langle 1-p_D, v\rangle^{n-(j+u)}}{\langle 1, v\rangle^n}e_j(\Upsilon(v,\mathrm{Z})) \tag{3.55}$$

is the likelihood that a set of measurements Z is obtained for a scene containing n objects, $\rho_C(\cdot)$ is the cardinality distribution of clutter measurements, and $P^n_j = \frac{n!}{(n-j)!}$ is the permutation coefficient. The intensity $\kappa(z) = \lambda_c c(z)$ of the clutter measurements and the detection probability $p_D(x)$ yield the measurement likelihood

$$\psi_z(x) = \frac{\langle 1, \kappa\rangle}{\kappa(z)}p_D(x)g(z|x). \tag{3.56}$$

Further, $e_j(\mathrm{Y})$ is the j-th order elementary symmetric function of a finite set Y of real numbers which is defined by

$$e_j(\mathrm{Y}) = \sum_{S\subseteq\mathrm{Y},|S|=j}\prod_{\xi\in S}\xi, \tag{3.57}$$

where $e_0(\mathrm{Y}) = 1$ by definition, and $\Upsilon(v,\mathrm{Z}) = \{\langle v, \psi_z\rangle \ : \ z\in\mathrm{Z}\}$. An efficient implementation of the elementary symmetric function is possible using a double recursion [Mah07a, pp. 640–641]. Further, an alternative implementation method based on Vieta's Theorem [BE95] is proposed in [VVC07].

Similar to the PHD filter, the CPHD filter facilitates an implementation using sequential Monte-Carlo methods [Mah07a] or Gaussian mixtures [VVC07]. The computational complexity of the CPHD filter is $\mathcal{O}(m^3n)$, i.e. the CPHD filter is cubic in the number of measurements m while the PHD filter is linear. Compared to the PHD filter, the CPHD filter provides a more accurate and stable cardinality estimate since it estimates the cardinality distribution in addition to the first order multi-object moment while the PHD filter assumes a Poisson distributed multi-object posterior which implies a large standard deviation for the cardinality estimate.

In [FSU09], Fränken *et al.* observed the so-called ”spooky action at a distance” or spooky effect within the CPHD filter. The spooky effect is exemplified using a linear Gaussian example with two well separated objects, where object 1 is not detected at $k = 7$ and object 2 is not detected at $k = 13$. The detection probability in the example is given by $p_D = 0.98$. Figure 3.3 illustrates the weights of the extracted Gaussian components of the individual tracks and the sum of their weights. Obviously, the missed detection of one of the objects leads to an increasing weight for the detected object. The reason for the spooky effect is the scaling of a local estimate for the

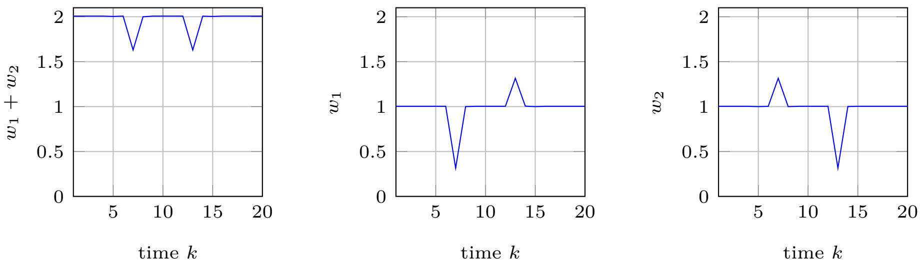


Figure 3.3: Track weights for the spooky effect example: sum of the weights of both objects (left), weight of object 1 (middle), weight of object 2 (right).

number of objects with a global cardinality distribution. An ad-hoc approach to eliminate the spooky effect is proposed in [FSU09] where individual CPHD filters are used for each track cluster. Within a PHD filter, the spooky action at a distance is not present since the PHD filter only estimates the intensity function.

Since the cardinality estimate of the CPHD filter is directly coupled with the false alarm process, an imprecise knowledge of the clutter rate leads to erroneous cardinality estimates. While the clutter rate is approximately constant in radar based air-surveillance systems, the assumption of a constant clutter rate is not fulfilled in a large number of applications like e.g. vehicular environment perception. The λ-CPHD filter proposed in [MVV11] circumvents this issue by additionally filtering the clutter rate. In [BVV13], Beard *et al.* show that the performance of the λ-CPHD filter is worse than the one of a CPHD filter with matching clutter rate. Thus, they propose a bootstrap Gaussian mixture cardinalized probability hypothesis density (GM-CPHD) filter which uses a λ-CPHD filter to estimate the clutter rate and passes the estimated clutter rate to a standard CPHD filter. The bootstrap GM-CPHD filter achieves similar performance to a GM-CPHD filter with matching clutter rate.

In the literature, several applications of the CPHD filter exist, including ground moving target tracking [PPR11; UEW07], vehicle environment perception [LCB12a; LCB12b], sonar applications [EWC08; GW12], passive radar data [PE13], and extended object tracking [LGO13]. An application of the λ-CPHD filter to road user tracking at an intersection using a network of laser range finders is presented in [RMD12].

3.7.3 Cardinality Balanced Multi-Target Multi-Bernoulli Filter

The cardinality balanced multi-target multi-Bernoulli (CB-MeMBer) filter [VVC09] conceptually differs from the PHD and CPHD filters introduced in the previous sections. While the PHD and the CPHD filter propagate the first moment of the multi-object probability density over time, the CB-MeMBer filter propagates the parameters of a multi-Bernoulli distribution. Hence, the CB-MeMBer filter requires the approximation of the multi-object posterior density by a multi-Bernoulli distribution at each time step. The prediction and the update step of the CB-MeMBer filter require similar assumptions as the PHD filter. However, the birth density has to follow a multi-Bernoulli RFS (instead of a birth intensity) which is independent of the multi-Bernoulli RFS of the surviving objects. Similar to the PHD filter, the clutter process has to follow a Poisson RFS but the approximations in the update step of the CB-MeMBer filter require a not too dense clutter distribution [VVC09].

The CB-MeMBer filter assumes that the multi-object posterior density is represented by a multi-Bernoulli distribution of the form

$$\pi = \left\{ \left(r^{(i)}, p^{(i)} \right) \right\}_{i=1}^{M} . \tag{3.58}$$

According to the standard multi-object motion model, objects may appear or disappear during the prediction of the multi-object state to the next time step. The state of each surviving object is predicted using a single-object motion model while new born objects are modeled using a multi-Bernoulli birth RFS with parameter set

$$\pi_B = \left\{ \left(r_B^{(i)}, p_B^{(i)} \right) \right\}_{i=1}^{M_B} , \tag{3.59}$$

where the existence probabilities $r_B^{(i)}$, the spatial distributions $p_B^{(i)}$, and the number of birth components M_B are application dependent. The expected number of new-born objects is obtained using (3.20):

$$\hat{N}_B = \sum_{i=1}^{M_B} r_B^{(i)}.$$

Since the multi-Bernoulli birth RFS is assumed to be independent of the multi-Bernoulli RFS of the surviving objects, the predicted multi-Bernoulli RFS is given by

$$\pi_+ = \left\{ \left(r_{+,S}^{(i)}, p_{+,S}^{(i)} \right) \right\}_{i=1}^{M} \cup \left\{ \left(r_B^{(i)}, p_B^{(i)} \right) \right\}_{i=1}^{M_B}, \tag{3.60}$$

where the parameters of the surviving objects are

$$r_{+,S}^{(i)} = r^{(i)} \eta_S(i), \tag{3.61}$$

$$p_{+,S}^{(i)}(x) = \frac{\left\langle f_+(x|\cdot), p^{(i)} p_S \right\rangle}{\eta_S(i)}, \tag{3.62}$$

$$\eta_S(i) = \left\langle p^{(i)}, p_S \right\rangle. \tag{3.63}$$

Here, $f_+(x|\cdot)$ denotes the single-object state transition density and $p_S(x)$ is the state dependent survival probability.

Due to the birth process, the total number of predicted Bernoulli distributions increases to $M_+ = M + M_B$ and the predicted multi-target density again is a multi-Bernoulli distribution of the form

$$\pi_+ = \left\{ \left(r_+^{(i)}, p_+^{(i)} \right) \right\}_{i=1}^{M_+}.$$

Since the measurement update of the predicted multi-object density is not a multi-Bernoulli distribution, the posterior multi-object density is approximated by

$$\pi \approx \left\{ \left(r_L^{(i)}, p_L^{(i)} \right) \right\}_{i=1}^{M_+} \cup \left\{ \left(r_U(z), p_U(\cdot; z) \right) \right\}_{z \in Z}, \tag{3.64}$$

where the subscript L denotes the legacy tracks and the subscript U denotes the measurement updated tracks. The legacy tracks correspond to the possibility that a track is not detected and a measurement updated Bernoulli distribution is obtained for each measurement z.

Using the state dependent detection probability $p_D(x)$, the existence probability and the spatial distribution of the legacy tracks are updated by

$$r_L^{(i)} = r_+^{(i)} \frac{1 - \left\langle p_+^{(i)}, p_D \right\rangle}{1 - r_+^{(i)} \left\langle p_+^{(i)}, p_D \right\rangle}, \tag{3.65}$$

$$p_L^{(i)}(x) = p_+^{(i)}(x) \frac{1 - p_D(x)}{1 - \left\langle p_+^{(i)}, p_D \right\rangle}. \tag{3.66}$$

The existence probability and the spatial distribution of the measurement updated tracks are approximated by

$$r_U(z) = \frac{\sum_{i=1}^{M_+} \frac{r_+^{(i)}(1-r_+^{(i)})\left\langle p_+^{(i)}, \psi_z \right\rangle}{\left(1-r_+^{(i)}\left\langle p_+^{(i)}, p_D \right\rangle\right)^2}}{\kappa(z) + \sum_{i=1}^{M_+} \frac{r_+^{(i)}\left\langle p_+^{(i)}, \psi_z \right\rangle}{1-r_+^{(i)}\left\langle p_+^{(i)}, p_D \right\rangle}}, \tag{3.67}$$

$$p_U(x;z) = \frac{\sum_{i=1}^{M_+} \frac{r_+^{(i)}}{1-r_+^{(i)}} p_+^{(i)}(x)\psi_z(x)}{\sum_{i=1}^{M_+} \frac{r_+^{(i)}}{1-r_+^{(i)}} \left\langle p_+^{(i)}, \psi_z \right\rangle}, \tag{3.68}$$

where

$$\psi_z(x) = g(z|x)p_D(x),$$

$g(z|x)$ is the single object measurement likelihood, $p_D(x)$ is the state dependent detection probability, and $\kappa(\cdot)$ denotes the intensity of the Poisson distributed clutter process. The parameters $r_U(z)$ and $p_U(x;z)$ of the Bernoulli approximation are chosen to match the PHD or intensity function of the posterior probability generating functional (PGFL). However, the approximation requires a high detection probability and not too dense clutter [VVC09].

The CB-MeMBer filter facilitates an implementation using SMC methods. The SMC implementation is commonly used in case of highly non-linear systems and the spatial distribution of each Bernoulli distribution is approximated by a set of weighted particles:

$$p^{(i)}(x) = \sum_{j=1}^{\nu^{(i)}} w^{(i,j)} \delta_{x^{(i,j)}}(x). \tag{3.69}$$

In contrast to the PHD and CPHD filter, the weights $w^{(i,j)}$ in (3.69) have to be normalized to ensure a valid probability density, i.e. $\int p^{(i)}(x)dx = 1$.

In case of state independent detection and survival probabilities as well as linear motion and measurement models, the CB-MeMBer filter can be implemented using Gaussian mixtures, where the spatial distribution of each Bernoulli distribution (including the birth components) has to be a mixture of Gaussians:

$$p^{(i)}(x) = \sum_{j=1}^{J^{(i)}} w^{(i,j)} \mathcal{N}\left(x; \hat{x}^{(i,j)}, \underline{\mathrm{P}}^{(i,j)}\right). \tag{3.70}$$

Here, $w^{(i,j)}$ are the normalized weights of the Gaussians with mean value $\hat{x}^{(i,j)}$ and covariance matrix $\underline{\mathrm{P}}^{(i,j)}$. The Gaussian mixture implementation may also be used in case of slightly non-linear motion or measurement models by using an EKF or a UKF implementation. Explicit equations for the GM and SMC implementation of the CB-MeMBer filter are given in [VVC09].

Compared to the PHD and CPHD filter, the principal advantage of the CB-MeMBer filter is the intuitive and straightforward extraction of track estimates due to the representation of each object by a Bernoulli distribution. Especially in SMC implementations of the PHD and the CPHD filter, the extraction of track estimates is more challenging since computationally expensive and error-prone clustering algorithms are required. An additional advantage of the CB-MeMBer filter is the estimation of object individual existence probabilities which are desired in applications like vehicle environment perception. Similar to the PHD filter, the complexity of the CB-MeMBer filter is linear in the number of measurements and linear in the number of objects.

While the PHD and CPHD filters have been applied to a huge amount of applications, the CB-MeMBer filter is rarely used in the literature. Among the applications of the CB-MeMBer filter are multi-object tracking using sensor networks [WZ09a; WZ09b; WZ10] and video images [HVV13; HVVS11; VVPS10]. In [VVHM13], the robust multi-Bernoulli filter is proposed which additionally estimates the clutter density and the detection probability. Further, an adaptive birth distribution for the CB-MeMBer filter is introduced in [RMWD13]. Similar to the adaptive birth intensities for the PHD and CPHD filter, the adaptive birth distribution for the CB-MeMBer filter concentrates around the measurements which are far away from any existing tracks. In [DK13], an extension of the CB-MeMBer filter to support multiple motion models is proposed.

A possible reason for preferring the PHD and CPHD filters are the issues of the CB-MeMBer filter with a low signal to noise ratio. Specifically, the CB-MeMBer filter tends to overestimate the cardinality in scenarios with a high clutter rate, which is due to the assumptions used in the update step of the filter.

3.8 Performance Evaluation of Multi-Object Tracking Algorithms

3.8.1 Optimal Subpattern Assignment Metric

In order to evaluate the performance of multi-object tracking algorithms, a metric to compare the estimated set of objects with the set of existing objects is necessary. Thus, in addition to the state estimation error considered by single-object miss distances like the Euclidean or Mahalanobis distance, the required metric has to incorporate the cardinality error. One of the first approaches for a multi-object miss distance is the Optimal Mass Transfer (OMAT) metric [HM04b] which is based on the Wasserstein distance and resolves some issues of the Hausdorff metric in multi-object filtering applications. In [SVV08], Schuhmacher *et al.* pointed out some weaknesses of the OMAT. For example, the OMAT is undefined if one of the two sets is empty. Further, the OMAT distance does not penalize multiple estimates for a single object. In order to resolve the described weaknesses of the OMAT, the optimal subpattern assignment (OSPA) metric is proposed in [SVV08].

In the OSPA metric, the distance of two state vectors x and y is calculated by

$$d_c(x, y) = \min(c, d(x, y)), \tag{3.71}$$

where $c > 0$ is the cut-off parameter. For two finite sets $\mathrm{X} = \{x^{(1)}, \ldots, x^{(m)}\}$ and $\mathrm{Y} = \{y^{(1)}, \ldots, y^{(n)}\}$, $m \leq n$, the OSPA distance of order p is given by

$$d_p^{(c)}(\mathrm{X}, \mathrm{Y}) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d_c(x^{(i)}, y^{(\pi(i))})^p + c^p (n - m) \right) \right)^{\frac{1}{p}}, \tag{3.72}$$

where Π_k is the set of permutations on $\{1, 2, \ldots, k\}$. For $m > n$, the parameters X and Y of equation (3.72) have to be switched.

Obviously, a higher order p assigns more weight to large distances [SVV08] which leads to a higher penalization of outliers. In the OSPA metric, the effect of higher orders p is attenuated due to the cut-off parameter c which restricts the maximum distance. Further, the cut-off parameter c prevents that a high distance between an estimated state and the true position leads to a larger OSPA distance than a cardinality error due to a missed detection.

During the development of multi-object tracking algorithms, the combined localization and cardinality error represented by the OSPA metric conceals the reason for the large distance. In order to determine the main contribution to the OSPA distance, the metric may be split up into a localization error

$$d_p^{(c,loc)}(\mathrm{X}, \mathrm{Y}) = \left(\frac{1}{n} \cdot \min_{\pi \in \Pi_n} \sum_{i=1}^{m} d_c(x^{(i)}, y^{(\pi(i))})^p \right)^{\frac{1}{p}} \tag{3.73}$$

and a cardinality error

$$d_p^{(c,card)}(\mathrm{X}, \mathrm{Y}) = \left(\frac{c^p(n-m)}{n} \right)^{\frac{1}{p}}. \tag{3.74}$$

Although the separated errors are no longer a metric on the space of finite subsets, they provide additional information for further improvements of the multi-object tracking algorithms [SVV08].

For an efficient computation of the OSPA distance, the following procedure is suggested in [SVV08]: expand the smaller set X by adding $n - m$ state vectors $x^{(j)}$, $j = m+1, \ldots, n$ which satisfy the condition

$$d(x^{(j)}, y^{(i)}) > c \; \forall \; i = 1, \ldots, n. \tag{3.75}$$

Thus, each of the added state vectors increases the OSPA distance by exactly the same value as a cardinality error. Afterwards, the p-th order distance between all points of the two subsets has to be calculated. Instead of evaluating (3.72) for all possible permutations, an optimal assignment algorithm like the Hungarian method [Kuh55] is used to determine the permutation which obtains the minimum distance.

3.8.2 Track Based Optimal Subpattern Assignment

The OSPA metric evaluates the result of a multi-object tracking system with respect to the number of objects and the residual between the estimated and the true object states for a single time step. Thus, the OSPA does not incorporate information about track continuity or switching of track IDs.

In [RVCV11], Ristic *et al.* proposed the optimal subpattern assignment for tracks (OSPAT), which extends the OSPA metric by incorporating labeling information in the distance calculation. The distance of two labeled state vectors $\boldsymbol{x} = (x, l)$ and $\boldsymbol{y} = (y, s)$ in the OSPAT is given by

$$d(\boldsymbol{x}, \boldsymbol{y}) = \Big(d(x,y)^p + \big(\alpha(1 - \delta_l(s))\big)^p \Big)^{\frac{1}{p}}, \tag{3.76}$$

where l and s are the labels of the tracks and the parameter $\alpha \in [0, c]$ impacts the penalty due to labeling errors.

Choosing $\alpha = 0$, the second term of (3.76) disappears. Consequently, the OSPAT distance is equivalent to the OSPA distance in this case and labeling errors are neglected. In case of $\alpha = c$, the distance (3.76) of two tracks with labels $l \neq s$ is $d(\boldsymbol{x}, \boldsymbol{y}) = c$ and the mismatching labels are penalized like a missed detection. The optimal assignment of the labels is determined using standard two-dimensional assignment algorithms [RVCV11].

3.8.3 Multi-Object Divergence Detectors

In single-object tracking, online consistency checks of tracking filters are usually performed using the NIS introduced in Section 2.5. In [Mah13a], Mahler proposes to interpret the NIS given by (2.46) as an approximation of the generalized normalized innovation squared (GNIS)

$$\mathrm{GNIS}(z) = -2 \log f(z), \tag{3.77}$$

where

$$f(z) \triangleq \int g(z|x) p_+(x) dx \tag{3.78}$$

is the normalization constant of the Bayes filter update (2.8).

In case of a Kalman filter, the GNIS is given by [Mah13a]

$$\begin{aligned} \mathrm{GNIS}(z) &= -2 \log f(z) \\ &= -2 \log \left(\frac{1}{\sqrt{\det 2\pi \underline{\mathrm{S}}}} \cdot \exp\left(-\frac{1}{2} \gamma^{\mathrm{T}} \underline{\mathrm{S}}^{-1} \gamma \right) \right) \\ &= \log \det(2\pi \underline{\mathrm{S}}) + \gamma^{\mathrm{T}} \underline{\mathrm{S}}^{-1} \gamma = \log \det(2\pi \underline{\mathrm{S}}) + \mathrm{NIS}(z), \end{aligned} \tag{3.79}$$

where $\underline{\mathrm{S}}$ is the innovation covariance and $\gamma = z - z_+$ is the residual between the predicted and the actual measurement. Since the NIS only penalizes the residual γ, it successfully detects underestimated process or measurement noises. However, the NIS does not explicitly enforce accurate state estimates since the magnitude of the innovation covariance matrix is neglected. The GNIS additionally penalizes high uncertainties within the innovation covariance $\underline{\mathrm{S}}$. Thus, more accurate state estimates result in a diminishing value of the GNIS. In contrast to the NIS, the GNIS is not χ^2 distributed any more due to the additional term $\log \det(2\pi \underline{\mathrm{S}})$.

The definition of the GNIS using the normalization constant of the Bayes filter enables a rigorous extension of the GNIS to multi-object tracking. Using the normalization constant

$$f(\mathrm{Z}) \triangleq \int g(\mathrm{Z}|\mathrm{X})\pi_{+}(\mathrm{X})\delta\mathrm{X} \tag{3.80}$$

of the multi-object Bayes filter, the multi-object generalized normalized innovation squared (MGNIS) is given by [Mah13a]

$$\mathrm{MGNIS}(Z) = -2\log f(Z). \tag{3.81}$$

Although closed-form expressions for the MGNIS have been proposed for the PHD and CPHD filters in [Mah13a], no results using simulated or real data have been presented for these filters so far. The properties of the MGNIS as well as its application to tracking algorithms using labeled RFSs are investigated in Section 5.4.

Chapter 4

Multi-Object Bayes Filter Incorporating Object Interactions

The sequential Monte-Carlo (SMC) implementation of the multi-object Bayes filter has not attracted much attention in the past since the enormous computational complexity prevents a real-time implementation in applications comprising a huge number of objects. However, the full multi-object Bayes filter facilitates the incorporation of object interactions in the prediction step of the filter due to the representation of the entire environment using random finite sets (RFSs). Compared to the PHD, CPHD, and CB-MeMBer filter, this is a significant benefit. While the PHD and CPHD filter lose the set representation of the multi-object state in consequence of the moment approximation, the representation of the multi-object state by statistically independent Bernoulli distributions prevents object interactions in the CB-MeMBer filter. Especially in scenarios with closely spaced objects, the incorporation of object interactions is expected to provide more accurate and physically meaningful predicted multi-object states. Moreover, a simplification of the data association in the measurement update is achieved. In order to enable a real-time capable implementation of the SMC multi-object Bayes filter, an approximation of the multi-object likelihood function is proposed which significantly reduces the computational complexity of the measurement update. Further, an approach to determine the existence probability of an object within the SMC multi-object Bayes filter is introduced. The approach is built upon the proposed track extraction algorithm which uniquely assigns a track ID within a realization of an RFS.

First, the SMC implementation of the full multi-object Bayes filter is reviewed. Afterwards, several possibilities to incorporate object interdependencies into the prediction step of the multi-object Bayes filter are proposed. Based on the improved prediction step, an approximation of the multi-object likelihood function which allows for a real-time capable implementation of the multi-object Bayes filter is presented. Finally, a track extraction scheme is proposed which additionally facilitates the determination of an object's existence probability.

4.1 Sequential Monte Carlo Implementation of the Multi-Object Bayes Filter

The sequential Monte-Carlo (SMC) implementation of the full multi-object Bayes filter is closely related to the SMC implementation of the single-object Bayes filter. A particle of the single-object Bayes filter is a sample of the spatial distribution $p(x)$, i.e. usually a vector $x^{(i)} \in \mathbb{R}^n$. In contrast, a multi-object particle of the multi-object Bayes filter is a sample of the multi-object probability density $\pi(\mathrm{X})$ and is given by a set of state vectors

$$\mathrm{X}^{(i)} \triangleq \left\{x^{(1)}, \ldots, x^{(n)}\right\}, \tag{4.1}$$

where n denotes the number of objects which are represented by the multi-object particle $\mathrm{X}^{(i)}$. A state vector $x^{(i)}$ within a multi-object particle is conveniently called "particle" in the following. Finally, the multi-object probability density function is approximated using ν multi-object particles:

$$\pi(\mathrm{X}) \cong \sum_{i=0}^{\nu} w^{(i)} \cdot \delta_{\mathrm{X}^{(i)}}(\mathrm{X}). \tag{4.2}$$

In the following, the SMC implementation of the full multi-object Bayes filter is summarized. For further details, refer to [Mah07a; MVSB06; SW03; VSD05].

4.1.1 Prediction

The predictor step of the multi-object Bayes filter is introduced in Section 3.6 and is given by

$$\pi_+(\mathrm{X}_+) = \int f_+(\mathrm{X}_+|\mathrm{X}) \cdot \pi(\mathrm{X})\delta\mathrm{X}. \tag{4.3}$$

While it is sufficient to predict each particle according to the Markov motion model in the single-object Bayes filter, the prediction step of the multi-object Bayes filter has to represent the multi-object motion model introduced in Section 3.5. Consequently, the prediction of the multi-object particles has to incorporate object appearance and disappearance in addition to the motion of persisting objects. Thus, a predicted multi-object particle is given by the union

$$\mathrm{X}_{+}^{(i)} = \mathrm{X}_{+,S}^{(i)} \cup \mathrm{X}_{B}^{(i)} \tag{4.4}$$

where $\mathrm{X}_{+,S}^{(i)}$ is the set of surviving particles and $\mathrm{X}_{B}^{(i)}$ is the set of new born particles.

In order to obtain the persisting multi-object particle, the set of persisting particles within the multi-object particle $\mathrm{X}^{(i)} = \{x^{(1)}, \ldots, x^{(n)}\}$ is required. The persisting particles are determined using a multi-Bernoulli distribution where the persistence probability $p_S(x^{(j)})$ is used as a parameter. Due to the multi-Bernoulli distribution, the persistence probabilities of the objects within the multi-object particle are assumed to be statistically independent. The probability that a subset $\{x^{(1)}, \ldots, x^{(n')}\}$ persists is given by

$$\pi\left(\left\{x^{(1)}, \ldots, x^{(n')}\right\} \middle| \mathrm{X}^{(i)}\right) = \prod_{x \in \mathrm{X}^{(i)}} (1 - p_S(x)) \cdot \prod_{x \in \left\{x^{(1)}, \ldots, x^{(n')}\right\}} \frac{p_S(x)}{1 - p_S(x)}. \tag{4.5}$$

Using (4.5), the probability that all of the particles within the multi-object particle persist is given by

$$\pi\left(\left\{x^{(1)}, \ldots, x^{(n)}\right\} \middle| \mathrm{X}^{(i)}\right) = p_S(x^{(1)}) \cdots p_S(x^{(n)})$$

while the probability that none of the particles survives is given by

$$\pi\left(\emptyset | \mathrm{X}^{(i)}\right) = (1 - p_S(x^{(1)})) \cdots (1 - p_S(x^{(n)})).$$

Since a multi-Bernoulli distribution is the union of M independent Bernoulli distributions, the persistence of each particle is determined independently instead of drawing the persisting objects directly from (4.5). Thus, a uniformly distributed random number $\zeta^{(j)}$ is drawn for each of the particles in $\mathrm{X}^{(i)}$ and the set of persisting particles is given by

$$\mathrm{X}_{S}^{(i)} = \left\{x : \zeta^{(j)} < p_S(x^{(j)}) \ \forall\ j = 1, \ldots, |\mathrm{X}^{(i)}|\right\}, \tag{4.6}$$

i.e. a particle persists if the drawn random number is smaller than the state dependent survival probability. Afterwards, each of the $j = 1, \ldots, n'$ persisting particles is predicted to the next time step according to the single-object Markov transition density using

$$x_{+}^{(j)} \sim f_{+}\left(\cdot | x^{(j)}\right) \tag{4.7}$$

and the predicted set of surviving particles is given by

$$\mathrm{X}_{+,S}^{(i)} = \left\{ x_{+}^{(1)}, \ldots, x_{+}^{(n')} \right\}. \tag{4.8}$$

The set of new born particles $\mathrm{X}_{B}^{(i)}$ is obtained by drawing the number of appearing objects n_B from the cardinality distribution

$$\rho_B(n) = \frac{e^{-\lambda_B} \lambda_B^n}{n!}, \tag{4.9}$$

where λ_B is the expected number of new born objects. Afterwards, the state of each particle is given by sampling from the birth distribution p_B:

$$x_{+}^{(j)} \sim p_B(\cdot) \; \forall \; j = 1, \ldots, n_B. \tag{4.10}$$

Obviously, it is not always possible to sample the predicted particles and the new born particles directly from the distribution. In these cases, the samples have be obtained using importance sampling (see Section 2.3).

4.1.2 Update

In order to perform the measurement update step of the SMC multi-object Bayes filter using the corrector equation

$$\pi(\mathrm{X}) = \frac{g(\mathrm{Z}|\mathrm{X}_{+}) \cdot \pi_{+}(\mathrm{X}_{+})}{\pi(\mathrm{Z})} \tag{4.11}$$

introduced in Section 3.6, the evaluation of the multi-object likelihood function $g(\mathrm{Z}|\mathrm{X}_{+})$ is required. Following Section 3.4, the multi-object likelihood function for scenarios with missed detections and false alarms is given by

$$g(\mathrm{Z}|\mathrm{X}_{+}) = \pi_C(\mathrm{Z}) \pi(\emptyset|\mathrm{X}_{+}) \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_D\left(x_{+}^{(i)}\right) \cdot g\left(z_{\theta(i)}|x_{+}^{(i)}\right)}{\left(1 - p_D(x_{+}^{(i)})\right) \cdot \lambda_c c(z_{\theta(i)})}, \tag{4.12}$$

where

$$\pi(\emptyset|\mathrm{X}_{+}) = \prod_{i=1}^{n} \left(1 - p_D\left(x_{+}^{(i)}\right)\right), \tag{4.13}$$

$$\pi_C(\mathrm{Z}) = e^{-\lambda_c} \prod_{z\in\mathrm{Z}} \lambda_c c(z), \tag{4.14}$$

and λ_c is the expected number of Poisson distributed false alarms which follow the spatial distribution $c(z)$. Further, $p_D(\cdot)$ denotes the state dependent detection probability and $g(\cdot|\cdot)$ is the single-object measurement likelihood. The multi-object likelihood function averages over all possible association hypotheses $\theta : \{1,\ldots,n\} \rightarrow \{0,1,\ldots,m\}$ for n objects and m measurements where "0" represents the missed detection. By using the average, all association hypotheses are taken into account equally which corresponds to the absence of knowledge about the correct association. Further, each summand in (4.12) represents one association hypothesis θ.

An intuitive representation of all association hypotheses is a hypotheses tree [Mah09b; Mas04; MMD10]. Figure 4.1 illustrates the hypotheses tree for an example with two objects and two measurements. Each association hypothesis is represented by a path

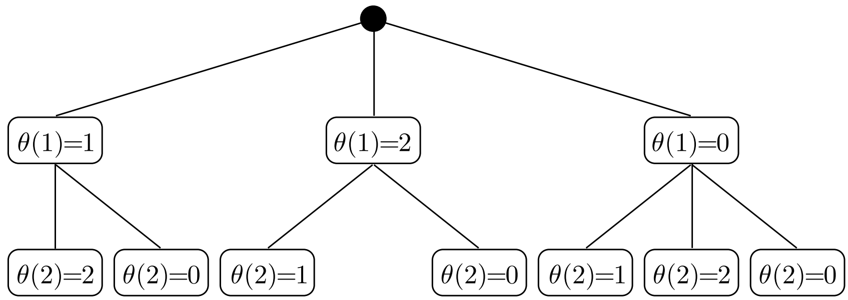


Figure 4.1: Hypotheses tree for two objects and two measurements: a node $\theta(i) = j$ denotes the association of object i to measurement j, where $j = 0$ denotes the missed detection.

from the root of the tree to a leaf. Compared to the trees for the JIPDA algorithm in [Mah09b; MMD10], the following differences are observed: First, the nodes for assigning measurements to the clutter source disappeared, since the factor $\pi_C(\mathrm{Z})$ already assigns all measurements to the clutter source and the track to measurement associations cancel out the clutter terms for the associated measurements. Second, the assignment of a track to non-existence is not required any more, since the non-existence of an object is represented by other multi-object particles.

The association tree enables a straight-forward recursive implementation of the multi-object likelihood (4.12). Each edge of the association tree represents the likelihood of an association $\theta(i)$. One summand in (4.12) is determined by the product of the edge likelihoods from the root of the tree to a leaf. Finally, the likelihoods for all paths are accumulated and multiplied with the factor $\pi_C(\mathrm{Z})\pi(\emptyset|\mathrm{X}_+)$ to obtain the multi-object likelihood $\pi(\mathrm{Z}|\mathrm{X}_+)$.

Similar to the SMC implementation of the single-object Bayes filter, the update step only changes the weight of the multi-object particles without modifying the states. Consequently, the state of an updated multi-object particle is given by

$$\mathrm{X}^{(i)} \triangleq \mathrm{X}_+^{(i)}. \tag{4.15}$$

and its weight is obtained by normalizing the multi-object likelihoods of all ν multi-object particles:

$$w^{(i)} \triangleq \frac{g\left(\mathrm{Z}|\mathrm{X}_+^{(i)}\right)}{\sum\limits_{e=0}^{\nu} \pi\left(\mathrm{Z}|\mathrm{X}_+^{(e)}\right)}. \tag{4.16}$$

Similar to the SMC implementation of the single-object Bayes filter, the SMC multi-object Bayes filter requires the use of resampling approaches since the weights tend to concentrate on a few multi-object particles after a few cycles. However, the SMC multi-object Bayes filter facilitates the application of the well-known resampling approaches introduced in Section 2.3.

4.1.3 Discussion

The computationally intensive part of the filter is the calculation of the multi-object likelihood. The complexity of the association tree for n tracks and m measurements, $m \geq n$, is given by

$$\prod_{i=1}^{n}(m-i+1) \in O(m^n) \tag{4.17}$$

if the nodes for missed detections are neglected. Consequently, the complexity in (4.17) can be considered as a lower bound since the nodes for missed detections significantly increase the number of hypotheses. Due to the exponential growth of the complexity, an evaluation of the multi-object likelihood is only feasible for a small number of measurements and tracks. Similar to JPDA, JIPDA, and MHT, the assumption that a measurement is generated by at most one object affects the computational complexity. The influence of this assumption is illustrated in Figure 4.1. If measurement ”1” is assigned to track ”1” in the first stage ($\theta(1) = 1$), it may not be associated to track ”2” in the second stage (missing node $\theta(2) = 1$). Consequently, the possible associations in a stage of the tree depend on the associations in the previous stages.

Although a multi-object particle in the SMC implementation of the multi-object Bayes filter represents the complete environment, the prediction step described in Section 4.1.1 propagates the state of an object independent of the other objects' states within the multi-object particle. In scenarios with closely spaced targets, neglecting the presence of other objects in the proximity during prediction may lead to physically impossible states. Hence, an integration of dependencies between objects into the prediction step is expected to ensure valid multi-object states and to simplify the data association.

In the following sections, enhancements of the SMC implementation of the multi-object Bayes filter are proposed which address the integration of dependencies between the objects, the reduction of the computational complexity and the incorporation of a state dependent detection probability. Finally, a track extraction scheme is proposed which additionally facilitates the determination of an existence probability for the individual objects.

4.2 Methods to Model the Motion of Extended Objects

In the standard multi-object motion model introduced in Section 3.5, the state transition of all objects in the scene is assumed to be statistically independent. Using this assumption, the motion of an object depends only on its current state and its motion model. However, neglecting the influence of other nearby objects may lead to physically impossible predicted multi-object states in applications with extended objects. Figure 4.2 illustrates three predicted multi-object states of three circular objects with identical diameter. Obviously, the predicted state in the middle depicts a valid multi-object state while the predicted states on the left and the right are invalid due to the collision of two of the objects. Additionally, the two invalid predicted multi-object states increase the ambiguity of the data association in the measurement update due to the small distance between the centers of the colliding objects. Hence, the integration of object interactions in the prediction step is expected to provide a significant performance gain.

In general, the multi-object Markov density facilitates the incorporation of object interactions which is for example used to model a coordinated multi-object motion

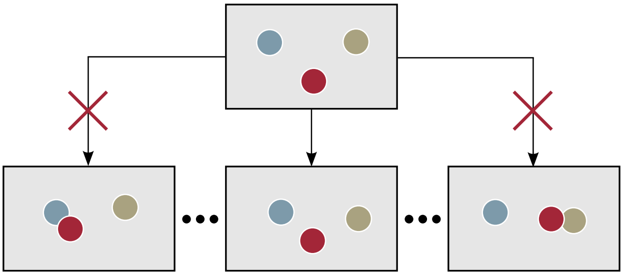
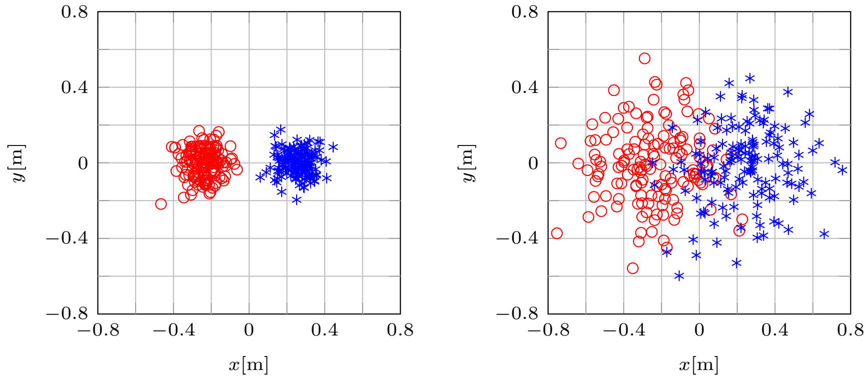


Figure 4.2: Prediction of closely spaced objects: prior multi-object state at the top, possible predicted states at the bottom.

in the virtual leader follower model [Mah07a, pp. 478ff.]. Within the virtual leader follower model, each object retains the same relative position (up to a random variation) to the centroid of a group since all objects are predicted using a joint mean velocity. However, the model does not ensure valid multi-object states due to the additional random variation of the points. Consequently, the incorporation of object interactions in the prediction step requires the definition of transition densities which ensure valid multi-object predictions. In order to prevent a computationally demanding determination of these transition densities, the objects within a multi-object particle are predicted independently and the predicted multi-object particles are validated using a distance constraint.

In order to simplify notations, the object interactions are considered on a two dimensional space and all objects are assumed to be of circular shape with approximately the same diameter. Due to the identical size of the objects, a point target representation within the SMC multi-object Bayes filter is still feasible. However, an extension to objects with arbitrary shape and size by appending the necessary information to the state vectors of the particles is straightforward.

The example in Figure 4.3 is considered to exemplify the benefit of incorporating object interactions and to demonstrate the differences between the proposed approaches: Two objects with a radius of $r_p = 0.2$ m are located at $(x, y) = (-0.25, 0)$ m and $(x, y) = (0.25, 0)$ m. The initial uncertainty is Gaussian distributed with standard deviation $\sigma_x = \sigma_y = 0.07$ m and the multi-object density is represented by $\nu = 150$ multi-object particles. Each multi-object particle contains a state vector for both of the objects and the corresponding prior distribution is illustrated in Figure 4.3a. Within the multi-object Markov density, each object is assumed to follow a random walk model with a Gaussian distributed uncertainty of $\sigma_x = \sigma_y = 0.2$ and the survival probability is $p_S = 1$. The prediction using the standard multi-object motion model results in the multi-object particles illustrated in Figure 4.3b. Obviously, the particles



(a) Prior multi-object particles. (b) Independent prediction.

Figure 4.3: Independent prediction of the multi-object particles of two closely spaced objects.

of the two objects mingle around the origin of the coordinate system. Further, the independent prediction of the particles within a multi-object particle obtains invalid predicted states since the distance between the centers of the objects drops below the physically impossible distance of $d = 2r_p = 0.4$ m in some of the multi-object particles.

In the following, two methods for the incorporation of the constraints into the multi-object Bayes filter are presented. Beside thinning using hard-core point processes and a set based weight adaption, a possible integration of destinations in the prediction is discussed briefly.

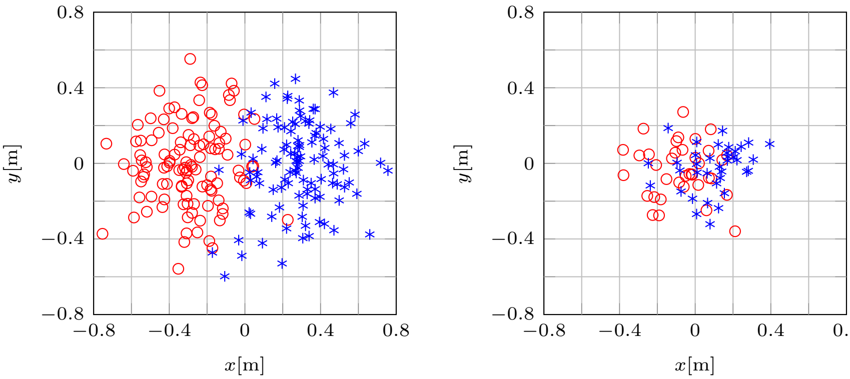
4.2.1 Thinning Using Hard-Core Point Processes

In stochastic geometry, a hard-core point process [DV03; Mat60; Mat86; SKM95; Str10] is a point process which ensures a minimum distance between individual points. Obviously, the points of a hard-core point process are statistically dependent since the state of one point restricts the possible states for all other points. The prior multi-object state at the top of Figure 4.2 as well as the predicted multi-object state in the middle of the bottom row are realizations of a hard-core point process since the extended objects do not overlap. However, the other two predicted multi-object states are not a realization of a hard-core point process. In [Mat60; Mat86], Matérn proposes two thinning procedures which remove points from a homogeneous Poisson point process (PPP) until the process is a realization of a hard-core point process.

Matérn Method I

The Matérn method I determines all pairs of points in a realization of a point process which do not fulfill the distance constraint. After the distances for all pairs of points are evaluated, points with a minimum distance of $d \geq 2r_p$ to all other points are retained while points with a distance of $d < 2r_p$ to any other point are deleted. Since the Matérn thinning is applied to each multi-object particle separately, its complexity is quadratic in the number of objects per multi-object particle.

Using the Matérn I thinning to ensure a minimum distance of $d = 0.4$ m between the particles of a multi-object particle, the multi-object particles of Figure 4.4a are obtained for the predicted multi-object density of Figure 4.3b. While the particles of the two objects are mingled in Figure 4.3b around the origin of the coordinate system, the particles of the two objects are well separated in Figure 4.4a after removing a large number of particles (Figure 4.4b).



(a) Retained points. (b) Deleted points.

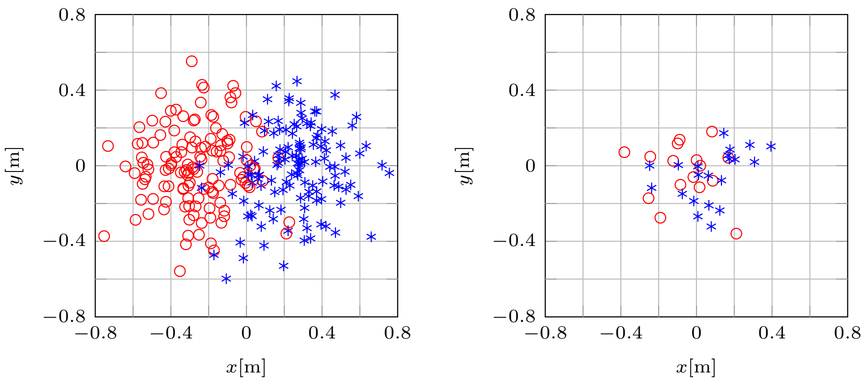
Figure 4.4: Result of the application of Matérn I thinning to the predicted multi-object particles depicted by Figure 4.3b.

Matérn Method II

The Matérn method II is intended to obtain a hard-core point process with a higher density compared to Matérn I. While Matérn I removes both points of a pair which does not satisfy the required minimum distance, Matérn II keeps one of the points to increase the density of the resulting hard-core process.

Thus, Matérn II requires to add an additional mark $u^{(i)}$ to each of the points $x^{(i)}$ of the point process. In general, the marks are random numbers of a uniform distribution [SKM95; Str10]. A point $x^{(i)}$ is retained in the hard-core point process, if it does not overlap with the sphere of a point $x^{(j)}$ with a smaller mark $u^{(j)} < u^{(i)}$. Similar to Matérn I, each point is determined to be deleted or to be retained before the actual thinning is executed.

Applying the Matérn II thinning to the predicted multi-object density of Figure 4.3b results in the hard-core processes illustrated by Figure 4.5a. Compared to the result of Matérn I thinning in Figure 4.4a, the particles of the two objects are not equally well separated. However, the number of thinned particles (Figure 4.5b) is significantly smaller than the number of thinned particles for Matérn I.



(a) Retained points.

(b) Deleted points.

Figure 4.5: Result of the application of Matérn II thinning to the predicted multi-object particles depicted by Figure 4.3b.

Weight Adaption

A thinning of the predicted multi-object particles using Matérn I or II implies a correlation of the spatial proximity of the objects and the predicted mean cardinality, since the thinning effectively reduces the cardinality of the multi-object particles. However, a correlation between spatial proximity and persistence of an object is only appropriate in a small number of applications (e.g. a person getting on a vehicle). Hence, a penalization of the thinned multi-object particles using the survival probability $p_S(x^{(j)})$ is appropriate. Since the thinning of object $x^{(j)}$ contradicts the survival of object $x^{(j)}$ in the prediction step of the SMC multi-object Bayes filter, the weight of a thinned multi-object particle is given by

$$\widetilde{w}_{+}^{(i)} = w_{+}^{(i)} \cdot \prod_{x \in \mathrm{D}^i} \frac{1 - p_S(x)}{p_S(x)}, \tag{4.18}$$

where the set D^i denotes the set of thinned particles of the multi-object particle $\mathrm{X}^{(i)}$. Since $p_S(x) \gg 0.5$ in most applications, (4.18) significantly reduces the weight of thinned multi-object particles.

4.2.2 Social Force Model

In the early 1970's, Henderson [Hen71] investigated the motion of pedestrians and observed significant correlations with fluid dynamics. Using fluid dynamics corresponds to a macroscopic formulation, i.e. it delivers quantities like e.g. the mean velocity of a group and its density. An application of fluid dynamics to traffic simulations is discussed in [GH75]. In [HM95], Helbing and Molnar proposed a microscopic model for pedestrian dynamics, the *Social Force Model*, which is based on the approach of Lewin [Lew51] and models behavioral changes using social forces. Within this model, it is assumed that the behavior of a human changes based on the current environment and its personal aims. The influences of the environment as well as the personal aims are modeled using repellent and attractive forces. The model additionally assumes, that each object wants to reach a specific destination on the shortest possible path, which is given by a straight line between the current position and the destination or by a polygon which avoids obstacles. Additionally, the model captures that an object generally prefers to move with a desired velocity which results in a force vector modeling the necessary acceleration or deceleration to reach this speed. The forces of the social force model are transferable to other applications by an adaptation of the occurring forces. An application to the automotive environment perception may for example use the forces to keep the vehicles on their lanes and to keep a minimum distance to other vehicles.

Set Based Weight Adaption

An application of the entire social force model to multi-object tracking is not possible in general, since quantities like the destination or the desired velocity of an object are usually not available. Thus, the proposed set based weight adaption uses only the distances between the particles of a multi-object and the desired distances to other objects.

The repellent forces due to other objects are commonly modeled using exponential functions [PESG09]. Consequently, the likelihood of a multi-object particle comprising two particles s and t is given by

$$\Lambda_d(x^{(s)}, x^{(t)}) = \begin{cases} 0 & \text{if } d(x^{(s)}, x^{(t)}) < 2r_p \\ 1 - \exp\left(-\frac{(d(x^{(s)},x^{(t)})-2r_p)^2}{2\sigma_d^2}\right) & \text{otherwise} \end{cases} \tag{4.19}$$

where r_p denotes the radius of the circular objects and $d(x^{(s)}, x^{(t)})$ denotes their Euclidean distance. While the thinning methods in Section 4.2.1 use a hard distance threshold, the likelihood (4.19) uses the exponential function with parameter σ_d to obtain a smooth transition between impossible object states and the preferred inter-object distance. Subsequently, the likelihood Λ_d is used to adapt the weight of the multi-object particle $\mathrm{X}^{(i)}$ using

$$\widetilde{w}_+^{(i)} = \min_{s=1,\dots,|\mathrm{X}^{(i)}|} \left(\min_{t=1,\dots,|\mathrm{X}^{(i)}|, t\neq s} \left(\Lambda_d(x^{(s)}, x^{(t)}) \right) \right) \cdot w_+^{(i)} \ , \tag{4.20}$$

i.e. the minimum likelihood of all possible pairs (s,t) is used to modify the weight. Thus, the weight of a multi-object particle with a minimum distance $d(x^{(s)}, x^{(t)}) < 2r_p$ is set to zero, while the weight is unchanged in case of $d(x^{(s)}, x^{(t)}) \gg 2r_p$.

Obviously, the set based weight adaption assigns a weight of $\widetilde{w}_+^{(i)} = 0$ to all multi-object particles which are deleted by the Matérn I thinning (see Figure 4.4b). However, the weight of retained multi-object particles with a minimum distance $d < 2r_p + 3\sigma_d$ between two of the points is reduced while $\widetilde{w}_+^{(i)} \approx w_+^{(i)}$ for $d \geq 2r_p + 3\sigma_d$. Using the parameter $\sigma_d = 0.05$ m, Figure 4.6 shows all retained multi-object particles whose weight is adapted to $0 < \widetilde{w}_+^{(i)} < 0.95 \cdot w_+^{(i)}$.

While the results of set based weight adaption and Matérn I thinning are identical (except for the reduced weight of some retained multi-object particles), significant differences arise in case of multi-object particles with larger cardinality: On the one hand, the set based weight adaption completely removes invalid multi-object particles by setting the weight to zero. On the other hand, the Matérn I thinning just removes the colliding particles within the corresponding multi-object particles (and reduces the weight if the weight adaption is applied).

Integration of Destinations

As mentioned above, the destination of an object is generally not known in multi-object tracking applications. However, in some applications an imprecise knowledge about

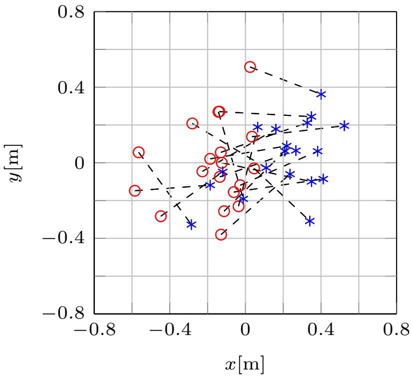


Figure 4.6: Retained particles of the set based weight adaption with an adapted weight of $0 < \widetilde{w}_{+}^{(i)} < 0.95 \cdot w_{+}^{(i)}$. The dashed lines connect the particles of a multi-object particle to illustrate the distance between the two states.

possible destinations of the objects is available and may be used to improve the state prediction in case of low measurement rates or occlusions. Although the integration of destinations is not investigated in this thesis, an example for future research is discussed in the following. Figure 4.7 depicts an example for the incorporation of destinations in the context of automotive environment perception. Using the context information given by the road network, there are three possible situations: The blue car

- turns right after the beige car passed by,
- waits at the stop line until both cars passed by,
- turns left after the beige car passed by.

Obviously, a critical situation only arises if the driver of the blue car fails to notice the red car and starts a left turn immediately behind the beige car. In this example, an adapted social force model which incorporates the lane constraints and the multiple destinations facilitates an improved state prediction for the blue car during the occlusion. An improved prediction is expected to facilitate an earlier intervention of a driver assistance system in case of the critical situation compared to a system using standard motion models.

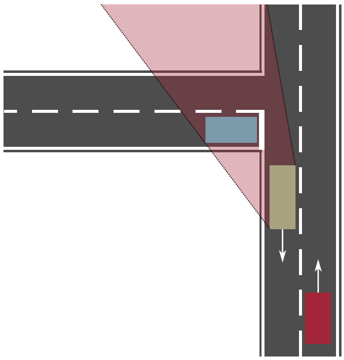


Figure 4.7: Exemplary scenario with three cars (red, blue, beige) where the integration of destinations is promising. Using the road network information, the blue car is expected to turn left, turn right, or to wait at the stop line. Obviously, the left turn may result in a critical situation for the red and blue car.

4.3 Approximation of the Multi-Object Likelihood

The evaluation of the multi-object likelihood function is only possible for a small number of tracks and measurements due to the combinatorial complexity and the limited computation time. As mentioned in Section 4.1.3, the assumption that a measurement is generated by at most one object is responsible for the exponential complexity. By neglecting this assumption, the measurement to track associations at each stage are independent of the associations at previous stages. Figure 4.8 illustrates the corresponding modifications of the association tree. Since a measurement may now be associated to more than one track, two additional nodes are added to the tree (marked by dashed lines).

Using this approximation, the calculation of the multi-object likelihood (4.12) simplifies to

$$\widetilde{g}(\mathrm{Z}|\mathrm{X}_+) = \pi_C(\mathrm{Z})\pi(\emptyset|\mathrm{X}_+) \cdot \prod_{i=1}^{n} \left(1 + \sum_{j=1}^{m} \frac{p_D\left(x_+^{(i)}\right) \cdot g\left(z_j|x_+^{(i)}\right)}{\left(1 - p_D(x_+^{(i)})\right) \lambda_c c(z_j)} \right), \tag{4.21}$$

where $\pi_C(\mathrm{Z})$ and $\pi(\emptyset|\mathrm{X}_+)$ are given by (4.14) and (4.13). Equation (4.21) simply multiplies the multi-object likelihoods for all particles i, where the value one in each factor represents the possibility of a missed detection and the sum accumulates the single-object likelihoods of all measurements for the current particle i.

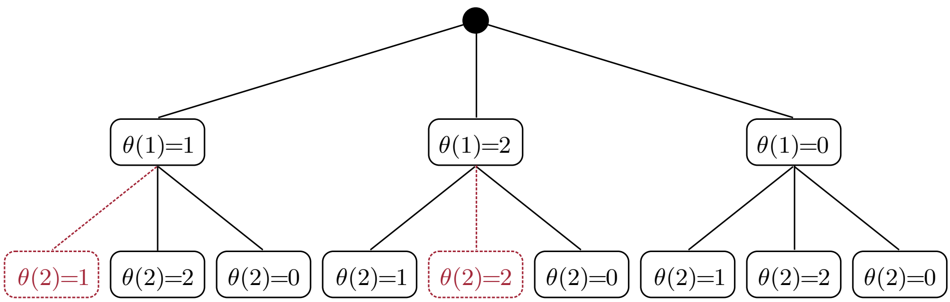


Figure 4.8: Approximation of the multi-object likelihood: Two additional nodes (marked by red dashed lines) are added to the tree of Figure 4.1 to model multiple associations.

Due to the additional nodes, the complexity of the approximate multi-object likelihood (4.21) is linear in the number of objects as well as in the number of measurements ($O(mn)$) and facilitates an application of the SMC multi-object Bayes filter to scenarios with a higher number of tracks and measurements. Further, (4.21) allows for a straightforward implementation using two for-loops.

4.3.1 Approximation Error

The approximation error of (4.21) depends on the uncertainty of the track to measurement associations. Considering again the example depicted by Figure 4.8, it is apparent that measurements with a high spatial likelihood for more than one object lead to a large approximation error, e.g. the association hypothesis with $\theta(1) = 1$ and $\theta(2) = 1$. However, the approximation error is negligible if each measurement has a high spatial likelihood for at most one of the objects. In this case, the proposed approximation handles the uncertainty due to several feasible measurements for each of the objects equally well. Further, all additional hypotheses created by the approximation have negligible weight.

If an application uses one of the models presented in Section 4.2 to incorporate the dependencies between the motion of the extended objects, a minimum distance between all objects of a multi-object particle is ensured. In this case, the approximation directly depends on the ratio of the measurement noise to the object size. Figure 4.9 illustrates two objects of radius r_p whose centers exactly keep the minimum

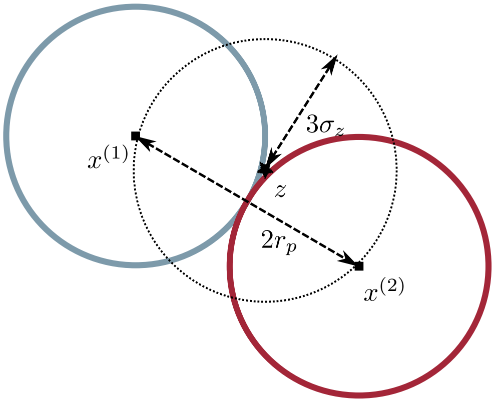


Figure 4.9: Dependence of approximation error on the measurement noise to object size ratio: two circular objects $x^{(1)}$ and $x^{(2)}$ with radius r_p and a Gaussian distributed measurement z with standard deviation σ_z, where $3\sigma_z = r_p$.

distance. Additionally, a Gaussian distributed measurement z with measurement noise covariance $R = \sigma_z^2 I_2$ is shown, where I_2 is the two dimensional identity matrix. The standard deviation of the measurement noise is $3\sigma_z = r_p$ which corresponds to the limiting case that a measurement has a very small likelihood for both of the objects. If $3\sigma_z < r_p$, at most one object may be located within the $3\sigma_z$ bound of a measurement and the approximation error is negligible. However, in applications with a measurement noise of $3\sigma_z > r_p$ the approximation error becomes apparent since the centers of more than one object may be situated within the $3\sigma_z$ bound of a measurement.

4.3.2 Marginalization

As shown above, the approximation error is negligible if the object size to measurement noise ratio is large enough. However, this assumption is obviously not met for all applications and the approximate multi-object likelihood function (4.21) may lead to a bias in the cardinality estimate if a measurement has a high spatial likelihood for more than one particle within a multi-object particle.

In [ML08], the linear multi-target integrated probabilistic data association (LM-IPDA) algorithm is proposed as an computationally efficient approximation of the JIPDA algorithm. Similar to the approximation of the multi-object likelihood in (4.21), the LM-IPDA reduces the complexity by permitting the association of a measurement to more than one track. In [ML08], other objects in the proximity of a track are used to adapt the clutter distribution, i.e. the other objects are considered as scatterers.

The promising simulation results of the LM-IPDA encourage the application of a similar approach to reduce the approximation error of (4.21). Hence, the probability for assigning a measurement z_j to an object $x^{(i)}$ is obtained using marginalization:

$$p(x^{(i)}|z_j) = \frac{g(z_j|x^{(i)})}{\sum_{i=1}^{n(i)} g(z_j|x^{(i)})}, \tag{4.22}$$

where $n(i) = |\mathrm{X}^{(i)}|$ and

$$\sum_{i=1}^{n(i)} p(x^{(i)}|z_j) = 1. \tag{4.23}$$

The approximation error of (4.21) is reduced by multiplying each track to measurement association with the marginal probability $p(x^{(i)}|z_j)$, i.e.

$$\widetilde{g}^m(\mathrm{Z}|\mathrm{X}_+) = \pi_C(\mathrm{Z})\pi(\emptyset|\mathrm{X}_+) \cdot \prod_{i=1}^{n} \left(1 + \sum_{j=1}^{m} \frac{p_D\left(x_+^{(i)}\right) \cdot g\left(z_j|x_+^{(i)}\right) \cdot p(x^{(i)}|z_j)}{\left(1 - p_D(x_+^{(i)})\right) \lambda_c c(z_j)} \right). \tag{4.24}$$

Compared to the approximate multi-object likelihood (4.21), which explicitly allows to assign a measurement to more than one track, the marginalized multi-object likelihood (4.24) reduces the likelihood of a track to measurement association in ambiguous situations where a measurement has a high likelihood for more than one track. Thus, the marginalized likelihood reduces the approximation error in applications where the minimum distance between the objects does not ensure a negligible approximation error. For $3\sigma_z \ll r_p$, the marginalized multi-object likelihood is identical to the approximate multi-object likelihood, since there is only one track which has a significant likelihood for a measurement z_j.

4.3.3 Real-Time Implementation

The SMC multi-object Bayes filter allows for a parallel implementation of the prediction and the update steps since the calculations for the individual multi-object particles are independent of each other. In contrast, a parallelization of the resampling step is more challenging since it typically requires the weights of all particles[1]. Hence, a graphics processing unit (GPU) is well suited for an efficient implementation of the prediction and update steps of the filter.

In the author's workshop paper [RDH11], the first real-time capable SMC implementation of the multi-object Bayes filter using a GPU is proposed. Within the CUDA programming language [Nvi14], the usage of recursive functions is restricted. Thus, a GPU implementation of the exact multi-object likelihood function (4.12) is challenging since the enumeration of all possible associations is typically realized using recursive functions. In contrast, the proposed approximations (4.21) and (4.24) of the multi-object likelihood function facilitate a GPU implementation using two nested for loops. The resampling step is realized as follows: First, the cumulative sum of the weight vector is copied from the GPU to the CPU. Based on the weights, the resampling is performed on the CPU and an index vector is generated. Finally, the index vector is transferred to the GPU again and the multi-object particles are copied to the positions determined by the index vector. In [RWW$^+$13], the GPU implementation is shown to be able to process the prediction and update step (including the track extraction) for a scenario with up to seven objects in less than 40 ms on a Tesla C2075 GPU (using $N = 25000$ multi-object particles). Since the sensors operate with 12.5 Hz, the proposed implementation is real-time capable.

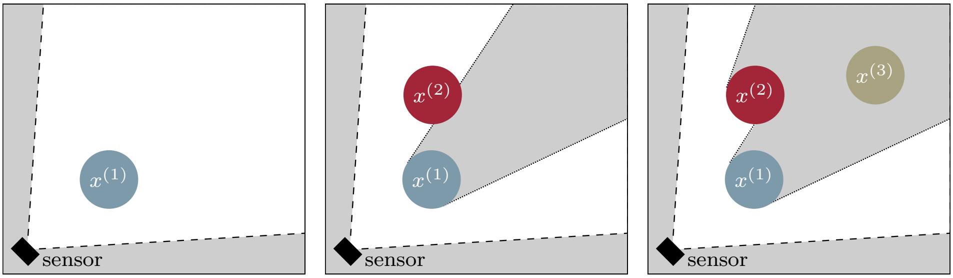
4.4 State Dependent Detection Probability

In most multi-object tracking algorithms, the detection probability is considered to be state independent within the field of view (FOV) of the sensor. However, the detection probability is often affected by static obstacles in the FOV, e.g. due to buildings or bushes in automotive environment perception or tunnels in ground moving target indicator (GMTI) applications [UK06]. In such situations, it is common to represent the state dependent detection probability using a static map [HM11; UK06] or to estimate the detection probability only based on the occupancy grid mapping approach [TBF05]. Further, the detection of an object is also obscured by the clutter notch and the minimum detectable velocity [KK01; UEW07; UK06] in GMTI applications. Consequently, applications using state dependent detection probabilities enable the use of "negative" sensor information [Koc04], e.g. the missed detection of an object in the proximity of an occluded area incorporates the information that the object is probably located within the occluded area.

The incorporation of extended, tracked objects into the determination of a state

[1] An approach for a parallel implementation of the resampling step is e.g. given by [HKG10]

dependent detection probability is far more complex, since a multi-object state includes the uncertainty in the number of objects and their individual states. However, the representation of the entire environment using multi-object particles facilitates a straightforward determination of a state dependent detection probability since the uncertainty about the number of objects and their state is represented by other multi-object particles.



(a) (b) (c)

Figure 4.10: State dependent detection probability of three objects within a multi-object particle. Grey areas indicate occluded parts of the measurement space. Objects are inserted by means of rising distance (a)-(c) and the area behind an object is indicated as occluded when the next object is inserted.

Figure 4.10 illustrates a procedure to obtain an object individual detection probability in case of a sensor with a radial measurement principle. First, it is necessary to sort the objects by means of rising distance to the sensor. Afterwards, the closest object is inserted into the map for the state dependent detection probability (Figure 4.10a). Obviously, the detection probability of the closest object is not affected by other objects. The detection probability of object $x^{(2)}$ in Figure 4.10b is affected by the area which is occluded due to the existence of object $x^{(1)}$. In Figure 4.10b, the detection probability depends on the detection principle of the sensor. On the one hand, $x^{(2)}$ has a high detection probability if the sensor data preprocessing delivers detections for partially occluded objects. On the other hand, a low detection probability is obtained if the detection algorithm is not able to detect partially occluded objects. Finally, object $x^{(3)}$ has a vanishing detection probability since it is located in the occluded area behind $x^{(1)}$. A state dependent detection probability for the current situation may obtained by averaging over the detection probability maps of all multi-object particles.

4.5 Adaptive Birth Density

In applications with diffuse birth densities, a huge amount of multi-object particles is required for an adequate representation of the birth density. In [RCV10; RCVV12], an adaptive birth intensity for the SMC implementation of the PHD and CPHD filter is proposed which significantly reduces the required number of particles. The adaptive birth intensity concentrates around the measurements with a small spatial likelihood for the persisting particles within the filter.

Since each multi-object particle of the SMC multi-object Bayes filter represents a possible multi-object state, an adaptive birth density has to be determined for each of the multi-object particles. Using the exact multi-object likelihood function (4.12), the contribution of a measurement to an updated multi-object particle at time k is given by the association probability

$$r_{U,k}(z_j|\mathrm{Z},\mathrm{X}_+) = \frac{\sum_{\theta} 1_{\theta}(j) \prod_{i:\theta(i)>0} \frac{p_D\left(x_+^{(i)}\right) \cdot g\left(z_{\theta(i)}|x_+^{(i)}\right)}{\left(1-p_D(x_+^{(i)})\right) \cdot \lambda_c c(z_{\theta(i)})}}{\sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_D\left(x_+^{(i)}\right) \cdot g\left(z_{\theta(i)}|x_+^{(i)}\right)}{\left(1-p_D(x_+^{(i)})\right) \cdot \lambda_c c(z_{\theta(i)})}}. \tag{4.25}$$

While the nominator contains only hypotheses θ which comprise an association of measurement z_j to any of the objects, the denominator contains all association hypotheses. For the approximate multi-object likelihood function (4.21), the association probability of measurement z_j is given by

$$r_{U,k}(z_j|\mathrm{Z},\mathrm{X}_+) = \frac{\pi_C(\mathrm{Z})\pi(\emptyset|\mathrm{X}_+)\Lambda_U(z_j|\mathrm{Z},\mathrm{X}_+)}{\widetilde{g}(\mathrm{Z}|\mathrm{X}_+)} \tag{4.26}$$

where the nominator is obtained by the recursion

$$\begin{aligned}\Lambda_U(z_j|\mathrm{Z},\mathrm{X}_+) = \Lambda_U(z_j|x_+^{(1)}) \prod_{i=2}^{n} \left(1 + \sum_{l=1}^{m} \Lambda_U(z_l|x^{(i)})\right) + \\ \left(1 + \sum_{z\in \mathrm{Z}\setminus\{z_j\}} \Lambda_U(z_l|x_+^{(i)})\right)\Lambda_U(z_j|\mathrm{Z},\mathrm{X}_+\setminus\{x_+^{(1)}\})\end{aligned} \tag{4.27}$$

with

$$\Lambda_U(z|x_+) = \frac{p_D\left(x_+\right) \cdot g\left(z|x_+\right)}{\left(1-p_D(x_+)\right)\lambda_c c(z)}. \tag{4.28}$$

Obviously, the calculation of (4.26) is combinatorial. Hence, the association probability is approximated by

$$r_{U,k}(z_j|\mathrm{Z},\mathrm{X}_+) \approx \frac{1}{g_{max}} \cdot \max_{x\in\mathrm{X}_+} g(z_j|x), \tag{4.29}$$

where g_{max} denotes the maximum value of the measurement likelihood g which is usually obtained for $z_+ = z$ in case of Gaussian distributed measurements.

Similar to the birth process in Section 4.1.1, the number of new-born particles n_B per multi-object particle at time $k+1$ is drawn from a Poisson distribution with a mean value of λ_B. On the one hand, a measurement which is not associated to any of the existing objects during the update at time k is likely to correspond to a new born object or to the clutter process. On the other hand, a measurement which is associated to an object in all hypotheses with a significant weight is not likely to belong to a new born object. Hence, the probability that measurement z_j is used to create one of the n_B new born objects at time $k+1$ is given by

$$r_{B,k+1}(z_j) = \frac{1 - r_{U,k}(z_j)}{\sum\limits_{\xi\in\mathrm{Z}_k} 1 - r_{U,k}(\xi)}. \tag{4.30}$$

Using the probabilities of (4.30), a sample of a uniform distribution may be utilized to determine which measurement should be used to sample the state of a new born object. The state of the new born object is finally drawn from a birth distribution $p_B(x|z)$ which depends on the application, the measurement z, and the measurement noise of the sensor. Obviously, for more than one new born object per multi-object particle, the used measurements have to be distinct.

4.6 Track Extraction and Existence Estimation

Since the multi-object Bayes filter avoids an explicit data association by averaging over all possible associations, the filter delivers an estimate for the multi-object posterior but does not produce object tracks. In [Mah07a, pp. 497ff.], Mahler proposes two Bayes-optimal multi-object state estimators, the marginal multi-object estimator and the joint multi-object estimator, to determine the number of objects and the individual object states. While the joint multi-object estimator jointly estimates the number of objects and their states, the marginal multi-object estimator estimates the number of objects $\hat{N}$ using the cardinality distribution and subsequently determines the $\hat{N}$ states. In general, the joint multi-object estimator outperforms the marginal multi-object estimator since the estimated mean cardinality may differ from the number of objects during object birth or in case of track existence probabilities which are significantly smaller than one.

The track extraction scheme proposed in this section can be considered as an computationally efficient approximation of the joint multi-object estimator. The algorithm consists of two parts: First, an adapted k-means clustering is applied which additionally estimates the number of cluster centroids. Afterwards, the clustering result is validated to obtain unique track IDs within each multi-object particle. The unique track IDs additionally facilitate the determination of a track existence probability.

4.6.1 Particle Labeling

In order to simplify the track extraction, a label or track ID ℓ is appended to the individual object states ([Mah07a, pp. 505–508], [MVSB06]). Using the notation of the class of labeled RFS (see Section 3.3), a labeled multi-object particle is denoted by

$$\mathbf{X}^{(i)} = \left\{ \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \right\} \tag{4.31}$$

where

$$\boldsymbol{x}^{(i)} = \left\{ x^{(i)}, \ell^{(i)} \right\}. \tag{4.32}$$

The appended label information is neglected in the prediction and update steps of the SMC multi-object Bayes filter to prevent the impact of incorrect clustering results on the performance of the filter. However, attaching the label information to each of the particles significantly simplifies the extraction of individual tracks and their trajectories. The labels allow for an accurate initialization of the clustering algorithm and ensures that the same label is attached to an object in successive time steps. In order to simplify the differentiation between persisting and new born particles, the label $\ell_B = -1$ is assigned to each new born particle. Similar to the class of labeled RFSs, the labels within a multi-object particle $\mathbf{X}^{(i)}$ are required to be distinct.

4.6.2 Track Existence Probability

The representation of the multi-object state using labeled multi-object particles $\mathbf{X}^{(i)}$ allows for the calculation of an object individual existence probability $r^{(\ell)}$ if the labels within each multi-object particle are distinct. Similar to the existence probability of an object in a δ-GLMB RFS, the existence probability of track ℓ is given by

$$r^{(\ell)} = \sum_{i=1}^{\nu} w^{(i)} 1_{\mathcal{L}(\mathbf{X}^{(i)})}(\ell). \tag{4.33}$$

where $\mathcal{L}(\mathbf{X}^{(i)})$ is defined by (3.21) and simply returns the set of labels of the multi-object particle. While the existence probability (3.35) of a track in a δ-GLMB RFS is obtained by accumulating the weights of all hypotheses (I, ξ) with $\ell \in I$, the existence probability (4.33) accumulates the weight of all multi-object particles ν which contain a particle with label ℓ.

4.6.3 k-Means Clustering

The clustering of the particles is performed using a modified version of the k-means clustering [Bis06]. Instead of a random initialization of the cluster centroids, the proposed algorithm uses the clustering result of the previous time step for initialization and automatically adjusts the number of clusters. Algorithm 4.1 illustrates a complete cycle of the proposed clustering algorithm.

The clustering algorithm uses an internal member variable $k = |\mathbb{L}|$ to store the previous number of clusters where $|\mathbb{L}|$ denotes the number of track labels. If $k = 0$, the first centroid is initialized using a randomly chosen particle of a non-empty multi-object particle. If the clustering algorithm is already initialized and the previous scene was not empty ($k > 0$), the centroids of the clusters are initialized using the clustering result of the previous time step. Hence, the updated centroid of a cluster $l \in \mathbb{L}$ is given by the mean value of all particles with label l:

$$\hat{x}(l) = \frac{1}{\nu(l)} \sum_{i=1}^{\nu} \sum_{j=1}^{|\mathbf{X}^{(i)}|} \delta_l(\ell^{(i,j)}) \cdot x^{(i,j)} \tag{4.34}$$

where

$$\nu(l) = \sum_{i=1}^{\nu} 1_{\mathcal{L}(\mathbf{X}^{(i)})}(l). \tag{4.35}$$

In order to handle disappearing objects, all clusters l with $\nu(l) = 0$ are removed using the *removeEmptyClusters()* method in Algorithm 4.1.

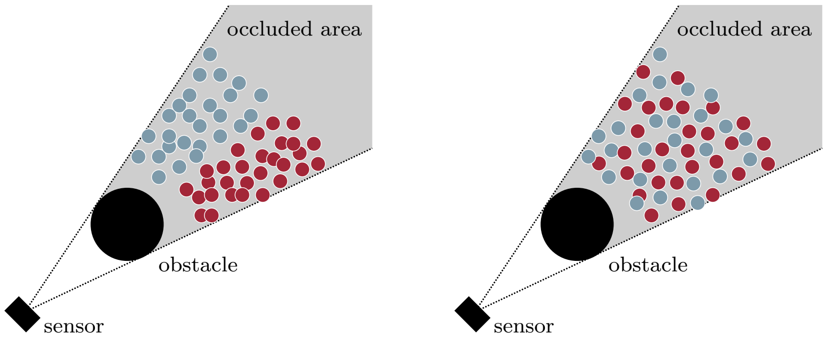
After initialization, the k-means algorithm is applied to update the k centroids (lines 9–14). First, all particles are assigned to their respective centroid using the *findAssignment()* method which performs a nearest neighbor association based on the

Algorithm 4.1 `ClusterParticles()`

```
if k = 0 then
    addCentroid(x)
else
    adjustCentroids()
    removeEmptyClusters()
end if
while finished = 0 do
    finished = 1
    while converged = 0 do
        converged = findAssignment()
        if converged = 0 then
            adjustCentroids()
        end if
    end while
    [d_max, x] = getMaxDistanceToCentroid()
    if d_max > d_split then
        addCentroid(x)
        finished = 0
    end if
    if minCentroidDist > getMinCentroidDistance() then
        mergeClusters()
        finished = 0
    end if
end while
```

Euclidean distance. If the labels of all particles are unchanged by *findAssignment()*, the k-means clustering converges. Otherwise, the new labels of the particles are exploited to calculate the updated state of the centroids using (4.34).

In the standard implementation of the k-means algorithms, the *findAssignment()* method is based on the Euclidean distance between the current centroids and the particles. However, an incorporation of the state dependent detection probability is required to obtain realistic clustering results for occluded objects. Consider the following scenario for illustration: Two objects enter an occluded area, one from the left side, the other one from the right side. Figure 4.11a shows the standard clustering result for the two objects after several time steps. Since the clustering is initialized at every cycle with the result of the previous time step, the particles are still separated by a hard boundary. If the size of the occluded area is large enough to allow the objects to pass each other, a clustering result like the one in Figure 4.11b meets one's expectations since it is not clear if the objects switched their position in the mean time or not. The result in Figure 4.11b is easily obtained by retaining the labels of occluded particles within the *findAssignment()* method.



(a) Standard clustering using Euclidean distance.

(b) No update of labels for particles located in an occluded area.

Figure 4.11: Example for the clustering result several measurement cycles after the two objects entered the occluded area: clustering with (b) and without (a) using the information about the detection probability.

Obviously, the k-means algorithm assigns all new born particles to any of the already existing clusters. This behavior is desired, since a new born particle in one of the multi-object particles may correspond to an already existing cluster. Assuming that the new born particle does not belong to any of the existing clusters, an additional procedure is required which initializes a new centroid for this particle. Hence, the *getMaxDistanceToCentroid()* method is used to determine the maximum distance of any new born particle to the assigned centroid. If the distance exceeds a given threshold d_{max}, an additional centroid is initialized using the state vector x of the new born particle. Obviously, the threshold d_{max} depends on the application and especially on the measurement noise. In case of Gaussian measurements, it is common to use the standard deviation $\sigma_{\mathbf{z}}$ of the measurements to choose the threshold, e.g. $d_{max} = 3\sigma_{\mathbf{z}}$.

In Sections 4.2 and 4.3, all objects are assumed to have a circular shape with approximately the same diameter. Thus, the object extension limits the minimum distance between any of the centroids. If the distance between two centroids is smaller than a minimum distance *minCentroidDist*, the respective clusters are merged using *mergeClusters()*.

Obviously, if a new cluster is added or two existing clusters are merged, the k-means algorithm as well as the subsequent adding or merging of clusters have to be repeated until none of them changes the clustering result any more. In order to prevent endless loops, it is recommended to ensure that a maximum number of iterations is not exceeded.

4.6.4 Set Based Validation of the Clustering Result

The k-means clustering of the previous section assigns a label to each of the particles but does not ensure unique labels within a multi-object particle. The set based validation of the k-means clustering results is performed using Algorithm 4.2.

Algorithm 4.2 `SetBasedClusterValidation`()

$maxDimension$ = getDimensionOfLargestSet()
if $k < maxDimension$ **then**
    addCentroidsSet()
    while $converged = 0$ **do**
        $converged$ = findAssignment()
        if $converged = 0$ **then**
            adjustCentroids()
        end if
    end while
end if
validateClusterResults()
adjustCentroids()

Obviously, the estimated number of clusters k has to be at least as large as the dimension of the largest multi-object particle. If k is smaller, the largest multi-object particle is used to initialize additional centroids and the clustering result is obtained by an additional run of the k-means algorithm.

After ensuring the required number of clusters, the labels within each multi-object particle are verified using the *validateClusterResults()* method in order to detect duplicate assignments of a label. If the labels within a multi-object particle are not unique, a global nearest neighbor algorithm is applied to find the optimal assignment of the particles to the existing centroids. Since *validateClusterResults()* possibly changes the labels of the particles, the position of the centroids has to be updated again to achieve the final clustering result.

4.7 Discussion

The proposed approximation of the multi-object likelihood function facilitates a real-time implementation of the SMC multi-object Bayes filter. Further, the set representation of the objects within the multi-object Bayes filter is used to enable the incorporation of dependencies between the objects in the prediction step which is not possible in the standard multi-object tracking algorithms introduced in Section 2.4. Using the distance constraints of the thinning and the set based weight adaption, an approximation of the obtained error, based on the ratio of the measurement noise to the object size, is possible. The set based track extraction algorithm additionally allows for the determination of an object individual existence probability which is desired in a large amount of applications.

However, the representation of n objects within a multi-object particle requires to draw samples from a space with dimension $n \cdot |x|$, where $|x|$ is the dimension of the state vector of a single object. Hence, the maximum number of objects within the SMC multi-object Bayes filter is limited due to the required number of samples for an adequate representation of the multi-object posterior.

The application of an individual SMC multi-object Bayes filter to each group of objects generally provides the possibility to track a higher number of objects. Since the group compositions are time varying, an additional approach for splitting and merging groups is required. The labeled multi-Bernoulli (LMB) filter proposed in the next chapter addresses this problem based on the notation of labeled RFSs.

Chapter 5

Multi-Object Tracking Using Labeled Random Finite Sets

The absence of an explicit data association in the SMC multi-object Bayes filter as well as in the approximate multi-object filtering algorithms introduced in Section 3.7 requires an additional track extraction algorithm in the post-processing to obtain trajectories of the individual objects. Especially in SMC implementations of the PHD and CPHD filter, the track extraction is computationally expensive and error-prone.

The class of labeled random finite sets (see Section 3.3) enables the estimation of object trajectories since a label ℓ is appended to each of the state vectors and unique labels are ensured within a realization of a labeled RFS $\mathbf{X}$. In [VV13b], the δ-generalized labeled multi-Bernoulli (δ-GLMB) distribution is utilized to derive the δ-GLMB filter which implicitly creates individual object tracks.

In Section 5.1, the δ-generalized labeled multi-Bernoulli (δ-GLMB) filter is summarized and the issues concerning its computational complexity are discussed. Afterwards, the labeled multi-Bernoulli (LMB) filter and its implementation using SMC and GM methods are proposed. Additionally, a grouping procedure which significantly reduces the computational complexity of the LMB filter is introduced. Further, an adaptive birth density, a track extraction scheme, and the multi-target generalized NIS for the LMB are proposed.

5.1 The δ-Generalized Labeled Multi-Bernoulli Filter

In [VV13b], the δ-GLMB RFS is shown to be a conjugate prior [RS68] with respect to the update of the multi-object Bayes filter using the standard multi-object likelihood function (3.38). Since a δ-GLMB RFS is also closed under the prediction using the multi-object motion model given by (3.41), a δ-GLMB representation of the multi-object state allows for an analytic implementation of the multi-object Bayes filter, the δ-GLMB filter.

As mentioned in Section 3.3.3, the δ-GLMB filter uses a specific version of the δ-GLMB RFS, where the discrete space Ξ represents the history of track label to measurement associations. In this case, the multi-object density of the δ-GLMB RFS in (3.31) simplifies to

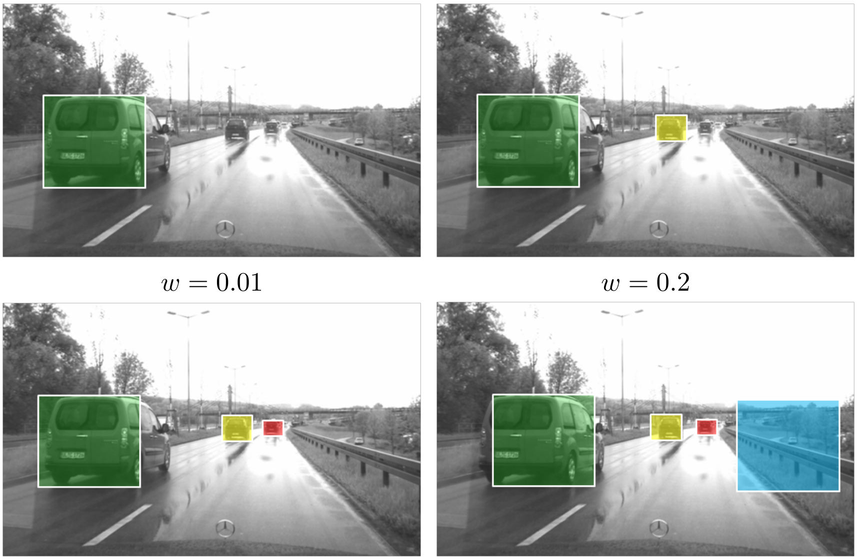
$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} \delta_I(\mathcal{L}(\mathbf{X})) \left[p^{(\xi)}\right]^{\mathbf{X}}, \tag{5.1}$$

where $p^{(I,\xi)}$ is abbreviated by $p^{(\xi)}$. The abbreviation is possible since the set of track labels I is implicitly contained in the history of track label to measurement associations ξ. Within a hypothesis (I,ξ), only tracks with a label $\ell \in I$ exist and the history of track label to measurement associations is given by $\xi \in \Xi$. Assuming that the associations between track labels and measurements at time k are given by an association map θ_k, the history of track label to measurement associations for a set I is given by concatenating all association maps up to time k:

$$\xi = (\theta_1, ..., \theta_k). \tag{5.2}$$

In the prediction step to time $k+1$, measurements up to time k have been associated to the tracks. Thus, the association maps are available up to θ_k. In the update step at time $k+1$, the track label to measurement associations for time $k+1$ are explicitly used and the history of association maps consequently contains associations up to time θ_{k+1}. The normalized weight $\omega^{(I,\xi)}$ can be interpreted as the probability of hypothesis (I,ξ). The spatial distribution of a track ℓ depends on all associations up to the current time step and is given by $p^{(\xi)}(x,\ell)$. Obviously, a history of association maps ξ may be inconsistent with a set of track labels I, e.g. if one of the association maps does not contain an association for one of the tracks. Hence, inconsistent pairs (I,ξ) have zero weight.

Figure 5.1 exemplarily illustrates the δ-GLMB representation of the multi-object posterior where a total number of four hypotheses is used to represent the uncertainty about the true multi-object state. In general, the δ-GLMB distribution facilitates several hypotheses for the same set of track labels, if the track label to measurement association is ambiguous.



$w = 0.01$ $w = 0.2$

$w = 0.74$ $w = 0.05$

Figure 5.1: Visualization of the δ-GLMB representation of the multi-object posterior using an example of a vehicle tracking application.

5.1.1 Prediction

In [VV13b], the δ-GLMB RFS is shown to be closed under the prediction using the standard multi-object motion model (3.41), i.e. the prediction of a δ-GLMB distributed multi-object prior from time k to time $k+1$ is again a δ-GLMB RFS. Hence, if the multi-object prior at time k is given by (5.1), the predicted multi-object distribution is given by the δ-GLMB RFS

$$\boldsymbol{\pi}_+(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I_+,\xi)\in\mathcal{F}(\mathbb{L}_+)\times\Xi} w_+^{(I_+,\xi)} \delta_{I_+}(\mathcal{L}(\mathbf{X})) \left[p_+^{(\xi)}\right]^{\mathbf{X}}. \tag{5.3}$$

Since the multi-object motion model incorporates object persistence and object birth, the predicted set of track labels I_+ consists of the set of surviving tracks $L = I_+ \cap \mathbb{L}$

and the set of new born tracks $J = I_+ \cap \mathbb{B}$. The label space $\mathbb{B}$ for new born targets at the current time step has to be disjoint with the label space $\mathbb{L}$, i.e. $\mathbb{L} \cap \mathbb{B} = \varnothing$. Consequently, the label space of the predicted set of track labels I_+ is given by $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$.

The state of a persisting track $\ell \in \mathbb{L}$ with association map ξ is predicted using the single-object motion model $f_+(x|\cdot, \ell)$ and the state dependent survival probability $p_S(x, \ell)$. Thus, the predicted state is given by

$$p_S^{(\xi)}(x, \ell) = \frac{\langle p_S(\cdot, \ell) f_+(x|\cdot, \ell), p^{(\xi)}(\cdot, \ell)\rangle}{\eta_S^{(\xi)}(\ell)} \tag{5.4}$$

where the normalization constant

$$\eta_S^{(\xi)}(\ell) = \int \left\langle p_S(\cdot, \ell) f_+(x|\cdot, \ell), p^{(\xi)}(\cdot, \ell)\right\rangle dx \tag{5.5}$$

corresponds to the state independent survival probability of track ℓ. The predicted set of track labels I_+ may be obtained from all sets I which include the set $L = I_+ \cap \mathbb{L}$ of surviving tracks. In order to calculate the weight of the predicted set I_+, the probability that a track $\ell \notin I_+$ disappears during prediction is required:

$$q_S^{(\xi)}(\ell) = \left\langle q_S(\cdot, \ell), p^{(\xi)}(\cdot, \ell)\right\rangle. \tag{5.6}$$

Here, $q_S(x, \ell) = 1 - p_S(x, \ell)$ is the state dependent probability that a track disappears. Using (5.5) and (5.6), the weight of a persisting set of track labels L is given by

$$w_S^{(\xi)}(L) = [\eta_S^{(\xi)}]^L \sum_{I \subseteq \mathbb{L}} 1_I(L) [q_S^{(\xi)}]^{I-L} w^{(I,\xi)}. \tag{5.7}$$

Similar to other RFS based tracking algorithms, the δ-GLMB filter uses a birth distribution in the prediction step. Following [VV13b], GLMB birth distributions, which cover both labeled Poisson and labeled multi-Bernoulli distributions, are applicable to the δ-GLMB filter. In this work, only labeled multi-Bernoulli (LMB) birth distributions

$$\boldsymbol{\pi}_B(\mathbf{X}) = \left\{ r_B^{(\ell)}, p_B^{(\ell)} \right\}_{\ell \in \mathbb{B}} \tag{5.8}$$

are considered where the number of new born tracks $|\mathbb{B}|$, the initial existence probabilities $r_B^{(\ell)}$ and the spatial distributions $p_B^{(\ell)}$ depend on the application. Using (3.20), the expected number of new born objects is $\hat{N}_B = \sum_{\ell \in \mathbb{B}} r_B^{(\ell)}$ and (3.26) yields the probability for a subset $J \in \mathbb{B}$ of new born tracks:

$$w_B(J) = \prod_{i \in \mathbb{B}(M)} \left(1 - r_B^{(i)}\right) \prod_{\ell \in J} \frac{1_{\mathbb{B}(M)}(\ell) r_B^{(\ell)}}{1 - r_B^{(\ell)}}. \tag{5.9}$$

Using (5.7) and (5.9), the weight of a predicted hypothesis (I_+, ξ) is given by

$$w_+^{(I_+,\xi)} = w_B(I_+ \cap \mathbb{B}) w_S^{(\xi)}(I_+ \cap \mathbb{L}) \tag{5.10}$$

and the predicted spatial distribution of a track with label ℓ follows

$$p_+^{(\xi)}(x, \ell) = 1_{\mathbb{L}}(\ell) p_S^{(\xi)}(x, \ell) + 1_{\mathbb{B}}(\ell) p_B(x, \ell). \tag{5.11}$$

Using the inclusion functions $1_{\mathbb{L}}(\ell)$ and $1_{\mathbb{B}}(\ell)$, (5.11) returns either the spatial distribution of the persisting track ($\ell \in \mathbb{L}$) or the spatial distribution of a new born track ($\ell \in \mathbb{B}$). The history of association maps is not affected by the prediction.

Figure 5.2 illustrates the δ-GLMB filter prediction for a hypothesis (I, ξ). Due to appearance and disappearance of objects, the predicted set of track labels I_+ differs from I in all but one of the predicted hypotheses.

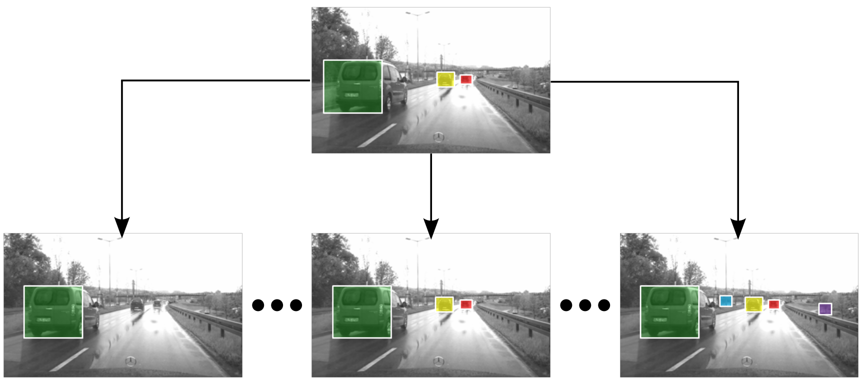


Figure 5.2: Example for the δ-GLMB prediction of a single hypothesis (I, ξ): prior hypothesis at the top, possible predicted hypotheses at the bottom.

5.1.2 Update

In [VV13b], the δ-GLMB RFS is shown to be a conjugate prior with respect to the update of the multi-object Bayes filter using the multi-object likelihood function, i.e. the multi-object posterior of a δ-GLMB RFS is again a δ-GLMB RFS. However, due to the association uncertainties in the update step, the number of hypotheses increases. Consequently, a truncation of the distribution is required to ensure computational tractability. The multi-object posterior of the predicted δ-GLMB RFS (5.3) is given by the δ-GLMB RFS

$$\boldsymbol{\pi}(\mathbf{X}|\mathrm{Z}) = \Delta(\mathbf{X}) \sum_{(I_+,\xi)\in\mathcal{F}(\mathbb{L}_+)\times\Xi} \sum_{\theta\in\Theta} w^{(I_+,\xi,\theta)}(\mathrm{Z})\delta_{I_+}(\mathcal{L}(\mathbf{X})) \left[p^{(\xi,\theta)}(\cdot|\mathrm{Z})\right]^{\mathbf{X}}, \quad (5.12)$$

where Θ denotes the space of track label to measurement associations $\theta : \mathbb{L}_+ \to \{0, 1, ..., |\mathrm{Z}|\}$, where 0 represents the missed detection. Since a measurement is assumed to be generated by at most one object, the property $\theta(i) \neq \theta(j) > 0$ has to be fulfilled for all $i \neq j$. Observe that the posterior sets of track labels are equivalent to the predicted sets of track labels, i.e. $I = I_+$.

Depending on the association map θ, a track ℓ is either assumed to be updated by a measurement $z_{\theta(\ell)}$ ($\theta(\ell) > 0$) or to be miss detected ($\theta(\ell) = 0$). The updated spatial distribution of a track ℓ is given by

$$p^{(\xi,\theta)}(x,\ell|\mathrm{Z}) = \frac{p_+^{(\xi)}(x,\ell)\psi_{\mathrm{Z}}(x,\ell;\theta)}{\eta_{\mathrm{Z}}^{(\xi,\theta)}(\ell)} \quad (5.13)$$

where the normalization constant

$$\eta_{\mathrm{Z}}^{(\xi,\theta)}(\ell) = \left\langle p_+^{(\xi)}(\cdot,\ell), \psi_{\mathrm{Z}}(\cdot,\ell;\theta)\right\rangle \quad (5.14)$$

corresponds to the likelihood of the track label to measurement association. The generalized measurement likelihood $\psi_{\mathrm{Z}}(x,\ell;\theta)$ in (5.13) is defined by

$$\psi_{\mathrm{Z}}(x,\ell;\theta) = \delta_0(\theta(\ell))q_D(x,\ell) + (1-\delta_0(\theta(\ell)))\frac{p_D(x,\ell)g(z_{\theta(\ell)}|x,\ell)}{\kappa(z_{\theta(\ell)})} \quad (5.15)$$

and consequently incorporates both missed detection and measurement update. Here, the state dependent detection probability of the track is denoted by $p_D(x,\ell)$ and the probability of a missed detection is $q_D(x,\ell) = 1 - p_D(x,\ell)$. Further, $\kappa(z) = \lambda_c c(z)$ represents the intensity function of the Poisson distributed clutter process with a mean number of λ_c clutter measurements and $g(z_{\theta(\ell)}|x,\ell)$ is the spatial likelihood of the measurement $z_{\theta(\ell)}$ given a track with label ℓ.

The updated weight of a hypothesis (I, ξ, θ) is finally obtained using the association likelihoods η_Z:

$$w^{(I,\xi,\theta)}(Z) = \frac{\delta_{\theta^{-1}(\{0:|Z|\})}(I) w_+^{(I,\xi)} [\eta_Z^{(\xi,\theta)}]^I}{\sum\limits_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum\limits_{\theta\in\Theta} \delta_{\theta^{-1}(\{0:|Z|\})}(I) w_+^{(I,\xi)} [\eta_Z^{(\xi,\theta)}]^I}. \tag{5.16}$$

Here, the denominator normalizes the weights to ensure that they sum up to one.

Figure 5.3 illustrates the δ-GLMB filter update for a hypothesis (I_+, ξ). The hypothesis on the left illustrates the correct association (two objects are detected, one object is not detected, and the two additional measurements are considered as false alarms). Due to the different track label to measurement associations, each image in the bottom row corresponds to a distinct posterior hypothesis (I, ξ).

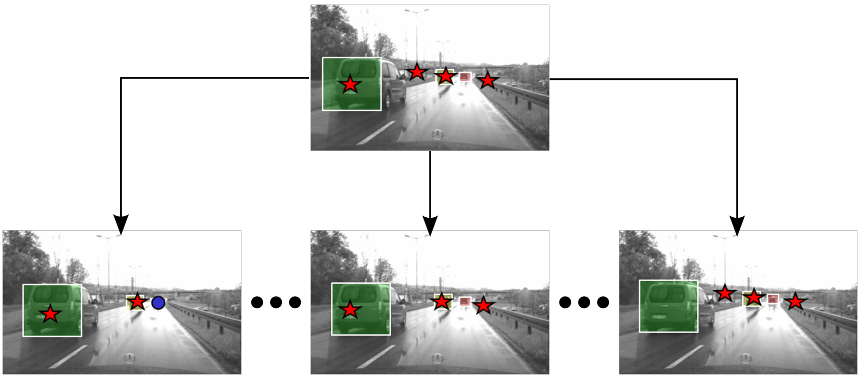


Figure 5.3: Example for the δ-GLMB update of a predicted hypothesis (I_+, ξ) shown in the upper row. Bottom row illustrates three possible track label to measurement associations where red stars depict the measurements and blue circles represent missed detections.

5.1.3 Discussion

An analytical implementation of the δ-GLMB filter is possible using both GM and SMC methods. In [VV11; VV12; VV13b], first simulation results of the GM and SMC implementations are presented which indicate that the δ-GLMB filter significantly outperforms the CPHD filter. Since the prediction and update steps of the δ-GLMB filter are combinatorial, the number of hypotheses increases exponentially. Thus, it is necessary to prune hypotheses with insignificant weights. A straight-forward solution is to evaluate all hypotheses and subsequently prune the ones with weights $w < \vartheta$ where ϑ is the pruning threshold. A computationally more efficient solution is the use of Murty's algorithm [Mur68] (see Section 2.4.3, page 17), which is also used in MHT implementations to reduce the computational complexity. Murty's algorithm can be applied in the prediction step to find the M most likely predicted sets I_+ for a set I and in the update step to find the M best association maps θ for a hypothesis (I, ξ). The main benefit of Murty's algorithm is that it provides the M best solutions without evaluation of all possible solutions. In the prediction step, the truncation can also be performed using the k shortest paths algorithm [Epp98; Yen71], which is computationally more efficient than Murty's algorithm.

Although the truncation using Murty's algorithm significantly reduces the computational costs, an implementation of the δ-GLMB filter is only possible for a limited number of objects. Consider the following example for illustration: Assume a prior set of track labels I with cardinality $|I| = 20$. If the predicted hypotheses have to represent that up to five objects out of I may disappear, a total number of

$$N_{hyp} = \sum_{i=15}^{20} \binom{20}{i} = 21700$$

hypotheses is required to cover all possible predicted sets I_+. Hence, significant cardinality changes in combination with a high number of objects considerably increase the required number of hypotheses and consequently the computational complexity of the δ-GLMB filter.

At a conceptual level, the δ-GLMB filter resembles the track-oriented multi-hypothesis tracking (MHT). Within the measurement update, both algorithms require the computation of data associations. Further, the multi-object likelihood function only differs from the MHT likelihood function by a constant factor in case of a constant detection probability [Mah07a, pp. 422ff.]. However, the prediction step of the δ-GLMB filter differs, since it generates multiple sets of predicted tracks for a set of posterior tracks and explicitly adds new tracks using a birth model. The main benefit of the δ-GLMB filter is that no additional techniques like track or hypothesis scoring for track maintenance are required since the problem is completely cast into a fully probabilistic framework. Hence, the δ-GLMB filter proposed in [VV13b] is the first Bayes optimal approach for multi-object detection and tracking since it estimates both states and identities of the objects and provides a benchmark for approximate multi-object tracking algorithms. The probabilistic framework allows for further approximations along the way and still enables the characterization of the obtained error.

5.2 The Labeled Multi-Bernoulli Filter

The concept behind the labeled multi-Bernoulli (LMB) filter is the representation of the predicted and posterior multi-object densities using an LMB RFS. The main advantage of an LMB representation compared to a δ-GLMB representation is the linear increase of the number of components in the number of objects. Hence, an LMB approximation is expected to significantly reduce the computational cost in scenarios with a large number of objects.

The idea to represent the predicted and posterior multi-object densities using the parameters of a multi-Bernoulli RFS was already considered in the derivation of the multi-target multi-Bernoulli (MeMBer) filter [Mah07a] and the CB-MeMBer filter [VVC09]. Although the CB-MeMBer filter drastically reduces the cardinality error of the MeMBer filter, the approximations within the derivation of the update step still require a considerably high detection probability together with a relatively low false alarm rate to prevent a bias in the cardinality estimate. The aim of the proposed LMB filter is to provide a significantly better tracking accuracy compared to the CB-MeMBer filter due to a more precise update step and the integrated estimation of the track identity.

5.2.1 Prediction

In the LMB filter, the multi-object posterior is assumed to follow an LMB RFS with state space $\mathbb{X}$ and label space $\mathbb{L}$ which is given by

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) w(\mathcal{L}(\mathbf{X})) p^{\mathbf{X}}, \tag{5.17}$$

where

$$w(L) = \prod_{i \in \mathbb{L}} \left(1 - r^{(i)}\right) \prod_{\ell \in L} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)}}{1 - r^{(\ell)}}, \tag{5.18}$$

$$p(x, \ell) = p^{(\ell)}(x). \tag{5.19}$$

Further, the multi-object birth density is an LMB RFS with state space $\mathbb{X}$ and label space $\mathbb{B}$ which is given by

$$\boldsymbol{\pi}_B(\mathbf{X}) = \Delta(\mathbf{X}) w_B(\mathcal{L}(\mathbf{X})) \left[p_B\right]^{\mathbf{X}}, \tag{5.20}$$

where

$$w_B(I) = \prod_{i \in \mathbb{B}} \left(1 - r_B^{(i)}\right) \prod_{\ell \in L} \frac{1_{\mathbb{B}}(\ell) r_B^{(\ell)}}{1 - r_B^{(\ell)}}, \tag{5.21}$$

$$p_B(x, \ell) = p_B^{(\ell)}(x). \tag{5.22}$$

Obviously, the labels $\ell \in \mathbb{B}$ of new born objects have to be distinct to ensure unique track labels. Further, the sets $\mathbb{L}$ and $\mathbb{B}$ have to be disjoint, i.e. $\mathbb{L} \cap \mathbb{B} = \emptyset$.

Although the prediction of a GLMB or a δ-GLMB RFS is closed under the prediction using the Chapman Kolmogorov equation [VV13b], it does not imply that an LMB RFS is also closed under the prediction. However, using the property that an LMB RFS is a special case of a GLMB RFS, the prediction of an LMB RFS is given by the GLMB RFS

$$\boldsymbol{\pi}_+(\mathbf{X}_+) = \Delta(\mathbf{X}_+) w_+(\mathcal{L}(\mathbf{X}_+)) \left[p_+\right]^{\mathbf{X}_+} \tag{5.23}$$

with state space $\mathbb{X}$ and finite label space $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$ where

$$w_+(I_+) = w_B(I_+ \cap \mathbb{B}) w_S(I_+ \cap \mathbb{L}), \tag{5.24}$$

$$p_+(x, \ell) = 1_{\mathbb{L}}(\ell) p_{+,S}(x, \ell) + 1_{\mathbb{B}}(\ell) p_B(x, \ell), \tag{5.25}$$

$$p_{+,S}(x, \ell) = \frac{\left\langle p_S(\cdot, \ell) f_+(x|\cdot, \ell), p(\cdot, \ell) \right\rangle}{\eta_S(\ell)}, \tag{5.26}$$

$$\eta_S(\ell) = \int \left\langle p_S(\cdot, \ell) f_+(x|\cdot, \ell), p(\cdot, \ell) \right\rangle dx, \tag{5.27}$$

$$w_S(L) = [\eta_S]^L \sum_{I \supseteq L} [1 - \eta_S]^{I - L} w(I), \tag{5.28}$$

$p_S(x, \ell)$ denotes the state dependent survival probability, and $f_+(x|\cdot, \ell)$ is the single-object Markov transition density. Equations (5.23)–(5.28) directly correspond to the prediction equations of the δ-GLMB filter but, due to the LMB prior, the history of association maps ξ is missing. Following the discussion in Section 3.3.2 and Section 3.3.3, the sum over all hypotheses (I, ξ) in (5.3) reduces to the single term in (5.23).

Proposition 5.1. *The predicted multi-object density* (5.23) *is the GLMB representa-*

tion of the LMB RFS

$$\boldsymbol{\pi}_+ = \left\{ \left(r_+^{(\ell)}, p_+^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}_+} = \left\{ \left(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}} \cup \left\{ \left(r_B^{(\ell)}, p_B^{(\ell)} \right) \right\}_{\ell \in \mathbb{B}}, \tag{5.29}$$

where

$$r_{+,S}^{(\ell)} = \eta_S(\ell) r^{(\ell)}, \tag{5.30}$$

$$p_{+,S}^{(\ell)} = \left\langle p_S(\cdot,\ell) f_+(x|\cdot,\ell), p(\cdot,\ell) \right\rangle / \eta_S(\ell), \tag{5.31}$$

$p_S(x,\ell)$ *denotes the state dependent survival probability,* $f_+(x|\cdot,\ell)$ *is the single-object Markov transition density, and* $\eta_S(\ell)$ *is given by* (5.27)*.*

Proof. Since the GLMB RFS (5.23) comprises only a single component, (5.23) is an LMB RFS if the weight (5.24) follows the weight of an LMB RFS. Obviously, the weights $w_B(I_+ \cap \mathbb{B})$ of the new born objects follow an LMB RFS since they are given by (5.21). Thus, if the predicted weights $w_S(L)$ in (5.28) can be rewritten in the LMB form

$$\begin{aligned} w_S^{\mathrm{LMB}}(L) &= \prod_{i \in \mathbb{L}} \left(1 - r^{(i)} \eta_S(i) \right) \prod_{\ell \in L} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)} \eta_S(\ell)}{1 - r^{(\ell)} \eta_S(\ell)} \\ &= (1 - r^{(\cdot)} \eta_S)^{\mathbb{L}} \left(\frac{r^{(\cdot)} \eta_S}{1 - r^{(\cdot)} \eta_S} \right)^L, \end{aligned} \tag{5.32}$$

the predicted GLMB RFS (5.23) is equivalent to (5.29).

The weight $w_S(L)$ denotes the probability that all tracks with label $\ell \in L$ of a superset $I \supseteq L$ survive while all tracks of the supersets with a label $\ell \notin L$ disappear. By enumerating the labels $\mathbb{L} = \{\ell_1, ..., \ell_M\}$, the weight $w_S(L)$ can be rewritten in the following form using (5.18):

$$w_S(L) = \sum_{I \supseteq L} (1 - r^{(\cdot)})^{\mathbb{L}} \eta_S^L \frac{(1 - \eta_S)^I}{(1 - \eta_S)^L} \left(\frac{r^{(\cdot)}}{1 - r^{(\cdot)}} \right)^I. \tag{5.33}$$

Next, the equality of (5.32) and (5.33) is shown by induction. Since $I = L$ is the only superset of $L = \mathbb{L}$, (5.33) simplifies in this case to

$$w_S(L) = \left(\eta_S r^{(\cdot)} \right)^L.$$

Obviously, this result is equivalent to the result of (5.32). Having proven the equality for $L = \mathbb{L}$, the inductive step is to proof that the proposition for a set $L = \{\ell_1, ..., \ell_n\}$ implies the validity of the proposition for a set $L \backslash \{\ell_n\}$. For notational convenience the summands of (5.33) are abbreviated by

$$f^{(L)}(I) \triangleq (1 - r^{(\cdot)})^{\mathbb{L}} \eta_S^L \frac{(1-\eta_S)^I}{(1-\eta_S)^L} \left(\frac{r^{(\cdot)}}{1 - r^{(\cdot)}} \right)^I \tag{5.34}$$

in the following. For any set $L \backslash \{\ell_n\}$, the weight $w_S(L)$ is given by

$$w_S(L \backslash \{\ell_n\}) \triangleq \sum_{I \supseteq L \backslash \{\ell_n\}} f^{(L \backslash \{\ell_n\})}(I) \tag{5.35}$$

$$= \sum_{I \supseteq L} \left[f^{(L \backslash \{\ell_n\})}(I) + f^{(L \backslash \{\ell_n\})}(I \backslash \{\ell_n\}) \right], \tag{5.36}$$

where the second equality follows from the fact, that only the sets $I \supseteq \{\ell_1, ..., \ell_n\}$ and the sets $I \backslash \{\ell_n\}$ contain the set $L = \{\ell_1, ..., \ell_{n-1}\}$. Using (5.34), the summands in (5.36) can be expressed as functions of $f^{(L)}(I)$:

$$f^{(L \backslash \{\ell_n\})}(I) = \frac{1 - \eta_S(\ell_n)}{\eta_S(\ell_n)} f^{(L)}(I) = \frac{r^{(\ell_n)} - r^{(\ell_n)} \eta_S(\ell_n)}{r^{(\ell_n)} \eta_S(\ell_n)} f^{(L)}(I), \tag{5.37}$$

$$f^{(L \backslash \{\ell_n\})}(I \backslash \{\ell_n\}) = \frac{1 - r^{(\ell_n)}}{r^{(\ell_n)} \eta_S(\ell_n)} f^{(L)}(I). \tag{5.38}$$

The weight of the set $L \backslash \{\ell_n\}$ can be rewritten in terms of the weight of the set L by applying (5.37) and (5.38) to (5.36):

$$w_S(L \backslash \{\ell_n\}) = \frac{1 - r^{(\ell_n)} \eta_S(\ell_n)}{r^{(\ell_n)} \eta_S(\ell_n)} \sum_{I \supseteq L} f^{(L)}(I). \tag{5.39}$$

Representing the weight of the set L using the proposition (5.32), the weight of the set $L \backslash \{\ell_n\}$ simplifies to

$$\begin{aligned} w_S(L \backslash \{\ell_n\}) &= \left(\frac{1 - r^{(\ell_n)} \eta_S(\ell_n)}{r^{(\ell_n)} \eta_S(\ell_n)} \right) (1 - r^{(\cdot)} \eta_S)^{\mathbb{L}} \left(\frac{r^{(\cdot)} \eta_S}{1 - r^{(\cdot)} \eta_S} \right)^L \\ &= (1 - r^{(\cdot)} \eta_S)^{\mathbb{L}} \left(\frac{r^{(\cdot)} \eta_S}{1 - r^{(\cdot)} \eta_S} \right)^{L \backslash \{\ell_n\}} . \end{aligned}$$

Thus, the weight $w_S(I_+ \cap \mathbb{L})$ in (5.24) follows an LMB RFS and (5.23) is an LMB RFS. □

Since the prediction using (5.29) is equivalent to (5.23), an LMB RFS is closed under the prediction using the standard multi-object motion model. Further, the LMB prediction is identical to the prediction step of the unlabeled CB-MeMBer filter in Section 3.7.3, if the component indices in (3.60) are interpreted as track labels.

5.2.2 Update

The standard multi-object measurement model assumes that a measurement is generated by at most one object. Consequently, the association of a measurement m to a track ℓ_n (i.e. $\theta(\ell_n) = m$) within a set of track labels I_+ implies that the measurement may not be associated to any other track (i.e. $\theta(\ell_i) \neq m \; \forall \; i \neq n$) within the same association map. Hence, the possible track to measurement associations depend on all tracks of the set I_+ and consequently the tracks are not independent of each other any more. Since an LMB RFS is defined as the union of independent Bernoulli components, an LMB RFS is not a conjugate prior with respect to the update using the multi-object likelihood function. Consequently, an LMB approximation of the multi-object posterior density is required.

Proposition 5.2. *If the predicted multi-object density follows an LMB RFS with state space $\mathbb{X}$, finite label space $\mathbb{L}_+$, and parameter set $\boldsymbol{\pi}_+ = \{(r_+^{(\ell)}, p_+^{(\ell)})\}_{\ell\in\mathbb{L}_+}$, the LMB RFS*

$$\boldsymbol{\pi}(\mathbf{X}|\mathrm{Z}) = \{(r^{(\ell)}, p^{(\ell)})\}_{\ell\in\mathbb{L}_+} \tag{5.40}$$

with parameters

$$r^{(\ell)} = \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell) \tag{5.41}$$

$$p^{(\ell)}(x) = \frac{1}{r^{(\ell)}} \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell) p^{(\theta)}(x,\ell) \tag{5.42}$$

and

$$w^{(I_+,\theta)}(\mathrm{Z}) = \frac{\delta_{\theta^{-1}(\{0:|\mathrm{Z}|\})}(I_+) w_+(I_+) [\eta_\mathrm{Z}^{(\theta)}]^{I_+}}{\sum\limits_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} \delta_{\theta^{-1}(\{0:|\mathrm{Z}|\})}(I_+) w_+(I_+) [\eta_\mathrm{Z}^{(\theta)}]^{I_+}}, \tag{5.43}$$

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \frac{p_+(x,\ell)\psi_\mathrm{Z}(x,\ell;\theta)}{\eta_\mathrm{Z}^{(\theta)}(\ell)}, \tag{5.44}$$

$$\eta_\mathrm{Z}^{(\theta)}(\ell) = \left\langle p_+(\cdot,\ell), \psi_\mathrm{Z}(\cdot,\ell;\theta) \right\rangle, \tag{5.45}$$

$$\psi_{\mathrm{Z}}(x,\ell;\theta) = \delta_0(\theta(\ell))q_D(x,\ell) + (1-\delta_0(\theta(\ell)))\frac{p_D(x,\ell)g(z_{\theta(\ell)}|x,\ell)}{\kappa(z_{\theta(\ell)})} \tag{5.46}$$

exactly matches the first moment, i.e. the PHD, of each individual track and consequently the PHD of the multi-object posterior density. Here, Θ_{I_+} *denotes the space of mappings* $\theta : I_+ \to \{0,1,\ldots,|\mathrm{Z}|\}$ *where* $\theta(i) \neq \theta(j) > 0$ *implies* $i = j$*, i.e. the mapping ensures that one measurement is generated by at most one object. Further,* $p_D(x,\ell)$ *denotes the state dependent detection probability of the track,* $q_D(x,\ell) = 1 - p_D(x,\ell)$ *is the missed detection probability,* $\kappa(z) = \lambda_c c(z)$ *represents the intensity function of the Poisson distributed clutter process with a mean number of* λ_c *clutter measurements, and* $g(z_{\theta(\ell)}|x,\ell)$ *is the spatial likelihood of the measurement* $z_{\theta(\ell)}$ *given a track with label* ℓ*.*

Proof. Using (3.36), the predicted LMB RFS (5.29) can be equivalently rewritten in δ-GLMB form:

$$\boldsymbol{\pi}_+(\mathbf{X}_+) = \Delta(\mathbf{X}_+) \sum_{I_+ \in \mathcal{F}(\mathbb{L}_+)} w_+(I_+)\delta_{I_+}(\mathcal{L}(\mathbf{X}_+)) \left[p_+\right]^{\mathbf{X}_+}. \tag{5.47}$$

Since a δ-GLMB RFS is a conjugate prior with respect to the update of the multi-object Bayes filter using the multi-object likelihood function, the multi-object posterior is again a δ-GLMB RFS. Due to the LMB representation of the tracks, the history of association maps is not available any more in (5.47). Hence, the δ-GLMB update (5.12) introduced in Section 5.1.2 simplifies to

$$\boldsymbol{\pi}(\mathbf{X}|\mathrm{Z}) = \Delta(\mathbf{X}) \sum_{(I_+,\theta) \in \mathcal{F}(\mathbb{L}_+)\times\Theta_{I_+}} w^{(I_+,\theta)}(\mathrm{Z})\delta_{I_+}(\mathcal{L}(\mathbf{X})) \left[p^{(\theta)}(\cdot|\mathrm{Z})\right]^{\mathbf{X}} \tag{5.48}$$

and the measurement updated weights $w^{(I_+,\theta)}(\mathrm{Z})$ and spatial distributions $p^{(\theta)}(\cdot|\mathrm{Z})$ are given by (5.43) and (5.44) which correspond to the update equations of the δ-GLMB filter if no history of association maps is available.

Since the LMB filter requires the representation of the multi-object posterior distribution by an LMB RFS, an approximation of the measurement updated δ-GLMB RFS (5.48) is necessary. An intuitive approximation is the representation of the posterior distribution using its PHD which corresponds to the first statistical moment. Using (3.33), the PHD of (5.48) is given by

$$v(x) = \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta} \sum_{\ell\in\mathbb{L}_+} p^{(\theta)}(x,\ell) \sum_{L\subseteq\mathbb{L}_+} 1_L(\ell) w^{(I_+,\theta)}(\mathrm{Z})\delta_{I_+}(L)$$

$$= \sum_{\ell \in \mathbb{L}_+} \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell) p^{(\theta)}(x,\ell). \tag{5.49}$$

As introduced in Section 3.3.3, the unlabeled PHD of a δ-GLMB distribution comprises the PHDs of the individual tracks. Thus, the PHD of track ℓ is given by

$$v^{(\ell)}(x) = \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell) p^{(\theta)}(x,\ell). \tag{5.50}$$

The integral over the PHD of an individual track ℓ may be interpreted as the existence probability of the track ℓ. Using (3.35) and (5.50), the existence probability of a track ℓ is given by

$$r^{(\ell)} = \left\langle v^{(\ell)}, 1 \right\rangle = \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell), \tag{5.51}$$

and its spatial distribution is obtained by normalization:

$$p^{(\ell)}(x) = \frac{v^{(\ell)}(x)}{\left\langle v^{(\ell)}, 1 \right\rangle} = \frac{1}{r^{(\ell)}} \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+)\times\Theta} w^{(I_+,\theta)}(\mathrm{Z}) 1_{I_+}(\ell) p^{(\theta)}(x,\ell). \tag{5.52}$$

Following (3.19), the PHD of the unlabeled version of (5.40) is

$$v^{\mathrm{LMB}}(x) = \sum_{\ell\in\mathbb{L}_+} r^{(\ell)} p^{(\ell)}(x) = \sum_{\ell\in\mathbb{L}_+} \left\langle v^{(\ell)}, 1 \right\rangle \frac{v^{(\ell)}(x)}{\left\langle v^{(\ell)}, 1 \right\rangle} = \sum_{\ell\in\mathbb{L}_+} v^{(\ell)}(x). \tag{5.53}$$

which matches the PHD (5.49) of the full multi-object posterior. The matching PHDs of the distributions comprise identical mean cardinalities, i.e. both distributions deliver the same estimate for the number of objects. Further, the proposed LMB approximation preserves the spatial distribution of each track. □

The main difference between the LMB and the δ-GLMB filter is the representation of the data association uncertainties. While the δ-GLMB filter represents the uncertainty using hypothesis (I, ξ), the LMB filter captures the association uncertainty within the spatial distributions of the individual tracks. Compared to the δ-GLMB filter, the LMB approximation loses information about the cardinality distribution due to the approximation of the multi-object posterior by its PHD. The cardinality distribution of (5.40) is given by the cardinality distribution of a multi-Bernoulli RFS (3.18) which is unimodal with a variance of (Lemma A.1, Appendix A)

$$\sigma^2_{\mathrm{MB}} = \sum_{i=1}^{M} (1 - r^{(i)}) \cdot r^{(i)}. \tag{5.54}$$

In contrast, the cardinality distribution (3.32) of a δ-GLMB RFS is not restricted to a specific distribution. Thus, the cardinality distribution may be multi-modal and the variance may be lower or higher than the one of the LMB approximation.

While the approximations in the derivation of the CB-MeMBer filter update require a high detection probability and a relatively low clutter density, the LMB approximation using the first moment of the full multi-object posterior is not restricted to scenarios which fulfill these restrictions. Thus, the LMB filter is expected to significantly outperform the CB-MeMBer filter in scenarios with low detection probability and high clutter density. Further, the less drastic approximations in the derivation of the LMB filter update promise more accurate estimation results without a bias in the cardinality estimate at the cost of a higher computational complexity.

5.2.3 Approximation Error

The following example illustrates the obtained error due to the approximation of a δ-GLMB RFS by an LMB RFS with matching PHD. Assume a scenario comprising two tracks with labels ℓ_1 and ℓ_2 whose spatial distributions are represented by a Gaussian distribution. Figure 5.4 shows an arbitrary δ-GLMB distribution of the two tracks which represents all possible hypotheses and their corresponding weights. The upper plot illustrates the hypothesis for $\mathbf{X} = \emptyset$, the two plots in the middle show the two hypotheses for cardinality $|\mathbf{X}| = 1$, and the hypothesis for cardinality $|\mathbf{X}| = 2$ is represented by the plot at the bottom. Using (3.32), the cardinality distribution of the δ-GLMB RFS in Figure 5.4 is

$$\rho(n) = \begin{cases} 0.2 & n = 0, \\ 0.3 & n = 1, \\ 0.5 & n = 2, \end{cases}$$

which corresponds to a mean cardinality of $\hat{N} = 1.3$ with a variance of $\sigma^2_{\mathrm{DGLMB}} = 0.61$.

The approximation of the δ-GLMB distribution depicted by Figure 5.4 using an LMB RFS with matching PHD and mean cardinality results in the tracks ℓ_1 and ℓ_2 shown in Figure 5.5. The spatial distributions and the existence probabilities of the tracks are calculated using (5.52) and (5.51). Due to the matching PHD, the spatial distribution of each track is not affected by the approximation. However, the

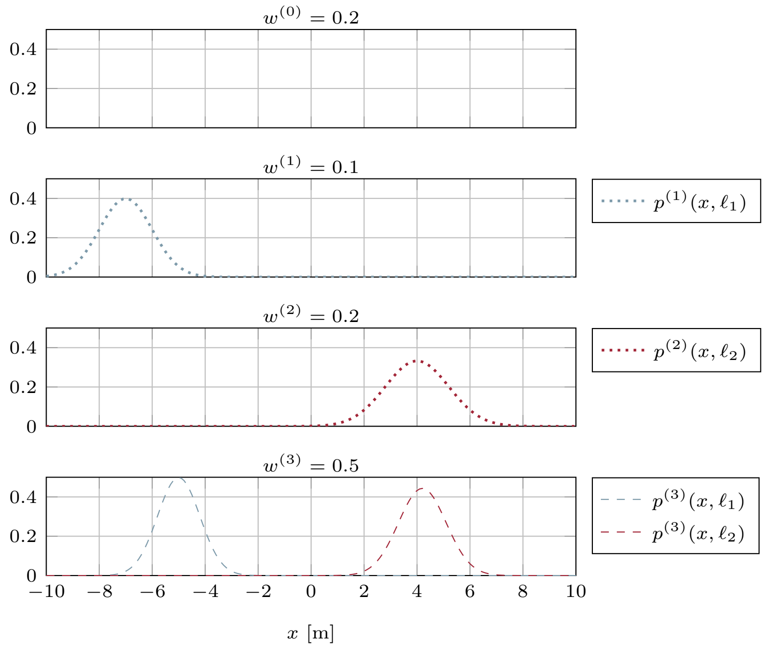


Figure 5.4: δ-GLMB representation of two tracks with labels ℓ_1 and ℓ_2.

cardinality distribution of the approximate LMB process follows (3.18) and is given by

$$\rho(n) = \begin{cases} 0.12 & n = 0, \\ 0.46 & n = 1, \\ 0.42 & n = 2. \end{cases}$$

Obviously, the LMB approximation of a δ-GLMB RFS implicates a loss of information which results in a different cardinality distribution. As expected due to the matching PHD, the mean cardinality $\hat{N} = 1.3$ of the LMB approximation matches the mean cardinality of the original δ-GLMB distribution. In this example, the variance of the estimated number of objects decreases to $\sigma^2_{\mathrm{LMB}} = 0.45$ due to the LMB approximation. However, it is also possible that the variance increases due to the approximation.

Using (3.37), the LMB approximation may be equivalently rewritten in δ-GLMB form. The obtained δ-GLMB RFS is illustrated in Figure 5.6. The cardinality distribution of the δ-GLMB form is equivalent to the one of the LMB distribution and consequently differs from the original cardinality distribution. In contrast to the original δ-GLMB process in Figure 5.4, the spatial distribution of each track ℓ is identical in each of the

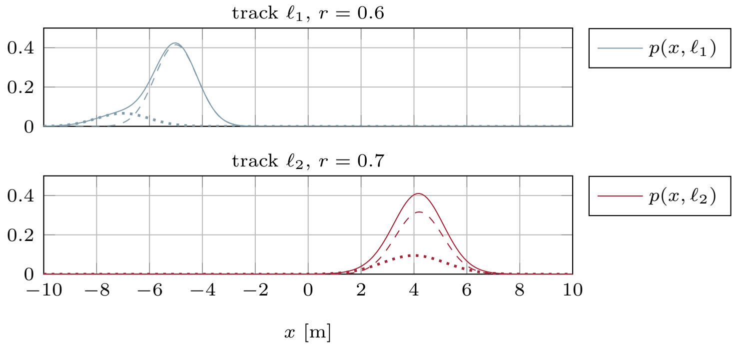


Figure 5.5: LMB approximation of the δ-GLMB distribution shown in Figure 5.4. Both tracks are represented by a mixture of Gaussian distributions and a corresponding existence probability.

hypotheses depicted by Figure 5.6. The spatial distribution of a track is given by the weighted sum of its distributions in the original δ-GLMB process. Both effects are due to the approximation of the original δ-GLMB by an LMB.

5.3 Implementation of the Labeled Multi-Bernoulli Filter

Although the proposed labeled multi-Bernoulli (LMB) filter significantly reduces the computational complexity of the prediction step, the representation of the predicted LMB RFS in δ-GLMB form still requires a huge amount of hypotheses in scenarios with a large number of objects. However, the LMB representation allows for principled approximations which significantly reduce the computational complexity. Figure 5.7 conceptually illustrates the proposed implementation of the LMB filter.

In order to reduce the complexity of the update step, a grouping procedure partitions the predicted LMB density in groups of closely spaced objects and their associated measurements. Using a sufficiently large gating value in the grouping procedure, the groups can be considered to be statistically independent. Hence, the LMB filter update may be applied to each group instead of the entire multi-object state which significantly reduces the required number of hypotheses. The LMB filter update of each group consists of three steps: First, the LMB RFS has to be represented in δ-GLMB form. Afterwards, the full δ-GLMB update is performed which results in a

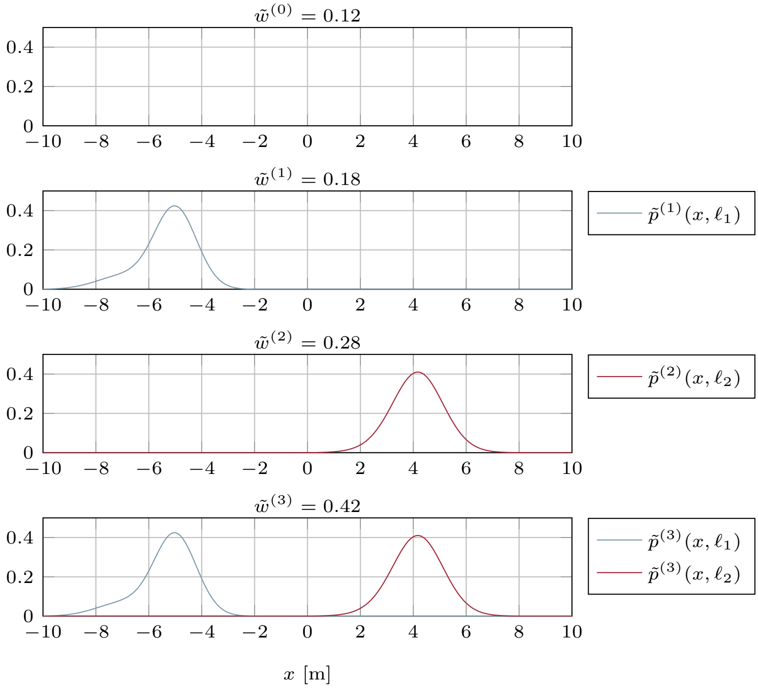


Figure 5.6: δ-GLMB representation of the LMB tracks shown in Figure 5.5.

δ-GLMB posterior. Finally, the δ-GLMB posterior is approximated by an LMB RFS which exactly matches the first multi-object moment. Since the groups are considered to be statistically independent, the posterior LMB RFS for the entire multi-object state is simply given by the union of the LMB RFSs of the N groups. Afterwards, the track management is applied to extract the track estimates and to prune tracks with a small existence probability.

The following subsections provide full details of the LMB filter implementation as well as explicit equations for the sequential Monte-Carlo (SMC) and Gaussian mixture (GM) implementation. The notation of the multi-object densities in Figure 5.7 corresponds to the notation used in the following subsections.

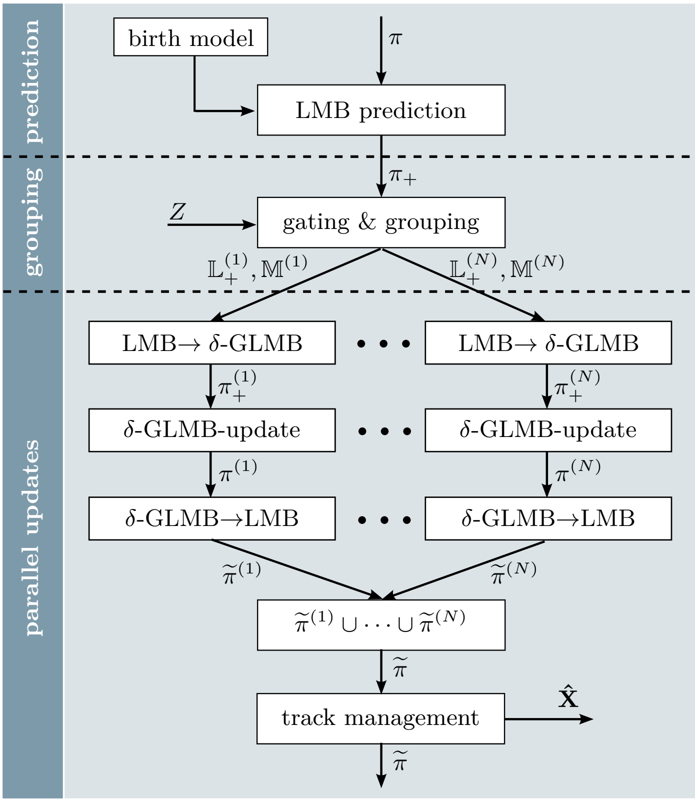


Figure 5.7: LMB filter schematic (cf. [RVVD14]).

5.3.1 Prediction

The prediction step of the LMB filter directly implements the result of Section 5.2.1. Hence, if the multi-object posterior is an LMB RFS with label space $\mathbb{L}$ and parameter set

$$\boldsymbol{\pi} = \left\{ \left(r^{(\ell)}, p^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}},$$

the predicted LMB distribution with label space $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$ is given by

$$\boldsymbol{\pi}_+ = \left\{ \left(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}} \cup \left\{ \left(r_B^{(\ell)}, p_B^{(\ell)} \right) \right\}_{\ell \in \mathbb{B}}.$$

Here, the first parameter set corresponds to the surviving Bernoulli tracks while the second parameter set represents new born objects. While the existence probability and the spatial distribution of each surviving Bernoulli track is modified using the multi-object motion model, the label of a surviving track is unchanged. The labels of the new born tracks have to be distinct and the parameters of the distribution depend on the application.

Sequential Monte Carlo Implementation

In an SMC implementation of the LMB filter, the posterior spatial distribution $p^{(\ell)}$ of a track ℓ is represented by a set of weighted samples $\delta_{x^{(\ell,j)}}(x)$:

$$p^{(\ell)} = \sum_{j=1}^{\nu} w^{(\ell,j)} \delta_{x^{(\ell,j)}}(x).$$

Using (5.30) and (5.31) of Proposition 5.1, the predicted existence probability and the predicted spatial distribution of each track are obtained by:

$$r_{+,S}^{(\ell)} = r^{(\ell)} \sum_{j=1}^{\nu} w^{(\ell,j)} p_S(x^{(\ell,j)}), \tag{5.55}$$

$$p_{+,S}^{(\ell)}(x) = \sum_{j=1}^{\nu} w_{+,S}^{(\ell,j)} \delta_{x_{+,S}^{(\ell,j)}}(x). \tag{5.56}$$

If the predicted samples

$$x_{+,S}^{(\ell,j)} \sim f_+(\cdot|x^{(\ell,j)}),\ \forall\ j = 1, ..., \nu$$

can be drawn directly from the distribution (e.g. for Gaussian and uniform transition densities $f_+(\cdot|x^{(\ell,j)})$, the weights are unchanged during prediction. Otherwise, the predicted samples

$$x_{+,S}^{(\ell,j)} \sim q^{(\ell)}(\cdot|x^{(\ell,j)}),\ \forall\ j = 1, ..., \nu$$

have to be drawn from a proposal density $q^{(\ell)}(\cdot|x)$ which fulfills the condition

$$p_{+,S}^{(\ell)}(x) > 0 \Rightarrow q^{(\ell)}(\cdot|x) > 0\ \forall\ x \in \mathbb{X}.$$

Then, the importance weights $\widetilde{w}_{+,S}^{(\ell,j)}$ of the predicted samples are given by

$$\widetilde{w}_{+,S}^{(\ell,j)} = \frac{w^{(\ell,j)} f_{+}\left(x_{+,S}^{(\ell,j)}|x^{(\ell,j)}\right) p_S(x^{(\ell,j)})}{q^{(\ell)}\left(x_{+,S}^{(\ell,j)}|x^{(\ell,j)}\right)}$$

and the normalized importance weights

$$w_{+,S}^{(\ell,j)} = \frac{\widetilde{w}_{+,S}^{(\ell,j)}}{\sum_{j=1}^{\nu} \widetilde{w}_{+,S}^{(\ell,j)}}$$

are used to weight the particles within the predicted spatial distribution $p_{+,S}^{(\ell)}(x)$.

The Bernoulli components for new born tracks are obtained using the birth model. The existence probability $r_B^{(\ell)}$ of each component as well as the number of birth components are parameters of the birth model. The spatial distribution of the birth components is approximated by a set of weighted samples:

$$p_B^{(\ell)}(x) \approx \sum_{j=1}^{\nu_B} w_B^{(\ell,j)} \delta_{x_B^{(\ell,j)}}(x). \tag{5.57}$$

If the birth density $p_B^{(\ell)}(x)$ follows a Gaussian or a uniform distribution, the samples may be directly drawn from the distribution. Otherwise, the samples are obtained using a proposal density $b^{(\ell)}(x)$ which fulfills the condition

$$p_B^{(\ell)}(x) > 0 \Rightarrow b^{(\ell)}(x) > 0 \;\forall\; x \in \mathbb{X}.$$

Hence, the samples for the spatial distribution of the birth components and their corresponding weights are given by

$$\begin{aligned}
x_B^{(\ell,j)} &\sim b^{(\ell)}(\cdot) \;\forall\; j = 1, ..., \nu_B, \\
\widetilde{w}_B^{(\ell,j)} &= \frac{p_B(x_B^{(\ell,j)})}{b(x_B^{(\ell,j)})}, \\
w_B^{(\ell,j)} &= \frac{\widetilde{w}_B^{(\ell,j)}}{\sum_{j=1}^{\nu_B} \widetilde{w}_B^{(\ell,j)}}.
\end{aligned}$$

Finally, the predicted LMB distribution is represented by the union of surviving and new born tracks:

$$\boldsymbol{\pi}_+ = \left\{ \left(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}} \cup \left\{ \left(r_B^{(\ell)}, p_B^{(\ell)} \right) \right\}_{\ell \in \mathbb{B}} . \tag{5.58}$$

Gaussian Mixture Implementation

In a GM implementation, the posterior probability densities $p^{(\ell)}$ of all labeled Bernoulli tracks $\ell \in \mathbb{L}$ are given by a mixture of Gaussians

$$p^{(\ell)}(x) = \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \hat{x}^{(\ell,j)}, \underline{\mathrm{P}}^{(\ell,j)}\right),$$

where $\hat{x}^{(\ell,j)}$ is the mean value of each Gaussian component and $\underline{\mathrm{P}}^{(\ell,j)}$ is the according estimation error covariance. Similar to the GM implementation of the CB-MeMBer filter [VVC09], the weights $w^{(\ell,j)}$ have to be normalized. The GM implementation requires each object to follow a linear Gaussian process model

$$f_+(x|\xi) = \mathcal{N}\left(x; \underline{\mathrm{F}}\xi, \underline{\mathrm{Q}}\right)$$

using the state transition matrix $\underline{\mathrm{F}}$ and the process noise covariance $\underline{\mathrm{Q}}$. Additionally, the survival probability is assumed to be state independent, i.e.

$$p_S(x) = p_S. \tag{5.59}$$

Using the derivations in Appendix B.2, the predicted existence probability (5.30) and the predicted spatial distribution (5.31) of a track ℓ in the GM implementation of the LMB filter are given by

$$r_{+,S}^{(\ell)} = r^{(\ell)} p_S, \tag{5.60}$$

$$p_{+,S}^{(\ell)}(x) = \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_+^{(\ell,j)}, \underline{\mathrm{P}}_+^{(\ell,j)}\right), \tag{5.61}$$

where

$$\hat{x}_+^{(\ell,j)} = \underline{\mathrm{F}}\hat{x}^{(\ell,j)},$$
$$\underline{\mathrm{P}}_+^{(\ell,j)} = \underline{\mathrm{F}}\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}^{\mathrm{T}} + \underline{\mathrm{Q}}.$$

Thus, each of the $J^{(\ell)}$ Gaussian distributions of a track ℓ is predicted using the standard

Kalman filter equations and the existence probability of the track is reduced by the multiplication with a survival probability $p_S < 1$. An application of the proposed prediction step to slightly non-linear motion models is possible using the EKF or the UKF implementation of the Kalman filter. The derivation of the corresponding prediction equations is analog to the ones for the standard Kalman filter [BF88; JU04]. The explicit equations are identical to the ones of the CB-MeMBer filter given by [Vo08, pp. 210ff.].

The number of new born tracks $|\mathbb{B}|$ as well as the according existence probabilities $r_B^{(\ell)}$, $\ell \in \mathbb{B}$, are application dependent. The GM implementation requires the spatial distributions of the new born Bernoulli tracks to be comprised of a mixture of Gaussian distributions

$$p_B^{(\ell)} = \sum_{j=1}^{J_B} w_B^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_B^{(\ell,j)}, \underline{\mathrm{P}}_B^{(\ell,j)}\right), \tag{5.62}$$

where the number of Gaussian components J_B, the states $\hat{x}_B^{(\ell,j)}$, and the according estimation error covariances $\underline{\mathrm{P}}_B^{(\ell,j)}$ are given by the birth model. The expected number of new born objects at time k is given by the sum of the existence probabilities $r_B^{(\ell)}$, i.e.

$$\hat{N}_B = \sum_{\ell \in \mathbb{B}} r_B^{(\ell)},$$

and the predicted LMB RFS (5.29) is again given by the union of the surviving and new born tracks.

5.3.2 Measurement Updates

In order to calculate the posterior existence probability (5.41) as well as the posterior spatial distribution (5.42) of the individual tracks in Proposition 5.2, the update step of the LMB filter requires the innovations of each track ℓ with all measurements $z_{\theta(\ell)}$. This subsection proposes explicit forms for SMC and GM implementation of the likelihoods

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \left\langle p_+(\cdot, \ell), \psi_{\mathrm{Z}}(\cdot, \ell; \theta)\right\rangle \tag{5.63}$$

and the measurement updated posterior distributions

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \frac{p_+(x,\ell)\psi_\mathrm{Z}(x,\ell;\theta)}{\eta_\mathrm{Z}^{(\theta)}(\ell)}, \tag{5.64}$$

given by (5.45) and (5.44), respectively. The obtained likelihoods are essential for the grouping procedure in Section 5.3.3 and for the calculation of the weights of the hypotheses (5.43) in Section 5.3.4. Thus, the likelihoods for all all association hypotheses are required, while the calculation of the updated posterior distribution is only necessary for association hypotheses which are actually used in the filter update in Section 5.3.4.

Sequential Monte Carlo Implementation

The predicted spatial distribution of each track ℓ in the SMC implementation of the LMB filter is represented by a set of weighted samples:

$$p_+^{(\ell)}(x) = \sum_{j=1}^{\nu^{(\ell)}} w_+^{(\ell,j)} \delta_{x_+^{(\ell,j)}}(x).$$

The likelihood (5.45) for a track label to measurement association $\theta(\ell) > 0$ is given by

$$\eta_\mathrm{Z}^{(\theta)}(\ell) = \frac{1}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{\nu^{(\ell)}} w_+^{(\ell,j)} p_D\left(x_+^{(\ell,j)}\right) g\left(z_{\theta(\ell)}|x_+^{(\ell,j)}\right), \tag{5.65}$$

where $g(z_{\theta(\ell)}|\cdot)$ is the spatial likelihood of the measurement $z_{\theta(\ell)}$ conditioned on the jth particle of track ℓ, $p_D(\cdot)$ is the state dependent detection probability, and $\kappa(\cdot)$ is the intensity of the Poisson clutter process. Given an association $\theta(\ell) > 0$, the updated spatial distribution (5.44) of a track ℓ is

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \sum_{j=1}^{\nu^{(\ell)}} w^{(\ell,j,\theta)}(\mathrm{Z}) \delta_{x^{(\ell,j)}}(x), \tag{5.66}$$

where

$$\begin{aligned} w^{(\ell,j,\theta)}(\mathrm{Z}) &= \frac{\frac{1}{\kappa(z_{\theta(\ell)})} \cdot w_+^{(\ell,j)} p_D\left(x_+^{(\ell,j)}\right) g\left(z_{\theta(\ell)}|x_+^{(\ell,j)}\right)}{\eta_\mathrm{Z}^{(\theta)}(\ell)}, \\ x^{(\ell,j)} &= x_+^{(\ell,j)}. \end{aligned}$$

The likelihood for a missed detection of track ℓ, i.e. $\theta(\ell) = 0$, is given by

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \sum_{j=1}^{\nu^{(\ell)}} w_+^{(\ell,j)} q_D(x^{(\ell,j)}), \tag{5.67}$$

where $q_D(x^{(\ell,j)}) = 1 - p_D(x^{(\ell,j)})$ is the state dependent probability of a missed detection. For $\theta(\ell) = 0$, the updated spatial distribution (5.44) of track ℓ follows

$$p^{(\theta)}(x, \ell|\mathrm{Z}) = \sum_{j=1}^{\nu^{(\ell)}} w^{(\ell,j,\theta)}(\mathrm{Z}) \delta_{x^{(\ell,j)}}(x), \tag{5.68}$$

where

$$w^{(\ell,j,\theta)}(\mathrm{Z}) = \frac{q_D(x^{(\ell,j)}) w_+^{(\ell,j)}}{\eta_{\mathrm{Z}}^{(\theta)}(\ell)}, \tag{5.69}$$

$$x^{(\ell,j)} = x_+^{(\ell,j)}. \tag{5.70}$$

Observe, that the state dependent missed detection probability affects the weights of the particles. In case of a state independent missed detection probability $q_D(x^{(\ell,j)}) = q_D$, the posterior weight (5.69) of each particle matches the predicted weight since the missed detection probability is canceled out. Consequently, the predicted and the updated spatial distribution are equivalent in this case:

$$p^{(\theta)}(x, \ell|\mathrm{Z}) \equiv p_+^{(\ell)}(x).$$

Gaussian Mixture Implementation

In a GM implementation of the LMB filter, the predicted spatial distribution of a track ℓ is a mixture of Gaussians:

$$p_+^{(\ell)}(x) = \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_+^{(\ell,j)}, \underline{\mathrm{P}}_+^{(\ell,j)}\right).$$

In order to simplify notations, the measurement matrix $\underline{\mathrm{H}}$ and the measurement noise $\underline{\mathrm{R}}$ are assumed to be constant. Similar to the GM implementations of the PHD, CPHD, and CB-MeMBer filters, the detection probability is assumed to be state independent, i.e.

$$p_D(x) = p_D. \tag{5.71}$$

The derivations of the proposed update equations in this subsection are given in Appendix B.3.

The likelihood for the assignment of measurement $z_{\theta(\ell)}$ to track ℓ is given by

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \frac{p_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right) \tag{5.72}$$

where the innovation covariance $\underline{\mathrm{S}}^{(\ell,j)}$ and the predicted measurement $z_+^{(\ell,j)}$ are calculated using the standard Kalman filter equations

$$\underline{\mathrm{S}}^{(\ell,j)} = \underline{\mathrm{H}}\underline{\mathrm{P}}_+^{(\ell,j)}\underline{\mathrm{H}}^{\mathrm{T}} + \underline{\mathrm{R}}, \tag{5.73}$$

$$z_+^{(\ell,j)} = \underline{\mathrm{H}}\hat{x}_+^{(\ell,j)} \tag{5.74}$$

and $\kappa(\cdot)$ is the intensity of the Poisson distributed false alarm process. The posterior spatial distribution for the update of track ℓ with measurement $z_{\theta(\ell)}$ is given by

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \sum_{j=1}^{J_+^{(\ell)}} w^{(\ell,j,\theta)}(\mathrm{Z}) \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right) \tag{5.75}$$

where the posterior weight of each Gaussian component j follows

$$w^{(\ell,j,\theta)}(\mathrm{Z}) = \frac{\frac{1}{\kappa(z_{\theta(\ell)})} p_D w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right)}{\eta_{\mathrm{Z}}^{(\theta)}(\ell)} \tag{5.76}$$

and the posterior mean value and estimation error covariance are obtained using the Kalman filter equations

$$\hat{x}^{(\ell,j,\theta)}(\mathrm{Z}) = \hat{x}_+^{(\ell,j)} + \underline{\mathrm{K}}^{(\ell,j)}\left(z_{\theta(\ell)} - z_+^{(\ell,j)}\right), \tag{5.77}$$

$$\underline{\mathrm{K}}^{(\ell,j)} = \underline{\mathrm{P}}_+^{(\ell,j)}\underline{\mathrm{H}}^{\mathrm{T}}\left[\underline{\mathrm{S}}^{(\ell,j)}\right]^{-1} \tag{5.78}$$

$$\underline{\mathrm{P}}^{(\ell,j)} = \underline{\mathrm{P}}_+^{(\ell,j)} - \underline{\mathrm{K}}^{(\ell,j)}\underline{\mathrm{S}}^{(\ell,j)}\left[\underline{\mathrm{K}}^{(\ell,j)}\right]^{\mathrm{T}}. \tag{5.79}$$

An application of the proposed measurement update equations to slightly non-linear motion models is possible using the EKF or the UKF implementation of the Kalman filter. The derivation of the EKF and the UKF measurement update is straight-forward and resembles the CB-MeMBer filter equations given by [Vo08, pp. 210ff.].

The likelihood for the association of track ℓ to a missed detection, i.e. $\theta(\ell) = 0$, corresponds to the missed detection probability:

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = q_D, \tag{5.80}$$

where $q_D = 1 - p_D$. Due to the state independent missed detection probability, the posterior spatial distribution of the track corresponds to the predicted spatial distribution and is given by

$$p^{(\theta)}(x, \ell|\mathrm{Z}) = \sum_{j=1}^{J_+^{(\ell)}} w^{(\ell,j,\theta)}(\mathrm{Z}) \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right), \tag{5.81}$$

where

$$w^{(\ell,j,\theta)}(\mathrm{Z}) = w_+^{(\ell,j)}, \tag{5.82}$$

$$\hat{x}^{(\ell,j,\theta)}(\mathrm{Z}) = \hat{x}_+^{(\ell,j)}, \tag{5.83}$$

$$\underline{\mathrm{P}}^{(\ell,j)} = \underline{\mathrm{P}}_+^{(\ell,j)}. \tag{5.84}$$

5.3.3 Gating and Grouping

Although the LMB prediction facilitates the direct application of the full δ-GLMB update, the calculation of the update step is only feasible in scenarios with a small number of objects. However, the LMB representation of the predicted multi-object density enables a partitioning of the LMB parameters into mutually exclusive subsets which are obtained using an additional partitioning of the set of measurements. The partitioning of the measurements is performed using a standard gating procedure. Using sufficient gate sizes during partitioning, the obtained groups of tracks and measurements have insignificant influence on each other and facilitate a parallel update. Observe that the proposed grouping procedure requires birth distributions with moderate spatial uncertainties to prevent huge partitions. Hence, uniform birth densities over the complete state space contradict with the proposed grouping procedure and lead to a complexity of the LMB filter update which is similar to the one of the δ-GLMB filter since all tracks are within one partition. In case of a uniform birth density, an adaptive birth density (see Section (5.3.6)) may be used to obtain a meaningful grouping result.

Gating

The objective of the gating procedure is to provide a set of feasible measurements $\mathrm{Z}^{(\ell)}$ for each track ℓ where a feasible measurement has a non-zero spatial likelihood $g(z|x^{(\ell)})$ for track ℓ. For notational convenience, the set of measurements $\mathrm{Z} = \{z_1, \dots, z_m\}$ is enumerated by $\mathbb{M} = \{1, \dots, m\}$ where $m = |\mathrm{Z}|$. The set $\mathbb{M}^{(\ell)} \subseteq \mathbb{M}$ represents the set of measurements $\mathrm{Z}^{(\ell)}$ which are located within the gate of track ℓ.

In Kalman filter implementations, the Mahalanobis distance (MHD) is commonly used for gating (see Section 2.4.1). Within the GM implementation of the LMB filter, the spatial distribution of each track is represented by a mixture of Gaussians. A conservative strategy is to calculate the minimum MHD

$$d^2_{\mathrm{MHD}}(\ell) \triangleq \min_{j \in J^{(\ell)}} \left[\left(z_{\theta(\ell)} - z_+^{(\ell,j)} \right)^{\mathrm{T}} \underline{\mathrm{S}}^{(\ell,j)} \left(z_{\theta(\ell)} - z_+^{(\ell,j)} \right) \right] \tag{5.85}$$

over all Gaussian components j of a track ℓ, where the innovation covariance $\underline{\mathrm{S}}^{(\ell,j)}$ and the predicted measurement $z_+^{(\ell,j)}$ are given by (5.73) and (5.74), respectively. Hence, a measurement $z_{\theta(\ell)}$ is assumed to be within the gate of track ℓ if the condition $d^2_{\mathrm{MHD}}(\ell) < \vartheta$ is met for any of the Gaussian components $J^{(\ell)}$. In scenarios with low detection probabilities, this conservative strategy may result in unnecessarily large gates since the gating procedure is dominated by the Gaussian components with low weights and huge spatial uncertainties which correspond to several successive missed detections. In such scenarios, adaptive gating procedures which incorporate the weight of the Gaussian components in the gating threshold [MD12] may be used to reduce the influence of unlikely Gaussian components to the gating result.

In an SMC implementation, the gating for a track to measurement association $\theta(\ell)$ is performed using the following inequality:

$$\sum_{j=1}^{\nu^{(\ell)}} w_+^{(\ell,j)} g\left(z_{\theta(\ell)} | x_+^{(\ell,j)} \right) > \vartheta_{\mathrm{SMC}} \tag{5.86}$$

where ϑ_{SMC} is the gating threshold. If (5.86) is met, the measurement $z_{\theta(\ell)}$ is assumed to be within the gate of track ℓ. An alternative solution is the approximation of the spatial distribution by a mixture of Gaussians using the expectation maximization (EM) algorithm [DLR77]. The obtained mixture of Gaussians facilitates the usage of the MHD based gating procedure (5.85) for SMC implementations.

Grouping

The purpose of the grouping procedure is to obtain groups of tracks which share at least one measurement. The principle of the grouping procedure is illustrated in Figure 5.8. Obviously, if no measurement is located within the gating distance of a track, a group comprising a single track and no measurement is obtained. Further, a measurement is not necessarily contained in any of the groups. Thus, measurements with a vanishing likelihood for all tracks are assigned to the set of not assigned measurements $\mathbb{M}^0$.

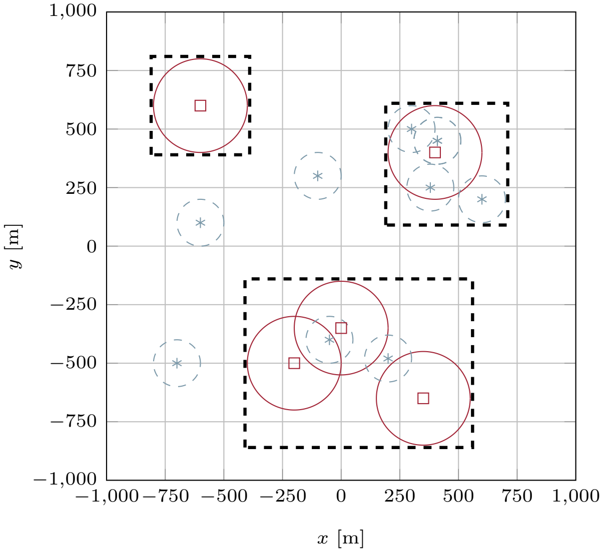


Figure 5.8: Grouping result (dashed rectangles) of the proposed partitioning scheme. Tracks are illustrated by red squares, measurements by blue stars.

The grouping procedure partitions the set of predicted track labels $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$ into N groups using

$$\mathbb{L}_+ = \bigcup_{n=1}^{N} \mathbb{L}_+^{(n)}.$$

A track label may only be contained in one of the partitions, i.e. $n \neq m$ implies $\mathbb{L}_+^{(n)} \cap \mathbb{L}_+^{(m)} = \emptyset$. The partitions $\mathbb{L}_+^{(n)}$ are obtained by an additional partitioning of the received measurements into the set of not assigned measurements $\mathbb{M}^0$ and the sets of measurements for the N partitions of track labels. The partitioning of the measurements is consequently given by

$$\mathbb{M} = \mathbb{M}^0 \cup \bigcup_{n=1}^{N} \mathbb{M}^{(n)}$$

where $n \neq m \geq 1$ implies $\mathbb{M}^{(n)} \cap \mathbb{M}^{(m)} = \emptyset$. Finally, a grouping $\mathcal{G}^{(n)}$ is defined by

$$\mathcal{G}^{(n)} = \left(\mathbb{L}_+^{(n)}, \mathbb{M}^{(n)}\right). \tag{5.87}$$

Obviously, there are several approaches to obtain the groupings $\mathcal{G}^{(n)}$. One solution is the clustering approach proposed in [DB93] which performs a clustering decomposition of the gating matrix in order to obtain subproblems with a smaller number of measurements and tracks. The approach used in this work is based on the sets of measurements within the gate of a track. Using the set representation significantly reduces the computational load in case of sparse gating matrices, since it only requires the calculation of the intersecting set of two measurement sets instead of comparing complete rows of the gating matrix.

The set based calculation of the groupings proceeds as follows: First, using the result of the gating in Section 5.3.3, a grouping for each LMB component ℓ is initialized using

$$\mathcal{G}^{(\ell)} = \left(\{\ell\}, \mathbb{M}^{(\ell)}\right) \tag{5.88}$$

where $\mathbb{M}^{(\ell)}$ is the set of measurements within the gate of track ℓ. Obviously, the obtained groupings do not fulfill the requirements, since the property $\mathbb{M}^{(n)} \cap \mathbb{M}^{(m)} = \emptyset$ is not ensured for all $n \neq m$. Thus, groupings which share at least one measurement have to be merged using

$$\mathcal{G}^{(i)} = \mathcal{G}^{(i)} \cup \left\{\mathcal{G}^{(j)} : \mathbb{M}^{(i)} \cap \mathbb{M}^{(j)} \neq \emptyset\right\}, \tag{5.89}$$

where the merging of two groupings i and j is defined by

$$\mathcal{G}^{(i)} \cup \mathcal{G}^{(j)} = \left(\mathbb{L}_+^{(i)} \cup \mathbb{L}_+^{(j)}, \mathbb{M}^{(i)} \cup \mathbb{M}^{(j)}\right). \tag{5.90}$$

After merging a grouping $\mathcal{G}^{(j)}$ with grouping $\mathcal{G}^{(i)}$, the grouping $\mathcal{G}^{(j)}$ is deleted. The merging procedure terminates if the sets of measurements for all groupings are disjoint.

Using the resulting number of N groupings, the predicted LMB distribution may also be partitioned into

$$\pi_+ = \bigcup_{i=1}^{N} \left\{ \left(r_{+,i}^{(\ell)}, p_{+,i}^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}_+^{(i)}} \tag{5.91}$$

where the subsets or groups of Bernoulli tracks are abbreviated by

$$\pi_+^{(i)} = \left\{ \left(r_{+,i}^{(\ell)}, p_{+,i}^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}_+^{(i)}} \tag{5.92}$$

in the following.

5.3.4 Group Updates

δ-GLMB Representation of the Predicted LMB RFS

The update of the LMB filter requires the representation of the predicted LMB density in δ-GLMB form. Thus, the predicted LMB density (5.92) of each group i has to be rewritten in the corresponding δ-GLMB form (5.47) in order to allow for the application of the δ-GLMB update. For a grouping $\mathcal{G}^{(i)}$ with the corresponding set of track labels $\mathbb{L}_+^{(i)}$, the predicted δ-GLMB density is given by

$$\boldsymbol{\pi}_+^{(i)}(\widetilde{\mathbf{X}}_+^{(i)}) = \Delta(\widetilde{\mathbf{X}}_+^{(i)}) \sum_{I_+ \in \mathcal{F}(\mathbb{L}_+^{(i)})} w_{+,i}^{(I_+)} \delta_{I_+}(\mathcal{L}(\widetilde{\mathbf{X}}_+^{(i)})) \left[p_+ \right]^{\widetilde{\mathbf{X}}_+^{(i)}} \tag{5.93}$$

where the weight of each set of track labels I_+ follows

$$w_{+,i}^{(I_+)} = \prod_{j \in \mathbb{L}_+^{(i)}} \left(1 - r_+^{(j)} \right) \prod_{\ell \in I_+} \frac{1_{\mathbb{L}_+^{(i)}}(\ell) r_+^{(\ell)}}{1 - r_+^{(\ell)}}$$

and the multi-object state $\widetilde{\mathbf{X}}_+^{(i)}$ is defined by

$$\widetilde{\mathbf{X}}_+^{(i)} = \left\{ \boldsymbol{x} : \mathcal{L}(\boldsymbol{x}) \in \mathbb{L}_+^{(i)} \ \forall \ \boldsymbol{x} \in \mathbf{X} \right\}.$$

The conversion of an LMB to a δ-GLMB distribution facilitates a straightforward parallelization since the parameters of the predicted δ-GLMB density for each grouping i may be calculated independently.

Section 5.2.3 already showed an example of representing an LMB RFS in δ-GLMB form. In this example, the hypotheses of the δ-GLMB distribution are enumerated in a brute-force way, i.e. a hypothesis is generated for all possible combinations of the predicted labels $\mathbb{L}_+^{(i)}$ and at least one hypothesis is generated for each cardinality $n = 0, 1, \ldots, |\mathbb{L}_+^{(i)}|$. Obviously, the brute-force approach is only feasible for small $|\mathbb{L}_+^{(i)}|$ since the number of hypotheses per cardinality n follows the binomial coefficient

$$C_n^{|\mathbb{L}_+^{(i)}|} = \frac{|\mathbb{L}_+^{(i)}|!}{n!(|\mathbb{L}_+^{(i)}| - n)!}$$

and the total number of hypotheses is given by $2^{|\mathbb{L}_+^{(i)}|}$. Since the number of hypotheses grows exponentially in the number of track labels, the group based conversion significantly reduces the number of hypotheses required for the δ-GLMB representation.

Although the grouping procedure in Section 5.3.3 is intended to obtain groups with a small number of track labels $|\mathbb{L}_+^{(i)}|$, it does not ensure tractable group sizes. Experimental results showed that the brute-force conversion is only suitable for groups with $|\mathbb{L}_+^{(i)}| \lesssim 10$. However, the threshold is highly dependent on the compute capabilities and the number of groups. Hence, an approximate δ-GLMB representation is required to ensure that a maximum number of hypotheses is not exceeded during conversion. An intuitive approximation is a truncation of the δ-GLMB distribution which preserves the hypotheses I_+ with the highest weights $w_+^{(i)}(I_+)$. The obtained approximation error of the truncation is negligible if the number of hypotheses is sufficient to represent all hypotheses with significant weights. The truncation may be realized using the k-shortest path algorithm [Epp98; Yen71]. An alternative to the approximation using the k-shortest path algorithm is to sample the desired amount of hypotheses. Algorithm 5.1 illustrates the sampling of a predicted hypothesis I_+. Obviously, the algorithm has to be repeated until the desired amount of unique hypotheses is obtained. A subsequent validation of the nth sampled hypothesis using

$$I_+^{(n)} \neq I_+^{(j)} \; \forall \; j < n$$

is required since the sampling procedure does not ensure unique hypotheses.

Algorithm 5.1 Hypothesis Sampling

```
I_+ = ∅;
for ℓ ∈ L_+^(i) do
    sample u ~ U(0,1);
    if u ≤ r_+^(ℓ) then
        I_+ = I_+ ∪ {ℓ};
    end if
end for
```

δ-GLMB Update

The predicted δ-GLMB distributions $\boldsymbol{\pi}_+^{(i)}(\widetilde{\mathbf{X}}_+^{(i)}),\ i = 1, \ldots, N$, facilitate the parallel computation of the update step for each grouping i. Using the predicted δ-GLMB RFSs (5.93), the posterior δ-GLMB RFS of each grouping i is obtained using the δ-GLMB update equation (5.48):

$$\boldsymbol{\pi}^{(i)}(\widetilde{\mathbf{X}}^{(i)}|\mathrm{Z}^{(i)}) = \Delta(\widetilde{\mathbf{X}}^{(i)}) \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+^{(i)})\times\Theta_{I_+}^{(i)}} w^{(I_+,\theta)}(\mathrm{Z}^{(i)})\delta_{I_+}(\mathcal{L}(\widetilde{\mathbf{X}}^{(i)})) \left[p^{(\theta)}(\cdot|\mathrm{Z}^{(i)})\right]^{\widetilde{\mathbf{X}}^{(i)}}. \tag{5.94}$$

Due to the grouping procedure, only the subset $\mathrm{Z}^{(i)}$ of measurements has to be considered for the update of $\widetilde{\mathbf{X}}^{(i)}$. Hence, the track to measurement associations are given by the mapping $\theta : \mathbb{L}_+^{(i)} \to \{0\} \cup \mathbb{M}^{(i)}$ where $\theta(i) = \theta(j) > 0$ implies $i = j$. The posterior weight for the update of a set of track labels I_+ using an association map θ is given by

$$w^{(I_+,\theta)}(\mathrm{Z}^{(i)}) = \frac{\delta_{\theta^{-1}(\{0\cup\mathbb{M}^{(i)}\})}(I_+)w_+^{(i)}(I_+)[\eta_{\mathrm{Z}^{(i)}}^{(\theta)}]^{I_+}}{\sum\limits_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+^{(i)})\times\Theta_{I_+}^{(i)}} \delta_{\theta^{-1}(\{0\cup\mathbb{M}^{(i)}\})}(I_+)w_+^{(i)}(I_+)[\eta_{\mathrm{Z}^{(i)}}^{(\theta)}]^{I_+}}. \tag{5.95}$$

Since a mapping θ only contains track to measurement associations for the current set of measurements $\mathrm{Z}^{(i)}$, the association likelihood and the measurement updated spatial distributions may be equivalently rewritten using the entire set of measurements Z:

$$\eta_{\mathrm{Z}^{(i)}}^{(\theta)}(\ell) = \eta_{\mathrm{Z}}^{(\theta)}(\ell),$$
$$p^{(\theta)}(x,\ell|\mathrm{Z}^{(i)}) = p^{(\theta)}(x,\ell|\mathrm{Z}).$$

Thus, the likelihood and the updated spatial distributions for given associations are already given by the results in Section 5.3.2.

In order to obtain the posterior existence probability (5.41) and the posterior spatial distribution (5.42) of each LMB track, the weights $w^{(I_+,\theta)}$ of the hypotheses have to be evaluated. The weight of all possible association hypotheses may be calculated using an association tree (see Figure 4.1 in Section 4.1.2). Following the discussion in Section 4.1.3, the complexity of the evaluation of all association hypotheses grows exponentially and an evaluation of all hypotheses is only possible for small sets of track labels I_+ and a small number of measurements $|\mathrm{Z}^{(i)}|$. Thus, the complexity of the calculation directly depends on the result of the grouping procedure.

In [VV13b], the usage of Murty's algorithm [Mur68] is proposed to truncate the posterior distribution (5.94). Murty's algorithm facilitates the evaluation of the $k^{(i)}$ most significant hypotheses for grouping i without calculating the weights of all possible hypotheses. Murty's algorithm starts with finding the best association map θ for a set of track labels I_+ using one of the optimal assignment algorithms introduced in 2.4.2, e.g. the Hungarian method [Kuh55]. Based on the best association map, additional association maps are obtained by removing a track to measurement association of the best association map as an assignment possibility. A detailed example of Murty's algorithm is given in [BP99, p. 346ff.]. The complexity of Murty's algorithms is only cubic [MSC97] which is significantly smaller than the complexity for evaluating the entire association tree.

Similar to the implementation of the δ-GLMB filter in [VV13b], the number of evaluated association maps $k^{(I_+)}$ per set of track labels I_+ is chosen to be proportional to the predicted weight $w_{+,i}^{(I_+)}$:

$$k^{(I_+)} = \left\lceil w_{+,i}^{(I_+)} \cdot k^{(i)} \right\rceil \tag{5.96}$$

where $k^{(i)}$ is the total number of calculated hypotheses for group i. The cost matrix for a predicted set $I_+ \in \mathcal{F}(\mathbb{L}_+^{(i)})$ containing n track labels and $m = |\mathrm{Z}^{(i)}|$ measurements is given

$$C_{\mathrm{Z}^{(i)}}^{(I_+)} = \begin{pmatrix} \widetilde{\eta}_{\mathrm{Z}}^{(1)}(\ell_1) & \widetilde{\eta}_{\mathrm{Z}}^{(2)}(\ell_1) & \cdots & \widetilde{\eta}_{\mathrm{Z}}^{(m)}(\ell_1) & \widetilde{\eta}_{\mathrm{Z}}^{(0)}(\ell_1) & \cdots & \widetilde{\eta}_{\mathrm{Z}}^{(0)}(\ell_1) \\ \widetilde{\eta}_{\mathrm{Z}}^{(1)}(\ell_2) & \widetilde{\eta}_{\mathrm{Z}}^{(2)}(\ell_2) & \cdots & \widetilde{\eta}_{\mathrm{Z}}^{(m)}(\ell_2) & \widetilde{\eta}_{\mathrm{Z}}^{(0)}(\ell_2) & \cdots & \widetilde{\eta}_{\mathrm{Z}}^{(0)}(\ell_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{\eta}_{\mathrm{Z}}^{(1)}(\ell_n) & \widetilde{\eta}_{\mathrm{Z}}^{(2)}(\ell_n) & \cdots & \widetilde{\eta}_{\mathrm{Z}}^{(m)}(\ell_n) & \widetilde{\eta}_{\mathrm{Z}}^{(0)}(\ell_n) & \cdots & \widetilde{\eta}_{\mathrm{Z}}^{(0)}(\ell_n) \end{pmatrix}, \tag{5.97}$$

where

$$\widetilde{\eta}_{\mathrm{Z}}^{(i)}(\ell_n) = -\log\left(w_+^{(i)}(I_+) \cdot \eta_{\mathrm{Z}}^{(\theta(\ell_n)=i)}(\ell_n)\right).$$

Observe that the cost matrix has $n = I_+$ rows (one for each track label) and $m + n$ columns, where the first m columns represent the association of a track label ℓ_n to a measurement $j = 1, \ldots, m$ whereas the additional n columns are required to be able to assign each track label to a missed detection.

Using the cost matrix and Murty's algorithm, the cost of the lth best track label to measurement assignment θ^l is given by

$$c_{\mathrm{Z}^{(i)}}^{(I_+)}(\theta^l) = \sum_{i=1}^{|I_+|} C_{\mathrm{Z}^{(i)}}^{(I_+)}\left(i, \theta^l(\ell_i)\right) \tag{5.98}$$

which corresponds to the hypothesis with the lth best weight

$$\widetilde{w}^{(I_+,\theta^l)}(\mathrm{Z}^{(i)}) = \exp\left(-c_{\mathrm{Z}^{(i)}}^{(I_+)}(\theta^l)\right). \tag{5.99}$$

The posterior weight of an association hypothesis θ^l is finally obtained by normalizing (5.99) using the weight of the k best associations:

$$w^{(I_+,\theta^l)}(\mathrm{Z}^{(i)}) = \frac{\widetilde{w}^{(I_+,\theta^l)}(\mathrm{Z}^{(i)})}{\sum\limits_{I_+\in\mathcal{F}(\mathbb{L}_+^{(i)})}\sum\limits_{j=1}^{k^{(I_+)}}\widetilde{w}^{(I_+,\theta^j)}(\mathrm{Z}^{(i)})}. \tag{5.100}$$

Labeled Multi-Bernoulli Approximation

As introduced in Section 5.2.2, the measurement updated δ-GLMB distributions have to be approximated by LMB RFSs in order to allow for the LMB recursion. Thus, the posterior δ-GLMB RFSs $\boldsymbol{\pi}^{(i)}(\cdot|\mathrm{Z}^{(i)})$ of all groups $\mathcal{G}^{(i)}$, $i = 1, ..., N$, are approximated by

$$\boldsymbol{\pi}^{(i)}(\cdot|\mathrm{Z}^{(i)}) \approx \widetilde{\boldsymbol{\pi}}^{(i)}(\cdot|\mathrm{Z}^{(i)}) = \left\{\left(r^{(\ell)}, p^{(\ell)}\right)\right\}_{\ell\in\mathbb{L}_+^{(i)}}.$$

The existence probability $r^{(\ell)}$ and the spatial distribution $p^{(\ell)}$ of each track ℓ are calculated according to (5.41) and (5.42). As discussed in Section 5.2.2, the LMB approximation matches the δ-GLMB RFS with respect to the unlabeled PHD which incorporates a matching mean cardinality.

Finally, the LMB approximation of the full multi-object posterior is given by the union of the LMB approximations $\widetilde{\boldsymbol{\pi}}^{(i)}$ of the N groups:

$$\boldsymbol{\pi}(\cdot|\mathrm{Z}) \approx \widetilde{\boldsymbol{\pi}}(\cdot|\mathrm{Z}) = \bigcup_{i=1}^{N}\left\{\left(r^{(\ell)}, p^{(\ell)}\right)\right\}_{\ell\in\mathbb{L}_+^{(i)}}. \tag{5.101}$$

Grouping Error

Since an LMB RFS is defined as the union of M independent labeled Bernoulli RFSs, the predicted LMB RFS can be equivalently rewritten by

$$\boldsymbol{\pi}_+(\mathbf{X}_+) = \boldsymbol{\pi}_+^{(1)}\left(\widetilde{\mathbf{X}}_+^{(1)}\right)\cdots\boldsymbol{\pi}_+^{(N)}\left(\widetilde{\mathbf{X}}_+^{(N)}\right). \tag{5.102}$$

For sufficiently large gating values, the single object measurement likelihood for a track $x \in \widetilde{\mathbf{X}}_+^{(i)}$ of group $\mathcal{G}^{(i)}$ and a measurement of any other group $\mathcal{G}^{(j)},\ j \neq i$, is negligible:

$$g(z_{\theta(\ell)}|x,\ell) \approx 0 \;\forall\; z_{\theta(\ell)} \notin \mathrm{Z}^{(i)},\ x \in \widetilde{\mathbf{X}}_+^{(i)}. \tag{5.103}$$

Additionally, a measurement of group $\mathcal{G}^{(i)}$ has an insignificant likelihood for all tracks of a grouping $\mathcal{G}^{(j)},\ j \neq i$:

$$g(z_{\theta(\ell)}|x,\ell) \approx 0 \;\forall\; z_{\theta(\ell)} \in \mathrm{Z}^{(i)},\ x \notin \widetilde{\mathbf{X}}_+^{(i)} \tag{5.104}$$

Under the assumptions (5.103) and (5.104), an approximate decomposition of the multi-object likelihood function is given by

$$g(\mathrm{Z}|\mathbf{X}) \approx g\left(\mathrm{Z}^{(1)} \mid \widetilde{\mathbf{X}}^{(1)}\right) \cdots g\left(\mathrm{Z}^{(N)} \mid \widetilde{\mathbf{X}}^{(N)}\right). \tag{5.105}$$

Using the alternative representation of the predicted LMB density (5.102) and the decomposed multi-object likelihood function (5.105), the multi-object posterior is given by

$$\begin{aligned}
\boldsymbol{\pi}(\mathbf{X}|\mathrm{Z}) &\propto \boldsymbol{\pi}_+(\mathbf{X}) g(\mathrm{Z}|\mathbf{X}) && (5.106)\\
&\approx \boldsymbol{\pi}_+^{(1)}\left(\widetilde{\mathbf{X}}^{(1)}\right) g\left(\mathrm{Z}^{(1)} \mid \widetilde{\mathbf{X}}^{(1)}\right) \cdots \boldsymbol{\pi}_+^{(N)}\left(\widetilde{\mathbf{X}}^{(N)}\right) g\left(\mathrm{Z}^{(N)} \mid \widetilde{\mathbf{X}}^{(N)}\right) && (5.107)\\
&= \boldsymbol{\pi}^{(1)}\left(\widetilde{\mathbf{X}}^{(1)} \mid \mathrm{Z}^{(1)}\right) \cdots \boldsymbol{\pi}^{(N)}\left(\widetilde{\mathbf{X}}^{(N)} \mid \mathrm{Z}^{(N)}\right). && (5.108)
\end{aligned}$$

Consequently, sufficiently large gating values in the grouping procedure ensure that the error due to the parallel group updates is negligible.

5.3.5 Track Extraction

Track extraction algorithms for RFS based multi-object filters commonly use the maximum a posteriori (MAP) estimate of the cardinality distribution to obtain the estimated number of objects $\hat{N}$. Hence, a straightforward track extraction procedure for a δ-GLMB filter is to pick the hypothesis (I, ξ) with the highest weight for the cardinality $\hat{N}$. Since the LMB filter does not provide a single δ-GLMB posterior, this procedure is only applicable to the posterior δ-GLMB distributions of the groupings $\boldsymbol{\pi}^{(i)}(\widetilde{\mathbf{X}}^{(i)}|\mathrm{Z}^{(i)})$.

In an LMB filter, the track extraction is not restricted to the approach based on the MAP estimate of the cardinality distribution. Similar to the track extraction for the JIPDA tracker in [Mun11], the posterior LMB distribution enables an intuitive and straightforward track extraction scheme which selects all tracks whose existence probability exceeds an application dependent threshold ϑ. Hence, the set of extracted tracks is given by

$$\hat{\mathbf{X}} = \{(\hat{x}, \ell)|r^{(\ell)} > \vartheta\}.$$

The application dependent threshold ϑ allows to adapt the output of the tracking system to the requirements of arbitrary applications. A high value for ϑ increases the delay for including a new born track in the output of the tracking system and decreases the number of false tracks. Smaller values for ϑ decrease the output delay for new born tracks in conjunction with an increasing number of false tracks.

The difference between cardinality based and existence based track extraction is illustrated using the example in Section 5.2.3. The MAP estimate for the cardinality in the example is $\hat{N} = 1$. A basic approach just picks the hypothesis with the highest weight for the estimated cardinality. Thus, only track ℓ_2 is extracted and reported to an operator or a higher level fusion system. Obviously, the cardinality based track extraction does not allow to adapt the output of the tracking system to the application dependent requirements. Using the existence based track extraction, three different outputs are possible:

$$\hat{\mathbf{X}} = \begin{cases} (\hat{x}_1, \ell_1) \cup (\hat{x}_2, \ell_2) & \text{if } \vartheta < 0.6, \\ (\hat{x}_2, \ell_2) & \text{if } 0.6 \leq \vartheta < 0.7, \\ \emptyset & \text{if } \vartheta \geq 0.7. \end{cases}$$

In applications with high detection probabilities, a single missed detection significantly reduces the existence probability of an object. In combination with a high value for ϑ, the missed detection may lead to a temporary rejection of the track by the track extraction algorithm. In order to obtain a continuous output of the tracks in these situations, a hysteresis is used within the track extraction. The output for a track ℓ_i is only activated, if its maximum existence probability $r_{\max}^{(\ell_i)}$ exceeds an upper threshold ϑ_u and its current existence probability $r^{(\ell_i)}$ is not below a lower threshold ϑ_l:

$$\hat{\mathbf{X}} = \left\{(\hat{x}, \ell)|r_{max}^{(\ell)} > \vartheta_u \wedge r^{(\ell)} > \vartheta_l\right\} \tag{5.109}$$

5.3.6 Adaptive Birth Distribution

In Section 5.3.1, the prediction step of the LMB filter assumed the availability of a priori distributions for new born objects. In applications like air surveillance, where new objects may only appear close to airports or at the borders of the FOV of the sensor, sufficient knowledge about the birth distribution is available. In applications like vehicular environment perception, the spatial distribution of the birth locations is usually unknown and also time varying. In GM implementations of the PHD and the CPHD filter, a common approach is to use uniform or widespread birth densities [BVVA12]. In the context of the proposed LMB filter, these densities contradict with the group oriented updates, since the large spatial uncertainties prohibit a partitioning into statistically independent groups.

In order to utilize the computational savings of group oriented updates in applications without accurate a priori information about possible birth locations, a measurement driven birth density is proposed. The measurement driven birth density is based on the concept of the adaptive birth intensities for the PHD and CPHD filters [RCVV12]. In contrast to the adaptive birth intensity which consists of a single intensity function, the adaptive birth distribution for the LMB filter is required to follow an LMB process which prevents a straightforward adaptation of [RCVV12].

The adaptive LMB birth distribution $\boldsymbol{\pi}_{B,k+1}$ used within the prediction to time $k+1$ is obtained using the set of measurements Z_k at time k. For each measurement, a Bernoulli birth component is initialized using the adaptive LMB birth distribution

$$\boldsymbol{\pi}_{B,k+1} = \left\{ r_{B,k+1}^{(\alpha_i)}(z_i), p_{B,k+1}^{(\alpha_i)}(x|z_i) \right\}_{i=1}^{|\mathrm{Z}_k|},$$

where the labels α_i are obtained from $\mathbb{B} = \{\alpha_i : i \in \{1, \ldots, |\mathrm{Z}_k|\}\}$. The spatial distribution $p_{B,k+1}^{(\alpha_i)}(x|z_i)$ of each Bernoulli component is obtained by transforming the measurement z_k and its according measurement noise into the state space $\mathbb{X}$. In [RCVV12], the adaptive birth intensity for the PHD and CPHD filter concentrates around the measurements which are far away from any of the existing tracks. In case of an LMB birth density, the existence probabilities $r_{B,k+1}^{(\ell)}(z_k)$ of the birth components have to represent this behavior.

For each grouping $\mathcal{G}^{(i)}$, a measurement $z \in \mathrm{Z}^{(i)}$ is assigned to an existing track with probability

$$r_{U,k}(z) = \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+^{(i)})\times\Theta_{I_+}^{(i)}} 1_\theta(z) w_k^{(I_+,\theta)}(\mathrm{Z}^{(i)}), \tag{5.110}$$

where $w_k^{(I_+,\theta)}(\mathrm{Z}^{(i)})$ is given by (5.95) and the inclusion function $1_\theta(z)$ ensures that the sum only incorporates the weight of the hypotheses which assigned the measurement z to any of the existing tracks. All measurements $z \in \mathrm{Z}^{(0)}$ are neglected in the track update due to the large distance to any of the tracks. Consequently, an association probability of

$$r_{U,k}(z) = 0 \tag{5.111}$$

is assigned to the measurements $z \in \mathrm{Z}^{(0)}$.

To concentrate the birth density around the measurements which are far away from any of the existing tracks, the existence probability of the birth component corresponding to measurement z has to be proportional to the probability that z is not assigned to any of the existing tracks. Hence, the existence probability of a birth component is given by

$$r_{B,k+1}(z) = \min\left(r_{B,k+1}^{\max} \, , \; \frac{1 - r_{U,k}(z)}{\sum\limits_{\xi \in \mathrm{Z}_k} 1 - r_{U,k}(\xi)} \cdot \lambda_B \right), \tag{5.112}$$

where λ_B is the expected number of new born objects. In case of $\lambda_B > 1$, the second term may be larger than one (e.g. if the birth density concentrates around a single measurement). Thus, it is necessary to apply the minimum operator to restrict the existence probabilities to $r_B^{\max} \in [0, 1]$. Further, using lower values for $r_B^{\max}$ reduces the number of false tracks at the cost of an increasing time until a track is confirmed. Due to the capping using $r_B^{\max}$, the expected number of new born objects is given by

$$\sum_{z \in \mathrm{Z}_k} r_{B,k+1}(z) \leq \lambda_B .$$

Obviously, equality applies if $r_{B,k+1}^{\max}$ is not used for any of the measurements $z \in \mathrm{Z}_k$.

The proposed adaptive birth density is not restricted to the application within the LMB filter. Since the calculation of the existence probabilities is based on the weights of the δ-GLMB update, the proposed LMB birth density may also be applied to the δ-GLMB filter. Further, the approach also applies to the CB-MeMBer filter which calculates $r_{U,k}(z)$ within the filter update by default. Simulation results for the CB-MeMBer filter with adaptive birth density are presented in the authors conference paper [RMWD13].

5.4 Multi-Object Divergence Detectors

5.4.1 Multi-Object Generalized NIS

In Section 3.8.3, the multi-object generalized normalized innovation squared (MGNIS) was introduced as a multi-object divergence detector. Since the update equation of the δ-GLMB filter provides an analytic solution for the normalization constant of the multi-object Bayes filter, the calculation of the MGNIS for the δ-GLMB filter is possible. Using the update equation (5.12), the normalization constant is given by the denominator of the updated weights (5.16). Within the derivation of the update equations in [VV13b], the clutter factor

$$\pi_C(\mathrm{Z}) = e^{-\lambda_c} \prod_{j=1}^{m} \kappa(z) \tag{5.113}$$

canceled out due to appearance in nominator and denominator. Thus, the normalization constant of the δ-GLMB filter is given by

$$f(\mathrm{Z}) = \pi_C(\mathrm{Z}) \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum_{\theta\in\Theta} \delta_{\theta^{-1}(\{0:|\mathrm{Z}|\})}(I) w_+^{(I,\xi)} [\eta_{\mathrm{Z}}^{(\xi,\theta)}]^I. \tag{5.114}$$

The MGNIS of the δ-GLMB filter is obtained by inserting (5.114) into (3.81):

$$\begin{aligned}
\mathrm{MGNIS}(\mathrm{Z}) &= -2\log(f(\mathrm{Z})) \\
&= -2\log \pi_C(\mathrm{Z}) \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum_{\theta\in\Theta} \delta_{\theta^{-1}(\{0:|\mathrm{Z}|\})}(I) w_+^{(I,\xi)} [\eta_{\mathrm{Z}}^{(\xi,\theta)}]^I \qquad (5.115) \\
&= -2\log \pi_C(\mathrm{Z}) - 2\log \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum_{\theta\in\Theta} \delta_{\theta^{-1}(\{0:|Z|\})}(I) w_+^{(I,\xi)} [\eta_{\mathrm{Z}}^{(\xi,\theta)}]^I, \qquad (5.116)
\end{aligned}$$

where each summand of the inner sum represents the contribution of a track to measurement association for a given set of track labels I. Although the MGNIS incorporates the target detection and the clutter process, (5.116) allows for a separation into a clutter part $\mathrm{MGNIS}_C(Z)$ and a object detection part $\mathrm{MGNIS}_T(Z)$ which are given by

$$\mathrm{MGNIS}_C(\mathrm{Z}) = -2\log \pi_C(\mathrm{Z}), \tag{5.117}$$

$$\mathrm{MGNIS}_T(\mathrm{Z}) = -2\log \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \sum_{\theta\in\Theta} \delta_{\theta^{-1}(\{0:|Z|\})}(I) w_+^{(I,\xi)} [\eta_{\mathrm{Z}}^{(\xi,\theta)}]^I. \tag{5.118}$$

Observe, that $\text{MGNIS}_T(Z)$ involves a dependency on clutter due to the appearance of $\kappa(z)$ in $\eta_{\text{Z}}^{(\xi,\theta)}(\ell)$ given by (5.14). Further, the $\text{MGNIS}_C(\text{Z})$ is calculated using all measurements and the number of object detections consequently impacts the $\text{MGNIS}_C(\text{Z})$.

The properties of the MGNIS are illustrated in Figure 5.9 for a scenario with a single object in clutter. At each time step, only one hypothesis (I, ξ) with $I = \{\ell_1\}$ exists, i.e. the existence probability of the track with label ℓ_1 is $r^{(\ell_1)} = 1$. Further, the detection probability is chosen to $p_D = 0.98$. The object is detected at every time step k with differing Mahalanobis distances between the predicted and the actual measurement (see Figure 5.9a). Additionally, a mean number of $\lambda_c = 10$ Poisson distributed clutter measurements with $c(z) = 0.001$ is received at every time step. Using only the detection dependent $\text{MGNIS}_T(\text{Z})$ in Figure 5.9b, the filter divergence for $50 < k < 70$ is easily detected. Due to the high variance of the Poisson distributed clutter ($\sigma^2 = \lambda_c$) and $\kappa \ll 1$, the clutter contribution $\text{MGNIS}_C(\text{Z})$ has an even higher variance (see Figure 5.9c). Figure 5.9d indicates that the clutter dominates the MGNIS since it is not possible to observe that the measurements for the tracked objects are diverging for $50 < k < 70$.

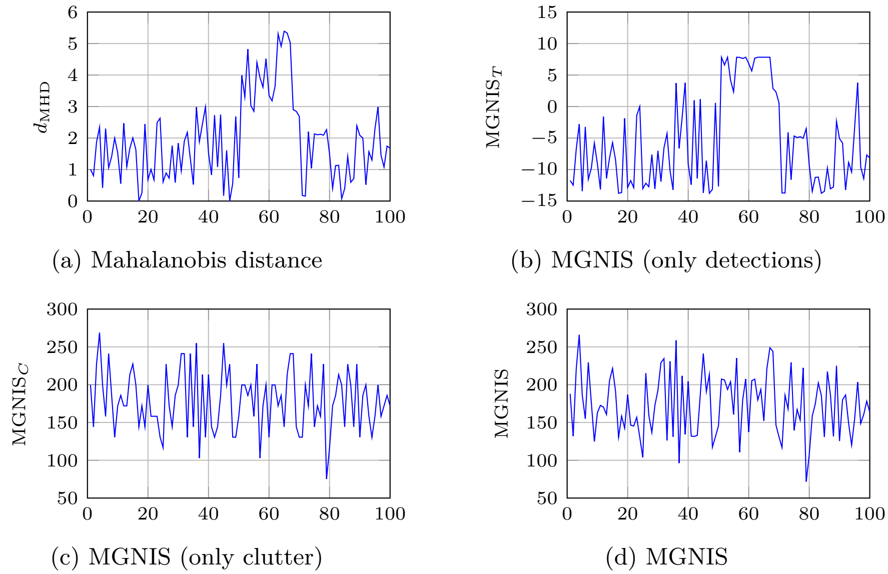


Figure 5.9: MGNIS result for different Mahalanobis distances between the predicted and actual measurement of a single object. Additionally, Poisson distributed clutter with $\lambda_c = 10$ occur and $p_D = 0.98$. The x-axis denotes the time steps k.

The LMB filter facilitates the evaluation of the MGNIS for each group of objects due to the usage of the δ-GLMB update equations. However, the determination of a single MGNIS value over all groups of objects may be desirable in some applications. Following (5.108), the multi-object posterior is given by the product of the multi-object posteriors of each group:

$$\boldsymbol{\pi}(\mathbf{X}|\mathrm{Z}) \approx \boldsymbol{\pi}^{(1)}\left(\widetilde{\mathbf{X}}^{(1)} \mid \mathrm{Z}^{(1)}\right) \cdots \boldsymbol{\pi}^{(N)}\left(\widetilde{\mathbf{X}}^{(N)} \mid \mathrm{Z}^{(N)}\right). \tag{5.119}$$

Consequently, the normalization constant of the LMB filter is given by the product of the normalization constants of the N groups:

$$f(\mathrm{Z}) = \pi_C(\mathrm{Z}) \cdot f^{(1)}(\mathrm{Z}^{(1)}) \cdots f^{(N)}(\mathrm{Z}^{(N)}). \tag{5.120}$$

where

$$f^{(i)}(\mathrm{Z}^{(i)}) = \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+^{(i)})\times\Theta_{I_+}^{(i)}} \delta_{\theta^{-1}(\{0\cup\mathbb{M}^{(i)}\})}(I_+) w_+^{(i)}(I_+)[\eta_{\mathrm{Z}^{(i)}}^{(\theta)}]^{I_+} \tag{5.121}$$

is the normalization constant of (5.95). Thus, the MGNIS of the LMB filter follows

$$\mathrm{MGNIS}(\mathrm{Z}) = \mathrm{MGNIS}_C(\mathrm{Z}) + \sum_{i=1}^{N} \mathrm{MGNIS}_T^{(i)}(\mathrm{Z}^{(i)}). \tag{5.122}$$

5.4.2 Approximate Multi-Target NIS

The approximate multi-target normalized innovation squared (AMNIS) is intended to be the multi-object analog of the single-object NIS with similar characteristics. Since an influence of the clutter process is not desired, the AMNIS is based on the detection part MGNIS_T of the MGNIS. In single-object tracking, the NIS approximates the GNIS by neglecting the magnitude of the innovation covariance matrix. Consequently, the NIS only penalizes the residuals between the measurement z and the predicted measurement z_+. Due to the approximation, the NIS follows a χ^2 distribution which facilitates an intuitive interpretation of the obtained NIS value.

In the following, only the AMNIS for the LMB filter is considered. In order to achieve similar characteristics to the NIS, the likelihood (5.72), which is given by

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \frac{p_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right) \tag{5.123}$$

for $\theta(\ell) > 0$, is approximated by

$$\bar{\eta}_{\mathrm{Z}}^{(\theta)}(\ell) = p_D \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \exp\left(\left(z_{\theta(\ell)} - z_+^{(\ell,j)}\right)^{\mathrm{T}} \left(\underline{\mathrm{S}}^{(\ell,j)}\right)^{-1} \left(z_{\theta(\ell)} - z_+^{(\ell,j)}\right)\right) \quad (5.124)$$

where the influence of the clutter intensity as well as the influence of the magnitude of the innovation covariance are neglected. For $\theta(\ell) = 0$, the likelihood is equivalent to (5.80):

$$\bar{\eta}_{\mathrm{Z}}^{(\theta)}(\ell) = 1 - p_D. \quad (5.125)$$

Using the approximate likelihoods (5.124) and (5.125), the AMNIS for a group $\mathcal{G}^{(i)}$ of the LMB filter is given by

$$\mathrm{AMNIS}^{(i)}(\mathrm{Z}^{(i)}) = -2\log \sum_{(I_+,\theta)\in\mathcal{F}(\mathbb{L}_+^{(i)})\times\Theta_{I_+}^{(i)}} \delta_{\theta^{-1}(\{0\cup\mathbb{M}^{(i)}\})}(I_+) w_+^{(i)}(I_+)[\bar{\eta}_{\mathrm{Z}^{(i)}}^{(\theta)}]^{I_+}, \quad (5.126)$$

while the AMNIS corresponding to the full multi-object distribution is obtained by

$$\mathrm{AMNIS}(\mathrm{Z}) = \sum_{i=1}^{N} \mathrm{AMNIS}^{(i)}(\mathrm{Z}^{(i)}). \quad (5.127)$$

5.4.3 Simplification of the AMNIS for High SNR Cases

The AMNIS follows a χ^2 distribution and reveals a behavior similar to the NIS in case of a high signal to noise ratio (SNR) which is characterized by

- the AMNIS(Z) is dominated by a single association hypothesis,
- the detection probability is $p_D \approx 1$ and all targets are detected.

The AMNIS is obviously dominated by a single association hypothesis (I_+, θ) if the objects are well separated and only a single measurement has non-zero likelihood for each track. In this case, the AMNIS simplifies to

$$\mathrm{AMNIS}^{(i)}(\mathrm{Z}^{(i)}) = -2\log[\bar{\eta}_{\mathrm{Z}^{(i)}}^{(\theta)}]^{I_+}, \quad (5.128)$$

Due to the high SNR, each track ℓ is usually represented by a single Gaussian component and all objects are detected. Using the assumption $p_D \approx 1$, the AMNIS for high SNR is given by

$$\mathrm{AMNIS}^{(i)}(\mathrm{Z}^{(i)}) = -2\log p_D^{|I_+|} \prod_{\ell\in I_+} \exp\left(\left(\gamma^{(\ell,\theta)}\right)^{\mathrm{T}} \left(\underline{\mathrm{S}}^{(\ell)}\right)^{-1} \gamma^{(\ell,\theta)}\right) \tag{5.129}$$

$$= -2|I_+|\log p_D + \sum_{\ell\in I_+} \left(\gamma^{(\ell,\theta)}\right)^{\mathrm{T}} \left(\underline{\mathrm{S}}^{(\ell)}\right)^{-1} \gamma^{(\ell,\theta)} \tag{5.130}$$

$$\approx \sum_{\ell\in I_+} \left(\gamma^{(\ell,\theta)}\right)^{\mathrm{T}} \left(\underline{\mathrm{S}}^{(\ell)}\right)^{-1} \gamma^{(\ell,\theta)}. \tag{5.131}$$

Obviously, each summand of (5.131) follows a χ^2 distribution since it matches the single-object NIS of the track ℓ. Since the χ^2 distribution is additive and the objects are assumed to be independent of each other, the AMNIS follows a χ^2 distribution in case of a high SNR. Consequently, the AMNIS is χ^2 distributed with $|I_+| \cdot \dim(\mathrm{Z})$ degrees of freedom for a measurement space with dimension $\dim(\mathrm{Z})$.

5.5 Discussion

The proposed LMB filter may be interpreted in two ways: On the one hand, the LMB filter may be considered as a more accurate approximation of the multi-object Bayes filter compared to the CB-MeMBer filter which additionally estimates the track labels. The improved accuracy of the LMB filter is due to the more precise update step which does not require restrictions of the detection probability or the false alarm density. Consequently, the LMB filter is not prone to the bias in the cardinality estimate which is observed for the CB-MeMBer filter in scenarios with a high false alarm density. However, the computational complexity of the LMB filter update is significantly higher than the one of the CB-MeMBer filter for groups with more than one track. On the other hand, the LMB filter may be considered as an approximation of the δ-GLMB filter which approximates the multi-object posterior using an LMB RFS in order to simplify the prediction step. Further, the LMB representation significantly reduces the computational complexity of the δ-GLMB filter update due to the possibility of applying a grouping procedure and the parallel group updates. Due to the approximation of the δ-GLMB posterior after each measurement update using an LMB RFS, the LMB filter loses information about the cardinality distribution while preserving the mean cardinality as well as the spatial distribution of each track. Additionally, the LMB filter does not provide a history of association maps since the association maps are dropped during the approximation of the δ-GLMB RFS by an LMB RFS. However, this is not a limiting factor in most applications since the previous association maps are not required for filtering the multi-object state. Appendix C provides a brief summary illustrating the main differences between the δ-GLMB, the LMB filter, and the approximations introduced in Section 3.7.

The LMB filter provides several interesting topics for future research projects: Compared to the SMC multi-object Bayes filter in Chapter 4, the proposed LMB filter does not facilitate the integration of object interactions at the moment since all tracks are predicted independently. Using an additional grouping procedure before the prediction step, possibly interacting tracks can be identified. The interacting tracks may be used to initialize the SMC multi-object Bayes filter. After prediction and update of the group, the posterior multi-object particles can be approximated by an LMB RFS again.

The proposed grouping procedure for the LMB filter does not ensure a maximum computing time per filter update due to the fixed gating threshold. In [Mun11], an agglomerative grouping procedure for the JIPDA filter is proposed which ensures a maximum number of nodes in the association trees using adaptive gating thresholds. An adaptation of the agglomerative grouping procedure to the LMB filter to ensure a maximum computing time is promising but the reduced gating thresholds affect the assumption of statistically independent groupings. Hence, a detailed evaluation of the corresponding effects will be required.

Further possible extensions of the LMB filter include the support of multiple-model approaches, the application to SLAM, and the integrated estimation of the clutter distribution.

Chapter 6

Evaluation

In this chapter, the performance of the proposed filter algorithms is evaluated using simulated and real world sensor data. Sections 6.1 and 6.2 present simulation results for the sequential Monte-Carlo multi-object Bayes (SMC-MOB) filter and the LMB filter, respectively. Afterwards, an approach to incorporate sensor specific true positive probabilities in multi-object Bayes filters is introduced in Section 6.3. The SMC-MOB filter is applied to pedestrian tracking in an indoor scenario which comprises closely spaced objects as well as occlusions. Finally, the performance of the LMB filter is compared to the JIPDA tracker for several typical scenarios in vehicle environment perception to illustrate possible performance gains. The tracking results of the algorithms are compared using the optimal subpattern assignment (OSPA) distance and the optimal subpattern assignment for tracks (OSPAT) distance which provide the average distance between the ground truth positions and the estimates in meters since they are solely calculated using the Cartesian x and y positions.

6.1 SMC Multi-Object Bayes Filter - Simulations

6.1.1 Multi-Object Likelihood Approximation

The approximation of the multi-object likelihood function in Section 4.3 allows for the assignment of a measurement z_j to multiple tracks within an association hypothesis. If the minimum distance between the objects is sufficiently larger than the standard deviation of the measurement noise, the approximation error is expected to be negligible. In order to illustrate the approximation error, a static scenario comprising five objects is investigated. Figure 6.1 depicts the positions of the objects for the three scenarios S_1, S_2, and S_3. The simulated sensor delivers Cartesian position measurements with a measurement noise of $\sigma = \sigma_x = \sigma_y = 0.07$ m. The SMC-MOB filter uses a constant detection probability of $p_D = 0.95$. The target dynamics follow a random walk model with a standard deviation of $\sigma_v = 0.08$ m/s. Object interactions are modeled using the set based weight adaption introduced in Section 4.2.2 which decreases the weight of a multi-object particle if the distance between any of the objects is smaller than the required minimum distance. The minimum object distance is set to $d = 0.3$ m for scenario S_1 and to $d = 0.4$ m for the other scenarios.

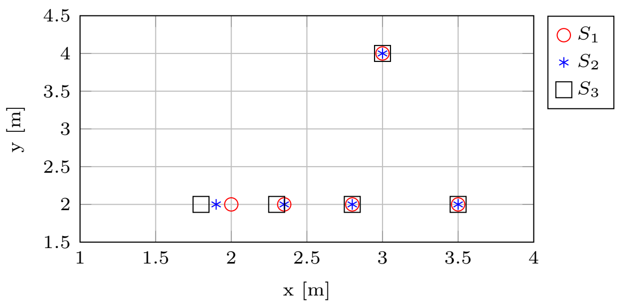


Figure 6.1: Ground truth object positions for the scenarios S_1, S_2, and S_3.

The results of the SMC-MOB filter with $N = 10000$ multi-object particles are summarized in Table 6.1 by means of the expectation value of the OSPA and the OSPAT distance. Obviously, the OSPA distance of the filters using the approximate multi-object likelihood is only slightly higher than the one of the filters which apply the computationally more expensive exact multi-object likelihood function. However, the OSPAT distance indicates that the approximation increases the possibility of switching track IDs in the scenarios S_1 and S_2 since a measurement may update more than one of the state vectors within a multi-object particle.

The approximate multi-object likelihood function (4.21) requires a minimum distance $d > 6\sigma$ between the objects to ensure negligible approximation errors. Although this requirement is not fulfilled for the scenario S_1 ($d \approx 4.3\sigma$) as well as the scenarios S_2 and S_3 ($d \approx 5.7\sigma$), the approximate and the exact multi-object likelihood function achieve approximately the same performance as the with respect to the OSPA distance. Hence, the utilization of the computationally less expensive approximation is suggested if continuous track IDs are not required and the minimum distance is slightly below the required minimum distance. However, the usage of the exact multi-object likelihood function is recommended for applications which require accurate track IDs and do not ensure negligible approximation errors (i.e. $d < 6\sigma$).

Scenario	Algorithm	E(OSPA) [m]	E(OSPAT) [m]
S_1	Exact Likelihood	0.0595	0.0608
S_1	Approx. Likelihood	0.0635	0.2674
S_2	Exact Likelihood	0.0588	0.0600
S_2	Approx. Likelihood	0.0621	0.1552
S_3	Exact Likelihood	0.0585	0.0587
S_3	Approx. Likelihood	0.0612	0.0815

Table 6.1: Comparison of the approximate and exact multi-object likelihood using scenarios S_1, S_2, and S_3: mean value and standard deviation of the OSPA distance (cut-off $c = 1$ m, order $p = 1$) and the OSPAT distance ($\alpha = c$) averaged over 50 Monte Carlo runs.

6.1.2 Object Interactions

In Section 4.2, several methods to model the motion of extended objects have been proposed. In contrast to the set based weight adaption, which adjusts the weight of a multi-object particle with respect to the minimum distance between any of the included objects, the thinning procedures based on the Matérn methods remove state vectors from the multi-object particles until they fulfill the distance constraint. The performance of the methods is compared using the scenario S_2 given in the previous subsection.

The mean values of the OSPA and the OSPAT distance for the five methods to model the motion of extended objects are given by Table 6.2. The SMC-MOB filter uses $N = 10000$ particles and the approximate multi-object likelihood function. The algorithms RFS Matérn I and RFS Matérn II perform only the thinning procedure, while the algorithms with "PS" also adapt the weights of the thinned multi-object particles. With respect to the OSPA distance, the algorithms using thinning obtain almost identical results while the set based weight adaption performs slightly better. If labeling errors are penalized using the OSPAT distance, the algorithm using set based weight adaption outperforms the thinning methods without weight adaption. Combining the thinning procedures with an additional weight adaption reduces the average OSPAT distance, but still leads to slightly worse performance than the set based weight adaption.

In applications which require continuous track IDs, the utilization of the set based weight adaption is recommended since the thinning procedures increase the possibility of track ID switches. The superior performance of the set based weight adaption is explained as follows: While the set based weight adaption invalidates a multi-object

Algorithm	E(OSPA) [m]	E(OSPAT) [m]
RFS Set Based Weight Adaption	0.0621	0.1552
RFS Matérn I	0.0696	0.2006
RFS Matérn I PS	0.0693	0.1725
RFS Matérn II	0.0674	0.1929
RFS Matérn II PS	0.0683	0.1728

Table 6.2: Simulation results for the methods to model the motion of extended objects: mean values of the OSPA and the OSPAT distance ($c = 1$ m, $p = 1$, $\alpha = c$) averaged over 50 Monte Carlo runs.

particle if two of the objects do not fulfill the distance constraint, the thinning methods only delete the colliding state vectors which increases the area of possible states of the retained state vectors. Thus, it is possible that one of the retained state vectors moves to a position which would be occupied by one of the deleted state vectors and obtains the ID of the deleted state vector after the measurement update.

6.1.3 Marginalization

In the scenarios S_1 and S_2, the approximation of the multi-object likelihood function leads to a higher OSPAT distance. Especially in scenario S_1, the minimum distance of the objects does not ensure a negligible approximation error due to $d < 6\sigma$. In Section 4.3.2, an alternative approximation of the multi-object likelihood function based on the marginal probability $p(x^{(i)}|z_j)$ is proposed to reduce the approximation error in these situations. Especially in situations where a measurement z_j has a significant likelihood for two or more tracks, the marginal probability reduces the approximation error by penalizing association hypotheses which assign the measurement z_j to multiple tracks.

The performance of the marginalized multi-object likelihood is evaluated using the scenario depicted by Figure 6.2. First, the five objects are well separated. At $k = 125$, three of the objects meet around the origin of the coordinate system and the two objects at the bottom pass each other. The minimum distance of the objects drops below 0.4 m at this time. The motion model is again a random walk model with $\sigma_v = 0.08$ m/s. The measurement noise of the Cartesian position measurements is $\sigma = 0.07$ m, the detection probability is $p_D = 0.95$, and a mean number of $\lambda_c = 10$ clutter measurements is obtained with each scan which are uniformly distributed over the region depicted by Figure 6.2.

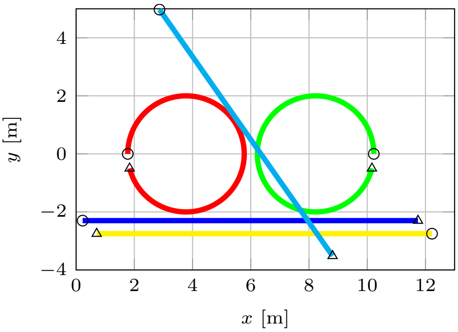
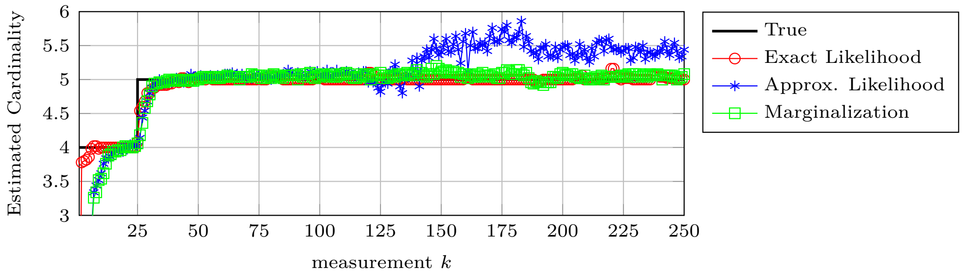


Figure 6.2: Ground truth trajectories of five objects: start position marked by a circle, end position by a triangle. The object starting at $y = 5$ m appears at $k = 25$, the other objects at $k = 0$.

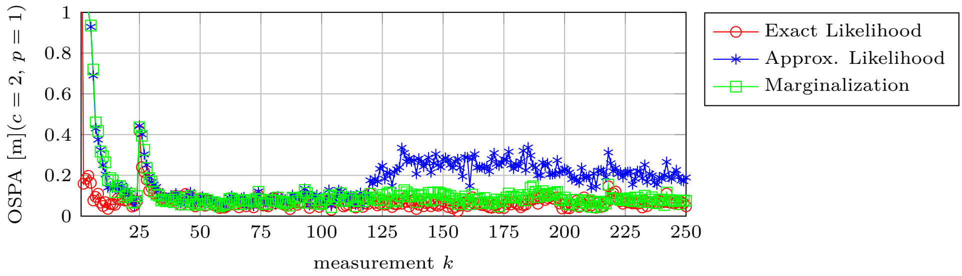
In order to illustrate the effect of the marginalization, a minimum distance of $d = 0.1$ m is used in the filter. Figure 6.3 depicts the simulation results for the exact, the approximate, and the marginalized multi-object likelihood function. Figure 6.3a indicates that the approximate likelihood tends to initialize an additional object after $k = 125$. Since the minimum distance is significantly smaller than 6σ and the approximation permits the association of a single measurement to more than one object, the filter may not get rid of the additional track. In contrast to the approximate multi-object likelihood function, the marginalized likelihood function has no bias in the cardinality estimate and its OSPA distance (see Figure 6.3b) is almost identical to the one obtained using the exact multi-object likelihood function. Considering track ID switches using the OSPAT distance, the filter using the marginalized multi-object likelihood function is only slightly outperformed by the filter using the exact likelihood function. However, the performance gain compared to the approximate multi-object likelihood is significant in this case.

The simulation results show that the approximation may lead to a bias in the cardinality estimate if the minimum distance does not ensure negligible approximation errors. By incorporating the marginal probability $p(x^{(i)}|z_j)$ into the approximation of the multi-object likelihood function, the bias in the cardinality estimate disappears and the approximation achieves nearly the performance of the exact multi-object likelihood function at a significantly lower computational complexity.

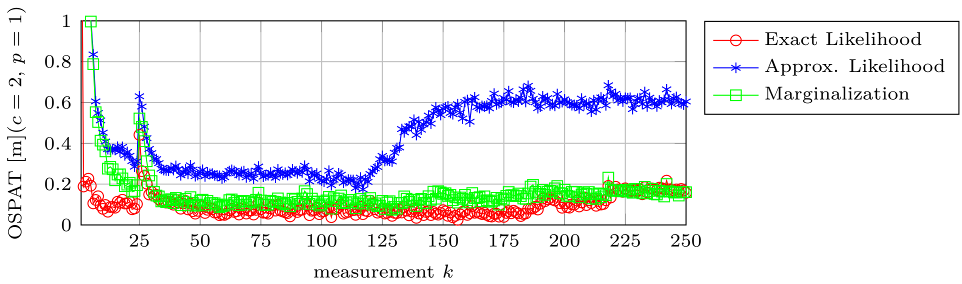
The marginalized multi-object likelihood function (4.24) is recommended for applications with a minimum distance $d < 6\sigma$ since it reduces the approximation error by incorporating the marginal probability. For higher minimum distances, the influence of the marginal probability on the multi-object likelihood is negligible since a measurement may not have a high likelihood for more than one track in this case. Thus, the



(a) Cardinality estimates.



(b) OSPA distances (order $p = 1$, cut-off $c = 300$ m).



(c) OSPAT distances (order $p = 1$, cut-off $c = 300$ m, $\alpha = c$).

Figure 6.3: Cardinality estimates and OSPA/OSPAT distances for the scenario depicted by Figure 6.2 averaged over 50 Monte Carlo runs. The SMC-MOB filter uses $N = 50000$ multi-object particles, the mean number of false alarms is $\lambda_c = 10$, and the minimum distance for the set based weight adaption is set to $d = 0.1$ m.

computationally less expensive approximate multi-object likelihood (4.21) achieves identical results for $d \gg 6\sigma$.

6.2 Labeled Multi-Bernoulli Filter - Simulations

6.2.1 Sequential Monte Carlo Implementation

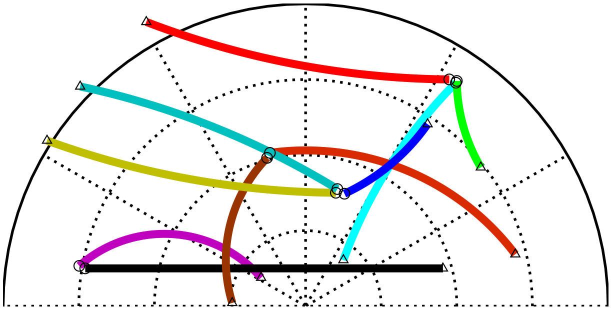


Figure 6.4: Object trajectories for the evaluation of the SMC implementation. Start points are marked by a circle, end points by a triangle. The half-disc has a radius of 2000 m and the sensor is located at the bottom center.

The performance of the SMC LMB filter is evaluated using a scenario comprising a total number of 10 objects and several objects appearances and disappearances. The trajectories of the objects are illustrated by Figure 6.4. The object states are given by the state vector $x = [\ p_x, v_x, p_y, v_y, \omega\ ]^{\mathrm{T}}$ which contains the two dimensional Cartesian positions p and velocities v as well as the turn rate ω. Within the coordinated turn process model [VV13b], the standard deviation of the acceleration noise is $\sigma_a = 15\ \mathrm{m/s}^2$ and the standard deviation of the yaw rate is $\sigma_\omega = \pi/180$ rad/s. Further, the survival probability is assumed to be state independent and is given by $p_S(x) = 0.99$.

Within the simulation, objects may only appear at four different locations which are modeled using a multi-Bernoulli birth RFS

$$\pi_B = \{(r_B^{(i)},\ p_B^{(i)})\}_{i=1}^4,$$

where the spatial distribution of each birth track is a Gaussian distribution $p_B^{(i)}(x) = \mathcal{N}(x; \hat{x}_B^{(i)}, \underline{\mathrm{P}}_B)$. The existence probabilities of the Bernoulli distributions and the mean values of the Gaussian distributions are given by

$$\begin{aligned} r_B^{(1)} &= 0.02, & \hat{x}_B^{(1)} &= [-1500, 0, 250, 0, 0]^{\mathrm{T}}, \\ r_B^{(2)} &= 0.02, & \hat{x}_B^{(2)} &= [-250, 0, 1000, 0, 0]^{\mathrm{T}}, \\ r_B^{(3)} &= 0.03, & \hat{x}_B^{(3)} &= [250, 0, 750, 0, 0]^{\mathrm{T}}, \end{aligned}$$

$$r_B^{(4)} = 0.03, \quad \hat{x}_B^{(4)} = [1000, 0, 1500, 0, 0]^{\mathrm{T}},$$

and the covariance matrix follows $\underline{P}_B = \mathrm{diag}([50, 50, 50, 50, 6(\pi/180)]^T)^2$.

The measurement vector $z = [\varphi, r]^T$ of the sensor comprises the angle φ and the range r. The simulated sensor is assumed to deliver measurements up to a distance of 2000 m and the opening angle is 180°, i.e. $\varphi = [-\pi/2, \pi/2]$. The measurement noise of the sensor follows a Gaussian distribution and the standard deviations for angle and range measurements are $\sigma_\varphi = (\pi/180)$ rad and $\sigma_r = 5$ m, respectively. The detection probability of the sensor is range dependent and is given by

$$\begin{aligned} p_D(x) &= p_{D,max} \exp\left([p_x, p_y]^{\mathrm{T}} \Sigma_D^{-1} [p_x, p_y]\right) \\ \Sigma_D &= 6000^2 \underline{I}_2 \end{aligned}$$

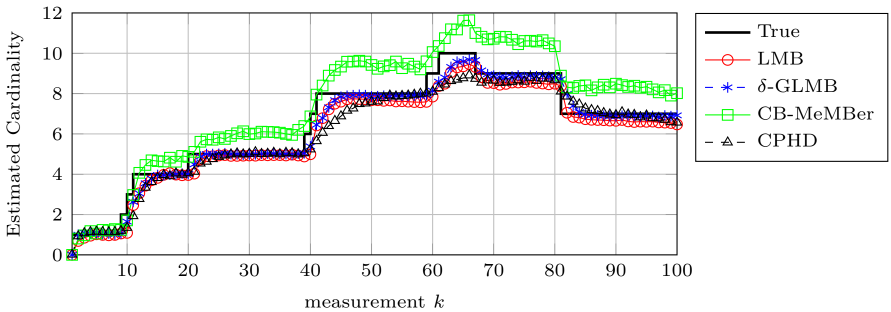
where $\underline{I}_2$ denotes the identity matrix of dimension two. In addition to the object measurements, the sensor delivers Poisson distributed clutter measurements with $\kappa(z) \approx 10^{-5}$ which coincides with a mean number of $\lambda_c = 60$ false alarms per scan.

The LMB filter performance is compared to the δ-GLMB filter, the CB-MeMBer filter, and the CPHD filter. Within the prediction and the update step, the δ-GLMB filter incorporates the 1000 components with the highest weights determined by the k-shortest paths and the Murty algorithm, respectively. Further, components with a weight $w < 0.00001$ are pruned. The δ-GLMB filter represents the spatial distribution of each track within a set of track labels I using $N = 1000$ particles. In contrast, the LMB filter and the CB-MeMBer filter use $N = 1000$ particles for each track and prune tracks with an existence probability $r < 0.001$. Further, the LMB filter uses only

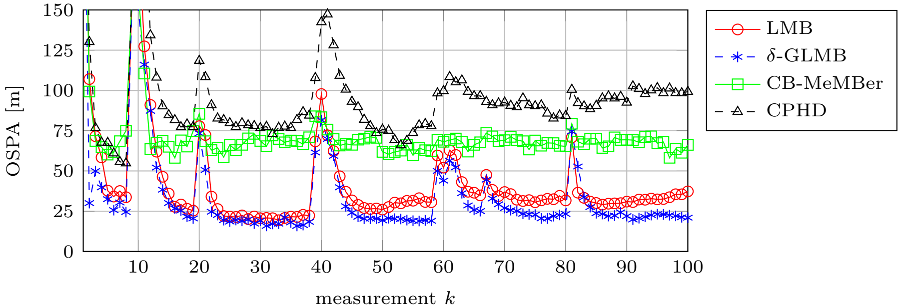
$$h^{(i)} = 1 + 2^{|\mathbb{L}_+^{(i)}|}$$

association hypotheses per group $\mathcal{G}^{(i)} = (\mathbb{L}_+^{(i)}, \mathbb{M}^{(i)})$ in the update step. The CPHD filter uses $N = 1000 \cdot \hat{N}$ particles where $\hat{N}$ represents the estimated number of objects. In contrast to the δ-GLMB, the LMB, and the CB-MeMBer filter, the CPHD filter does not facilitate the application of a multi-Bernoulli birth model. Hence, the CPHD filter uses a Poisson birth model whose intensity matches the one of the multi-Bernoulli birth model.

Figure 6.5a shows the cardinality estimates of the four filters for the scenario depicted by Figure 6.4. The δ-GLMB filter provides the most accurate cardinality estimates for the investigated scenario while the LMB filter slightly underestimates the cardinality for $k > 50$. The CPHD filter also delivers a satisfactory result, although the filtering of the cardinality distribution causes a low pass behavior of the cardinality estimate during object appearance and disappearance. As expected, the cardinality estimate of the CB-MeMBer filter is biased due to the high clutter density.



(a) Cardinality estimates.



(b) OSPA distances (order $p = 1$, cut-off $c = 300$ m).

Figure 6.5: Cardinality estimates and OSPA distance for the SMC implementations of the LMB, δ-GLMB, CB-MeMBer, and CPHD filter for the scenario depicted by Figure 6.4 with $\lambda_c = 60$ and $p_{D,max} = 0.98$ (averaged over 100 MC runs).

Figure 6.5b shows the corresponding OSPA distances of the filters. As expected, the δ-GLMB filter provides the smallest OSPA distance for almost all time steps. However, the OSPA distance of the LMB filter is only slightly larger than the one of the δ-GLMB filter which is expected due to the approximation of the δ-GLMB RFS by an LMB RFS. However, the OSPA distance is also influenced by the effectively used number of association hypotheses in the prediction and update step as well as the different number of particles per object. While the LMB filter represents each object using $N = 1000$ particles, the δ-GLMB filter represents an object's state with $N = 1000$ particles for each history of association maps. Thus, a total number of $(m+1) \cdot N$ particles is used to represent the ambiguity of a state update with m measurements inside the gate of a track. Due to the bias in the cardinality estimate, the OSPA distance of the CB-MeMBer filter is significantly higher than the ones of the LMB and the δ-GLMB filter. Although the CPHD filter provides a more accurate cardinality estimate than the CB-MeMBer filter, the OSPA distance of the CPHD filter is higher than the one of the CB-MeMBer filter due to the error-prone clustering algorithm which is required for track extraction.

While the δ-GLMB filter may use most of its association hypotheses to resolve the uncertainty about the track to measurement association for a small subset of tracks, the LMB filter uses a fixed amount of association hypotheses which only depends on the number of tracks within the group. Figure 6.6 illustrates the influence of the number of hypotheses per group on the tracking result. In three different experiments, the hypotheses per group are limited to

$$h_3^{(i)} = \min(1 + 3^{|\mathbb{L}_+^{(i)}|}, 1000), \quad h_5^{(i)} = \min(1 + 5^{|\mathbb{L}_+^{(i)}|}, 1000), \quad h_{1000}^{(i)} = 1000.$$

Since the number of tracks and measurements per group restrict the quantity of feasible hypotheses, the limits for the hypotheses per group denote an upper bound. Obviously, the filter using $h_5^{(i)}$ hypotheses per group significantly outperforms the one with $h_3^{(i)}$ hypotheses. However, increasing the number of hypotheses to up to 1000 per group does not provide a huge gain. Due to the higher amount of hypotheses, the LMB filter performance is close to the one of the δ-GLMB filter although the computational complexity is still significantly smaller due to the grouping procedure.

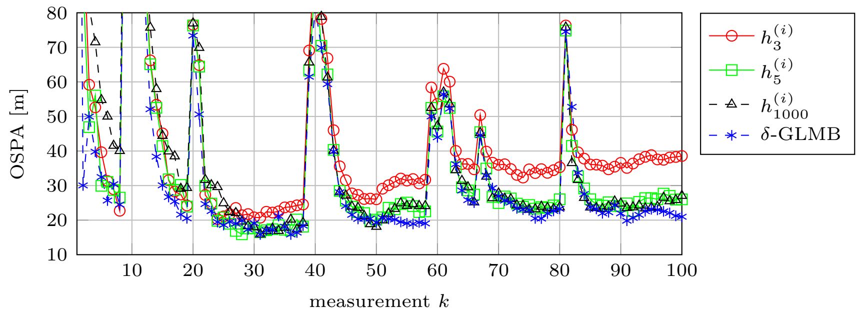


Figure 6.6: Comparison of the OSPA distances ($p = 1$, $c = 300$ m) of LMB filters with different numbers of evaluated association hypotheses with the OSPA distance of the δ-GLMB filter (averaged over 100 MC runs).

6.2.2 Gaussian Mixture Implementation

The Gaussian mixture (GM) implementation of the LMB filter is evaluated using a scenario on the two dimensional region $[-1000, 1000] \times [-1000, 1000]$. The trajectories of the 12 objects are depicted by Figure 6.7. Similar to the SMC scenario, the GM scenario comprises the appearance and disappearance of several objects. The motion of the objects follows a two dimensional CV model and the standard deviation of the process noise in x and y direction is $\sigma_a = 5$ m/s^2. Further, the survival probability $p_S = 0.99$ is assumed to be state independent. Within the observed region, the sensor delivers measurements for the x and y positions of the objects with a standard deviation of $\sigma_x = \sigma_y = 10$ m. The detection probability of the sensor is assumed to be state independent and is given by $p_D = 0.98$. Similar to the SMC implementation, clutter follows a Poisson distribution with intensity $\kappa(z) = 1.5 \cdot 10^{-5}$ which coincides again with a mean number of $\lambda_c = 60$ clutter measurements per scan.

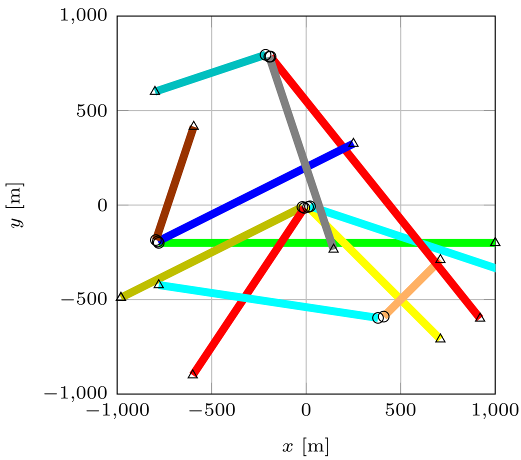


Figure 6.7: Object trajectories for the evaluation of the GM implementation. Start points are marked by a circle, end points by a triangle.

Similar to the SMC implementation, appearing objects are modeled using a multi-Bernoulli birth RFS

$$\pi_B = \{(r_B^{(i)}, p_B^{(i)})\}_{i=1}^4. \tag{6.1}$$

The spatial distribution of the birth tracks follows a Gaussian distribution $p_B^{(i)}(x) = \mathcal{N}(x; \hat{x}_B^{(i)}, \underline{\mathrm{P}}_B)$. The existence probabilities of the Bernoulli distributions and the mean values of the Gaussian distributions are given by

$$
\begin{aligned}
r_B^{(1)} &= 0.04, \quad & \hat{x}_B^{(1)} &= [0, 0, 0, 0]^{\mathrm{T}}, \\
r_B^{(2)} &= 0.03, \quad & \hat{x}_B^{(2)} &= [-200, 0, 800, 0]^{\mathrm{T}}, \\
r_B^{(3)} &= 0.03, \quad & \hat{x}_B^{(3)} &= [-800, 0, -200, 0]^{\mathrm{T}}, \\
r_B^{(4)} &= 0.02, \quad & \hat{x}_B^{(4)} &= [400, 0, -600, 0]^{\mathrm{T}},
\end{aligned}
$$

and the covariance matrix follows $\underline{\mathrm{P}}_B = \mathrm{diag}([10, 10, 10, 10]^T)^2$.

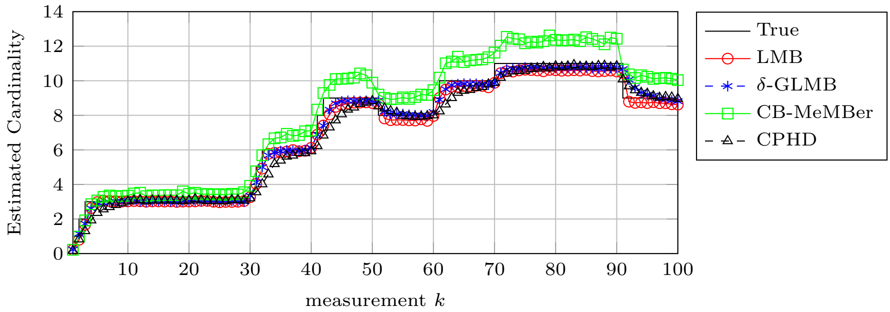
The GM implementation of the LMB filter is again compared to the δ-GLMB, CPHD, and CB-MeMBer filter. The LMB and the CB-MeMBer filter prune tracks with an existence probability $r < 0.001$. Additionally, Gaussians with a weight $w < 0.01$ are pruned and Gaussians with a Mahalanobis distance $d_{\mathrm{MHD}} < 0.5$ are merged. The CPHD filter prunes Gaussians with a weight of $w < 0.00001$ and uses the same threshold for merging as the LMB filter. The δ-GLMB filter uses a total number of 2000 hypotheses and prunes hypotheses with weight $w < 0.00001$. Figure 6.8a shows the cardinality estimates of the filters for the linear Gaussian scenario depicted by Figure 6.7. Again, the CB-MeMBer filter has a significant bias in the cardinality estimate while the other filters provide excellent cardinality estimates. Similar to the SMC results in Figure 6.5a, the estimate of the CPHD filter shows a low pass behavior. The δ-GLMB filter has some issues with disappearing objects (e.g. for $k > 90$) which is due to the restriction of the number of hypotheses to 2000 during prediction. The LMB filter slightly underestimates the cardinality since it only uses

$$h_0^{(i)} = 2^{|\mathbb{L}_+^{(i)}|}$$

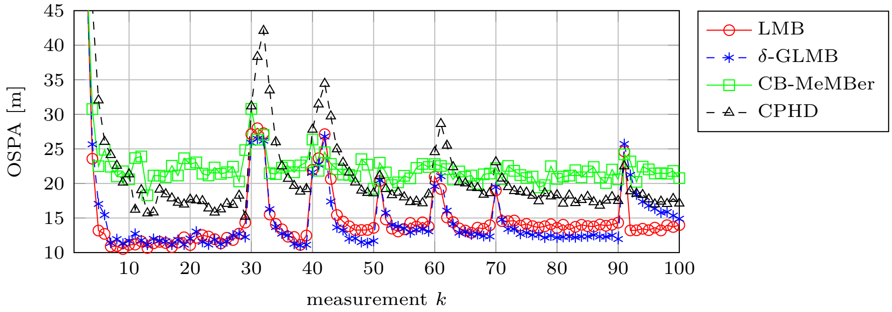
association hypotheses within the filter update.

The OSPA distances of the filters are depicted by Figure 6.8b. In contrast to the results for the SMC implementations, the CPHD filter outperforms the CB-MeMBer filter since the error-prone particle clustering is not required in a GM implementation. For $k \leq 90$, the δ-GLMB filter obtains the smallest OSPA distance, while the LMB filter performs best for $k > 90$. However, the δ-GLMB filter is expected to show at least the same accuracy for $k > 90$ if the number of components in the prediction step is significantly increased.

The GM implementations of the δ-GLMB filter and the LMB filter allow for a more precise comparison of the filters with respect to the number of association hypotheses used in the update step. While the δ-GLMB filter uses 2000 hypotheses, the LMB filter implementation is evaluated for the following upper bounds of hypotheses per



(a) Cardinality estimates.



(b) OSPA distances (order $p = 1$, cut-off $c = 100$ m).

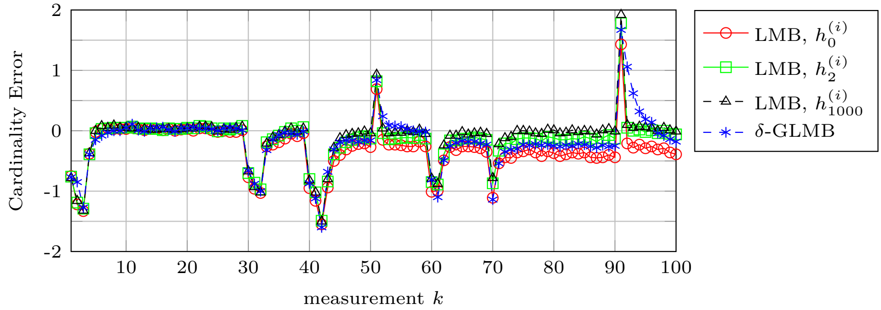
Figure 6.8: Cardinality estimates and OSPA distance for the GM implementations of the LMB, δ-GLMB, CB-MeMBer, and CPHD filter for the scenario depicted by Figure 6.7 with $\lambda_c = 60$ and $p_D = 0.98$ (averaged over 100 MC runs).

group:

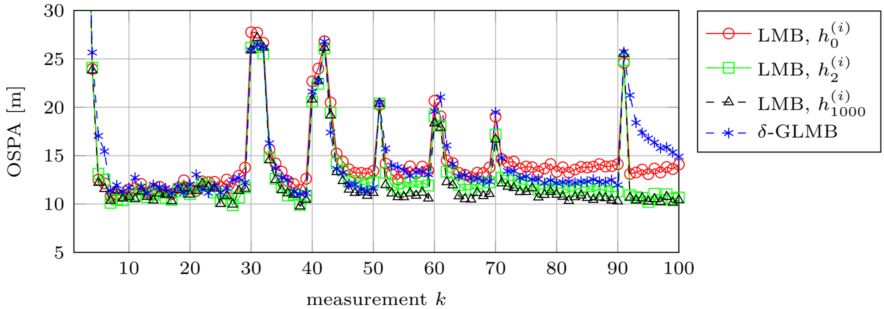
$$h_0^{(i)} = \min(2^{|\mathbb{L}_+^{(i)}|}, 1000), \quad h_2^{(i)} = \min(1 + 2^{|\mathbb{L}_+^{(i)}|}, 1000), \quad h_{1000}^{(i)} = 1000.$$

Figures 6.9a and 6.9b indicate that the cardinality error and the OSPA distance of an LMB filter are already smaller than the ones of the δ-GLMB filter if the LMB filter utilizes h_2 association hypotheses per group. Using h_{1000}, the LMB filter significantly outperforms the δ-GLMB filter although the posterior δ-GLMB density is approximated by an LMB RFS after each filter update.

Since the δ-GLMB filter is expected to show a superior performance compared to the LMB filter, a more detailed investigation is required. As indicated by the results in Figure 6.9, the number of association hypotheses within the filter update is essential



(a) Cardinality errors.



(b) OSPA distances (order $p = 1$, cut-off $c = 100$ m).

Figure 6.9: Comparison of the OSPA distances of LMB filters with different numbers of evaluated association hypotheses with the OSPA distance of the δ-GLMB filter (averaged over 100 MC runs).

for the filter performance. Figure 6.10 illustrates the total number of hypotheses

$$h_j = \prod_{i=1}^{|\mathcal{G}|} h_j^{(i)}$$

of the LMB filter where $|\mathcal{G}|$ denotes the number of groups. Although the LMB filter with h_0 already computes a significantly larger amount of hypotheses than the δ-GLMB filter in some cases, the δ-GLMB filter shows superior tracking results. Due to the high detection probability and the high false alarm rate, the number of measurements $|\mathbb{M}^{(i)}|$ for group i is usually higher than the number of track labels $|\mathbb{L}_+^{(i)}|$. Consequently, $h_0^{(i)}$ hypotheses are almost always feasible for group i. Consequently, the approximation of the posterior δ-GLMB RFS within the LMB filter is the main reason for the slightly worse performance in case of h_0 hypotheses.

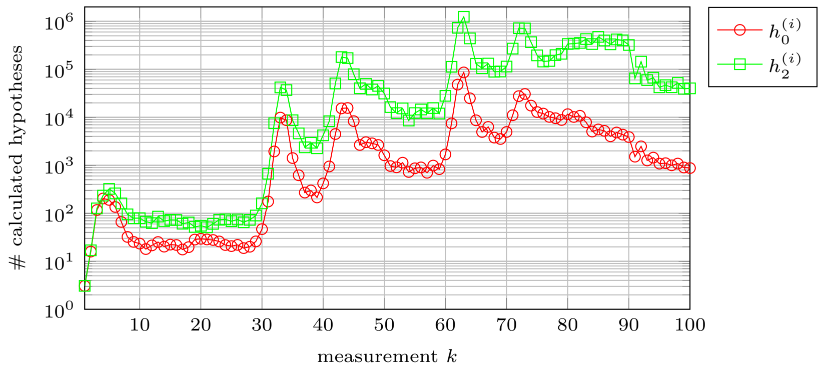


Figure 6.10: Amount of evaluated hypotheses in the update step of the LMB filter.

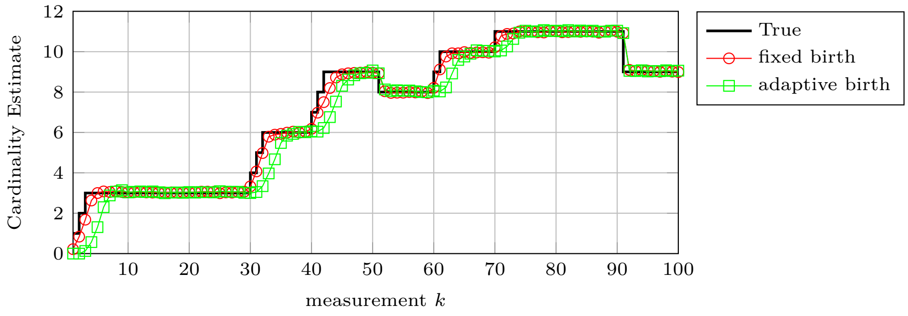
6.2.3 Adaptive Birth Distributions

The simulation results in the previous paragraphs are based on fixed birth locations which are concentrated at four different locations. If no context knowledge about possible birth locations is available, an alternative solution is to concentrate the birth distribution at the end of the sensor's FOV. However, both approaches do not facilitate the re-initialization of lost tracks at arbitrary positions which could be realized using wide-spread birth distributions covering the complete state space. Since a wide-spread birth distribution counteracts the computational savings of the grouping procedure, the LMB filter requires an adaptive birth distribution (see Section 5.3.6) to allow for new-born objects all over the state space. The linear simulation scenario of the previous subsection is used to illustrate the difference between an LMB filter with adaptive birth distribution and an LMB filter with fixed birth locations. The LMB filter evaluates (if possible) a total number of 1000 hypotheses per group.

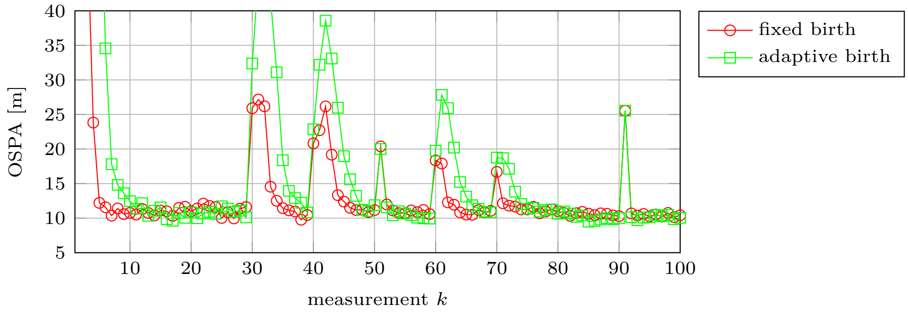
The adaptive birth distribution uses the parameter

$$\lambda_B = r_{B,k+1}^{\max} = \sum_{i=1}^{4} r_B^{(i)}$$

where $r_B^{(i)}$ are the existence probabilities of the multi-Bernoulli birth model given by (6.1). The existence probability of the individual birth tracks is calculated by (5.112). Since the sensor delivers x and y positions, the mean value of a track's spatial distribution is located at the position of the measurement and the velocity is set to zero. The spatial uncertainty of the birth track is represented by $\underline{P}_B = \mathrm{diag}([10^2, 10^2, 10^2, 10^2]^T)$, which is identical to the covariance used in the birth model with fixed locations.



(a) Cardinality estimates.



(b) OSPA distances (order $p = 1$, cut-off $c = 100$ m).

Figure 6.11: Comparison of fixed and adaptive birth model: Cardinality estimates and OSPA distance for the GM-LMB filter for the scenario depicted by Figure 6.7 with $\lambda_c = 60$ and $p_D = 0.98$ (averaged over 100 MC runs).

Figure 6.11a shows the cardinality estimates for both versions, the simulation parameters are identical to the ones of the previous paragraph. Obviously, the confirmation of new objects in case of an adaptive birth model is delayed by approximately two measurement cycles. While the LMB filter with fixed birth locations utilizes the first measurement of a new born track to update a component of the birth model, the adaptive birth distribution requires the first measurement to create a birth candidate. Further, the existence probability of a birth track is significantly lower in case of the adaptive birth model, since the expected number of new born objects λ_B is distributed over the set of all measurements at time k. Except for the delayed object initialization, the cardinality estimate of the LMB filter with adaptive birth matches the one of the filter with fixed birth locations. Thus, the LMB filter successfully prevents clutter tracks in scenarios with a huge number of false alarms even if tracks may appear all over the state space.

The OSPA distances for the different birth models are depicted by Figure 6.11b. Due to the delayed object confirmation, the OSPA distance of the filter with adaptive birth is significantly higher for a few measurement cycles after a new object appeared. However, the adaptive birth distribution obtains a slightly smaller OSPA distance for large k since it facilitates the re-initialization of a lost object which is not possible in case of fixed birth locations.

6.2.4 Approximate Multi-Object NIS

The results of the previous subsection required the availability of an accurate ground truth position of all objects to calculate the OSPA or the OSPAT. In online applications, a ground truth is in general not available. The AMNIS proposed in Section 5.4.2 facilitates a consistency check for the LMB filter in such situations.

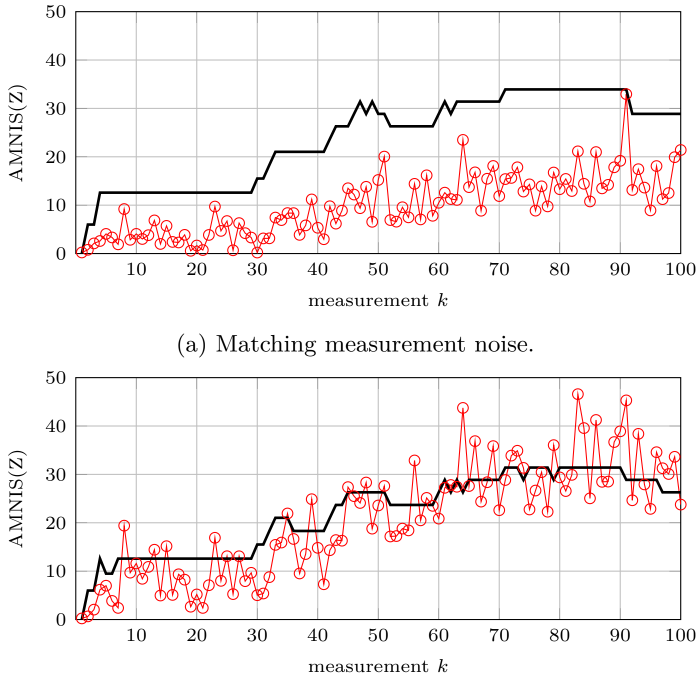
Figure 6.12a shows the AMNIS for a single run of the scenario depicted by Figure 6.7. The LMB filter uses a matching measurement noise and a mean number of $\lambda_c = 60$ Poisson distributed clutter measurements is obtained with each scan. The black line corresponds to the χ^2 value for a confidence level of 0.95. The χ^2 distribution has $2 \cdot \hat{N}$ degrees of freedom where $\hat{N}$ represents the estimated number of objects. The AMNIS indicates a consistent multi-object tracking algorithm since it is significantly below the χ^2 threshold.

Similar to the NIS, the AMNIS is expected to indicate an underestimated process or measurement noise. Figure 6.12b illustrates the AMNIS for an LMB filter which significantly underestimates the measurement noise using $\sigma_x = \sigma_y = 5$ m (the actual measurement noise is $\sigma_x = \sigma_y = 10$ m). Obviously, the AMNIS is significantly higher in this case and exceeds the χ^2 threshold for a huge number of measurement cycles.

The results in Figure 6.12a and 6.12b lead to the assumption that the approximate χ^2 property of the AMNIS may also be applicable to scenarios with low SNR although a high SNR is required by the definition in Section 5.4.3.

6.3 Multi-Bernoulli Clutter Model

The RFS based tracking algorithms conveniently model the clutter measurements using a Poisson distribution. While the Poisson distribution is adequate in air



(a) Matching measurement noise.

(b) Underestimated measurement noise.

Figure 6.12: AMNIS for LMB filters with matching process noise. The black line depicts the χ^2 threshold for a confidence level of 0.95 with $2 \cdot \hat{N}$ degrees of freedom.

surveillance applications using radar sensors, the clutter distribution in applications like vehicle environment perception significantly differs from a Poisson distribution with fixed parameters. For example, false alarms are often state dependent and the expected number of false alarms is usually time and environment dependent. However, sensors may also facilitate the determination of measurement specific true positive probabilities $p_{TP}(z_j)$ [Mah09b; Mun11]. The true positive probability is defined as the probability that the received measurement is originated by an object of interest, e.g. a car. In pattern classification, the receiver operating characteristics (ROC) [DHS01] is commonly used to obtain the true positive probability $p_{TP}(z_j)$ and the false positive probability is given by $p_{FP}(z_j) = 1 - p_{TP}(z_j)$. In [Mah09b; Mun11], the incorporation of a measurement's true positive probability into the calculation of the track to measurement associations of the JIPDA algorithm is proposed.

Similar to the object detection process, the probability that measurement z_j is a false positive measurement is modeled using the Bernoulli distribution

$$\pi_{FP}(\mathrm{Z}) = \begin{cases} 1 - p_{FP}(z_j), & \text{if } \mathrm{Z} = \emptyset, \\ p_{FP}(z_j) \cdot c(z_j), & \text{if } \mathrm{Z} = \{z_j\}. \end{cases} \tag{6.2}$$

Obviously, the false positive probability $p_{FP}(z_j)$ is only available after the current set of measurements is received. Thus, the spatial distribution $c(z)$ of a false positive measurement is concentrated at the position of the measurement z_j, i.e. it is given by the Dirac delta function $\delta_{z_j}(z)$. Assuming the received false positive measurements to be independent of each other, the probability density of the resulting multi-Bernoulli process is obtained by the union of $|\mathrm{Z}|$ independent Bernoulli distributions:

$$\pi_{FP}(\mathrm{Z}) = \{p_{FP}(\cdot), c(\cdot)\}_{z \in \mathrm{Z}} \,. \tag{6.3}$$

Following [Mah07a, Appendix G.18], the multi-object likelihood function for Poisson distributed clutter measurements is obtained by the convolution of the object detection process π_D and the clutter process π_C:

$$\pi(\mathrm{Z}|\mathrm{X}_+) = \sum_{\mathrm{W} \subseteq \mathrm{Z}} \pi_D(\mathrm{W}|\mathrm{X}_+)\pi_C(\mathrm{Z} - \mathrm{W}). \tag{6.4}$$

Here, the multi-Bernoulli object detection process π_D assumes that each object is detected with probability $p_D(x)$ and does not generate a measurement with $1 - p_D(x)$. In (6.4), each measurement $z \in \mathrm{Z}$ is exclusively assigned to the object detection or to the Poisson clutter process. Since the false positive distribution π_{FP} depends on the set of all measurements, π_{FP} and π_D are not independent. Consequently, replacing π_C with π_{FP} in (6.4) affects the validity of the convolution equation [Mah07a, pp. 385–386]. By neglecting the correlation of the multi-object probability densities, the multi-object likelihood for measurements with false positive probabilities results in

$$\pi(\mathrm{Z}|\mathrm{X}_+) \approx \sum_{\mathrm{W} \subseteq \mathrm{Z}} \pi_D(\mathrm{W}|\mathrm{X}_+)\pi_{FP}(\mathrm{Z} - \mathrm{W}), \tag{6.5}$$

where a measurement is either an object detection or a false positive. Since the false positive measurements follow a multi-Bernoulli distribution, the false positive density in (6.5) may be equivalently rewritten by

$$\pi_{FP}(\mathrm{Y}) = (1 - p_{FP})^{\mathrm{Z}} \cdot \left(\frac{p_{FP}}{1 - p_{FP}}\right)^{\mathrm{Y}} \tag{6.6}$$

$$= (1 - p_{FP})^{\mathrm{Z}} \cdot \left(\frac{p_{FP}}{1 - p_{FP}}\right)^{\mathrm{Z}} \cdot \left(\frac{1 - p_{FP}}{p_{FP}}\right)^{\mathrm{W}} \tag{6.7}$$

$$= (p_{FP})^{\mathrm{Z}} \cdot \left(\frac{1-p_{FP}}{p_{FP}}\right)^{\mathrm{W}} , \tag{6.8}$$

where $\mathrm{Y} = \mathrm{Z} - \mathrm{W}$ and $c(z) = 1$. Consequently, the probability that all measurements are false positives is given by

$$\pi_{FP}(\mathrm{Z}) = (p_{FP})^{\mathrm{Z}} . \tag{6.9}$$

Using (6.8) and (6.9), the convolution formula in (6.5) can be rearranged:

$$\pi(\mathrm{Z}|\mathrm{X}_+) \approx \pi_{FP}(\mathrm{Z}) \cdot \sum_{\mathrm{W}\subseteq\mathrm{Z}} \pi_D(\mathrm{W}|\mathrm{X}_+) \left(\frac{p_{TP}}{p_{FP}}\right)^{\mathrm{W}} , \tag{6.10}$$

where $p_{TP}(\cdot) = 1 - p_{FP}(\cdot)$.

Analog to [Mah07a, Appendix G.18], the probability that a set X_+ of tracks does not generate any measurement is given by

$$\pi(\emptyset|\mathrm{X}_+) = \prod_{x\in\mathrm{X}_+} (1 - p_D(x)) \tag{6.11}$$

which is independent of the false positive process since $\pi_{FP}(\emptyset) = 1$. Finally, the multi-object likelihood function (6.10) is rewritten by means of the proof in [Mah07a, Appendix G.18]:

$$\pi(\mathrm{Z}|\mathrm{X}_+) \approx \pi_{FP}(\mathrm{Z})\pi(\emptyset|\mathrm{X}_+) \cdot \sum_{\theta} \prod_{i:\theta(i)>0} \frac{p_D(x_+^{(i)}) \cdot g(z_{\theta(i)}|x_+^{(i)}) p_{TP}(z_{\theta(i)})}{(1 - p_D(x_+^{(i)})) \cdot p_{FP}(z_{\theta(i)})} . \tag{6.12}$$

The properties of the proposed multi-object likelihood function (6.12) are illustrated using a predicted multi-object state which is given by the Bernoulli distribution

$$\pi_+(\mathrm{X}_+) = \begin{cases} 1 - r_+, & \text{if } \mathrm{X}_+ = \emptyset, \\ r_+ \cdot p_+(x), & \text{if } \mathrm{X}_+ = \{x\} \end{cases} \tag{6.13}$$

with $r_+ = 0.4$. The sensor delivers only a single measurement and the detection probability is $p_D = 0.9$. Figure 6.13 depicts the posterior existence probability r of the object depending on the false positive probabilities p_{FP} and the spatial likelihood $g(z|x_+)$. A measurement with a false positive probability $p_{FP} > 0.5$ reduces the posterior existence probability of the object for all $0 < g(z|x_+) < 1$. In contrast, a false positive probability $p_{FP} < 0.5$ increases the object's existence probability if the spatial likelihood exceeds a threshold which depends on p_{FP}. For $p_{FP} = 0.5$, the posterior existence probability matches the predicted existence probability if $g(z|x_+) = 1$. In GM implementations, a limitation of the spatial likelihood to $0 < g(z|x_+) < 1$ is obtained by neglecting the normalizing constant of the Gaussian distribution $\mathcal{N}(z; z_+, \underline{S})$.

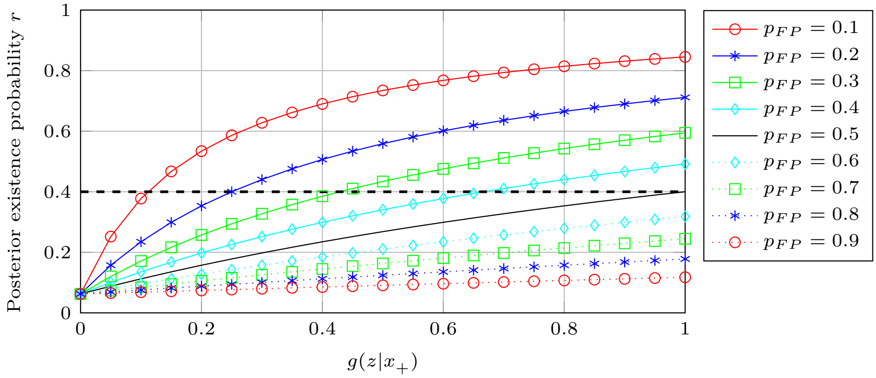


Figure 6.13: Posterior existence probability r of an object with $r_+ = 0.4$ for different measurement likelihoods $g(z|x_+)$ and false positive probabilities p_{FP}. The detection probability is $p_D = 0.9$.

The observed results meet one's expectations and indicate that (6.12) is a suitable approximation for applications where the clutter process does not follow a Poisson distribution and the sensor measurements provide a false positive probability. Further, the result of the update using the proposed multi-object likelihood function (6.12) matches the result of the JIPDA update for the example in [Mun11, pp. 46ff.] which comprises only one track and one measurement.

6.4 Pedestrian Tracking in Indoor Applications

The SMC multi-object Bayes filter is applied to pedestrian tracking in an indoor application where two laser range finders are mounted in the corners of a room. The static sensor setup facilitates a background suppression using occupancy grid mapping [TBF05] in order to reduce the number of measurements. However, the short distances between the sensors and the objects lead to multiple measurements per object.

6.4.1 System Setup

Segmentation

In order to be able to use the point target assumption within the SMC-MOB filter, a segmentation algorithm has to be applied to the raw data of the laser range finders to obtain object hypotheses. In scenarios with closely spaced objects, it is not possible to determine a single distance threshold for the segmentation algorithm which ensures that each object generates exactly one object hypothesis. While a distance threshold delivers the correct segmentation result in one situation, it may lead to over or under segmentation in the next situation. Obviously, any false decision in the segmentation algorithm directly affects the performance of the tracking algorithm.

In the author's conference paper [RD09], a fuzzy segmentation approach is proposed which avoids hard decisions in the segmentation algorithm by returning several segmentation hypotheses. Further, the fuzzy segmentation approach assigns a confidence value to each hypothesis which depends on the spatial proximity of the measurements in a hypothesis and the dimension of the obtained segment. Using the angular measurement principle of the laser range finders, the likelihood that two succeeding measurements belong to the same object depends on the radial and the angular distance. While the maximum radial distance of two succeeding measurements depends on the object type, the maximum angular distance depends on the detection probability of the sensor. The likelihood is obtained using two fuzzy membership functions which model the connectedness of the measurements based on their radial and angular distance. Afterwards, the set of all measurement points is split recursively at the positions with the smallest likelihood until the minimum likelihood for the set of measurements is above an application specific threshold. During the splitting procedure, the likelihoods are used to attach a confidence value to each of the obtained subsets. If the minimum likelihood for a set of points is very small, a high confidence value is assigned to the subsets obtained by the splitting and vice versa. Since the fuzzy segmentation is applied separately to the data of the two laser range finders, a subsequent cross-validation is used to associate corresponding subsets of both sensors and to obtain segmentation hypotheses and their corresponding confidence values $c_c(z_j)$.

Within the SMC-MOB filter, the confidence values of the segmentation algorithm are interpreted as true positive probabilities of the segmentation hypotheses. To avoid numerical instability, the true positive probability is defined as

$$p_{TP}(z_j) = \begin{cases} \epsilon & \text{if } c_c(z_j) < \epsilon \\ c_c(z_j) & \text{if } \epsilon < c_c(z_j) < 1-\epsilon \\ 1-\epsilon & \text{if } c_c(z_j) > 1-\epsilon, \end{cases} \tag{6.14}$$

where the parameter ϵ ensures that the true positive probability is in the open interval $(0, 1)$. Within the SMC-MOB filter, using several segmentation hypotheses does not lead to a bias in the cardinality estimate since a multi-object particle with several objects at approximately the same position obtains a reduced weight in the prediction step in case of the set based weight adaption. Additionally, the adaptive birth intensity assigns a small birth probability $r_B(z_j)$ to measurements which are close to existing objects and consequently mitigates the initialization of new objects in this case.

State Dependent Detection and Survival Probabilities

In Section 4.4, an approach for the calculation of a state dependent detection probability is proposed. However, the approach requires the calculation of the detection probability for each multi-object particle. Since the considered application is only intended to track pedestrians, it facilitates the approximation of the state dependent detection probability using a measurement grid [KSD10; TBF05]. Since the background subtraction using an occupancy grid already requires the calculation of the measurement grid, the computational cost is negligible compared to the separate calculation for each multi-object particle.

A measurement grid facilitates the partitioning of the grid cells into the three categories occupied, free, and hidden. In principle, the measurement grid is built up as follows: First, all grid cells m_i are initialized with $p_m(m_i) = 0.5$. Afterwards, all measurements have to be incorporated in the measurement grid considering the corresponding measurement uncertainties. Hence, the value of a grid cell is increased, if it is located within the measurement uncertainty of one of the measurements. Grid cells with a value $p_m(m_i) > 0.5$ are treated as occupied. All grid cells between the sensor and occupied cells are considered as free space. In [KSD10], several possibilities for free space models are presented. However, for the considered application a free space modeling using a constant value of $p_m(m_i) = 0.1$ is sufficient. The occupancy grid is updated with the measurement grid using the binary Bayes filter [TBF05] which calculates the occupancy probability of each grid cell independently.

Based on the current measurement, the measurement grid represents hidden or occluded grid cells m_i by the probability $p_m(m_i) = 0.5$. Thus, the measurement grid approximately represents the state dependent detection probability which is desired for the SMC-MOB filter. Since laser range finders only measure object contours and the resolution of the grid cells is 5 cm, all grid cells within the extend of an object are marked as hidden. In order to mitigate this effect, the average size of pedestrians (circular shape with diameter 40 cm) is integrated into the measurement grid and only grid cells which are more than 40 cm behind a measurement are marked as hidden. Thus, an application of this approach to tracking systems for multiple object classes is not possible in general and the detection probability has to be determined using the approach proposed in Section 4.4.

The sensor setup comprises two laser range finders and the measurement grid as well as the adapted measurement grid for the representation of the detection probability are calculated independently for both sensors. Since the segmentation approach allows the generation of an object hypothesis if an object is only visible for one of the sensors, a grid cell is only considered as occluded if both sensors are not able to detect objects in this cell. Thus, the state dependent detection probability of each grid cell is given by

$$p_D(m_i) = \begin{cases} 1 - p_D & \text{if } p_m^{(1)}(m_i) = p_m^{(2)}(m_i) = 0.5, \\ p_D & \text{otherwise,} \end{cases} \tag{6.15}$$

where $p_m^{(1)}$ and $p_m^{(2)}$ denote the sensor individual measurement grids. Further, p_D represents the state independent detection probability of the laser range finders.

The state dependent detection probability $p_D(m_i) = 1 - p_D$ is visualized in Figure 6.14 using red circles. The measurements of the laser range finders are separated into static (marked by black diamonds) and dynamic (marked by blue and green dots) points using the occupancy grid. Five pedestrians are located in the visible area and the black ellipses illustrate the result of the fuzzy segmentation algorithm. Another person stands in the occluded area at $(x, y) \approx (-0.9\text{m}, 5.4\text{m})$. As expected, a small detection probability is assigned to the area next to the occluded pedestrian. The area covered by the five visible pedestrians obtains a high detection probability since the object extend is considered in the detection probability model.

The state dependent detection probability given by (6.15) does not deliver an accurate estimate for the detection probability of an object whose center is located close to a transition between a high and a low detection probability. Assuming again that pedestrians have approximately a circular shape with a diameter of 40 cm, the object extend is incorporated in the detection probability grid using a convolution with a circular kernel function [RD10] which smooths the transition between high and low detection probability.

Due to the static obstacles in the environment, a constant survival probability within the field of view of the sensors is generally not sufficient. However, the occupancy grid provides the required information to obtain a state dependent survival probability. Since pedestrians may be located at grid cells which are not occupied by a static obstacle, a survival probability of $p_S = 0.995$ is assigned to all objects whose corresponding grid cell m_i has an occupancy probability of $p_o(m_i) < 0.5$. In contrast, a pedestrian may not be located at grid cells which are occupied by a static obstacle. Thus, a survival probability of $p_S = 0.1$ is assigned if $p_o(m_i) > 0.5$. In addition to free and occupied grid cells, the occupancy grid usually comprises grid cells which

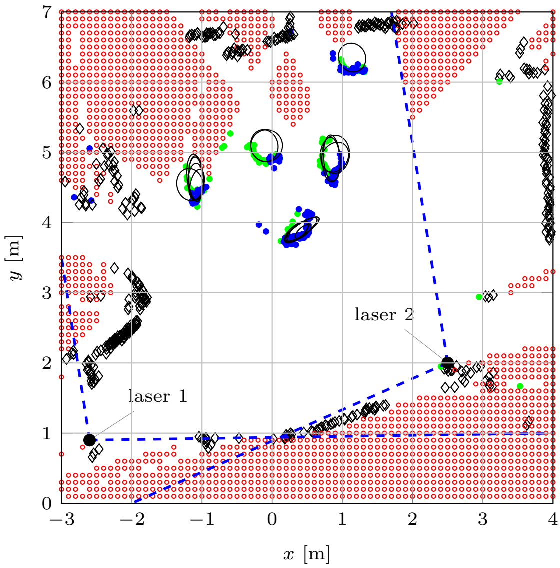


Figure 6.14: Pedestrian tracking scenario: black diamonds depict measurements of static objects, blue and green dots represent measurements of dynamic objects. Black ellipses illustrate the results of the segmentation algorithm. Areas which are occluded for both sensors are marked by red circles. The dashed blue lines depict the FOV of the sensors.

have never been observed by the sensor. In order to allow for the continuous tracking of an object over a small unobservable area and to mitigate objects spinning around in unobservable areas for a long time, a survival probability of $p_S = 0.8$ is assigned to objects located within these cells. Obviously, the choice of p_S depends on the size of the areas which are outside of the sensor's field of view. Empirical tests have shown, that $p_S = 0.8$ is a good choice for the considered sensor setup, since the unobservable areas are quite small and the pedestrians typically need only a few measurement cycles to pass these areas.

Adaptive Birth Probability in Scenarios with Occlusion

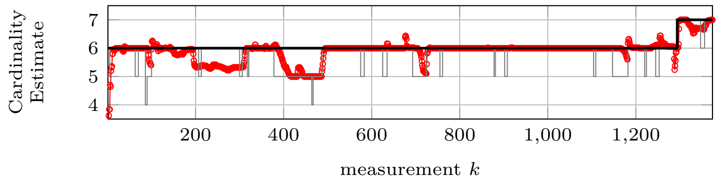
Scenarios with state dependent detection probabilities and occlusions pose a challenge for the birth model which is part of the prediction step. The adaptive birth density proposed in Section 4.5 tends to initialize a new born object and to keep the occluded object when the occluded object reappears. Similar effects are also expected in case of a static birth density. Since the particles corresponding to the occluded object are scattered over the occluded area, a multi-object particle may contain a particle for the object at one end of the occluded area while the object reappears at the other end. Thus, the measurement of a reappearing object may have an insignificant likelihood for the existing particle of the object since the spatial uncertainty of the object is not represented within a single multi-object particle. Consequently, the measurement tends to initialize a new object in this situation.

In order to mitigate this effect, the label information is incorporated into the calculation of the birth probability in case of a state dependent detection probability. Thus, a very low birth probability is assigned to a measurement z_j if the measurement has a high likelihood for an object with label ℓ in any of the multi-object particles and the current multi-object particle already contains an object with label ℓ. In contrast, a high birth probability is assigned if a measurement has only high likelihoods for objects whose label is not contained in the current multi-object particle.

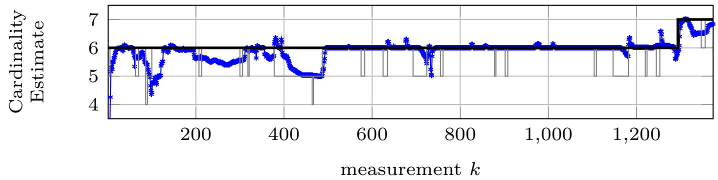
6.4.2 Results

The SMC-MOB filter is evaluated for a scene with up to seven pedestrians and several occlusions. The sequence is labeled in order to obtain ground truth positions of the objects. Using a state dependent detection probability, the SMC-MOB filter is expected to keep track of objects which are occluded for a short time. Since the scenario only contains occlusions with less than 0.8 seconds or more than 2.4 seconds, the OSPAT distance assumes the filters to be able to deliver continuous tracks for objects which are occluded for at most 0.8 seconds. In other words, track ID switches after an occlusion of less than 0.8 seconds are penalized while a track ID switch is expected after an occlusion with more than 2.4 seconds.

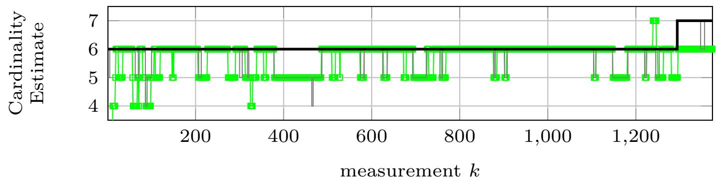
The results of the SMC-MOB filter which uses a state dependent detection and survival probability are compared to the results of a GM-LMB filter with state independent detection and survival probabilities. Furthermore, two variants of the SMC-MOB filter are compared. Both versions use $N = 25000$ multi-object particles. One of the SMC-MOB filters uses the approximate multi-object likelihood function. Due to the approximation, the filter is capable to handle all object hypotheses of the fuzzy segmentation algorithm and runs in real-time on a Nvidia Tesla C2075 GPU. The second SMC-MOB filter uses the exact multi-object likelihood function and is not real-time capable. Since the filter is not capable to handle the huge number of object hypotheses of the fuzzy segmentation algorithm, the object hypotheses are merged to reduce the computational load. The LMB filter also uses the merged hypotheses.



(a) SMC-MOB filter using approximate multi-object likelihood function.



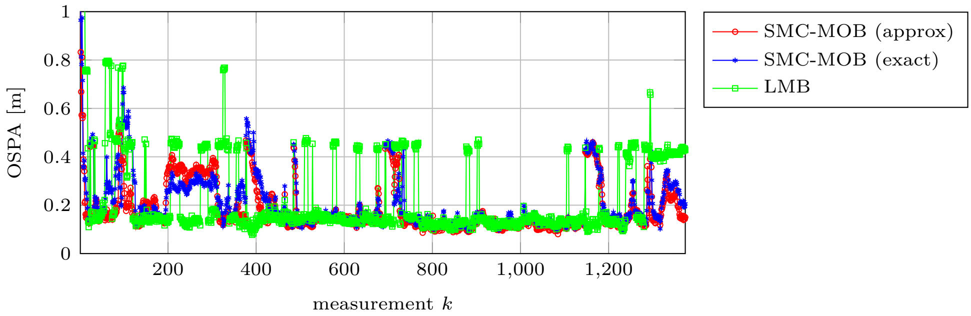
(b) SMC-MOB filter using exact multi-object likelihood function.



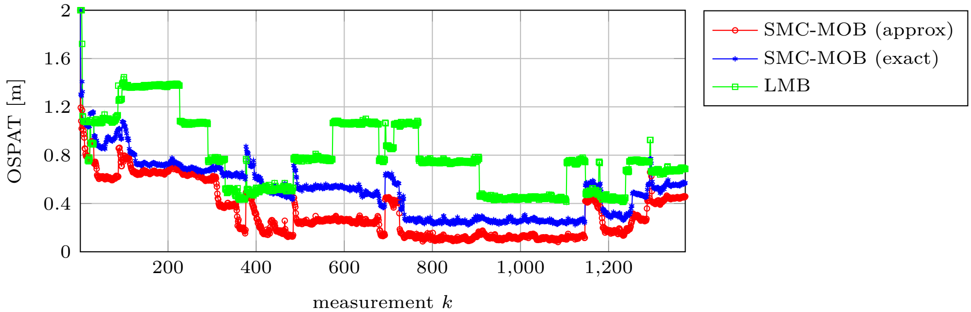
(c) LMB filter.

Figure 6.15: Cardinality estimates of the SMC-MOB filter (with exact and approximate likelihood calculation) and the LMB filter for the pedestrian tracking scenario averaged over 50 Monte Carlo runs. The thick black line depicts the number of objects in the scene, while the thin grey line illustrates the number of objects which are not occluded for the sensors.

Figure 6.15 shows the cardinality estimates of the three tracking algorithms. Due to the state dependent detection probability, the SMC-MOB filters are capable to keep track of an object which is occluded for a short time (e.g. around $k \approx 900$). The cardinality estimates for the SMC-MOB filters also illustrate that the filters continue to track occluded objects for quite a long time (e.g. before $k = 1200$). In contrast, the GM-LMB filter with constant detection probability loses these tracks in the same situation. Since the SMC-MOB filter with approximate likelihood calculation uses all object hypotheses obtained from the fuzzy segmentation algorithm, its cardinality estimates are more accurate at the beginning of the scenario ($k < 200$) than the ones of the other filters. Between $k \approx 200$ and $k \approx 300$, the SMC-MOB filters lose one of



(a) OSPA distances (order $p = 1$, cut-off $c = 2$ m).



(b) OSPAT distances (order $p = 1$, cut-off $c = 2$ m, $\alpha = c$).

Figure 6.16: OSPA and OSPAT distances for the pedestrian tracking scenario (averaged over 50 MC runs).

the tracks in approximately every second run since one of the persons leans against the back of a chair which is registered in the occupancy grid as a static obstacle. Hence, approximately half of the particles representing this person are located within occupied grid cells and obtain a small survival probability which consequently leads to the track loss.

The OSPA and OSPAT distances of the three algorithms are depicted by Figure 6.16. The OSPA distance in Figure 6.16a indicates that all algorithms provide almost identical results in situations without occlusions (e.g. $900 < k < 1100$). However, the GM-LMB filter is prone to lose tracks during short-term occlusions which is e.g. illustrated by the small peaks around $k = 900$. Since the SMC-MOB filter uses a state dependent detection probability, it outperforms the GM-LMB filter in these situations. In case of long-term occlusions (e.g. after $k \approx 700$), the SMC-MOB filters tend to have a higher OSPA distance than the GM-LMB filter due to the fact that they may still keep track of the occluded object although it is already occluded for more than nine measurement cycles. Penalizing switches of the track IDs using the OSPAT distance in Figure 6.16b, the SMC-MOB filters significantly outperform the GM-LMB filter due to the capability of keeping track of objects which are occluded for a short time. Additionally, the SMC-MOB filter using the approximate multi-object likelihood function has a smaller OSPAT distance than the SMC-MOB filter which uses the exact multi-object likelihood function. This behavior is due to the fact that the filter with the approximate likelihood calculation uses all object hypotheses of the fuzzy segmentation algorithm which significantly reduces the loss of information due to pre-processing.

6.5 Multi-Object Tracking in Automotive Applications

First, two additional simulations for typical scenarios in automotive applications are investigated to illustrate the possible performance gain of an LMB filter compared to the JIPDA filter. Afterwards, the system setup of the experimental vehicle is introduced. In Section 6.5.3, the JIPDA filter and the LMB filter are applied to two typical scenarios for automatic cruise control systems to illustrate the almost identical performance of the algorithms in unambiguous situations. Afterwards, three examples using unresolved measurements, challenging track initialization, and state dependent detection probabilities are presented to illustrate the capabilities of the LMB filter and the possible performance gains in automotive applications.

6.5.1 Simulations

Split and merge scenarios are challenging in multi-object tracking applications. Figure 6.17 depicts a typical scenario, where the starting positions of the objects are marked by a square and the destinations are illustrated by a triangle. Both objects move with an identical velocity of $v_x = 10$ m/s in x direction. The distance between the objects in y direction is decreasing from 9 m to a minimum of 3 m. Afterwards, the objects move parallel to each other for 40 measurement cycles ($k = 60$ to $k = 99$) before they are separating again. The objects are assumed to follow a constant velocity model with a standard deviation of the process noise of $\sigma_a = 4$ m/s^2 in x and y direction. The simulated sensor delivers Cartesian x and y position measurements at a frequency of 10 Hz. The standard deviation of the measurement noise is $\sigma_x = \sigma_y = 1$ m, the state independent detection probability is chosen to $p_D = 0.9$, and a true positive probability of $p_{TP} = 0.95$ is assigned to all measurements.

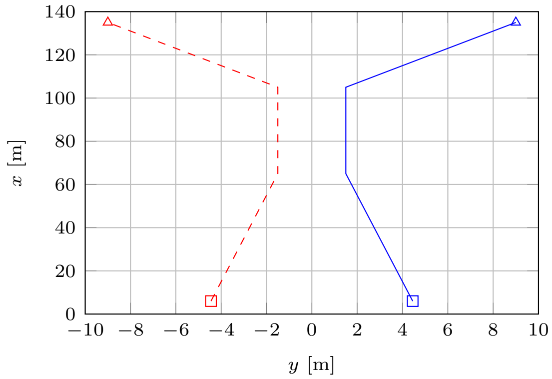
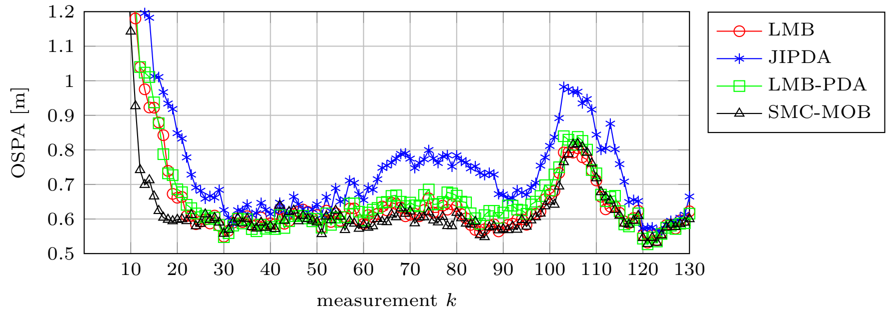


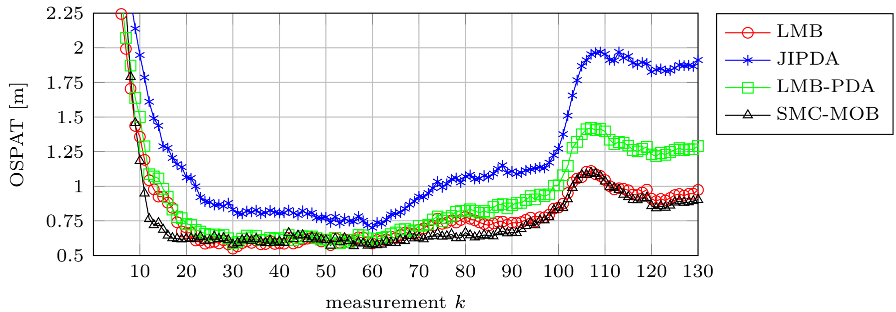
Figure 6.17: Ground truth for the split and merge scenario with two objects. Starting positions are marked by a square, destinations by a triangle.

The scenario is used to compare the performance of the LMB filter with the JIPDA and the SMC-MOB filter. All filters use matching process and measurement noises and the track extraction algorithm only outputs tracks with a maximum existence probability of $r_{max} > 0.5$ and a current existence probability $r > 0.05$. The LMB filter merges all Gaussian distributions with a Mahalanobis distance $d_{\mathrm{MHD}} < 0.2$ and prune Gaussians with a weight $w < 0.01$. One of the main differences between the JIPDA filter and the LMB filter is the representation of the posterior PDF of a track. While the JIPDA filter approximates the posterior by a single Gaussian distribution, the LMB filter retains a mixture of Gaussians. In order to illustrate the influence of this approximation, the LMB-PDA filter approximates the posterior using a single Gaussian distribution whose parameters are calculated using the PDA equations (2.40) and (2.41).

Figure 6.18a shows the OSPA distances of the five filters for the split and merge scenario. Obviously, the performance of LMB, LMB-PDA, and SMC-MOB is almost identical. However, the SMC-MOB filter has a slightly smaller OSPA distance during $60 \leq k < 100$ since it incorporates a minimum distance of the two objects of $d_{min} = 2.5$ m. The JIPDA filter obtains a marginally higher OSPA distance for $60 \leq k < 120$, i.e. while the objects move parallel to each other and start to diverge from each other. Since the LMB-PDA outperforms the JIPDA filter, the approximation of the posterior PDF using a single Gaussian distribution is not the only reason for the performance drop. The significant difference between the JIPDA filter and the LMB-PDA filter is mainly due to the different birth models. While the JIPDA implementation [MMD10] directly incorporates birth nodes in the hypotheses tree,



(a) OSPA distances.



(b) OSPAT distances.

Figure 6.18: OSPA and OSPAT distance of the filters for the split and merge scenario using a cut-off value of $c = 5$ m and order $p = 1$ (averaged over 100 MC runs). The OSPAT distance penalizes Track ID switches like missed detections using $\alpha = c$.

the LMB-PDA uses the adaptive birth model introduced in Section 5.3.6. The birth nodes of the JIPDA implementation require an accurate spatial birth density $p_B(x)$ which depends on the scenario and the current track estimates. However, an adequate modeling of $p_B(x)$ is very difficult in scenarios with closely spaced objects and an erroneous $p_B(x)$ may prevent the initialization of an appearing object.

The investigated split and merge scenario is typically prone to swap the track IDs of the objects. Figure 6.18b shows the OSPAT distance which additionally penalizes the track ID switches using the parameter $\alpha = c$, i.e. a switching track ID is penalized like a missed detection. In contrast to the OSPA results in Figure 6.18a, the OSPAT distance of the LMB-PDA filter significantly deviates from the results of the LMB and the SMC-MOB filter. Thus, an approximation of a track's posterior PDF using the PDA equations increases the number of track ID switches in the considered scenario.

The split and merge scenario depicted by Figure 6.17 starts with well separated objects. Thus, the filters are not required to handle ambiguities during object initialization. However, systems like adaptive cruise control or emergency break assists are required to initialize tracks at high distances. Due to the radial measurement principle of radar sensors, the standard deviation of a measurement increases with the range. Consequently, the track to measurement association for two closely spaced objects at a high distance is often ambiguous. Consider a scenario where the ego vehicle is heading towards the end of a traffic jam. Assume that the sensor is able to detect two non-moving cars at the end of the traffic jam whose centers are initially located at $(x, y) = (200, 3.5)$ m and $(x, y) = (200, -3.5)$ m. The ego vehicle is driving towards them with a velocity of $v_x = 55$ m/s. The detection probability is $p_D = 0.9$ and the standard deviations of the measurements for angle and range are given by $\sigma_\varphi = 1.25°$ and $\sigma_r = 0.2$ m. As in the previous scenario, a true positive probability of $p_{TP} = 0.95$ is assigned to all of the received sensor measurements.

Figure 6.19 shows the OSPA distance for the LMB, JIPDA, and LMB-PDA filter for the traffic jam scenario. Obviously, the LMB filters outperform the JIPDA filter which is mainly due to the different birth models. The LMB filter and the LMB-PDA filter deliver almost identical results for distances $x < 60$ m and $x > 160$ m between the ego vehicle and the two objects. However, for 60 m $< x <$ 160 m the LMB filter shows superior performance compared to the LMB-PDA filter since the more accurate representation of the posterior within the LMB filter facilitates the separation of the two objects at a higher distance.

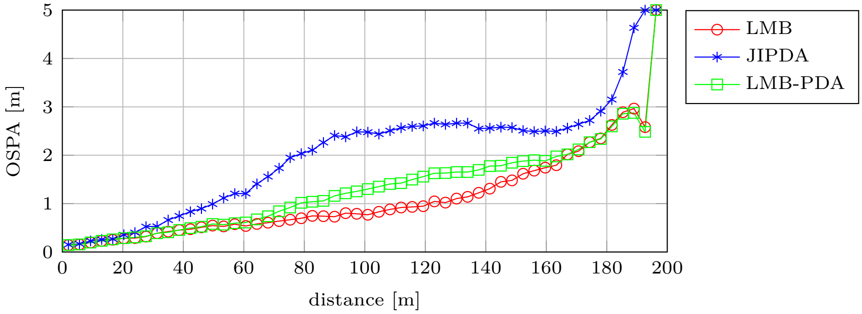


Figure 6.19: OSPA distances for LMB, JIPDA, and LMB-PDA filter for the traffic jam scenario plotted against the true distance between ego vehicle and the objects (averaged over 100 MC runs).

Assuming a width of 2 m for each of the cars, there is no need for an emergency break since there is enough space between the two cars. Consequently, it is crucial to know as soon as possible that the space between the cars is sufficient to pass by with a reduced velocity. Figure 6.20 shows the distances for 100 Monte Carlo runs, after which the distance between the objects (including one standard deviation of the uncertainty in y position) does not fall below a threshold of 2.4 m. In general, the LMB filter is able to separate the two cars at a higher distance since the LMB-PDA filter tends to have both tracks around $y = 0$ at high distances due to the approximation of the posterior PDF using a single Gaussian distribution.

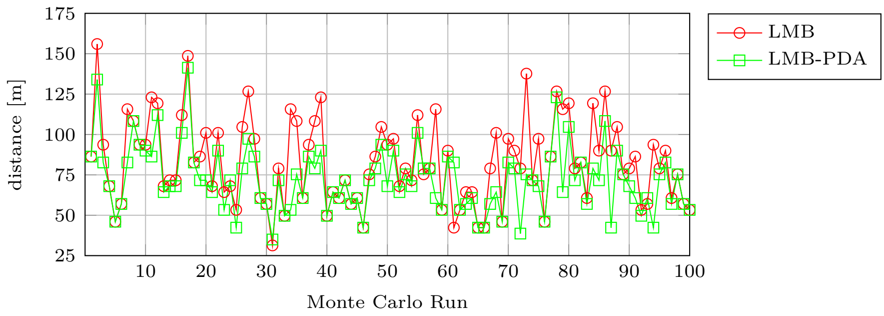


Figure 6.20: Distance to the objects after which the minimum space between the objects does not fall below 2.4 m (including one standard deviation).

6.5.2 System Setup

The GM-LMB and the JIPDA filter applied to vehicle environment perception in this section use an EKF implementation due to the non-linear constant turn rate and constant velocity (CTRV) motion model. Additionally, the measurement models of some of the sensors are non-linear and require a linearization. In order to obtain comparable results, both filters use identical measurement models. For tracking performance evaluation, ground truth information is required. The reference data is recorded using three vehicles equipped with an automotive dynamic motion analyzer (ADMA). The ADMA is a differential global positioning system (GPS) combined with a high precision inertial measurement unit. Under good receiving conditions, the ADMA achieves an accuracy of 0.02 m, which is approximately an order of magnitude more precise than the sensor data.

Ego Motion

Automotive tracking applications require the consideration of the vehicle's ego motion which causes an additional relative motion of the tracked objects between succeeding sensor measurements. The relative motion of the ego vehicle between two sensor measurements is usually represented using the absolute velocity and the yaw rate. The estimation of the ego motion is realized using the EKF implementation proposed in [Mah09b]. The measurement vector contains the wheel speeds of all wheels as well as the yaw rate which are available from the vehicle's electronic stability control system. Additionally, the pitching of the car is modeled by a stochastic noise process. The compensation of the ego motion is performed as follows: First, the filter algorithm predicts each object using its motion model. Afterwards, the motion of the ego vehicle is compensated using an additional state transition and the estimation error covariance $\underline{P}_+$ of the tracked objects is increased by the uncertainty of the ego motion.

Sensor Setup

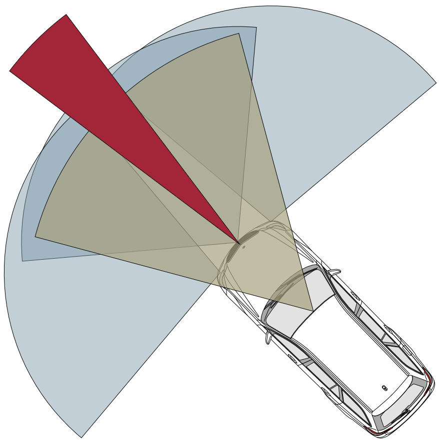


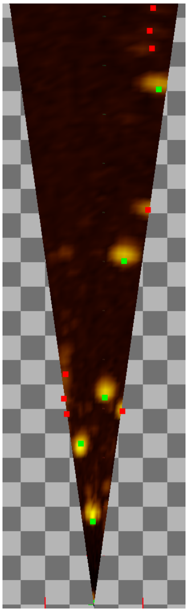
Figure 6.21: Sensor setup of the experimental vehicle, colored areas illustrate the fields of view of the sensors: laser range finders (blue), camera (beige), long-range radar (red). Maximum distance of the sensors is scaled down.

The experimental vehicle is equipped with a complementary sensor setup including radar, laser, and video sensors. Figure 6.21 illustrates the fields of view of the sensors and their mounting positions. Figure 6.22 shows an image of the gray scale video camera mounted behind the windshield of the car. The red rectangles depict the measurements obtained by the vehicle detector [GLW+13]. The detector is based on the architecture proposed by Viola and Jones in [VJ01] which uses Haar features and a cascaded processing scheme to obtain high detection rates in combination with tractable computational cost. The detector additionally assigns a true positive probability $p_{TP}(z)$ to each of the detections which corresponds to the inverse probability that the detection actually represents a car.

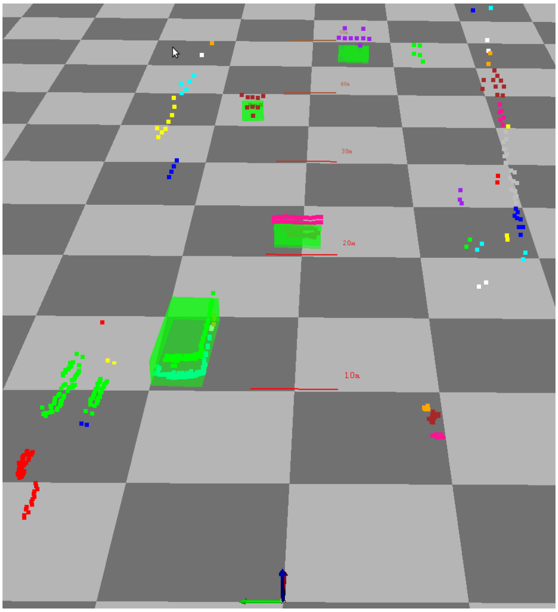
Figure 6.22: Video image of a test-run in a suburban area. Detections of the classifier are illustrated by red boxes.

Figure 6.23a shows the raw data of the imaging radar sensor Continental ARS 310 mounted in the grill of the experimental vehicle. Using the Doppler effect, the radar is able to distinguish between stationary and dynamic objects. While the video camera provides a high angular resolution, the radar provides very accurate range measurements. Consequently, a multi-sensor tracking system with radar and video sensors combines the strengths of both sensor types and achieves accurate estimates in range and angle.

The measurements of the three IBEO LUX laser range finders, which are integrated in the front bumper, are depicted by Figure 6.23b. Similar to the radar, the laser range finders deliver accurate range measurements. While the advantage of the radar sensor is the higher range coverage, the laser range finders provide a significantly higher angular resolution. Since the sensor returns multiple measurements per object, a segmentation algorithm is applied which fits boxes or lines into the raw data. For additional details about the segmentation algorithm, refer to [Mun11].



(a) Radar image.



(b) Laser point cloud and object hypotheses.

Figure 6.23: Raw data and pre-processing results of the radar and the laser range finder for the scene in Figure 6.22: (a) shows the raw data of the imaging radar and the extracted peaks of stationary (red) and dynamic (green) objects. (b) shows the raw measurements of the three laser range finders where the colors illustrate the clustering result and the object hypotheses are represented by green boxes.

Temporal Alignment and Spatial Calibration of the Sensors

A crucial point in multi-sensor tracking systems is the integration of the measurements in chronological order. Thus, a time stamp of a common time base has to be assigned to each of the measurements. Even if a correct time stamp is assigned to each measurement, it is still possible that the measurements do not arrive in correct order at the tracking system due to transmission latencies or varying computation time during pre-processing. In tracking applications, these measurements are commonly called out-of-sequence measurements [BWT11]. In order to ensure the correct order, a buffering strategy is used (e.g. [KD03; Mun11]). The tracking system only processes the oldest measurement in the buffer, if the buffer contains at least one measurement of each sensor. This ensures the correct temporal order of the measurements at the cost of a delay which depends on the minimum measurement frequency of all sensors and the latencies. In applications requiring minimum delays like e.g. a vehicle's pre-crash system [MAZ+10], a buffering strategy is usually not sufficient and more sophisticated approaches like reprocessing or retrodiction [Bar02; BCM04; MAZ+10; WDM+12] have to be applied to handle out-of-sequence measurements.

In addition to the temporal alignment of the sensor measurements, a precise spatial calibration of the sensors is crucial for a multi-sensor tracking system. For details about the calibration, refer to [Kam07; Mun11].

Extended Object Tracking Using Reference Points

In vehicle environment perception, the commonly used point target assumption is not feasible since some of the sensors return more than one measurement per object. Similar to [SWBH12], cars are represented using a rectangle with nine reference points: one at the center of the rectangle, four at the corners, and another four at the center of each side of the rectangle. The object dimensions are represented by the length and the width of the rectangle.

The reference point approach requires additional information about the measured reference points. The video detector for rear and front views of a car measures either rear center or front center. In parallel traffic, the radar sensor also returns measurements of rear center and front center while the possible reference points for the laser range finder are more complicated. If the laser range finder obtains an L-shape for the object, the most accurate reference point is the corner of the object. In case of an I-shape, the center of the line is the best choice. Consequently, the measurement models for the sensors are required to calculate predicted measurements for all possible reference points.

6.5.3 Comparison of LMB and JIPDA in Basic Applications

In this section, the performance of the LMB filter and the JIPDA filter is compared for two typical applications of an automatic cruise control system. In both scenarios, the distance between the vehicle in front and the ego vehicle varies between 10 m an 100 m. Due to high detection rates, accurate measurements, and a low object density in these scenarios, there is almost no ambiguity in the track to measurement association. Consequently, the posterior distribution of the track is supposed to be dominated by a single Gaussian distribution and the impact of the PDA approximation within the JIPDA filter is expected to be negligible in these scenarios. Thus, the performance of the JIPDA implementation proposed in [Mun11] should be almost identical to the one of the LMB filter in these situations. Both filters prune tracks with an existence probability $r < 0.001$. Additionally, the LMB filter prunes Gaussian mixtures with a weight $w < 0.01$ and merges Gaussian distributions with MHD $d_{\mathrm{MHD}} < 0.3$.

Figure 6.24 shows the OSPA distance for a 25 minute run on a highway where the two other vehicles equipped with an ADMA enter and leave the observed area several times. The huge peaks of the OSPA distance are either due to cardinality errors during object appearance and disappearance or due to inaccurate reference positions provided by the ADMA system. Obviously, the OSPA distances of the LMB and the JIPDA filter are almost identical in this scenario.

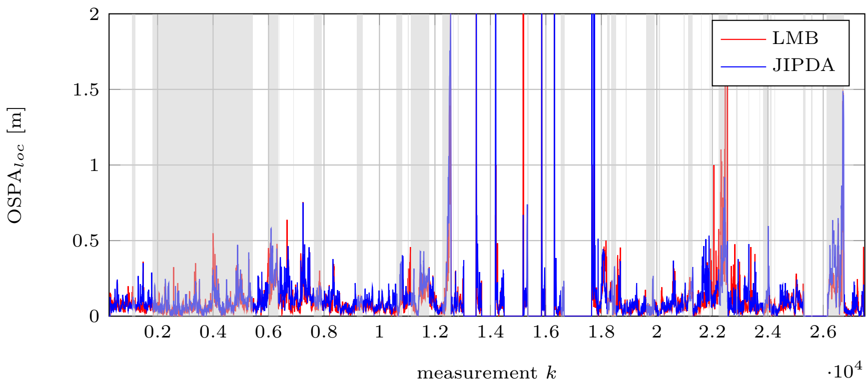
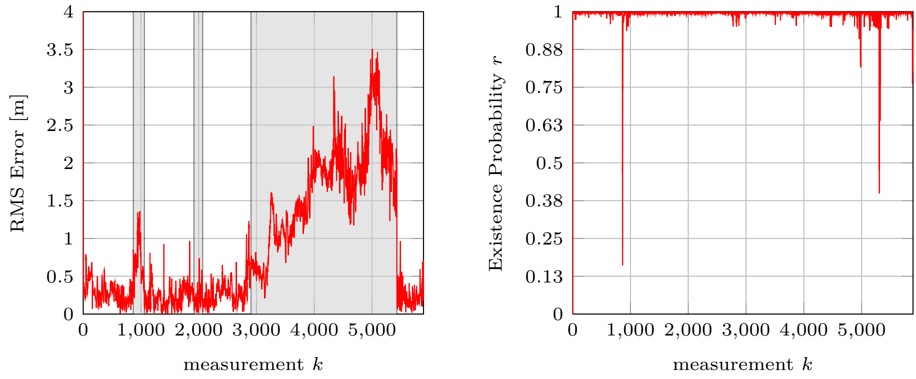


Figure 6.24: OSPA distances (order $p = 1$, cut-off $c = 2$ m) of the LMB filter and the JIPDA filter for the vehicle in front of the observing vehicle. The gray background indicates time steps where the ADMA systems did not provide the highest possible accuracy.

Figure 6.25 depicts the RMS error of the LMB filter for an 8 minute run on a winding rural road including three roundabouts. The LMB filter is able to track the vehicle in front continuously. Due to the lack of highly accurate reference positions of the ADMA systems for a long time, the RMS error for $2905 < k < 5429$ is significant. However, the very small RMS error for $k \geq 5429$ indicates that the LMB filter delivered excellent state estimates for the track during the time steps with imprecise ADMA positions. Most of the time, the estimated existence probability of the tracked vehicle is close to one. The two significant spikes are due to a few missed detections in a row.

In contrast to the highway scenario depicted by Figure 6.24, the run on the rural road comprises several situations with high accelerations in longitudinal direction and in orientation. Figure 6.26a shows the difference between the RMS error of the



(a) Root mean square error (for x and y position errors).

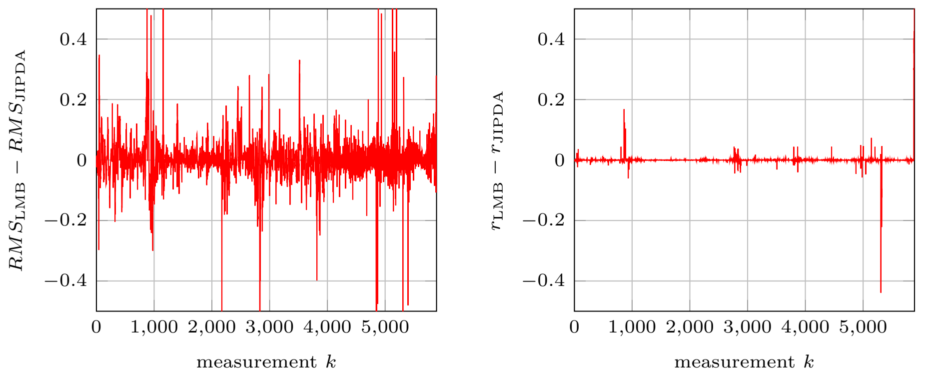
(b) Object existence probability.

Figure 6.25: Tracking results of the LMB filter for a test run on a winding rural road with three roundabouts. The gray background again depicts measurement times with inaccurate ADMA reference positions.

LMB and the JIPDA filter. The noisy difference between the two RMS errors is due to the different ways to obtain the state estimates. While the LMB filter simply returns the Gaussian distribution with the highest weight, the JIPDA filter returns the mean value of the Gaussian distribution obtained by the PDA approximation. The negligible difference between the RMS errors indicates comparable tracking result of the LMB and the JIPDA filter during $2905 < k < 5429$. Additionally, Figure 6.26b illustrates that the existence probabilities of the LMB and the JIPDA filter are almost identical. However, there are four time windows with slightly higher variance between the existence probabilities. The time slots correspond to the three roundabouts and a sharp corner where the deviation between the predicted state and the actual measurement is higher and a higher number of false alarms is returned by the sensor.

6.5.4 Partially Unresolved Measurements in Ambiguous Situations

In vehicle environment perception, tracking two cars driving side by side at a high distance is a challenging scenario. Since current radar sensors are not able to separate the two objects due to identical radial distance and velocity and a small angular difference, the sensor is likely to return only a single measurement for the two cars



(a) Difference of the RMS errors (b) Difference of the existence probabilities

Figure 6.26: Comparison of the tracking results of the LMB filter and the JIPDA filter for the test run on a winding rural road with three roundabouts.

which may be located on one of the cars or even between the cars. Thus, this scenario is even more challenging than the split and merge scenario evaluated in Section 6.5.1. Figure 6.27 exemplarily shows the estimated y positions for two cars obtained by an LMB filter using only radar measurements. Obviously, the filter is only able to separate the tracks during the first seconds. Afterwards, the tracks are switching several times due to the unresolved measurements. Further, the estimates for the car on the left lane (corresponding to the higher y value) tend to even higher y values due to reflections at the guardrail which occur at the same time as the unresolved measurements.

Since the video sensor has a significantly higher angular resolution, a multi-sensor tracking algorithm using radar and video is supposed to show significantly better tracking results in this situation. Due to the ambiguities in the scenario, a filter which facilitates a state representation using a Gaussian mixture is expected to show superior performance compared to a filter which represents the posterior using a single Gaussian distribution. In order to prevent the influence of the birth model used in the JIPDA implementation, the LMB filter is compared to the LMB-PDA filter.

Figure 6.28 shows the results for the LMB and the LMB-PDA filter for a scenario whose distance profile is depicted by Figure 6.28b. The LMB filter merges all Gaussian distributions with a Mahalanobis distance below 0.3. While the LMB filter is able to track both objects continuously (see Figure 6.28c), the LMB-PDA filter loses and re-initializes the track with the smaller y position (see Figure 6.28d) due to the decreasing distance between the objects. In addition to several unresolved radar

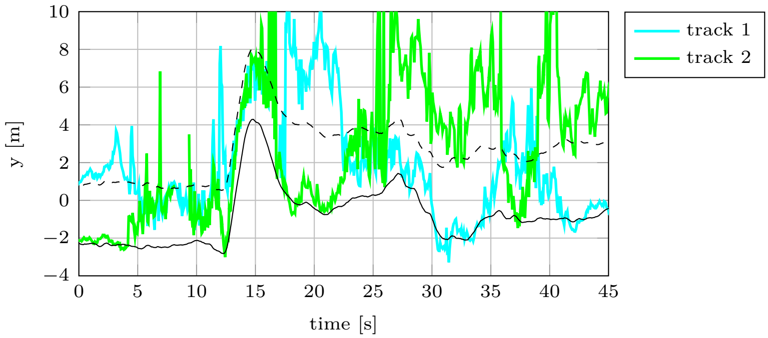
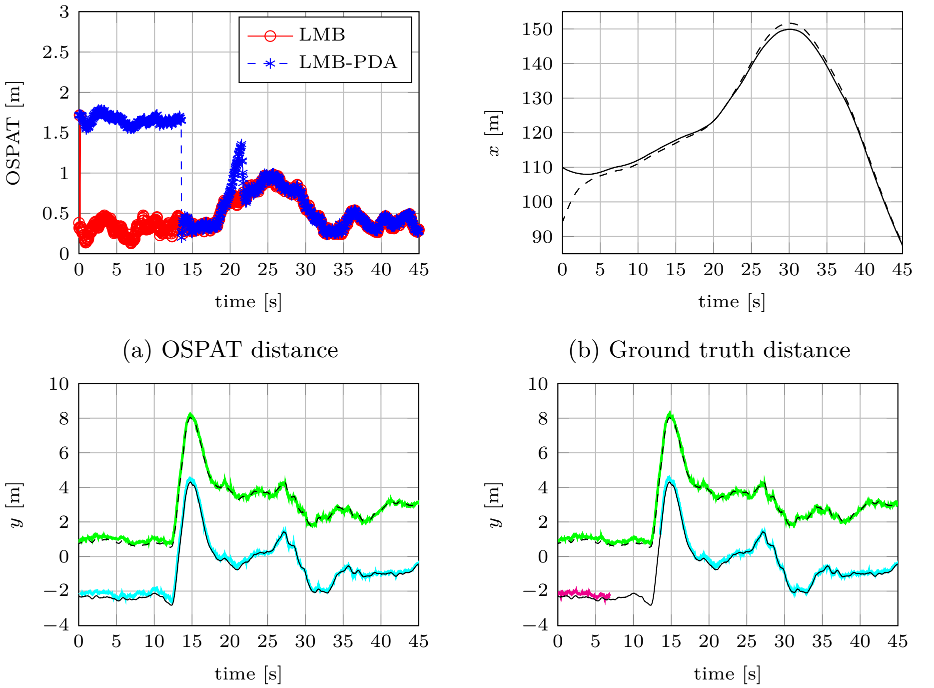


Figure 6.27: Radar only tracking using the LMB filter in case of unresolved measurements. Black lines illustrate the ground truth y positions of the two cars, green and blue line depict the estimated y positions.

measurements, several succeeding detections of the video sensor in between the two cars increase the ambiguity of the scenario at the time of the track loss. Due to the more accurate representation of the posterior PDFs of the tracks, the LMB filter significantly outperforms the LMB-PDA filter with respect to the OSPAT distance depicted by Figure 6.28a. Since the OSPAT determines the best match of ground truth tracks and estimated tracks, the blue and the green track are assigned to the two cars. Consequently, the OSPAT distance of the LMB-PDA filter during the first 15 seconds is significantly higher than the one of the LMB filter due to the mis-matching label of the red track and the subsequent cardinality error.

6.5.5 Multi-Hypotheses Track Initialization

In object tracking applications, sensors commonly observe only a part of the object's state vector. If the sensor is not able to measure the object's velocity, tracks are often initialized using a two-step initialization, which calculates the velocity of an object using two successive position measurements. However, the two-step initialization requires accurate range measurements to obtain a suitable velocity estimate. Since a video camera projects the three dimensional world to the two dimensional image plane, the range information is lost. However, the distance of an object detected in a video image is proportional to the width of the corresponding detection (see Figure 6.22). Using a prior width w_{car} for vehicles, the object's distance is determined using



(a) OSPAT distance

(b) Ground truth distance

(c) Estimated y positions LMB filter

(d) Estimated y positions LMB-PDA filter

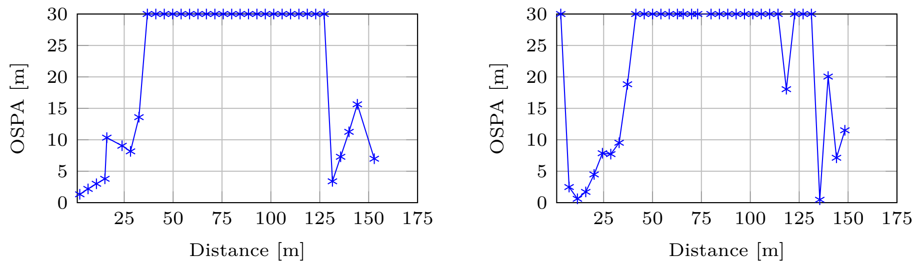
Figure 6.28: Results for a highway scenario with two cars driving side by side at a high distance. In (c) and (d), the black lines illustrate the ground truth y positions. The green, blue, and red lines depict the estimated y positions where each color represents a specific track ID.

the width of the current detection[1]. While the obtained distance is quite accurate for objects at low distances, the calculated distances of objects which are far apart are very uncertain. Consequently, a two-step initialization is likely to initialize objects with erroneous velocities which may even point to the opposite direction.

Figure 6.29 shows the tracking results of an LMB filter with two step initialization for an oncoming vehicle using only a mono video camera. The relative speed of the oncoming vehicle is approximately 55 m/s, which is typical for rural roads. Although the filter initializes a track at a distance of about 150 meters, it is not able to deliver

[1]Using the flat-world assumption, the intrinsic and the extrinsic camera parameters also facilitate the calculation of an object's distance. However, this approach delivers inaccurate range estimates at high distances if the flat-world assumption is not fulfilled (e.g. due to pitching).

satisfactory state estimates due to the error-prone initialization of the vehicle's velocity.



(a) Sequence 1 (b) Sequence 2

Figure 6.29: OSPA distance ($c = 30$ m, $p = 1$) for video-only tracking of an oncoming vehicle using an LMB filter with two-step initialization. The x axis illustrates the ground truth distance of the other vehicle.

Since the spatial distribution of each LMB track comprises a mixture of Gaussians, an LMB RFS enables a track initialization using multiple hypotheses, where each hypothesis represents a possible object state. The possible velocities of the objects depend on the permitted maximum velocity for the current road. For a rural road, the absolute velocity of other road users is approximately in the range $[0, 30]$ m/s. To handle preceding and oncoming vehicles, five hypotheses with the velocities

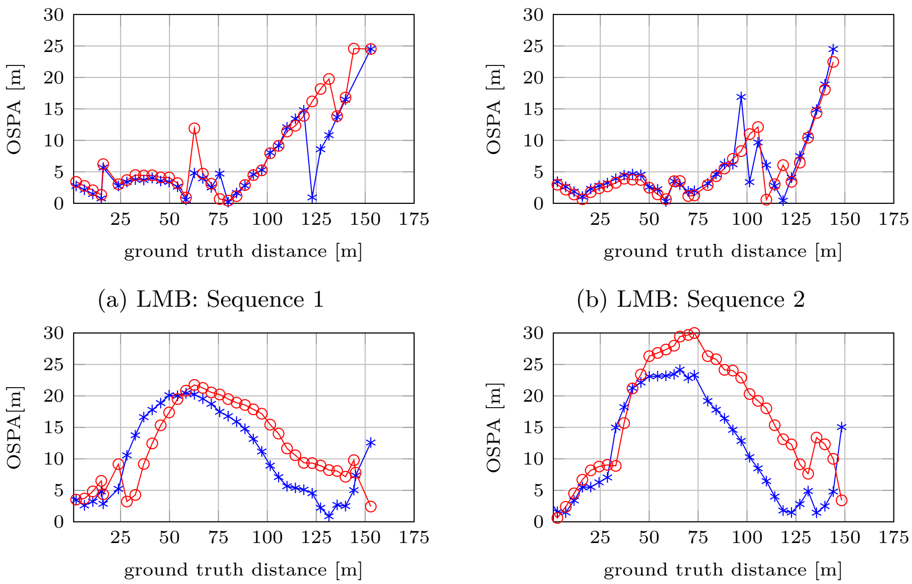
$$v_1 = -\frac{100}{3.6} \text{ m/s}, \quad v_2 = -\frac{50}{3.6} \text{ m/s}, \quad v_3 = 0 \text{ m/s}, \quad v_4 = \frac{50}{3.6} \text{ m/s}, \quad v_5 = \frac{100}{3.6} \text{ m/s}$$

are used and the standard deviation of the velocity is chosen to $\sigma_v = 25$ m/s. In addition to the velocities of the objects, the width is also unknown. To handle the range between small cars and trucks, four different object widths are used:

$$w_1 = 1.6 \text{ m}, \quad w_2 = 1.8 \text{ m}, \quad w_3 = 2 \text{ m}, \quad w_4 = 2.5 \text{ m}.$$

The corresponding standard deviation for the width is chosen to $\sigma_w = 0.1$ m. Consequently, each new born object is initialized using a total number of 20 hypotheses with equally distributed weights.

Figures 6.30a and 6.30b show the video only tracking results of the LMB filter in case of an initialization with several hypotheses. The number of Gaussian distributions is truncated using a merging procedure which merges Gaussians with Mahalanobis distance $d_{\text{MHD}} < 0.3$. Additionally, mixture components with weight $w < 0.01$ are pruned. Due to the improved track initialization, the LMB filter is able to track the oncoming vehicle continuously. The OSPA distance for the oncoming vehicle is below 5 m for ground truth distances below 80 m. Since the video sensor does not provide range information, the OSPA distance is dominated by the uncertain range estimate. In addition, Figures 6.30c and 6.30d depict the results of the LMB-PDA filter which approximates the posterior distribution of each track using a single Gaussian distribution. Obviously, the approximation significantly affects the filter performance since the multi-modal PDFs of the tracks are approximated using a single Gaussian distribution.



(a) LMB: Sequence 1 (b) LMB: Sequence 2

(c) LMB-PDA: Sequence 1 (d) LMB-PDA: Sequence 2

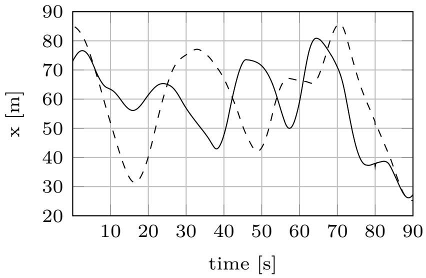
Figure 6.30: OSPA distance ($c = 30$ m, $p = 1$) for video-only tracking of an oncoming vehicle using an LMB filter with multi-hypotheses initialization. The x axis illustrates the ground truth distance of the other vehicle. Results depicted by a red circle use all widths w_1 to w_4 while only w_1 to w_3 are used for the results with the blue star.

6.5.6 Using State Dependent Detection Probabilities in Multi-Sensor Systems

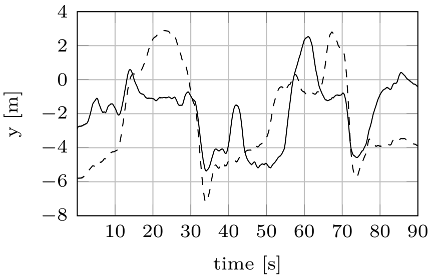
In multi-sensor fusion systems with complementary sensor setup, the measurements of two sensor types are often contradictory due to different detection capabilities. A typical example is given by the following situation: two observed cars are driving in a row. If the ego vehicle is located on the same lane, the foremost car is occluded for the video camera due to the other car. However, the radar sensor is usually able to deliver detections for both of the cars. In this scenario, the association of a detection in the video image is ambiguous if a track for both objects is initialized and a state independent detection probability is used for the video camera.

The update step of the LMB filter allows for an explicit modeling of a state dependent detection probability by incorporating the occlusion due to other objects. By representing the predicted LMB distribution in δ-GLMB form (see Section 5.3.4), the considered scenario results in four different predicted sets of track labels I_+. Obviously, the sets with cardinality $|I_+| \leq 1$ do not require the calculation of a state dependent detection probability, since at most one object exists. In case of $|I_+| = 2$, the detection probability of the furthermost car is significantly reduced due to the occlusion by the other car. In order to limit computational cost, the calculation of the state dependent detection probability neglects the state uncertainty of the objects. Hence, the predicted measurements for the video sensor are simply given by unrotated rectangles in the image plane. Consequently, an object is marked as occluded for the video sensor if its corresponding rectangle intersects with at least one rectangle of an object located closer to the sensor.

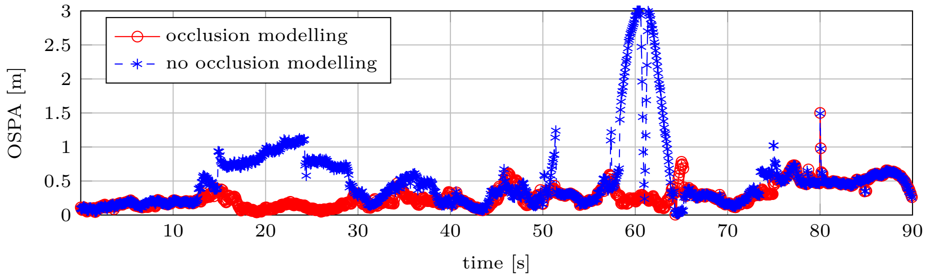
In the considered scenario, the video sensor has a detection probability of $p_D^V = 0.9$ and the detection probability of an occluded object is set to $p_D^{occ} = 0.01 p_D^V$. Figure 6.31 depicts the ground truth positions and the tracking results for two vehicles in front which overtake each other several times on a highway. The LMB filter with occlusion modeling significantly outperforms the filter with state independent detection probability in terms of the OSPA and the OSPAT distance. The OSPA distance in Figure 6.31c illustrates that the LMB filter with state independent detection probability has major difficulties to resolve the two occlusions starting at $t \approx 15$ s and $t \approx 57$ s. These difficulties are due to additional detections of the video sensor within the gate of the occluded track. Additionally, the OSPAT distance in Figure 6.31d indicates that the LMB filter without occlusion modeling loses the foremost track at $t \approx 64$ s due to several missed detections of the radar. In contrast, the LMB filter with occlusion modeling is capable to keep track of the object since the reduced detection probability of the video sensor results in a higher existence probability of the occluded object.



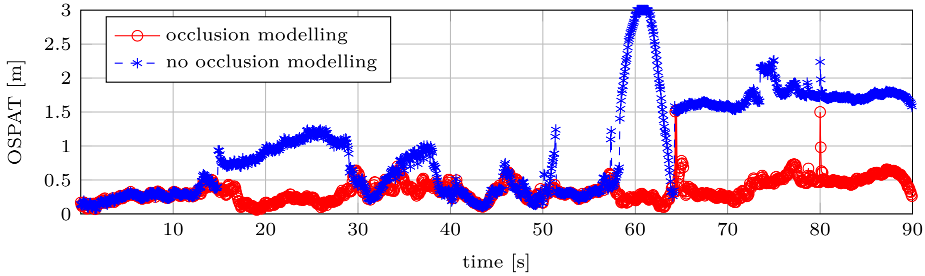
(a) Ground truth x position.



(b) Ground truth y position.



(c) OSPA distance ($c = 3$ m, $p = 1$).



(d) OSPAT distance ($c = 3$ m, $p = 1$, $\alpha = 3$).

Figure 6.31: Results for a highway scenario with several occlusions due to two vehicles in front which overtake each other several times. The OSPA and OSPAT distances compare the performance of an LMB filter with occlusion modeling to one with state independent detection probability.

6.6 Discussion

The results for the SMC-MOB filter illustrate the possible performance gains due to the incorporation of object interactions and state dependent detection probabilities. Further, the proposed approximations of the multi-object likelihood function are shown to achieve comparable results to the exact multi-object likelihood function if the minimum distances of the objects are large enough or the marginalized multi-object likelihood is used. In the application of the SMC-MOB filter to the pedestrian tracking scenario, the SMC-MOB filter outperforms the LMB filter due to its ability to model interactions and the very accurate representation of the state dependent detection probability.

The LMB filter is shown to achieve approximately the same performance as the computationally more complex δ-GLMB filter. As expected, the LMB filter significantly outperforms the CB-MeMBer and the CPHD filter. Using real-world sensor data, the LMB filter and the JIPDA filter yield approximately identical tracking results in unambiguous situations where the posterior of each track is dominated by a single Gaussian. In more challenging scenarios, the PDA approximation of the posterior distribution significantly impacts the performance of the tracking algorithms and the LMB filter outperforms the LMB-PDA filter in case of unresolved measurements and multi-hypotheses track initialization. Further, the δ-GLMB representation of the tracks within the measurement update of the LMB filter facilitates a straightforward calculation of a state dependent detection probability for each set of predicted track labels which involves improved tracking results for a video and radar fusion system during occlusions.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The key contributions of this dissertation are a real-time capable implementation of the SMC multi-object Bayes filter and a novel multi-object tracking algorithm, the labeled multi-Bernoulli (LMB) filter. The SMC implementation of the multi-object Bayes filter uses an approximate multi-object likelihood function which significantly reduces the computational complexity. Additionally, the filter explicitly models interactions using the social force model or hard-core point processes. Both approaches avoid physically impossible object states after the prediction step which additionally simplifies the data association within the filter update. The simulation results of the SMC multi-object Bayes filter as well as the results using real world sensor data indicate an excellent performance of the algorithm. However, the required number of particles for the representation of the multi-object states limits the maximum number of objects which may be tracked with the filter.

The LMB filter is an approximation of the δ-GLMB filter which approximates the posterior δ-GLMB distribution by an LMB RFS with matching intensity function. While the prediction of a δ-GLMB RFS is combinatorial, the LMB prediction features a linear computational complexity. Additionally, the LMB representation facilitates the partitioning of the predicted set of tracks and the set of measurements into independent groups, which significantly reduces the computational complexity of the filter update if the partitioning results in more than a single group. The LMB filter may also be considered as a generalized version of the CB-MeMBer filter which is not prone to a bias in the cardinality estimate due to the more accurate approximation of the multi-object posterior. Compared to the JIPDA algorithm, the LMB filter is superior since it does not require the approximation of the multi-object posterior by a single Gaussian distribution. A benefit of the LMB filter in comparison with MHT approaches is the enclosed estimation of the track existence probability and the usage of an explicit model for new born objects. The LMB filter significantly outperforms the PHD, the CPHD, and the CB-MeMBer filters and achieves approximately the same performance as the δ-GLMB filter at a significantly lower computational cost. Within the context of vehicle environment perception, the LMB filter and the JIPDA filter obtain almost identical results in situations with high detection probabilities and accurate sensor measurements. However, in challenging situations the LMB filter provides more accurate state estimates due to the more precise approximations.

7.2 Future Work

The incorporation of object destinations in the SMC multi-object Bayes filter has not been considered yet since destinations of pedestrians are not available in general. However, in the context of "Companion"-systems [WB12] a knowledge base is available which has information about upcoming activities of a user that may be utilized in the prediction step of the filter. In the context of vehicle environment perception, the destinations of the other road users are restricted by the road network. Hence, the topology of the road network as well as other information sources like the current state of traffic lights could improve the predicted object states.

The proposed implementations of the SMC multi-object Bayes filter and the LMB filter represent extended objects as circles or rectangles and require error-prone segmentation algorithms for the pre-processing of the sensor data. In case of elliptical objects, a representation of the object's size using random matrices [Koc08] is suitable and computationally tractable. In [GO12; LGO13], an incorporation of the random matrices approach into the PHD and the CPHD filter is presented and an extended object tracking using random matrices within the LMB filter is expected to show superior tracking results. However, ellipsoids are not a suitable representation for cars and trucks in vehicle environment perception applications due to the required state estimation accuracy and the approximately rectangular shape of the objects. An approach for elliptical an rectangular objects, which explicitly uses sensor characteristics to obtain the predicted measurements, is proposed in [GLO11]. In [BH11], Baum *et al.* propose a shape tracking algorithm using star-convex random hypersurface models which is able to represent arbitrary convex shapes. Within both approaches, the initialization of new born objects is critical if the objects are closely spaced. However, both approaches avoid error-prone segmentation algorithms and a combination of the approaches with the LMB filter is promising.

In addition to extended object tracking, an adequate modeling of unresolved objects is essential for further improvements of the tracking performance. The radar only tracking example in Section 6.5.4 illustrates the effect of unresolved objects: the radar sensor is not able to separate the two objects due to their small angular distance and approximately equal range and velocity. Since each object exists with a probability of one in a hypothesis of the LMB filter update, the structure of the LMB filter facilitates the modeling of unresolved measurements which is expected to provide a significant performance gain in such situations. An incorporation of object interactions in the prediction step of the LMB filter is also expected to improve the tracking results. Obviously, this requires δ-GLMB distributions during prediction which may be realized by a grouping procedure prior to the prediction step. An alternative solution is the utilization of an SMC multi-object Bayes filter for each group of interacting objects.

In addition to the state estimates provided by multi-object tracking algorithms, object classes present useful information for higher level applications like situation analysis or decision-making. Hence, multi-object tracking algorithms with integrated object identification and classification are desirable. In [VV13a], a first theoretical approach using the notation of labeled random finite sets is proposed. A conceivable approach to incorporate object classification in an LMB filter is the Dempster-Shafer [DLR77; Sha76] approach introduced in [Mun11]. Further, the classification may also be utilized to obtain class specific detection probabilities.

Finally, a comparison of the LMB filter as well as the δ-GLMB filter with MHT implementations is of great interest. The huge amount of possible MHT implementations available in the literature is counterproductive since it renders a fair comparison of the MHT filter with the LMB filter impossible. Hence, a public database containing simulated or real sensor data for a range of typical tracking scenarios would be beneficial. The experts for specific tracking algorithms may provide their simulation results for the reference data sets to ensure a fair comparison using sophisticated implementations of the algorithms.

Appendix A

Mathematical Derivations

Lemma A.1. *The variance of a multi-Bernoulli distribution* $\{(r^{(i)}, p^{(i)})\}_{i=1}^{M}$ *is given by*

$$\sigma_{\mathrm{MB}}^2 = \sum_{i=1}^{M} r^{(i)}(1 - r^{(i)}).$$

Proof. The expected number of objects represented by a Bernoulli distribution

$$\pi(\mathrm{X}) = \begin{cases} 1 - r, & \text{if } \mathrm{X} = \emptyset, \\ r \cdot p(x), & \text{if } \mathrm{X} = \{x\}. \end{cases} \tag{A.1}$$

is given by

$$\hat{N} = (1 - r) \cdot 0 + r \cdot 1 = r. \tag{A.2}$$

Thus, the variance of the number of objects is given by

$$\begin{aligned} \sigma_{\mathrm{Ber}}^2 &= (1 - r) \cdot (0 - \hat{N})^2 + r \cdot (1 - \hat{N})^2 \\ &= (1 - r) \cdot r^2 + r \cdot (1 - r)^2 \\ &= r^2 - r^3 + r - 2r^2 + r^3 \\ &= r(1 - r). \end{aligned} \tag{A.3}$$

Since the multi-Bernoulli distribution is defined as the union of M independent Bernoulli distributions, the estimated number of objects of a multi-Bernoulli distribution is given by the sum of the expected values of all M Bernoulli distributions

$$\hat{N} = \sum_{i=1}^{M} r^{(i)}, \tag{A.4}$$

and the variance is given by

$$\sigma_{\mathrm{MB}}^2 = \sum_{i=1}^{M} (1 - r^{(i)}) \cdot r^{(i)}. \tag{A.5}$$

Under the assumption that the existence probability of all Bernoulli distributions is identical, (A.5) simplifies to the variance of a Binomial distribution $\sigma_{\mathrm{Bin}}^2 = M \cdot (1 - r) \cdot r$. $\square$

Appendix B

Derivation of the Gaussian Mixture Labeled Multi-Bernoulli Filter

B.1 Gaussian Identities

The derivation of the Gaussian mixture LMB filter extensively uses the following Gaussian identities which have been used in [HL64] to derive the Kalman filter from a Bayesian point of view.

Lemma B.1. *Using a linear Gaussian process model with system matrix $\underline{\mathrm{F}}$ and covariance matrix $\underline{\mathrm{Q}}$, the prediction of a Gaussian distribution with mean $\hat{x}$ and covariance $\underline{\mathrm{P}}$ is given by*

$$\int \mathcal{N}\left(x; \underline{\mathrm{F}}\zeta, \underline{\mathrm{Q}}\right) \mathcal{N}\left(\zeta; \hat{x}, \underline{\mathrm{P}}\right) d\zeta = \mathcal{N}\left(x; \underline{\mathrm{F}}\hat{x}, \underline{\mathrm{F}}\underline{\mathrm{P}}\underline{\mathrm{F}}^{\mathrm{T}}\right) \tag{B.1}$$

Lemma B.2. *Using a linear Gaussian measurement model with measurement matrix $\underline{\mathrm{H}}$ and measurement noise covariance $\underline{\mathrm{R}}$, the innovation of a Gaussian distributed predicted object state with mean $\hat{x}_+$ and covariance matrix $\underline{\mathrm{P}}_+$ is given by*

$$\mathcal{N}\left(z; \underline{\mathrm{H}}x, \underline{\mathrm{R}}\right) \mathcal{N}\left(x; \hat{x}_+, \underline{\mathrm{P}}_+\right) = q(z)\mathcal{N}\left(x; \hat{x}, \underline{\mathrm{P}}\right) \tag{B.2}$$

where

$$\begin{aligned}
q(z) &= \mathcal{N}\left(z; \underline{\mathrm{H}}\hat{x}_+, \underline{\mathrm{H}}\underline{\mathrm{P}}_+\underline{\mathrm{H}}^{\mathrm{T}} + \underline{\mathrm{R}}\right) \\
\hat{x} &= \hat{x}_+ + \underline{\mathrm{K}}(z - \underline{\mathrm{H}}\hat{x}_+) \\
\underline{\mathrm{P}} &= \underline{\mathrm{P}}_+ - \underline{\mathrm{K}}\underline{\mathrm{S}}\underline{\mathrm{K}}^{\mathrm{T}} \\
\underline{\mathrm{S}} &= \underline{\mathrm{H}}\underline{\mathrm{P}}_+\underline{\mathrm{H}}^{\mathrm{T}} + \underline{\mathrm{R}} \\
\underline{\mathrm{K}} &= \underline{\mathrm{P}}_+\underline{\mathrm{H}}^{\mathrm{T}}\underline{\mathrm{S}}^{-1}
\end{aligned}$$

B.2 Derivation of the GM-LMB Prediction

Assume that the prior density of a track with label ℓ is given by a Gaussian mixture

$$p^{(\ell)}(x) = \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \hat{x}^{(\ell,j)}, \underline{\mathrm{P}}^{(\ell,j)}\right) \tag{B.3}$$

with mean values $\hat{x}^{(\ell,j)}$ and covariance matrices $\underline{\mathrm{P}}^{(\ell,j)}$. Further, the linear Gaussian process model follows

$$f_+(x|\xi) = \mathcal{N}\left(x; \underline{\mathrm{F}}\xi, \underline{\mathrm{Q}}\right) \tag{B.4}$$

with the state transition matrix $\underline{\mathrm{F}}$ and the according process noise covariance $\underline{\mathrm{Q}}$. Additionally, the survival probability is assumed to be state independent:

$$p_S(x) = p_S. \tag{B.5}$$

Using equations (B.3)–(B.5) and the Gaussian identity (B.1), the nominator of the predicted density $p_{+,S}(x,\ell)$ in (5.26) simplifies to

$$\begin{aligned}
&\left\langle p_S(\cdot,\ell) f_+(x|\cdot,\ell), p(\cdot,\ell)\right\rangle = \\
&\quad = \int p_S \mathcal{N}\left(x; \underline{\mathrm{F}}\xi, \underline{\mathrm{Q}}\right) \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(\xi; \hat{x}^{(\ell,j)}, \underline{\mathrm{P}}^{(\ell,j)}\right) d\xi && \text{(B.6)} \\
&\quad = p_S \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \int \mathcal{N}\left(x; \underline{\mathrm{F}}\xi, \underline{\mathrm{Q}}\right) \mathcal{N}\left(\xi; \hat{x}^{(\ell,j)}, \underline{\mathrm{P}}^{(\ell,j)}\right) d\xi && \text{(B.7)} \\
&\quad = p_S \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \underline{\mathrm{F}}\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}^{\mathrm{T}} + \underline{\mathrm{Q}}\right). && \text{(B.8)}
\end{aligned}$$

Consequently, the survival probability (5.27) of the track is obtained using integration:

$$\begin{aligned}
\eta_S(\ell) &= \int \left\langle p_S(\cdot,\ell) f_+(x|\cdot,\ell), p(\cdot,\ell)\right\rangle dx && \text{(B.9)} \\
&= \int p_S \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \underline{\mathrm{F}}\hat{x}^{(\ell,j)}, \underline{\mathrm{F}}\underline{\mathrm{P}}^{(\ell,j)}\underline{\mathrm{F}}^{\mathrm{T}} + \underline{\mathrm{Q}}\right) dx && \text{(B.10)}
\end{aligned}$$

$$= p_S \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \int \mathcal{N}\left(x; \underline{F}\hat{x}^{(\ell,j)}, \underline{F}\underline{P}^{(\ell,j)}\underline{F}^{\mathrm{T}} + \underline{Q}\right) dx \tag{B.11}$$

$$= p_S \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} = p_S. \tag{B.12}$$

Finally, the predicted PDF (5.26) of track ℓ is given by

$$p_{+,S}(x,\ell) = \frac{\left\langle p_S(\cdot,\ell) f_+(x|\cdot,\ell), p(\cdot,\ell)\right\rangle}{\eta_S(\ell)} \tag{B.13}$$

$$= \frac{p_S \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \underline{F}\hat{x}^{(\ell,j)}, \underline{F}\underline{P}^{(\ell,j)}\underline{F}^{\mathrm{T}} + \underline{Q}\right)}{p_S} \tag{B.14}$$

$$= \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N}\left(x; \underline{F}\hat{x}^{(\ell,j)}, \underline{F}\underline{P}^{(\ell,j)}\underline{F}^{\mathrm{T}} + \underline{Q}\right). \tag{B.15}$$

B.3 Derivation of the GM-LMB Update

Assume that the predicted density of a track with label ℓ is a Gaussian mixture

$$p_+^{(\ell)}(x) = \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_+^{(\ell,j)}, \underline{P}_+^{(\ell,j)}\right) \tag{B.16}$$

with mean values $\hat{x}_+^{(\ell,j)}$ and covariance matrices $\underline{P}_+^{(\ell,j)}$. The linear Gaussian measurement model is given by

$$g(z|x) = \mathcal{N}(z; \underline{H}x, \underline{R}) \tag{B.17}$$

using the measurement matrix $\underline{H}$ and the measurement noise covariance $\underline{R}$. Further, the detection probability is assumed to be state independent:

$$p_D(x) = p_D. \tag{B.18}$$

If a measurement is associated to track ℓ, i.e. $\theta(\ell) > 0$, the measurement likelihood

(5.46) is given by

$$\psi_{\mathrm{Z}}(x,\ell;\theta) = \frac{p_D(x,\ell)g(z_{\theta(\ell)}|x,\ell)}{\kappa(z_{\theta(\ell)})} \tag{B.19}$$

$$= \frac{p_D}{\kappa(z_{\theta(\ell)})}\mathcal{N}\left(z_{\theta(\ell)};\underline{\mathrm{H}}x,\underline{\mathrm{R}}\right). \tag{B.20}$$

Using the Gaussian identity (B.2), the multiplication of the predicted density (B.16) with the measurement likelihood (B.20) yields

$$p_+(x,\ell)\cdot\psi_{\mathrm{Z}}(x,\ell;\theta) = \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)}\mathcal{N}\left(x;\hat{x}_+^{(\ell,j)},\underline{\mathrm{P}}_+^{(\ell,j)}\right)\cdot\frac{p_D}{\kappa(z_{\theta(\ell)})}\mathcal{N}\left(z_{\theta(\ell)};\underline{\mathrm{H}}x,\underline{\mathrm{R}}\right) \tag{B.21}$$

$$= \frac{p_D}{\kappa(z_{\theta(\ell)})}\sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)}\mathcal{N}\left(z_{\theta(\ell)};z_+^{(\ell,j)},\underline{\mathrm{S}}^{(\ell,j)}\right)\mathcal{N}\left(x;\hat{x}^{(\ell,j,\theta)},\underline{\mathrm{P}}^{(\ell,j)}\right), \tag{B.22}$$

where

$$\underline{\mathrm{S}}^{(\ell,j)} = \underline{\mathrm{H}}\underline{\mathrm{P}}_+^{(\ell,j)}\underline{\mathrm{H}}^{\mathrm{T}} + \underline{\mathrm{R}}, \tag{B.23}$$

$$z_+^{(\ell,j)} = \underline{\mathrm{H}}\hat{x}_+^{(\ell,j)} \tag{B.24}$$

$$\hat{x}^{(\ell,j,\theta)}(\mathrm{Z}) = \hat{x}_+^{(\ell,j)} + \underline{\mathrm{K}}_{k+1}^{(\ell,j)}\left(z_{\theta(\ell)} - z_+^{(\ell,j)}\right), \tag{B.25}$$

$$\underline{\mathrm{K}}^{(\ell,j)} = \underline{\mathrm{P}}_+^{(\ell,j)}\underline{\mathrm{H}}^{\mathrm{T}}\left[\underline{\mathrm{S}}^{(\ell,j)}\right]^{-1} \tag{B.26}$$

$$\underline{\mathrm{P}}^{(\ell,j)} = \underline{\mathrm{P}}_+^{(\ell,j)} - \underline{\mathrm{K}}^{(\ell,j)}\underline{\mathrm{S}}^{(\ell,j)}\left[\underline{\mathrm{K}}^{(\ell,j)}\right]^{\mathrm{T}}. \tag{B.27}$$

The normalization constant (5.45) is given the integral over (B.22):

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \left\langle p_+(\cdot,\ell),\psi_{\mathrm{Z}}(\cdot,\ell;\theta)\right\rangle \tag{B.28}$$

$$= \int \frac{p_D}{\kappa(z_{\theta(\ell)})}\sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)}\mathcal{N}\left(z_{\theta(\ell)};z_+^{(\ell,j)},\underline{\mathrm{S}}^{(\ell,j)}\right)\mathcal{N}\left(x;\hat{x}^{(\ell,j)},\underline{\mathrm{P}}^{(\ell,j)}\right)dx \tag{B.29}$$

$$= \frac{p_D}{\kappa(z_{\theta(\ell)})}\sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)}\mathcal{N}\left(z_{\theta(\ell)};z_+^{(\ell,j)},\underline{\mathrm{S}}^{(\ell,j)}\right)\int\mathcal{N}\left(x;\hat{x}^{(\ell,j)},\underline{\mathrm{P}}^{(\ell,j)}\right)dx \tag{B.30}$$

$$= \frac{p_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right). \tag{B.31}$$

Finally, the measurement updated posterior distribution (5.44)

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \frac{p_+(x,\ell) \cdot \psi_{\mathrm{Z}}(x,\ell;\theta)}{\eta_{\mathrm{Z}}^{(\theta)}(\ell)} \tag{B.32}$$

$$= \frac{\frac{p_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right) \mathcal{N}\left(x; \hat{x}^{(\ell,j)}, \underline{\mathrm{P}}^{(\ell,j)}\right)}{\frac{p_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right)} \tag{B.33}$$

$$= \sum_{j=1}^{J_+^{(\ell)}} w^{(\ell,j,\theta)}(\mathrm{Z}) \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right) \tag{B.34}$$

is again a Gaussian mixture where the posterior weights are given by

$$w^{(\ell,j,\theta)}(\mathrm{Z}) = \frac{\frac{p_D}{\kappa(z_{\theta(\ell)})} w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, \underline{\mathrm{S}}^{(\ell,j)}\right)}{\eta_{\mathrm{Z}}^{(\theta)}(\ell)}. \tag{B.35}$$

The update of track ℓ for the association of a missed detection, i.e. $\theta(\ell) = 0$, differs from the update using one of the received measurements. The measurement likelihood (5.46) is simply given by the missed detection probability:

$$\psi_{\mathrm{Z}}(x,\ell;\theta) = q_D(x,\ell) = q_D, \tag{B.36}$$

where $q_D = 1 - p_D$. The multiplication of the predicted density (B.16) with the measurement likelihood yields

$$p_+(x,\ell) \cdot \psi_{\mathrm{Z}}(x,\ell;\theta) = \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_+^{(\ell,j)}, \underline{\mathrm{P}}_+^{(\ell,j)}\right) \cdot q_D \tag{B.37}$$

$$= q_D \sum_{j=1}^{J_+^{(\ell)}} w^{(\ell,j,\theta)} \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right) \tag{B.38}$$

where the updated weight, mean value, and covariance correspond to the predicted values:

$$w^{(\ell,j,\theta)} \equiv w_+^{(\ell,j)}, \tag{B.39}$$

$$\hat{x}^{(\ell,j,\theta)}(\mathrm{Z}) \equiv \hat{x}_+^{(\ell,j)}, \tag{B.40}$$

$$\underline{\mathrm{P}}^{(\ell,j)} \equiv \underline{\mathrm{P}}_+^{(\ell,j)}. \tag{B.41}$$

Due to the constant detection probability, the normalization constant (5.45) simplifies to

$$\eta_{\mathrm{Z}}^{(\theta)}(\ell) = \left\langle p_+(\cdot,\ell), \psi_{\mathrm{Z}}(\cdot,\ell;\theta) \right\rangle \tag{B.42}$$

$$= \int q_D \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(x; \hat{x}^{(\ell,j)}, \underline{\mathrm{P}}^{(\ell,j)}\right) dx \tag{B.43}$$

$$= q_D \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \int \mathcal{N}\left(x; \hat{x}^{(\ell,j)}, \underline{\mathrm{P}}^{(\ell,j)}\right) dx \tag{B.44}$$

$$= q_D \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} = q_D. \tag{B.45}$$

Finally, the posterior distribution (5.44) of track ℓ for the update with the missed detection is given by

$$p^{(\theta)}(x,\ell|\mathrm{Z}) = \frac{p_+(x,\ell) \cdot \psi_{\mathrm{Z}}(x,\ell;\theta)}{\eta_{\mathrm{Z}}^{(\theta)}(\ell)} \tag{B.46}$$

$$= \sum_{j=1}^{J_+^{(\ell)}} w^{(\ell,j,\theta)} \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \underline{\mathrm{P}}^{(\ell,j)}\right). \tag{B.47}$$

Appendix C

Comparison of Random Finite Set Based Tracking Algorithms

The tables on the next pages illustrate the main differences of the random finite set based tracking algorithms used in this thesis. For additional details, refer to the corresponding sections within this thesis and the references therein.

Within the prediction step, all algorithms use the following properties:

- the motion of each object follows a single-object Markov transition density $f_+(x_+|x)$.
- an existing object with state x survives from time step k to $k+1$ with probability $p_S(x)$.

In the PHD and the CPHD filter, new objects appear at time $k+1$ according to the birth intensity $b(x)$ while the CB-MeMBer, the LMB, and the δ-GLMB filter use a multi-Bernoulli birth distribution π_B. In order to ensure comparable results, the intensity function (or PHD) of π_B is expected to match the birth intensity $b(x)$.

The update steps of the filter are based on the standard multi-object measurement model, i.e.:

- the likelihood that an object with state x generates a measurement z is given by the single-object measurement likelihood $g(z|x)$.
- an object with state x is detected by the sensor with detection probability $p_D(x)$.
- the false alarm process follows a Poisson distribution and creates an average number of λ_c false alarms whose spatial distribution is given by the probability density $c(z)$.

Observe that the CPHD filter uses the more general i.i.d. cluster process for the clutter distribution, i.e. the number of false alarms is not required to follow a Poisson distribution.

	PHD filter	CPHD filter	CB-MeMBer filter
Approximation Type	– first moment of multi-object density	– first moment of multi-object density and cardinality distribution	– parameters of a multi-Bernoulli distribution
Approximations	– predicted PHD approximately follows Poisson RFS	– predicted and posterior PHD approximately follow an i.i.d. cluster RFS	– PGFL approximation of measurement updated components
Complexity (prediction/update)	– linear / linear	– linear / cubic	– linear / linear
Pros	– low complexity – real-time tracking of a large number of objects	– accurate cardinality estimate	– track extraction in SMC implementations – estimation of track existence probability
Cons	– no track labeling – track extraction in SMC implementations – unstable cardinality estimate	– no track labeling – track extraction in SMC implementations – spooky effect	– no track labeling – biased cardinality estimate in scenarios with low p_D and high false alarm rate

	LMB	δ-GLMB	SMC-MOB (approx.)
Approximation Type	– parameters of a labeled Multi-Bernoulli distribution	– parameters of a δ-GLMB distribution	– multi-object particles
Approximations	– approximation of δ-GLMB RFS by LMB RFS – update truncation using Murty's algorithm	– truncation using k-shortest paths / Murty's algorithm	– exact representation requires infinite number of multi-object particles – likelihood approximation
Complexity (prediction/update)	– linear / cubic	– cubic / cubic	– linear (one prediction per multi-object particle) / linear
Pros	– track labeling – estimation of track existence probability – grouping reduces complexity – track extraction in SMC implementations	– track labeling – estimation of track existence probability – track extraction in SMC implementations	– modeling of object interactions – estimation of track existence probability
Cons	– track coalescence possible	– required number of hypotheses grows exponentially in number of objects	– required number of particles grows exponentially in number of objects – post-processing for track extraction required

Acronyms

ADMA	automotive dynamic motion analyzer	161
AMNIS	approximate multi-target normalized innovation squared	125
CA	constant acceleration	9
CB-MeMBer	cardinality balanced multi-target multi-Bernoulli	46
CPHD	cardinalized probability hypothesis density	43
CTRV	constant turn rate and constant velocity	9
CV	constant velocity	9
δ-GLMB	δ-generalized labeled multi-Bernoulli	32, 83
EKF	extended Kalman filter	9
EM	expectation maximization	111
FISST	finite set statistics	23
FOV	field of view	34
GLMB	generalized labeled multi-Bernoulli	31
GM	Gaussian mixture	39, 101, 139
GM-CPHD	Gaussian mixture cardinalized probability hypothesis density	45
GM-PHD	Gaussian mixture probability hypothesis density	41
GMTI	ground moving target indicator	73
GNIS	generalized normalized innovation squared	52
GNN	global nearest neighbor	15
GPS	global positioning system	161
GPU	graphics processing unit	73
i.i.d.	independent identically distributed	26
IPDA	integrated probabilistic data association	18
JIPDA	joint integrated probabilistic data association	18

JPDA	joint probabilistic data association	17
LM-IPDA	linear multi-target integrated probabilistic data association	18, 71
LMB	labeled multi-Bernoulli	30, 86, 91, 100
MAP	maximum a posteriori	119
MeMBer	multi-target multi-Bernoulli	91
MGNIS	multi-object generalized normalized innovation squared	53, 123
MHD	Mahalanobis distance	14, 111
MHT	multi-hypothesis tracking	19, 90
MM-PHD	multiple model probability hypothesis density	42
NEES	normalized estimation error squared	21
NIS	normalized innovation squared	21
NN	nearest neighbor	14
OMAT	Optimal Mass Transfer	50
OSPA	optimal subpattern assignment	50, 129
OSPAT	optimal subpattern assignment for tracks	51, 129
PDA	probabilistic data association	15
PDF	probability density function	5
PGFL	probability generating functional	48
PHD	probability hypothesis density	28, 39
PPP	Poisson point process	63
PPT	point process theory	39
RFS	random finite set	23
RMS	root mean square	21
ROC	receiver operating characteristics	146
SLAM	simultaneous localization and mapping	42
SMC	sequential Monte-Carlo	10, 56, 101
SMC-MOB	sequential Monte-Carlo multi-object Bayes	129
SNR	signal to noise ratio	126
UKF	unscented Kalman filter	9

List of Symbols

Notations

a	vector or scalar value
$\underline{\mathrm{A}}$	matrix
$\underline{\mathrm{A}}^{-1}$	inverse of a matrix
$\underline{\mathrm{A}}^{\mathrm{T}}$	transpose of a matrix
A	random finite set
$\boldsymbol{a}$	labeled vector or scalar
$\mathbf{A}$	labeled random finite set
$\lvert a \rvert$	dimension of a vector
$\lvert \mathrm{A} \rvert$	cardinality of a random finite set
$\langle a, b \rangle$	inner product of two functions a and b
p^{A}	multi-object exponential ($\prod_{a \in \mathrm{A}} p^a$)
$\delta_{\mathrm{A}}(\mathrm{B})$	generalized Kronecker delta function
$1_{\mathrm{A}}(\mathrm{B})$	inclusion function
θ	association hypothesis, e.g. track label to measurement
C_k^n	binomial coefficient
P_k^n	permutation coefficient

Subscripts

k	time step k
$+$	predicted value of a variable or density
B	variable or density corresponding to a new born object
C	variable or density corresponding to clutter process
S	variable or density corresponding to a surviving object

Scalars and vectors

γ	residual of predicted and actual measurement
J	number of Gaussian mixtures
ℓ	track label

λ	expected value of the Poisson distribution
M	number of multi-Bernoulli components
$\hat{N}$	estimated number of objects
ν	number of particles
w	weight (e.g. of a particle or a hypothesis)
x	state vector
$\hat{x}$	expected value of the spatial distribution $p(x)$
$\boldsymbol{x}$	labeled state vector
z	measurement vector
z_+	predicted measurement of for a state vector x

Matrices

$\underline{\mathrm{F}}$	process (or state transition) matrix
$\underline{\mathrm{H}}$	measurement matrix
$\underline{\mathrm{I}}$	identity matrix
$\underline{\mathrm{K}}$	Kalman gain
$\underline{\mathrm{P}}$	estimation error covariance matrix
$\underline{\mathrm{Q}}$	covariance matrix of the process noise
$\underline{\mathrm{R}}$	covariance matrix of the measurement noise
$\underline{\mathrm{S}}$	innovation covariance matrix

Sets

I	set of track labels (within a hypothesis)
L	set of track labels
X	set of state vectors
$\mathbf{X}$	set of labeled state vectors
Z	set of measurements at time $k+1$
$\mathrm{Z}_{1:k+1}$	set of all measurements up to time $k+1$
$\Delta(\mathbf{X})$	distinct label indicator ($\lvert\mathrm{X}\rvert = \lvert\mathcal{L}(\mathbf{X})\rvert$)

Spaces

$\mathbb{B}$	label space of new born objects
$\mathbb{C}$	index space
$\mathbb{L}$	label space of existing objects
$\mathbb{M}$	index space of current measurements
$\mathbb{N}$	space of natural numbers
$\mathbb{R}$	space of real numbers

Θ	space of track label to measurement assignments
$\mathbb{X}$	state space
Ξ	discrete space
$\mathbb{Z}$	measurement space
$\mathcal{F}(\mathbb{X})$	finite subsets of the space $\mathbb{X}$
$\mathcal{F}_n(\mathbb{X})$	finite subsets of the space $\mathbb{X}$ with cardinality n
$\mathcal{L}$	projection from $\mathbb{X} \times \mathbb{L}$ to $\mathbb{L}$

Densities

p	spatial distribution p
π	multi-object probability density
$\boldsymbol{\pi}$	labeled multi-object probability density
$\rho(n)$	cardinality distribution
$c(z)$	probability density of clutter measurements
$f_+(x\|\cdot)$	Markov transition density
$\kappa(z)$	probability hypothesis density of the clutter process
$v(x)$	probability hypothesis density (equivalent to intensity density)
$\mathcal{N}(x; \mu, \sigma)$	normal distribution over x with mean μ and standard deviation σ
$\Pr(\|\mathrm{X}\| = n)$	probability that the cardinality of set X is equal to n
q	proposal density for Markov transition
b	proposal density for birth distribution

Specific probabilities

$p_D(x)$	detection probability of an object with state x
$q_D(x)$	missed detection probability of an object with state x
$p_S(x)$	survival (or persistence) probability of an object with state x
$q_S(x)$	disappearance probability of an object with state x
$\eta_S(\ell)$	state independent survival probability of track ℓ
$p_{TP}(z)$	true positive probability of measurement z
$p_{FP}(z)$	false positive probability of measurement z
r	existence probability

Measurement likelihoods

$g(z\|x)$	likelihood for a measurement z given an object with state x
$\psi_Z(x; \theta)$	generalized measurement likelihood for object x and association θ
$\eta_Z^\theta(\ell)$	likelihood for assigning track label ℓ to measurement $\theta(\ell)$

Bibliography

[AMGC02] Arulampalam, M.; Maskell, S.; Gordon, N.; and Clapp, T.: *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*. In: *IEEE Transactions on Signal Processing*, volume 50, no. 2, pages 174–188, 2002.

[AS72] Alspach, D. and Sorenson, H.: *Nonlinear Bayesian estimation using Gaussian sum approximations*. In: *IEEE Transactions on Automatic Control*, volume 17, no. 4, pages 439–448, 1972.

[Bar02] Bar-Shalom, Y.: *Update with out-of-sequence measurements in tracking: exact solution*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 38, no. 3, pages 769–777, 2002.

[BCM04] Bar-Shalom, Y.; Chen, H.; and Mallick, M.: *One-step solution for the multistep out-of-sequence-measurement problem in tracking*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 40, no. 1, pages 27–37, 2004.

[BE95] Borwein, P. and Erdélyi, T.: *Polynomials and Polynomial Inequalities*. Springer, New York, 1995.

[Ber88] Bertsekas, D. P.: *The auction algorithm: a distributed relaxation method for the assignment problem*. In: *Annals of Operations Research*, volume 14, no. 1, pages 105–123, 1988.

[Ber90] Bertsekas, D. P.: *The auction algorithm for assignment and other network flow problems: a tutorial*. In: *Interfaces*, volume 20, no. 4, pages 133–149, 1990.

[BF88] Bar-Shalom, Y. and Fortmann, T.: *Tracking and Data Association*. Academic Press, Inc., 1988.

[BH11] Baum, M. and Hanebeck, U.: *Shape tracking of extended objects and group targets with star-convex RHMs*. In: *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, 2011.

[Bis06] Bishop, C. M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

[BL71] Bourgeois, F. and Lassalle, J.-C.: *An extension of the Munkres algorithm for the assignment problem to rectangular matrices*. In: *Communications of the ACM*, volume 14, no. 12, pages 802–804, 1971.

[BP99] Blackman, S. and Popoli, R.: *Design and Analysis of Modern Tracking Systems*. Artech House Publishers, 1999.

[BT75] Bar-Shalom, Y. and Tse, E.: *Tracking in a cluttered environment with probabilistic data association*. In: *Automatica*, volume 11, no. 5, pages 451–460, 1975.

[BVV13] Beard, M.; Vo, B.-T.; and Vo, B.-N.: *Multitarget filtering with unknown clutter density using a bootstrap GMCPHD filter*. In: *IEEE Signal Processing Letters*, volume 20, no. 4, pages 323–326, 2013.

[BVVA12] Beard, M.; Vo, B.-T.; Vo, B.-N.; and Arulampalam, S.: *Gaussian mixture PHD and CPHD filtering with partially uniform target birth*. In: *Proceedings of the 15th International Conference on Information Fusion*, pages 535–541, 2012.

[BWT11] Bar-Shalom, Y.; Willett, P. K.; and Tian, X.: *Tracking and Data Fusion*. YBS Publishing, Storrs, CT, USA, 2011.

[CH96] Cox, I. and Hingorani, S.: *An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 18, no. 2, pages 138–150, 1996.

[CRPB07] Clark, D.; Ruiz, I.; Petillot, Y.; and Bell, J.: *Particle PHD filter multiple target tracking in sonar image*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 43, no. 1, pages 409–416, 2007.

[DB93] Dezert, J. and Bar-Shalom, Y.: *Joint probabilistic data association for autonomous navigation*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 29, no. 4, pages 1275–1286, 1993.

[DC05] Douc, R. and Cappe, O.: *Comparison of resampling schemes for particle filtering*. In: *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69, 2005.

[DFG01] Doucet, A.; de Freitas, N.; and Gordon, N.: *Sequential Monte Carlo methods in practice*. Ed. by Doucet, A. Springer, New York, 2001.

[DHS01] Duda, R. O.; Hart, P. E.; and Stork, D. G.: *Pattern Classification*. Volume 2. John Wiley & Sons, Inc., 2001.

[DK13] Dunne, D. and Kirubarajan, T.: *Multiple model multi-Bernoulli filters for manoeuvering targets*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 49, no. 4, pages 2679–2692, 2013.

[DLR77] Dempster, A. P.; Laird, N. M.; and Rubin, D. B.: *Maximum likelihood from incomplete data via the EM algorithm*. In: *Journal of the Royal Statistical Society. Series B (Methodological)*, volume 39, no. 1, pages 1–38, 1977.

[DV03] Daley, D. and Vere-Jones, D.: *An Introduction to the Theory of Point Processes*. Volume 1: Elementary Theory and Methods. 2nd editon, Springer, 2003.

[DV88] Daley, D. and Vere-Jones, D.: *An Introduction to the Theory of Point Processes*. Springer, New York, 1988.

[EAGG13] Edman, V.; Andersson, M.; Granström, K.; and Gustafsson, F.: *Pedestrian group tracking using the GM-PHD filter*. In: *Proceedings of the European Signal Processing Conference*, Marrakech, Morocco, 2013.

[Epp98] Eppstein, D.: *Finding the k shortest paths*. In: *SIAM Journal on Computing*, volume 28, no. 2, pages 652–673, 1998.

[EWB05] Erdinc, O.; Willett, P.; and Bar-Shalom, Y.: *Probability hypothesis density filter for multitarget multisensor tracking*. In: *Proceedings of the 8th International Conference on Information Fusion*, pages 1–8, 2005.

[EWC08] Erdinc, O.; Willett, P.; and Coraluppi, S.: *The Gaussian mixture cardinalized PHD tracker on MSTWG and SEABAR'07 datasets*. In: *Proceedings of the 11th International Conference on Information Fusion*, pages 1–8, 2008.

[FBS83] Fortmann, T.; Bar-Shalom, Y.; and Scheffe, M.: *Sonar tracking of multiple targets using joint probabilistic data association*. In: *IEEE Journal of Oceanic Engineering*, volume 8, no. 3, pages 173–184, 1983.

[FSU09] Fränken, D.; Schmidt, M.; and Ulmke, M.: *''Spooky action at a distance" in the cardinalized probability hypothesis density filter*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 45, no. 4, pages 1657–1664, 2009.

[GH75] Gerlough, D. and Huber, M.: *Traffic flow theory: a monograph*. Volume 165 from the series *Special Report*. Transportation Research Board, National Research Council, 1975.

[GLO10] Granström, K.; Lundquist, C.; and Orguner, U.: *A Gaussian mixture PHD filter for extended target tracking*. In: *Proceedings of the 13th International Conference on Information Fusion*, pages 1–8, 2010.

[GLO11] Granström, K.; Lundquist, C.; and Orguner, U.: *Tracking rectangular and elliptical extended targets using laser measurements*. In: *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, 2011.

[GLW+13] Gabb, M.; Löhlein, O.; Wagner, R.; et al.: *High-performance on-road vehicle detection in monocular images.* In: *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*, pages 336–341, 2013.

[GMN97] Goodman, I.; Mahler, R.; and Nguyen, H.: *Mathematics of Data Fusion.* From the series *Theory and decision library: Series B : Mathematical and Statistical Methods.* Springer, 1997.

[GO12] Granström, K. and Orguner, U.: *A PHD filter for tracking multiple extended targets using random matrices.* In: *IEEE Transactions on Signal Processing*, volume 60, no. 11, pages 5657–5671, 2012.

[Gor97] Gordon, N.: *A hybrid bootstrap filter for target tracking in clutter.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 33, no. 1, pages 353–358, 1997.

[GSS93] Gordon, N.; Salmond, D.; and Smith, A. F. M.: *Novel approach to nonlinear/non-Gaussian Bayesian state estimation.* In: *IEE Proceedings Radar and Signal Processing*, volume 140, no. 2, pages 107–113, 1993.

[GW12] Georgescu, R. and Willett, P.: *The GM-CPHD tracker applied to real and realistic multistatic sonar data sets.* In: *IEEE Journal of Oceanic Engineering*, volume 37, no. 2, pages 220–235, 2012.

[Hen71] Henderson, L. F.: *The statistics of crowd fluids.* In: *Nature 229*, pages 381–383, 1971.

[HKG10] Hendeby, G.; Karlsson, R.; and Gustafsson, F.: *Particle filtering: the need for speed.* In: *EURASIP Journal on Advances in Signal Processing*, volume 2010, no. 1, pages 1–9, 2010.

[HL10] Houssineau, J. and Laneuville, D.: *PHD filter with diffuse spatial prior on the birth process with applications to GM-PHD filter.* In: *Proceedings of the 13th International Conference on Information Fusion*, pages 1–8, 2010.

[HL64] Ho, Y.-C. and Lee, R.: *A Bayesian approach to problems in stochastic estimation and control.* In: *IEEE Transactions on Automatic Control*, volume 9, no. 4, pages 333–339, 1964.

[HM04a] Hall, D. L. and McMullen, S. A. H.: *Mathematical Techniques in Multisensor Data Fusion.* Artech House Inc., Norwood, 2004.

[HM04b] Hoffman, J. and Mahler, R.: *Multitarget miss distance via optimal assignment.* In: *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, volume 34, no. 3, pages 327–336, 2004.

[HM11] Horridge, P. and Maskell, S.: *Using a probabilistic hypothesis density filter to confirm tracks in a multi-target environment.* In: *Proceedings of the 6th Workshop Sensor Data Fusion: Trends, Solutions, Applications*, 2011.

[HM95] Helbing, D. and Molnar, P.: *Social force model for pedestrian dynamics*. In: *PHYSICAL REVIEW E*, volume 51, pages 4282–4286, 1995.

[HVV13] Hoseinnezhad, R.; Vo, B.-N.; and Vo, B.-T.: *Visual tracking in background subtracted image sequences via multi-Bernoulli filtering*. In: *IEEE Transactions on Signal Processing*, volume 61, no. 2, pages 392–397, 2013.

[HVVS11] Hoseinnezhad, R.; Vo, B.-N.; Vo, B.-T.; and Suter, D.: *Bayesian integration of audio and visual information for multi-target tracking using a CB-MeMBer filter*. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2300–2303, 2011.

[IB98] Isard, M. and Blake, A.: *CONDENSATION - conditional density propagation for visual tracking*. In: *International Journal of Computer Vision*, volume 29, pages 5–28, 1998.

[JDSW12] Jonsson, R.; Degerman, J.; Svensson, D.; and Wintenby, J.: *Multitarget tracking with background discrimination using PHD filters*. In: *Proceedings of the 15th International Conference on Information Fusion*, pages 854–860, 2012.

[JU04] Julier, S. and Uhlmann, J.: *Unscented filtering and nonlinear estimation*. In: *Proceedings of the IEEE*, volume 92, no. 3, pages 401–422, 2004.

[JUD00] Julier, S.; Uhlmann, J.; and Durrant-Whyte, H.: *A new method for the nonlinear transformation of means and covariances in filters and estimators*. In: *IEEE Transactions on Automatic Control*, volume 45, no. 3, pages 477–482, 2000.

[JUD95] Julier, S.; Uhlmann, J.; and Durrant-Whyte, H.: *A new approach for filtering nonlinear systems*. In: *Proceedings of the 1995 American Control Conference*. Volume 3, pages 1628–1632, 1995.

[JV87] Jonker, R. and Volgenant, A.: *A shortest augmenting path algorithm for dense and sparse linear assignment problems*. In: *Computing*, volume 38, no. 4, pages 325–340, 1987.

[JZT+13] Jovanoska, S.; Zetik, R.; Thomä, R.; et al.: *Device-free indoor localization using a distributed network of autonomous UWB sensor nodes*. In: *Proceedings of the 8th Workshop Sensor Data Fusion: Trends, Solutions, and Applications*, pages 1–6, 2013.

[Kal60] Kalman, R. E.: *A new approach to linear filtering and prediction problems*. In: *Transactions of the ASME - Journal of Basic Engineering*, volume 82, no. Series D, pages 35–45, 1960.

[Kal84] Kalata, P. R.: *The tracking index: a generalized parameter for $\alpha - \beta$ and $\alpha - \beta - \gamma$ target trackers*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 20, no. 2, pages 174–182, 1984.

[Kam07] Kämpchen, N.: *Feature-level fusion of laser scanner and video data for advanced driver assistance systems.* PhD thesis, Ulm University, 2007.

[KD03] Kämpchen, N. and Dietmayer, K.: *Data synchronization strategies for multi-sensor fusion.* In: *10th World Congress on Intelligent Transportation Systems*, Madrid, Spain, 2003.

[KK01] Koch, W. and Klemm, R.: *Ground target tracking with STAP radar.* In: *IEE Proceedings on Radar, Sonar and Navigation*, volume 148, no. 3, pages 173–185, 2001.

[Koc04] Koch, W.: *On "negative" information in tracking and sensor data fusion: discussion of selected examples.* In: *Proceedings of the 7th International Conference on Information Fusion*, 2004.

[Koc08] Koch, J. W.: *Bayesian approach to extended object and cluster tracking using random matrices.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 44, Issue 3, pages 1042–1059, July 2008.

[KSD10] Konrad, M.; Szczot, M.; and Dietmayer, K.: *Road course estimation in occupancy grids.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 412–417, 2010.

[Kuh55] Kuhn, H. W.: *The Hungarian method for the assignment problem.* In: *Naval Research Logistics Quarterly*, volume 2, pages 83–97, 1955.

[LCB12a] Lamard, L.; Chapuis, R.; and Boyer, J.-P.: *A comparison of two different tracking algorithms is provided for real application.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 414–419, 2012.

[LCB12b] Lamard, L.; Chapuis, R.; and Boyer, J.-P.: *Dealing with occlusions with multi targets tracking algorithms for the real road context.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 371–376, 2012.

[Lew51] Lewin, K.: *Field Theory in Social Science.* Harper & Row, Publishers, Inc., 1951.

[LGO11] Lundquist, C.; Granström, K.; and Orguner, U.: *Estimating the shape of targets with a PHD filter.* In: *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, 2011.

[LGO13] Lundquist, C.; Granström, K.; and Orguner, U.: *An extended target CPHD filter and a Gamma Gaussian Inverse Wishart implementation.* In: *IEEE Journal of Selected Topics in Signal Processing*, volume 7, no. 3, pages 472–483, 2013.

[Mah01] Mahler, R.: *Multitarget moments and their application to multitarget tracking.* In: *Proceedings of the Workshop on Estimation, Tracking, and Fusion: A tribute to Y. Bar-Shalom*, pages 134–166, Naval Postgraduate School, Monterey, CA, 2001.

[Mah03] Mahler, R.: *Multitarget Bayes filtering via first-order multitarget moments.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 39, no. 4, pages 1152–1178, 2003.

[Mah04] Mahler, R.: *"Statistics 101" for multisensor, multitarget data fusion.* In: *IEEE Aerospace and Electronic Systems Magazine*, volume 19, no. 1, pages 53–64, 2004.

[Mah07a] Mahler, R.: *Statistical Multisource-Multitarget Information Fusion.* Artech House Inc., Norwood, 2007.

[Mah07b] Mahler, R.: *PHD filters of higher order in target number.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 43, no. 4, pages 1523–1543, 2007.

[Mah09a] Mahler, R.: *PHD filters for nonstandard targets, I: extended targets.* In: *12th International Conference on Information Fusion, Seattle, WA, USA*, 2009.

[Mah09b] Mählisch, M.: *Filtersynthese zur simultanen Minimierung von Existenz-, Assoziations- und Zustandsunsicherheiten in der Fahrzeugumfelderfassung mit heterogenen Sensordaten.* PhD thesis, Ulm University, 2009.

[Mah12] Mahler, R.: *On multitarget jump-Markov filters.* In: *Proceedings of the 15th International Conference on Information Fusion*, pages 149–156, 2012.

[Mah13a] Mahler, R.: *Divergence detectors for multitarget tracking algorithms.* In: *Proceedings SPIE 8745, Signal Processing, Sensor Fusion, and Target Recognition XXII*, 2013.

[Mah13b] Mahler, R.: *"Statistics 102" for multisource-multitarget detection and tracking.* In: *IEEE Journal of Selected Topics in Signal Processing*, volume 7, no. 3, pages 376–389, 2013.

[Mas04] Maskell, S.: *Sequentially Structured Bayesian Solutions.* PhD thesis, Cambridge University, 2004.

[Mat60] Matérn, B.: *Spatial variation - stochastic models and their application to some problems in forest surveys and other sampling investigations.* In: *Meddelanden från Statens Skogsförsöksanstalt*, volume 49, no. 5, 1960.

[Mat75] Matheron, G.: *Random sets and integral geometry.* Wiley, New York, 1975.

[Mat86] Matérn, B.: *Spatial Variation (Lecture Notes in Statistics).* Springer, 1986.

[MAZ+10] Muntzinger, M. M.; Aeberhard, M.; Zuther, S.; et al.: *Reliable automotive pre-crash system with out-of-sequence measurement processing.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1022–1027, 2010.

[MD12] Macagnano, D. and De Abreu, G. T. F.: *Adaptive gating for multitarget tracking with Gaussian mixture filters*. In: *IEEE Transactions on Signal Processing*, volume 60, no. 3, pages 1533–1538, 2012.

[ME04] Musicki, D. and Evans, R.: *Joint integrated probabilistic data association: JIPDA*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 40, no. 3, pages 1093–1099, 2004.

[MES94] Musicki, D.; Evans, R.; and Stankovic, S.: *Integrated probabilistic data association*. In: *IEEE Transactions on Automatic Control*, volume 39, no. 6, pages 1237–1241, 1994.

[ML08] Musicki, D. and La Scala, B.: *Multi-target tracking in clutter without measurement assignment*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 44, no. 3, pages 877–896, 2008.

[MMD10] Munz, M.; Mählisch, M.; and Dietmayer, K.: *Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driver assistance systems*. In: *IEEE Intelligent Transportation Systems Magazine*, volume 2, no. 1, pages 6–17, 2010.

[MRD13] Meissner, D.; Reuter, S.; and Dietmayer, K.: *Road user tracking at intersections using a multiple-model PHD filter*. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 377–382, 2013.

[MRWD13] Meissner, D.; Reuter, S.; Wilking, B.; and Dietmayer, K.: *Road user tracking using a Dempster-Shafer based classifying multiple-model PHD filter*. In: *Proceedings of the 16th International Conference on Information Fusion*, pages 1236–1242, 2013.

[MSC97] Miller, M.; Stone, H.; and Cox, I. J.: *Optimizing Murty's ranked assignment method*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 33, no. 3, pages 851–862, 1997.

[MSRD06] Mählisch, M.; Schweiger, R.; Ritter, W.; and Dietmayer, K.: *Multisensor vehicle tracking with the probability hypothesis density filter*. In: *Proceedings of the 9th International Conference on Information Fusion*, pages 1–8, 2006.

[Mun11] Munz, M.: *Generisches Sensorfusionsframework zur gleichzeitigen Zustands- und Existenzschätzung für die Fahrzeugumfelderkennung*. PhD thesis, Ulm University, 2011.

[Mun57] Munkres, J.: *Algorithms for the assignment and transportation problems*. In: *Journal of the Society for Industrial and Applied Mathematics*, volume 5, no. 1, pages 32–38, 1957.

[Mur68] Murty, K.: *An algorithm for ranking all the assignments in order of increasing cost*. In: *Operations Research*, volume 16, pages 682–687, 1968.

[MVAV11a] Mullane, J.; Vo, B.-N.; Adams, M.; and Vo, B.-T.: *A random-finite-set approach to Bayesian SLAM.* In: *IEEE Transactions on Robotics*, volume 27, no. 2, pages 268–282, 2011.

[MVAV11b] Mullane, J.; Vo, B.-N.; Adams, M.; and Vo, B.-T.: *Random Finite Sets for Robot Mapping and SLAM - New Concepts in Autonomous Robotic Map Representations.* From the series *Springer Tracts in Advanced Robotics*. Springer, 2011.

[MVSB06] Ma, W.-K.; Vo, B.-N.; Singh, S.; and Baddeley, A.: *Tracking an unknown time-varying number of speakers using TDOA measurements: a random finite set approach.* In: *IEEE Transactions on Signal Processing*, volume 54, pages 3291–3304, 2006.

[MVV11] Mahler, R.; Vo, B.-T.; and Vo, B.-N.: *CPHD filtering with unknown clutter rate and detection profile.* In: *IEEE Transactions on Signal Processing*, volume 59, no. 8, pages 3497–3513, 2011.

[Nvi14] Nvidia: *CUDA C Programming Guide - Version 5.0.* 2014.

[PE13] Pikora, K. and Ehlers, F.: *Analysis of the FKIE passive radar data set with GMPHD and GMCPHD.* In: *Proceedings of the 16th International Conference on Information Fusion*, pages 272–279, 2013.

[PESG09] Pellegrini, S.; Ess, A.; Schindler, K.; and van Gool, L.: *You'll never walk alone: modeling social behavior for multi-target tracking.* In: *Proceedings of the 12th IEEE International Conference on Computer Vision*, pages 261–268, 2009.

[PPP+09] Pollard, E.; Plyer, A.; Pannetier, B.; Champagnat, F.; and Le Besnerais, G.: *GM-PHD filters for multi-object tracking in uncalibrated aerial videos.* In: *Proceedings of the 12th International Conference on Information Fusion*, pages 1171–1178, 2009.

[PPR11] Pollard, E.; Pannetier, B.; and Rombaut, M.: *Hybrid algorithms for multitarget tracking using MHT and GM-CPHD.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 47, no. 2, pages 832–847, 2011.

[RAG04] Ristic, B.; Arulampalam, S.; and Gordon, N.: *Beyond the Kalman Filter: Particle Filters for Tracking Applications.* Artech House Inc., 2004.

[RCV10] Ristic, B.; Clark, D.; and Vo, B.-N.: *Improved SMC implementation of the PHD filter.* In: *Proceedings of the 13th International Conference on Information Fusion*, pages 1–8, 2010.

[RCVV12] Ristic, B.; Clark, D.; Vo, B.-N.; and Vo, B.-T.: *Adaptive target birth intensity for PHD and CPHD filters.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 48, no. 2, pages 1656–1668, 2012.

[RD09] Reuter, S. and Dietmayer, K.: *Fuzzy estimation and segmentation for laser range scans.* In: *Proceedings of the 12th International Conference on Information Fusion*, pages 1974–1981, 2009.

[RD10] Reuter, S. and Dietmayer, K.: *Adapting the state uncertainties of tracks to environmental constraints.* In: *Proceedings of the 13th International Conference on Information Fusion*, pages 1–7, 2010.

[RDH11] Reuter, S.; Dietmayer, K.; and Handrich, S.: *Real-time implementation of a random finite set particle filter.* In: *Proceedings of the 6th Workshop Sensor Data Fusion: Trends, Solutions, Applications*, 2011.

[Rei79] Reid, D.: *An algorithm for tracking multiple targets.* In: *IEEE Transactions on Automatic Control*, volume 24, no. 6, pages 843–854, 1979.

[RGV+13] Rezatofighi, S. H.; Gould, S.; Vo, B.-N.; et al.: *A multiple model probability hypothesis density tracker for time-lapse cell microscopy sequences.* In: *Information Processing in Medical Imaging (IPMI)*, 2013.

[RMD12] Reuter, S.; Meissner, D.; and Dietmayer, K.: *Multi-object tracking at intersections using the cardinalized probability hypothesis density filter.* In: *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1172–1177, 2012.

[RMWD13] Reuter, S.; Meissner, D.; Wilking, B.; and Dietmayer, K.: *Cardinality balanced multi-target multi-Bernoulli filtering using adaptive birth distributions.* In: *Proceedings of the 16th International Conference on Information Fusion*, pages 1608–1615, 2013.

[RS68] Raiffa, H. and Schlaifer, R.: *Applied statistical decision theory.* Ed. by Raiffa, H. and Schlaifer, R. MIT Pr., Cambridge, 1968.

[RVCV11] Ristic, B.; Vo, B.-N.; Clark, D.; and Vo, B.-T.: *A metric for performance evaluation of multi-target tracking algorithms.* In: *IEEE Transactions on Signal Processing*, volume 59, no. 7, pages 3452–3457, 2011.

[RVVD14] Reuter, S.; Vo, B.-T.; Vo, B.-N.; and Dietmayer, K.: *The labeled multi-Bernoulli filter.* In: *IEEE Transactions on Signal Processing*, volume 62, no. 12, pages 3246–3260, 2014.

[RWW+13] Reuter, S.; Wilking, B.; Wiest, J.; Munz, M.; and Dietmayer, K.: *Real-time multi-object tracking using random finite sets.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 49, no. 4, pages 2666–2678, 2013.

[SA71] Sorenson, H. and Alspach, D.: *Recursive Bayesian estimation using Gaussian sums.* In: *Automatica*, volume 7, no. 4, pages 465–479, 1971.

[Sam65] Samuels, S. M.: *On the number of successes in independent trials.* In: *The Annals of Mathematical Statistics*, volume 36, no. 4, pages 1272–1278, 1965.

[SGM+12] Schikora, M.; Gning, A.; Mihaylova, L.; Cremers, D.; and Koch, W.: *Box-particle PHD filter for multi-target tracking.* In: *Proceedings of the 15th International Conference on Information Fusion*, pages 106–113, 2012.

[Sha76] Shafer, G.: *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, 1976.

[Sid03] Sidenbladh, H.: *Multi-target particle filtering for the probability hypothesis density.* In: *Proceedings of the 6th International Conference on Information Fusion.* Volume 2, pages 800–806, 2003.

[SKM95] Stoyan, D.; Kendall, W.; and Mecke, J.: *Stochastic Geometry and its Applications.* John Wiley & Sons, 1995.

[SRW08] Schubert, R.; Richter, E.; and Wanielik, G.: *Comparison and evaluation of advanced motion models for vehicle tracking.* In: *Proceedings of the 11th International Conference on Information Fusion*, pages 1–6, 2008.

[SSM62] Smith, G. L.; Schmidt, S. F.; and Mc Gee, L. A.: *Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle.* Tech. rep. NASA Ames Research Center, Tech. Rep. TR R-135, 1962.

[Str10] Streit, R. L.: *Poisson Point Processes.* Springer, 2010.

[SVV08] Schuhmacher, D.; Vo, B.-T.; and Vo, B.-N.: *A consistent metric for performance evaluation of multi-object filters.* In: *IEEE Transactions on Signal Processing*, volume 56, no. 8, pages 3447–3457, 2008.

[SW03] Sidenbladh, H. and Wirkander, S.-L.: *Tracking random sets of vehicles in terrain.* In: *Conference on Computer Vision and Pattern Recognition Workshop*, page 98, 2003.

[SWBH12] Schueler, K.; Weiherer, T.; Bouzouraa, E.; and Hofmann, U.: *360 degree multi sensor fusion for static and dynamic obstacles.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 692–697, 2012.

[TBF05] Thrun, S.; Burgard, W.; and Fox, D.: *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* The MIT Press, 2005.

[UEW07] Ulmke, M.; Erdinc, O.; and Willett, P.: *Gaussian mixture cardinalized PHD filter for ground moving target tracking.* In: *Proceedings of the 10th International Conference on Information Fusion*, pages 1–8, 2007.

[UK06] Ulmke, M. and Koch, W.: *Road-map assisted ground moving target tracking.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 42, no. 4, pages 1264–1274, 2006.

[VJ01] Viola, P. and Jones, M.: *Rapid object detection using a boosted cascade of simple features.* In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518, 2001.

[VM06] Vo, B.-N. and Ma, W.-K.: *The Gaussian mixture probability hypothesis density filter.* In: *IEEE Transactions on Signal Processing*, volume 54, no. 11, pages 4091–4104, 2006.

[Vo08] Vo, B.-T.: *Random Finite Sets in Multi-Object Filtering.* PhD thesis, The University of Western Australia, School of Electrical, Electronic and Computer Engineering, 2008.

[VPT06] Vo, B.-N.; Pasha, A.; and Tuan, H. D.: *A Gaussian mixture PHD filter for nonlinear jump Markov models.* In: *45th IEEE Conference on Decision and Control*, pages 3162–3167, 2006.

[VSD05] Vo, B.-N.; Singh, S.; and Doucet, A.: *Sequential Monte Carlo methods for multitarget filtering with random finite sets.* In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 41, Issue 4, pages 1224–1245, 2005.

[VV11] Vo, B.-T. and Vo, B.-N.: *A random finite set conjugate prior and application to multi-target tracking.* In: *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 431–436, 2011.

[VV12] Vo, B.-T. and Vo, B.-N.: *The para-normal Bayes multi-target filter and the spooky effect.* In: *Proceedings of the 15th International Conference on Information Fusion*, pages 173–180, 2012.

[VV13a] Vo, B.-T. and Vo, B.-N.: *Tracking, identification, and classification with random finite sets.* In: *Proceedings of the SPIE 8745, Signal Processing, Sensor Fusion, and Target Recognition XXII.* International Society for Optics and Photonics, 2013.

[VV13b] Vo, B.-T. and Vo, B.-N.: *Labeled random finite sets and multi-object conjugate priors.* In: *IEEE Transactions on Signal Processing*, volume 61, no. 13, pages 3460–3475, 2013.

[VVC07] Vo, B.-T.; Vo, B.-N.; and Cantoni, A.: *Analytic implementations of the cardinalized probability hypothesis density filter.* In: *IEEE Transactions on Signal Processing*, volume 55, no. 7, pages 3553–3567, 2007.

[VVC09] Vo, B.-T.; Vo, B.-N.; and Cantoni, A.: *The cardinality balanced multi-target multi-Bernoulli filter and its implementations.* In: *IEEE Transactions on Signal Processing*, volume 57, no. 2, pages 409–423, 2009.

[VVHM13] Vo, B.-T.; Vo, B.-N.; Hoseinnezhad, R.; and Mahler, R.: *Robust multi-Bernoulli filtering.* In: *IEEE Journal of Selected Topics in Signal Processing*, volume 7, no. 3, pages 399–409, 2013.

[VVPS10] Vo, B.-N.; Vo, B.-T.; Pham, N.-T.; and Suter, D.: *Joint detection and estimation of multiple objects from image observations.* In: *IEEE Transactions on Signal Processing*, volume 58, no. 10, pages 5129–5141, 2010.

[Wan93] Wang, Y. H.: *On the number of successes in independent trials.* In: *Statistica Sinica*, volume 3, no. 2, pages 295–312, 1993.

[WB12] Wendemuth, A. and Biundo, S.: *A companion technology for cognitive technical systems.* In: *Proceedings of the EUCogII-SSPNET-COST2102 International Conference*, pages 89–103. *Lecture Notes in Computer Science*, 2012.

[WDM+12] Westenberger, A.; Duraisamy, B.; Munz, M.; et al.: *Impact of out-of-sequence measurements on the joint integrated probabilistic data association filter for vehicle safety systems.* In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 438–443, 2012.

[WZ09a] Wei, J. and Zhang, X.: *Dynamic node collaboration for mobile multi-target tracking in two-tier wireless camera sensor networks.* In: *Proceedings of the IEEE Military Communications Conference*, pages 1–7, 2009.

[WZ09b] Wei, J. and Zhang, X.: *Mobile multi-target tracking in two-tier hierarchical wireless sensor networks.* In: *Proceedings of the IEEE Military Communications Conference*, pages 1–6, 2009.

[WZ10] Wei, J. and Zhang, X.: *Sensor self-organization for mobile multi-target tracking in decentralized wireless sensor networks.* In: *Proceedings of the IEEE Wireless Communications and Networking Conference*, pages 1–6, 2010.

[Yen71] Yen, J. Y.: *Finding the k shortest loopless paths in a network.* In: *Management Science*, volume 17, no. 11, pages 712–716, 1971.

Publications

Parts of this thesis were pre-released in author's journal articles and conference papers.

Reuter, S. and Dietmayer, K.: *Fuzzy estimation and segmentation for laser range scans*. In: *Proceedings of the 12th International Conference on Information Fusion*, pages 1974–1981, 2009.

Reuter, S. and Dietmayer, K.: *Adapting the state uncertainties of tracks to environmental constraints*. In: *Proceedings of the 13th International Conference on Information Fusion*, pages 1–7, 2010.

Reuter, S. and Dietmayer, K.: *Pedestrian tracking using random finite sets*. In: *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, 2011.

Reuter, S.; Dietmayer, K.; and Handrich, S.: *Real-time implementation of a random finite set particle filter*. In: *Proceedings of the 6th Workshop Sensor Data Fusion: Trends, Solutions, Applications*, 2011.

Reuter, S.; Meissner, D.; and Dietmayer, K.: *Multi-object tracking at intersections using the cardinalized probability hypothesis density filter*. In: *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1172–1177, 2012.

Reuter, S.; Meissner, D.; Wilking, B.; and Dietmayer, K.: *Cardinality balanced multi-target multi-Bernoulli filtering using adaptive birth distributions*. In: *Proceedings of the 16th International Conference on Information Fusion*, pages 1608–1615, 2013.

Reuter, S.; Vo, B.-T.; Vo, B.-N.; and Dietmayer, K.: *The labeled multi-Bernoulli filter*. In: *IEEE Transactions on Signal Processing*, volume 62, no. 12, pages 3246–3260, 2014.

Reuter, S.; Vo, B.-T.; Wilking, B.; Meissner, D.; and Dietmayer, K.: *Divergence detectors for the δ-generalized labeled multi-Bernoulli filter*. In: *Proceedings of the 8th Workshop Sensor Data Fusion: Trends, Solutions, and Applications*, Bonn, Germany, 2013.

Reuter, S.; Wilking, B.; and Dietmayer, K.: *Methods to model the motion of extended objects in multi-object Bayes filters*. In: *Proceedings of the 15th International Conference on Information Fusion*, pages 527–534, 2012.

Reuter, S.; Wilking, B.; Wiest, J.; Munz, M.; and Dietmayer, K.: *Real-time multi-object tracking using random finite sets*. In: *IEEE Transactions on Aerospace and Electronic Systems*, volume 49, no. 4, pages 2666–2678, 2013.

Deusch, H.; Wiest, J.; Reuter, S.; et al.: *A random finite set approach to multiple lane detection*. In: *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, pages 270–275, 2012.

Geier, T.; Biundo, S.; Reuter, S.; and Dietmayer, K.: *Track-person association using a first-order probabilistic model*. In: *IEEE 24th International Conference on Tools with Artificial Intelligence*. Volume 1, pages 844–851, 2012.

Geier, T.; Reuter, S.; Dietmayer, K.; and Biundo, S.: *Goal-based person tracking using a first-order probabilistic model*. In: *Proceedings of the 9th UAI Bayesian Modeling Applications Workshop*, 2012.

Glodek, M.; Reuter, S.; Schels, M.; Dietmayer, K.; and Schwenker, F.: *Kalman filter based classifier fusion for affective state recognition*. In: *Multiple Classifier Systems*, pages 85–94, Springer Berlin Heidelberg, 2013.

Meissner, D.; Reuter, S.; and Dietmayer, K.: *Multi-Model-PHD-Filter zur Verfolgung von Verkehrsteilnehmern an Kreuzungen mit Laserscannern*. In: *8. Workshop Fahrerassistenzsysteme FAS*, pages 57–66, 2012.

Meissner, D.; Reuter, S.; and Dietmayer, K.: *Real-time detection and tracking of pedestrians at intersections using a network of laserscanners*. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 630–635, 2012.

Meissner, D.; Reuter, S.; and Dietmayer, K.: *Road user tracking at intersections using a multiple-model PHD filter*. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 377–382, 2013.

Meissner, D.; Reuter, S.; Wilking, B.; and Dietmayer, K.: *Road user tracking using a Dempster-Shafer based classifying multiple-model PHD filter*. In: *Proceedings of the 16th International Conference on Information Fusion*, pages 1236–1242, 2013.

Nuss, D.; Reuter, S.; Konrad, M.; Munz, M.; and Dietmayer, K.: *Using grid maps to reduce the number of false positive measurements in advanced driver assistance systems*. In: *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*, pages 1509–1514, 2012.

Nuss, D.; Wilking, B.; Wiest, J.; et al.: *Decision-free true positive estimation with grid maps for multi-object tracking*. In: *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*, pages 28–34, The Hague, Netherlands, 2013.

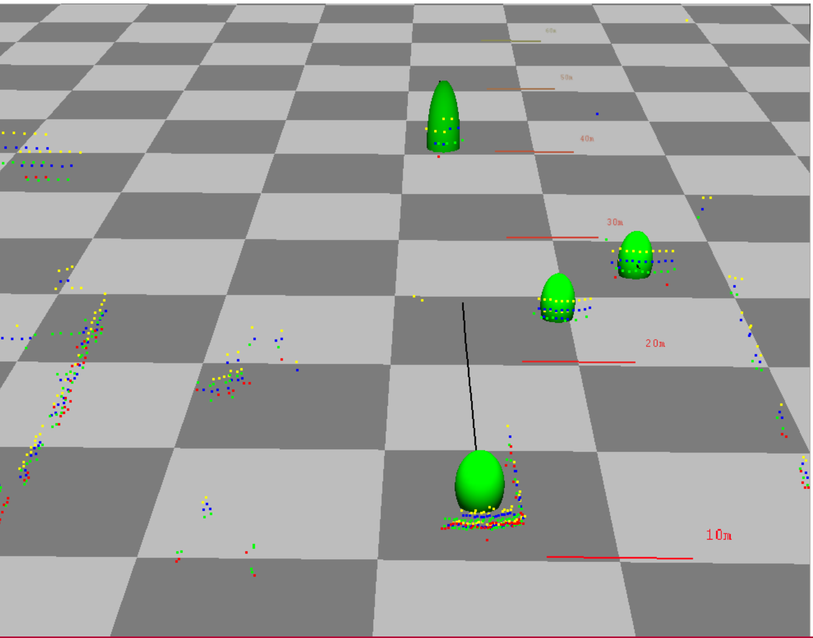
Wilking, B.; Reuter, S.; and Dietmayer, K.: *Probabilistic data association in information space for generic sensor data fusion*. In: *Proceedings of the 15th International Conference on Information Fusion*, pages 317–323, 2012.

Supervised Theses

Bach, M.: *Fahrstreifendetektion für ein autonomes Modellfahrzeug unter Verwendung eines Grafikprozessors*. Bachelor Thesis, Universität Ulm, 2013.

Götz, C.: *Entwicklung von Algorithmen zur Segmentierung mehrerer Personen in Bildern einer Time-Of-Flight Kamera*. Diploma Thesis, Universität Ulm, 2012.

Stern, S.: *Evaluierung von Kaskadenklassifikatoren zur Personendetektion in Videobildern*. Bachelor Thesis, Universität Ulm, 2010.



ISBN 978-3-941543-12-6

e-ISBN 978-3-941543-13-3