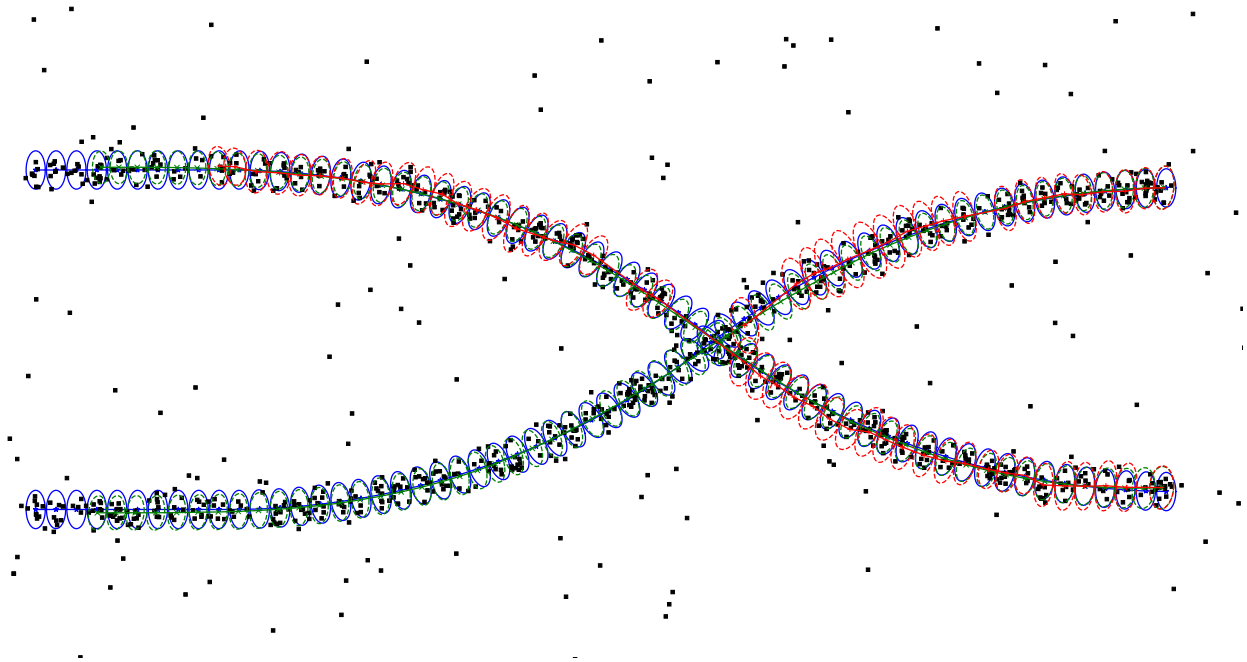
CHALMERS
UNIVERSITY OF TECHNOLOGY



Extended Object Tracking Using Sets of Trajectories with a PHD Filter

A Gamma Gaussian Inverse Wishart Trajectory PHD Filter with Smoothed Extent Estimates

Master's thesis in Systems, Control and Mechatronics

Martin Marcusson, Jakob Sjudin

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020
www.chalmers.se

Master's thesis 2020

Extended Object Tracking Using Sets of Trajectories with a PHD Filter

A Gamma Gaussian Inverse Wishart Trajectory PHD Filter with Smoothed Extent Estimates

Martin Marcusson
Jakob Sjudin

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2020

Extended Object Tracking Using Sets of Trajectories with a PHD Filter
A Gamma Gaussian Inverse Wishart Trajectory PHD Filter with Smoothed Extent Estimates

Martin Marcusson
Jakob Sjudin

© Martin Marcusson, Jakob Sjudin, 2020.

Supervisor: Johan Degerman, SafeRadar Research Sweden AB
Examiner: Lars Hammarstrand, Department of Electrical Engineering

Master's Thesis 2020
Department of Electrical Engineering
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A scenario where two objects cross paths, they start in the left and move to the right. Blue represents ground truth and the black dots are measurements. GGIWPHD filter estimates are red and GGIWTPHD filter estimates are green, where full lines are trajectories and dashed lines are estimated extent. Note that this scenario is not realistic, but it shows the capabilities of estimating trajectories with the GGIWTPHD filter.

Typeset in LaTeX
Gothenburg, Sweden 2020

Extended Object Tracking Using Sets of Trajectories with a PHD Filter
A Gamma Gaussian Inverse Wishart Trajectory PHD Filter with Smoothed Extent Estimates
Martin Marcusson, Jakob Sjudin
Department of Electrical Engineering
Chalmers University of Technology

Abstract

An important part of many automated systems today is to estimate where surrounding objects are located. To do this, different tracking algorithms can be used to gain more accurate estimates from noisy sensor measurements over time. Due to increased sensor resolution, it is becoming more common that multiple measurements can originate from the same object at each scan. If this is the case, the object is often referred to as an extended object. This introduces new challenges but allows for the estimation of an objects size and shape. Additionally, questions related to the history of an objects state can be of interest in tracking. For example, what is the probability that an object was at a certain position in the past given the current position. Thus, the trajectory of an object is needed. Previously, many filters used for extended object tracking (EOT) have not been able to reliably build trajectories.

This thesis proposes a new probability hypothesis density (PHD) filter used for EOT that is able to build trajectories which also include estimates of the object's size and shape. The new filter is called the Gamma Gaussian Inverse Wishart Trajectory PHD (GGIWTPHD) filter and has been developed with a heuristic approach extending previous work. This is done by extending the Gamma Gaussian Inverse Wishart PHD (GGIWPHD) filter by including results from the Trajectory PHD filter and introducing Bayesian smoothing of the extent estimates.

The performance of the GGIWTPHD filter is evaluated by comparing it to a GGIWPHD filter with labeled components using Monte Carlo simulation on simulated data. A metric is proposed for evaluating trajectories and is used to calculate a total root mean square (RMS) error in each time step. It is shown that in situations with spatially close targets the trajectory estimates from the GGIWTPHD filter are more accurate and reliable than trajectories built using estimates from a GGIWPHD filter with labeled components. The GGIWTPHD is also shown to yield better extent and position estimates in all demonstrated scenarios.

Keywords: Multiple object tracking, Extended object, Gamma gaussian inverse wishart, Trajectories, Bayesian smoothing, Random finite sets, Probability hypothesis density, PHD filtering.

Acknowledgements

We would like to thank Johan Degerman, SafeRadar Research AB who supervised the project and provided help when it was needed. We would also like to thank our examiner, Lars Hammarstrand who provided useful feedback throughout the project. Lastly, we would like to thank Lennart Svensson who provided support regarding some of the concepts dealt with in the thesis.

Martin Marcusson & Jakob Sjudin, Gothenburg, June 2020

Contents

List of Abbreviations

CT	coordinated turn
DA	data association
EOT	extended object tracking
GGIWPHD	Gamma Gaussian Inverse Wishart PHD
GGIWTPHD	Gamma Gaussian Inverse Wishart Trajectory PHD
lidar	light detection and ranging
MEOT	multiple extended object tracking
MOT	multiple object tracking
pdf	probability density function
PHD	probability hypothesis density
radar	radio detection and ranging
RFS	random finite set
RMS	root mean square

List of Figures

List of Tables

1

Introduction

In modern society it is becoming increasingly common to employ the help of automated systems to assist humans in tasks such as driving. In the same manner as humans use their senses to navigate in the world, these systems also need ways to perceive the surrounding environment. This is achieved by using a set of sensors, mainly radar, lidar, and cameras. Since the measurements from these sensors are often corrupted by noise, methods such as filtering and sensor fusion are needed in order to obtain more accurate observations.

For safety reasons, the filtering algorithms need to be as accurate and reliable as possible in order to not make decisions that could have unwanted results. In order to achieve this, multiple problems need to be solved. This includes associating measurements to objects while differentiating between multiple objects that appear and disappear in the surveillance area continuously. Additionally, when the size and shape of an object is of interest this problem is further complicated since this changes how the measurements are to be interpreted.

In the following parts of this chapter, background to the general tracking problem and the specific areas relevant to this thesis is presented first. This is followed by a definition of the purpose of this work. The overall goals are then presented together with the specific contributions to the field of tracking. Last, an overall outline of the rest of the report is presented.

1.1 Background

Tracking of moving objects is becoming more important in many different areas, e.g. surveillance of aircraft or autonomous drive systems where automation is becoming more common. Tracking in this context refers to estimating the kinematic properties such as position, velocity and heading of one or multiple objects that are in the field of view of one or more sensors. In many applications it is also of interest to track the trajectory of objects where a trajectory denotes a sequence of states that belong to the same object.

One of the more common sensors used in these areas is radar. Measurements obtained when using radar are often influenced by noise caused by inaccuracies in the

radar itself or by other factors imposed by the environment [1]. It is also possible for radars and other sensors to obtain measurements that do not originate from any object of interest. These measurements are often referred to as clutter measurements and can, e.g, be generated from the ground.

To interpret noisy measurements and obtain estimates of an objects state when the measurement to object association is unknown, tracking methods are utilized. The idea is to connect several noisy measurements over time combined with models that describe the evolution of the kinematic states, but also the intrinsic and extrinsic parameters of the sensor in order to get a more accurate estimate of the current state of a system.

Object tracking in clutter has been a research topic for many years, but until recently most of it operates under the assumption that an object yields at most a single measurement at each time step. This is reasonable for smaller objects or objects that are sufficiently far away to only occupy one resolution cell of a sensor. However, since the resolution of sensors is increasing, there are many cases where an object may span multiple resolution cells and thus yield multiple measurements at each scan. Such an object is often referred to as an extended object [2]. An example of an extended object is shown in Figure 1.1.

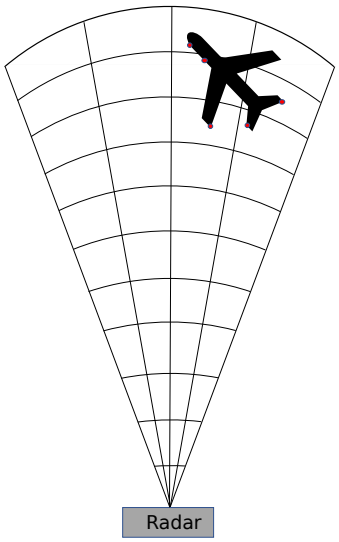


Figure 1.1: An example of radar measurements obtained from an extended object, i.e an object covering multiple resolution cells.

In the following subsections, some background regarding multiple object tracking (MOT) and extended object tracking (EOT) as well as tracking sets of trajectories is presented.

1.1.1 Multiple object tracking

MOT can in broad terms be defined as *"the process of successively determining the number and states of multiple dynamic objects based on noisy sensor measurements. [...] MOT based on the point-object assumption is a very complex problem due to sensor noise, missed detections, clutter detections, measurement origin uncertainty, and an unknown and time-varying number of objects."* [1]. This formulation implies a number of challenges, and many different filtering algorithms have been proposed in the literature. Some of these are:

- Multiple Hypothesis Tracking (MHT) [3]
- Joint Probabilistic Data Association (JPDA) [4]
- Probabilistic Multiple Hypothesis Tracking (PMHT) [5]
- Random Finite Set (RFS) methods [6]

The RFS-based methods which treat the tracking problem in a random finite set theoretical framework have been prominent in recent years. Briefly, a RFS can be described as a set where both the elements of the vectors in it as well as the total number of vectors are random. This is a very useful way to represent the multiple object distribution in MOT since the number of objects is in general not known and can vary over time.

An example of a RFS-based method is the probability hypothesis density (PHD) filter [7] which is in many aspects similar to the Kalman filter in that it approximates the exact multi-object posterior distribution by its first moment. Random finite sets and the probability hypothesis density will be discussed further in Section 2.2.

One method to implement a PHD filter is the so called Gaussian Mixture PHD (GMPHD) filter [8]. In this implementation the PHD is represented using a weighted mixture of Gaussian distributions, where each mixture component corresponds to a hypothesis in the PHD.

1.1.2 Extended object tracking

Normal point object trackers often have difficulty tracking extended objects, since multiple measurements originating from the same object causes a point object tracker to start many false tracks. One approach to tackle this problem is to utilize clustering in order to approximate which measurements originate from the same source. This has been done in an extended version of the GMPHD filter, which can be called the EGMPHD filter, proposed by Granström et al. [9].

One advantage of tracking extended objects is that in addition to estimating the kinematic properties of the object, the spatial extent as well as the orientation can be estimated. This allows for better classification of the objects where size may be a factor, and remedies the problem of false tracks as mentioned previously. However, estimation of the spatial extent is not done in the EGMPHD filter. Other filters such as the Gaussian inverse Wishart PHD (GIWPHD) filter and Gamma Gaussian inverse Wishart PHD (GGIWPHD) filter have been presented [10, 11] which are extensions of the GMPHD/EGMPHD filters that can estimate spatial extent.

A common way to represent the spatial property of an object is by estimating it as an ellipse, this allows the use of random matrices proposed by Koch [12]. Both the GIWPHD and GGIWPHD filters use an inverse Wishart distribution to represent the extent as a random matrix. The GGIWPHD filter additionally makes use of a Gamma distribution to model the number of expected measurements originating

from the corresponding extended object.

For further reading on extended object tracking and its applications, a comprehensive overview of the field is provided in [13].

1.1.3 Tracking sets of trajectories

PHD filters can not inherently build trajectories. Therefore, different methods to label object states and adding unique tags to the PHD components have been proposed [14, 15, 16]. This makes it possible to build trajectories with PHD filters, but it is not a mathematically sound way to provide trajectories and can, in some cases, be unstable. A major disadvantage of using labeled PHD components is that it can lead to more track switches, false targets, and missed detections.

Methods that provide trajectories from PHD filters without adding labels or tags have been presented, one of these is the trajectory PHD (TPHD) filter [17]. This approach is similar to regular PHD filtering but has one major difference, the filtering recursions are designed to estimate a set of trajectories instead of a set of object states at each time step.

1.2 Purpose

Tracking sets of trajectories have been shown to yield promising results [18, 17] and the purpose of this thesis is to present a method to track sets of trajectories in a filter capable of extended object tracking. To be able to build tracks with a PHD filter that can also estimate extent is of great interest since it could improve the performance of both the kinematic state estimation as well as the spatial extent estimation.

1.3 Objectives and contributions

With inspiration from the proposed trajectory PHD filter in [17], this thesis proposes a novel PHD filter for extended object tracking that produces trajectories with both kinematic state estimates and extent estimates. The new filter also includes smoothing of the extent estimates, which should give more accurate estimates of past states.

This is achieved by modifying the Gamma Gaussian Inverse Wishart PHD (GGIWPHD) filter proposed by Granström et al. [11] to handle sets of trajectories for the kinematic state as well as the extent state. The proposed filter is named the Gamma Gaussian Inverse Wishart Trajectory PHD (GGIWTPHD) filter.

The goals for the thesis are the following:

- Implement a GGIWTPHD filter by using a heuristic approach. For this filter the following conditions should be fulfilled:
  - It should be able to estimate the trajectory of an object with the same or better performance than already existing EOT filters that estimate sets of trajectories such as the GGIWPHD filter with labeled components [11].
  - It should be able to estimate the spatial extent of an object as well as or better than already existing EOT filters capable of estimating extent, such as the GGIWPHD filter.
  - The number of false, missed and switched tracks should be lower than the EOT filters used for comparison.
- Implement a GGIWPHD filter with labeled components that is to be used for comparison

In this case the performance of the filters will be evaluated using a slightly modified metric designed for evaluating MOT algorithms that uses sets of trajectories [19]. This metric also penalises missed objects, false tracks and track switching. Since extent is estimated in these filters, both the localisation and extent error is calculated using the Gaussian Wasserstein (GW) distance between the estimate and ground truth [20] which is a metric used when evaluating EOT algorithms.

1.4 Outline

This report is structured as follows. Chapter 2 presents some background theory needed to get a deeper understanding of the problem and how it can be solved. In addition, this chapter includes some specifics on metrics used for performance evaluation. In chapter 3 the implementation is specified in more detail. Next, the method for evaluation is explained and results and performance of the GGIWTPHD filter compared to the GGIWPHD filter can be seen in chapter 4. The final chapter discusses the results presented and summarizes the conclusions that can be drawn. The report concludes by proposing some future work.

2

Theory

This chapter introduces some of the theory that is needed in order to derive the algorithms presented in this thesis.

2.1 Bayesian state estimation

Bayesian state estimation is concerned with estimating a stochastic unknown state $\mathbf{x}_k$ given measurements $\mathbf{z}_{1:k}$. These measurements are often corrupted by noise which implies that the measurement can not be directly used to describe $\mathbf{x}_k$. Instead the states are viewed as a random distribution $p(\mathbf{x})$.

The goal is to obtain a posterior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ that describes the state $\mathbf{x}_k$ given a prior distribution $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ and a measurement $\mathbf{z}_k$ at time k. This can be done by applying the Chapman-Kolmogorov Equation together with Bayes' rule [21] as:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1} \tag{2.1}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k} \tag{2.2}$$

These equations describe the optimal Bayesian recursion that can be used to estimate the unknown and random state $\mathbf{x}_k$ given that the prior and posterior distribution belong to the same family of distributions.

2.1.1 Kalman filter

One of the most common filters based on Bayes' rule used today is the Kalman filter [22] which is an optimal minimum mean square error (MMSE) estimator under the following assumptions:

- The state transition is linear and the predicted density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is Gaussian: $p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; F_{k-1}\mathbf{x}_{k-1}, Q_{k-1})$ where F_{k-1} is a transition matrix and Q_{k-1} is a covariance matrix.

- The measurement likelihood function $h_k(\mathbf{z}_k|\mathbf{x}_k)$ is linear and Gaussian: $h_k(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; H_k\mathbf{x}_k, R_k)$
- The process and measurement noises w_k and v_k are normally distributed white noise processes, $w_k \sim \mathcal{N}(0, Q_k)$, $v_k \sim \mathcal{N}(0, R_k)$
- The system fulfills the Markov property

The Kalman filter calculates the posterior by propagating the mean and covariance that parametrizes its normal distribution through time. In more general terms the transition density and measurement likelihood functions can be described as:

- **Motion model** $f_{k|k-1}$
  A model which describes $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ based on prior knowledge about how the state is expected to evolve over time. This is used to make a guess on what $\mathbf{x}_k$ is before any observation $\mathbf{z}_k$ has been made. It also models the uncertainty of the state evolution when adding process noise.
- **Measurement model** h_k
  The measurement model is used to describe a mapping between the state space $\mathfrak{X}$ and measurement space $\mathfrak{Z}$. The model also captures the uncertainty distribution of the measurements. It is used to describe the distribution of the predicted measurement.

The closed-form equations to the Kalman filter are:

Prediction:

$$\hat{\mathbf{x}}_{k|k-1} = F_k\hat{\mathbf{x}}_{k-1|k-1} \tag{2.3}$$
$$P_{k|k-1} = F_kP_{k-1|k-1}F_k^T + Q_k \tag{2.4}$$

Update:

$$\mathbf{y}_k = \mathbf{z}_k - H_k\hat{\mathbf{x}}_{k|k-1} \tag{2.5}$$
$$S_k = H_kP_{k|k-1}H_k^T + R_k \tag{2.6}$$
$$K_k = P_{k|k-1}H_k^TS_k^{-1} \tag{2.7}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k\mathbf{y}_k \tag{2.8}$$
$$P_{k|k} = P_{k|k-1} + K_kS_kK_k^T \tag{2.9}$$

Where $\hat{\mathbf{x}}_{k|k-1}$ denotes the predicted state estimate and $\hat{\mathbf{x}}_{k|k}$ denotes the updated state estimate at time k.

The Kalman filter computes the posterior distribution by recursively propagating the mean and covariance in time. In the real world the assumptions made for the Kalman filter to be an optimal filter seldom hold, and because of this there are many different versions of the Kalman filter that can handle nonlinear motion- and

measurement models. Examples of these are the Extended Kalman Filter, Unscented Kalman filter and Cubature Kalman filter [23, 24]. The Extended Kalman filter approximates the transition and likelihood densities as Gaussian distributions. The Cubature- and Unscented Kalman filters however, uses methods to pick a set of sample points around the mean, called sigma points. These points are propagated through the nonlinear functions, from which new mean and covariance estimates are formed.

In tracking applications the Kalman filter can be used to describe the single-object posterior distribution under the assumption that there are no unknown associations between the measurements and the object states. In MOT the interest lies in estimating the multi-object posterior distribution which can be seen as a joint distribution of the single-object posteriors. This leads to the conclusion that the Kalman filter in its basic form is not sufficient for usage when tracking multiple objects in clutter. It is however a common starting point when deriving MOT filters, since many filters employ the strategy of dividing the problem into multiple single object tracking problems. The Kalman filter, or its nonlinear extensions, can then be used to propagate the single-object posterior distributions over time.

2.2 Random finite sets and the probability hypothesis density

In this section an overview of random finite sets (RFS) is presented. Random finite set theory is a useful tool in MOT since it can capture many of the difficulties introduced by tracking an unknown number of objects in clutter. The formal definition of an RFS is as follows:

Let $\mathfrak{X}$ and $\mathfrak{Z}$ denote the state and measurement spaces respectively. Then $\mathfrak{X}^{\infty}$ and $\mathfrak{Z}^{\infty}$ denote the hyperspaces of all finite subsets of $\mathfrak{X}$ and $\mathfrak{Z}$ with the empty set $\emptyset$ included. A random finite set is a random variable, denoted by Ψ, on $\mathfrak{X}$ or $\mathfrak{Z}$ [6].

An alternative explanation is that a random finite set can be described as a set $\Psi = \{\psi^i\}_{i=1}^{N}, \Psi \in \mathfrak{X}$ or $\mathfrak{Z}$ where both the elements of the vectors ψ^i and the total number of state vectors $N \in \mathbb{N} \cup 0$ are random. N is referred to as the cardinality of the RFS which is written as:

$$|\Psi| = N$$

A random set of states $\mathbf{X} = \{\mathbf{x}_k\}_{k=1}^{N}$ will have the following realisations:

$$\begin{aligned} \mathbf{X} &= \emptyset, \quad |\mathbf{X}| = 0 \\ \mathbf{X} &= \{\mathbf{x}_1\}, \quad |\mathbf{X}| = 1 \\ \mathbf{X} &= \{\mathbf{x}_1, \mathbf{x}_2\}, \quad |\mathbf{X}| = 2 \\ &\vdots \end{aligned}$$

Random finite sets are defined as unordered, meaning that $\{\mathbf{x}_1, \mathbf{x}_2\} = \{\mathbf{x}_2, \mathbf{x}_1\}$

RFS theory is a useful modelling tool since it handles many of the challenges of tracking an unknown and time-varying number of objects in a non-heuristic way. Since the multi object posterior distribution can be seen as an RFS, MOT can be performed by estimating this set. In point-object tracking the states often describe position, velocity and other kinematic properties but when the target is extended the state will also include parameters that determine the shape and orientation of an object's extent.

The first moment of an RFS is called the probability hypothesis density (PHD), in other words: The PHD is to an RFS as the expected value is to a random variable [25] and can be viewed as an intensity function $D(\cdot)$ defined over some space that the RFS belongs to. The local maxima of $D(\cdot)$ can be viewed as the different object estimates and the cardinality of an RFS $\mathbf{x}$ over some space $\mathcal{A}$ can be expressed as the integral over D:

$$\int_{\mathcal{A}} D(x)dx = \mathbb{E}[|\mathbf{x} \cap \mathcal{A}|]$$

Where $\mathbb{E}[\cdot]$ is the expected value operator.

2.3 Data association

A measurement set $\mathbf{Z}_k$ obtained at time k can be seen as a union:

$$\mathbf{Z}_k = \{\mathbf{z}_k^{(j)}\}_{j=1}^{m} \cup \mathbf{C}_k$$

where $\{\mathbf{z}_k^{(j)}\}_{j=1}^{m}$ is the set of measurements that originate from actual targets within the sensors field of view and $\mathbf{C}_k$ is the set of clutter measurements. A measurement is referred to as a clutter measurement if it originates from some source in the sensors field of view that is not of interest, e.g. reflections from the ground. In MOT only the target-generated measurements should be used when updating the multi-target posterior distribution which means that the set $\mathbf{C}_k$ must be filtered out. Assuming that the set of clutter measurements can be removed such that $\mathbf{Z}_k$ becomes:

$$\mathbf{Z}_k = \{\mathbf{z}_k^{(j)}\}_{j=1}^{m}$$

The association then concerns matching the i:th object state $\mathbf{x}_k^{(i)}$ with the j:th measurement $\mathbf{z}_k^{(j)}$. In point object tracking the number of associations grow very rapidly with the cardinality of the set $\mathbf{Z}_k$. In order to reduce the number of associations that can be made, unlikely measurements are often removed. One common method to reduce the number of available measurement to object associations is to use gating [26].

This is performed by forming a gate around each predicted measurement $\bar{\mathbf{z}}_k$ and disregarding the measurements outside these gates. For Gaussian distributions it is common to use ellipsoidal gates. To measure the distance d^2 between the predicted measurement $\bar{\mathbf{z}}_k = h_k(\hat{\mathbf{x}}_{k|k-1})$ and a measurement $\mathbf{z}_k$, the Mahalanobis distance is

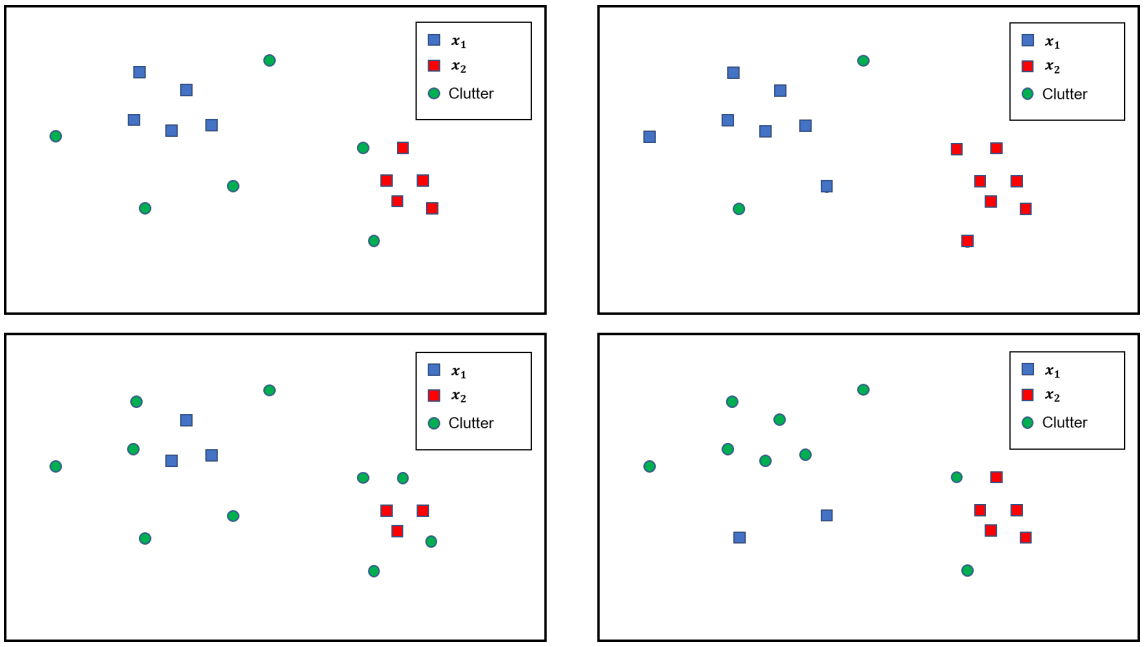


Figure 2.1: A measurement set where two extended objects, x_1 and x_2 are assumed to be present. Four scenarios that depict four different measurement set associations are demonstrated.

commonly used. If d^2 is larger than some threshold G then the measurement is disregarded as clutter. The Mahalanobis distance d^2 is expressed as:

$$d^2 = (\mathbf{z}_k - \bar{\mathbf{z}}_k)^T S_k^{-1} (\mathbf{z}_k - \bar{\mathbf{z}}_k) \tag{2.10}$$

Where S_k is the innovation covariance of the predicted measurement shown in Equation 2.6. One common way of selecting the threshold G is to set it according to a cumulative Chi distribution $\chi(n_z)$ with n_z degrees of freedom where n_z denotes the dimension of the measurements.

In multiple extended object tracking (MEOT) each object state $\mathbf{x}_k$ can generate more than one measurement per scan, and the data association can be divided into two sub-problems:

- Partitioning the measurement set $\mathbf{Z}_k$ into partitions $\mathcal{P}_i$
- Associating the partitions to object states $\mathbf{x}_k$

To demonstrate partitioning, an example from Mahler [27] is used. Given a measurement set $\mathbf{Z}_k = \{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\}$ this set can be partitioned in the following ways:

$$
\begin{aligned}
\mathcal{P}_1 &: \mathbf{W}_1^1 = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\right\} \\
\mathcal{P}_2 &: \mathbf{W}_1^2 = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}\right\}, \mathbf{W}_2^2 = \left\{\mathbf{z}_k^{(3)}\right\} \\
\mathcal{P}_3 &: \mathbf{W}_1^3 = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(3)}\right\}, \mathbf{W}_2^3 = \left\{\mathbf{z}_k^{(2)}\right\} \\
\mathcal{P}_4 &: \mathbf{W}_1^4 = \left\{\mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\right\}, \mathbf{W}_2^4 = \left\{\mathbf{z}_k^{(1)}\right\} \\
\mathcal{P}_5 &: \mathbf{W}_1^5 = \left\{\mathbf{z}_k^{(1)}\right\}, \mathbf{W}_2^5 = \left\{\mathbf{z}_k^{(2)}\right\}, \mathbf{W}_3^5 = \left\{\mathbf{z}_k^{(3)}\right\}
\end{aligned}
$$

Where $\mathbf{W}_i^j$ is the i:th cell of the j:th partition. The measurements in a cell $\mathbf{W}$ are the measurements that are associated with the object with index i.

An example demonstrating some different data associations for two extended objects are given in Figure 2.1. If all possible partitions of the measurement set are considered, the number of possible associations in MEOT is given by Bell's number [28]. If we assume that 130 measurements are received in a single scan this would be $B_{130} \propto 10^{161}$ [9] which is not computationally tractable. Therefore, methods that reduce the number of possible partitions and associations are needed.

Data association is often a large contributor to the time complexity of MOT/EOT algorithms and an important area for future research within the field of object tracking. In point-object tracking a common method is to assign measurements to object states using assignment algorithms like Murtys algorithm [29], Hungarian algorithm [30] or Gibbs sampling [31]. In the extended target case, clustering is often performed to find partitions that contain clusters which are assigned to object states. Stochastic optimization [32] has also been utilized as well as data association based on the likelihood function of the multi object posterior [33].

2.4 Bayesian filtering utilizing the probability hypothesis density

In Section 2.1 the equations for the optimal single object Bayes filter are presented. In the multi object case the equations have the following form:

$$p(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) = \int p(\mathbf{X}_k|\mathbf{X}_{k-1})p(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1})d\mathbf{X}_{k-1} \tag{2.11}$$

$$p(\mathbf{X}_k|\mathbf{Z}_{1:k}) = K^{-1}g_k(\mathbf{Z}_{1:k}|\mathbf{X}_k)p(\mathbf{X}_k|\mathbf{Z}_{1:k-1}) \tag{2.12}$$

$$K = \int g_k(\mathbf{Z}_{1:k}|\mathbf{X}_k)p(\mathbf{X}_k|\mathbf{Z}_{1:k})d\mathbf{X}_k \tag{2.13}$$

Where

- $\mathbf{X}_k$ represents the multi-object state at time step k and $\mathbf{X}_{k-1}$ represents the previous multi object state. The multi-object state $\mathbf{X}_k$ has an unknown number of object states $\mathbf{x}_k$.

- $\mathbf{Z}_k$ is the observation set containing all measurements obtained at a time step k and $g_k(\cdot)$ denotes the likelihood function of observing the measurements set $\mathbf{Z}_k$ given the multi object state $\mathbf{X}_k$.

This way of formulating multi-object filtering is theoretically correct, but is computationally challenging since it is difficult to represent the exact posterior distribution $p(\mathbf{X}_k|\mathbf{Z}_{1:k})$, and thus approximations must be made. In the following section a method for filtering is presented where the first-order statistical moment is propagated through time instead of the full posterior distribution.

2.4.1 Point objects

MOT under the point-object assumption using random finite sets was first proposed by Mahler [7] where both the set of object states $\mathbf{X}_k$ at time k and the measurement set $\mathbf{Z}_k$ are assumed to be Poisson multi-object RFSs where the random elements are independent and identically distributed (IID). It is also assumed that the measurement set $\mathbf{Z}_k$ contains clutter measurements distributed according to some density $\mathbf{c}(\mathbf{z})$. The cardinality of the clutter measurement set $\mathbf{C}_k$ is Poisson distributed with a constant intensity λ.

The proposed filter represents the multi-object state distribution $p(\mathbf{X}_k|\mathbf{Z}_k)$ by its first order moment which is the probability hypothesis density $D_{k|k}$. This is similar to the Kalman filter where the single object density $p(\mathbf{x}_k|\mathbf{z}_k)$ is represented by its first order moment which can be propagated through time.

A visual representation of the filter recursions from Mahler [7] is shown below.

$$\begin{array}{ccccccc} \cdots \rightarrow & p\left(\mathbf{X}_k|\mathbf{Z}_k\right) & \overset{\text{Time prediction}}{\rightarrow} & p\left(\mathbf{X}_{k+1}|\mathbf{Z}_k\right) & \overset{\text{Bayes' rule}}{\rightarrow} & p\left(\mathbf{X}_{k+1}|\mathbf{Z}_{k+1}\right) & \rightarrow \cdots \\ & \downarrow & & \downarrow & & \downarrow & \\ \cdots \rightarrow & D_{k|k} & \overset{\text{Prediction}}{\rightarrow} & D_{k+1|k} & \overset{\text{Correction}}{\rightarrow} & D_{k+1|k+1} & \rightarrow \cdots \end{array}$$

This filter is known as the probability hypothesis density (PHD) filter. One way to perform extraction in the PHD filter proposed by Mahler [7], is to estimate the expected number of objects in the surveillance area $\mathcal{A}$ given by the integral over D, $\int_{\mathcal{A}} D(x)dx = N$ and then extracting N maxima from D.

2.4.2 Extended objects

When tracking extended objects, similar approaches to point-object tracking can be used but with some extensions to the models. The main problems introduced when tracking extended objects are:

- How to group measurements that are likely to originate from the same extended object together. Commonly this is done by using various clustering methods in order to partition the measurement set into a set of partitions $\mathcal{P}$ where each partition contains cells $\mathbf{W}$. This topic is further discussed in Section 2.3.

- How to estimate the number of measurements that an object is likely to generate. Gilholm et al. [34] proposed that the number of detections can be modelled using a Poisson distribution with a state dependent rate $\gamma(\mathbf{x}_k)$. Because of the conjugacy between the Gamma distribution and the Poisson distribution Granström and Orguner have proposed a Bayesian recursion to estimate the Poisson rate γ_k recursively by using a Gamma distribution [35].

One important aspect of extended object tracking is how to model the extent of the object in the filter. Since measurements originating from extended objects usually do not reveal the true object extent, and because of increased complexity and run-times of filters, approximations are often needed. The different ways to model extended object can be divided into:

- Not modelling the objects shape at all, thus only estimating the kinematic properties of an extended object without explicitly describing its shape. An example if this type of filter is given in [9] where Granström et al. propose an extension to the PHD filter that can track extended objects without explicitly modelling its extent by using clustering.

- Modelling using geometric shapes such as ellipses [36], lines [37] or rectangles [38]. These approaches are attractive from a complexity standpoint, and can in many cases be a crude but useful representation of an objects extent.

- Modelling using a more general shape description. One example of this is using Star-Convex Shapes where the shape is parameterized by for example, a Fourier series expansion can be a suitable way to represent more general shapes [39, 40].

In the framework presented by Koch [12] the extent of an object is modelled by an inverse Wishart (IW) distribution denoted as:

$$X = \mathcal{IW}_d(X; v, V) \tag{2.14}$$

Where X denotes the positive-definite extent matrix, $v > p - 1$ denotes the degrees of freedom and $\lambda(V) > 0$, where V is a scale matrix of size $p \times p$ and $\lambda(\cdot)$ denotes eigenvalue operator. This representation of the extent is referred to as the random matrix model and the extent is represented by an ellipsoid.

2.4.3 Representing the probability hypothesis density

PHD filters often have to use approximations since representing the true PHD is computationally intractable. The most common approximation for a point target PHD filter was proposed by Vo and Ma [8] where the PHD is approximated by a mixture of weighted Gaussian distributions as:

$$D_{k|k} = \sum_{i=1}^{J_k} w_k^{(i)} \mathcal{N}(x; m_k^i, P_k^i) \tag{2.15}$$

Where J^k is the number of mixture components, $w_k^{(i)}$ is the weight of the i:th component and $\mathcal{N}(x; m_k^i, P_k^i)$ is a normal distribution that represents the i:th component. The prediction and update steps for each Gaussian is done by a simple Kalman prediction and update. This filter operates under similar assumptions that the single-object state densities and measurement likelihoods are Gaussian, but nonlinear versions of the GMPHD filter are also proposed by Vo and Ma. For more details regarding implementation, see [8].

One drawback of the GMPHD filter is that the number of mixture components J_k grows very quickly over time due to the nature of the filtering recursion and thus pruning[1], merging[2] and capping[3] is used to keep the filter computationally tractable.

In extended object tracking, similar filters that represent the PHD as a mixture of components have also been proposed. One example is the GGIWPHD filter. In this filter, the extended object state ξ_k is defined as the triple

$$\xi_k \triangleq (\gamma_k, \mathbf{x}_k, X_k) \tag{2.16}$$

The components used in a GGIWPHD filter are then assumed to be independent [25], i.e.:

$$\begin{aligned} p(\xi_k|\mathbf{Z}_k) &= p(\gamma_k|\mathbf{Z}_k)\, p(\mathbf{x}_k|\mathbf{Z}_k)\, p(X_k|\mathbf{Z}_k) \\ &= \mathcal{G}\left(\gamma_k; \alpha_{k|k}, \beta_{k|k}\right) \mathcal{N}\left(\mathbf{x}_k; m_{k|k}, P_{k|k}\right) \mathcal{IW}_d\left(X_k; v_{k|k}, V_{k|k}\right) \end{aligned} \tag{2.17}$$

Where

- $\mathbf{Z}_k$ is the set of measurements at time k.
- $p(\gamma_k|\mathbf{Z}_k)$ describes the distribution over the variable γ_k which represents the number of measurements that an object is expected to generate.
- $p(\mathbf{x}_k|\mathbf{Z}_k)$ describes the distribution over the kinematic state.
- $p(X_k|\mathbf{Z}_k)$ describes the distribution over the extent estimate.

Equation 2.17 can be written as

$$p(\xi_k|\mathbf{Z}_k) = \mathcal{GGIW}\left(\xi_k; \zeta_{k|k}\right) \tag{2.18}$$

where $\zeta_{k|k}$ is the set of GGIW density parameters, i.e.

$$\zeta_{k|k} = \left\{\alpha_{k|k}, \beta_{k|k}, m_{k|k}, P_{k|k}, v_{k|k}, V_{k|k}\right\} \tag{2.19}$$

$\alpha_{k|k}$ and $\beta_{k|k}$ are the shape and rate parameters of the Gamma distribution. $m_{k|k}$ and $P_{k|k}$ are the mean and variance of the normal distribution. $v_{k|k}$ and $V_{k|k}$ are the degrees of freedom and scale matrix in the inverse Wishart distribution.

[1] Removing components $w^{(i)}$ smaller than some threshold $\overline{w}$

[2] Combining mixture components that are similar by measuring e.g. Mahalanobis distance [41] between them

[3] Restricting the number of components in the mixture to some number M

Thus, the PHD in the GGIWPHD filter is represented as a weighted mixture of GGIW components:

$$D_{k|k} = \sum_{i=1}^{J_k} w_k^{(i)} \mathcal{GGIW}\left(\xi_k; \zeta_{k|k}^{(i)}\right) \tag{2.20}$$

In this model, the measurement rate γ_k, the kinematic state $\mathbf{x}_k$, and the extension state X_k are all modeled as independent. In reality, it would be reasonable to assume that the number of measurements generated from an object should depend on the size of the object as well as the distance to the sensor (i.e. the object position). Ignoring this dependence in the model may seem inadequate, however, in the general case this dependence is hard to model and it has been shown that the performance is often not affected to a large extent [35].

The kinematic state and the extension state are modeled as independent because it is difficult to handle nonlinear state vectors when there is dependence between the kinematic state $\mathbf{x}_k$ and the extent X_k. It can be shown that this assumption still yields good results and allows for more general kinematic state vector representations [42]. The necessary mutual dependence between the estimates of the position $\mathbf{x}_k$ and the extent X_k are provided through the time and measurement update in the filter, see Section 3.2.

2.5 Sets of trajectories

Many of the aforementioned approaches to object tracking are concerned with estimating a posterior distribution $p(\mathbf{X}_k|\mathbf{Z}_{1:k})$ where $\mathbf{X}_k$ denotes the set of object estimates at time step k. This means that previous estimates are not stored in $\mathbf{X}_k$. In many applications the previous states of $\mathbf{X}_k$ can also be of interest which means tracking the trajectory of a state with index j: $\mathcal{X}_k^{(j)} = \{x_1^{(j)}, x_2^{(j)}, \ ... \ , x_i^{(j)}\} \ \in \mathbf{X}_k$ over the lifespan i of the tracked object. Note that in this context, $\mathbf{X}_k$ denotes the set of trajectories at time step k.

An object state $\mathbf{x}_k \in \mathbb{R}^{n_x}$ contains properties related to the object such as position and velocity at time step k. A trajectory is then defined as a sequence of object states as $\mathcal{X} = (t, \mathbf{x}_{1:i})$ where t is the time of birth of the trajectory and i denotes the length of the trajectory. This implies that the dimension of the state vector changes as the trajectory grows longer.

In this formulation a trajectory is unknown before it comes into existence and after it has died. In filters that track sets of trajectories it is possible to store dead trajectories for later retrieval if that is of interest to the application.

It is also useful to define a function that selects the most previous entry in a trajectory $\mathcal{X}$, García-Fernández and Svensson [18] define a function $\tau_k(X)$ at time k as:

$$\tau_k(\mathcal{X}) = \begin{cases} \{x_{k+1-t}\}, & t \leq k \leq t+i-1 \\ \emptyset, & \text{elsewhere} \end{cases} \tag{2.21}$$

Given a set of trajectories $\mathbf{X}$ the set of object states $\tau_k(\mathbf{X})$ at time k is:

$$\tau_k(\mathbf{X}) = \bigcup_{\mathcal{X} \in \mathbf{X}} \tau_k(\mathcal{X}) \tag{2.22}$$

And $|\tau_k(\mathbf{X}_k)| = N_k$ where N_k represents the cardinality of $\mathbf{X}$ at time k.

2.6 Bayesian smoothing of extended object estimates

Smoothing denotes the practice of calculating the posterior distribution at time k conditioned on future information up until some future timestep $K > k$. Let $\vartheta = (\mathbf{x}_k,\ X_k)$ denote the extended object state and $\mathbf{Z}_{1:k}$ denote the set of measurements from timestep 1 up to, and including time k. The posterior distribution $p(\vartheta_k|\mathbf{Z}_{1:K})$ that is conditioned on both future and past measurements then becomes:

$$p(\vartheta_k|\mathbf{Z}_{1:K}) = p(\vartheta_k|\mathbf{Z}_{1:k}) \int \frac{p(\vartheta_{k+1}|\vartheta_k) p(\vartheta_{k+1}|\mathbf{Z}_{1:K})}{p(\vartheta_{k+1}|\mathbf{Z}_{1:k})} d\vartheta_{k+1} \tag{2.23}$$

Where

$$p(\vartheta_{k+1}|\mathbf{Z}_{1:k}) = \int p(\vartheta_{k+1}|\vartheta_k) p(\vartheta_k|\mathbf{Z}_{1:k}) d\vartheta_k$$

is the Chapman-Kolmogorov prediction and

$$p\left(\vartheta_{k+1}|\mathbf{Z}_{1:k+1}\right) = \frac{p\left(\mathbf{Z}_{k+1}|\vartheta_{k+1}\right) p\left(\vartheta_{k+1}|\mathbf{Z}_{1:k}\right)}{\int p\left(\mathbf{Z}_{k+1}|\vartheta_{k+1}\right) p\left(\vartheta_{k+1}|\mathbf{Z}_{1:k}\right) \mathrm{d}\vartheta_{k+1}}$$

is the Bayes update.

If a so-called factorized model is used, in where the extent state X_k is modelled such that it is dependent on the kinematic state $\mathbf{x}_k$ then the random matrix transition density is expressed as[4]:

$$p(\vartheta_{k+1}|\vartheta_k) = p(\mathbf{x}_{k+1}|\mathbf{x}_k) p(X_{k+1}|\mathbf{x}_k, X_k) \tag{2.24}$$

[4]Note that this is an assumption made in [43] and is not the true predicted density.

Using this transition density the equations for smoothing derived by Granström and Bramstång [43] are presented as follows:

$$v_{k|K} = v_{k|k} + \eta_2^{-1}\left(g - \frac{2(d+1)^2}{h+d+1}\right), \quad V_{k|K} = V_{k|k} + \eta_3^{-1} C_4 \tag{2.25a}$$

$$W = V_{k+1|K} - V_{k+1|k} \tag{2.25b}$$

$$w = v_{k+1|K} - v_{k+1|k}, \quad g = \eta_1^{-1}\left(w - \frac{2(d+1)^2}{n}\right) \tag{2.25c}$$

$$h = \frac{d+1}{d}\operatorname{Tr}\left\{C_3 C_4 \left(C_3 C_4 - \mathbf{I}_d\right)^{-1}\right\} \tag{2.25d}$$

$$\eta_1 = 1 + \frac{w - 3(d+1)}{n} \tag{2.25e}$$

$$\eta_2 = 1 + \frac{g - 3d + 1}{h+d+1} \quad \eta_3 = 1 + \frac{g-d-1}{h-d-1} \tag{2.25f}$$

$$C_3 = \mathbb{E}_{k|K}\left[\left(M_{\mathbf{x}}^{-1} W \left(M_{\mathbf{x}}^{-1}\right)^{\mathsf{T}}\right)^{-1}\right] = \mathbb{E}_{k|K}\left[M_{\mathbf{x}}^{\mathsf{T}} W^{-1} M_{\mathbf{x}}\right] \tag{2.25g}$$

$$C_4 = \mathbb{E}_{k|K}\left[M_{\mathbf{x}}^{-1} W \left(M_{\mathbf{x}}^{-1}\right)^{\mathsf{T}}\right] = \mathbb{E}_{k|K}\left[\left(M_{\mathbf{x}}^{\mathsf{T}} W^{-1} M_{\mathbf{x}}\right)^{-1}\right] \tag{2.25h}$$

Where

- n is a parameter that governs the uncertainty of the time evolution of the extent.
- $M_{\mathbf{x}}$ is a rotation matrix dependent on the kinematic state $\mathbf{x}$.
- d is the dimension of the extent matrix.

In order to compute the smoothed parameters $v_{k|K}$, $V_{k|K}$ in the $\mathcal{IW}_d$ distribution the expected value of $M_{\mathbf{x}} V M_{\mathbf{x}}^{\mathsf{T}}$ is needed. In [43] this value is approximated by using the third-order Taylor expansion which is defined as:

$$C_3 \approx \left(M_{\mathrm{x}}^{-1} V (M_{\mathrm{x}}^{-1})^{\mathsf{T}}\right)^{-1} + \sum_{i=1}^{n_x}\sum_{j=1}^{n_x} \frac{\mathrm{d}^2\left(M_{\mathrm{x}}^{-1} V (M_{\mathrm{x}}^{-1})^{\mathsf{T}}\right)^{-1}}{\mathrm{d}\mathbf{x}^{[i]}\mathrm{d}\mathbf{x}^{[j]}}\bigg|_{\mathbf{x}=m} P^{[i,j]} \tag{2.26}$$

Where $\mathbf{x}^{[i]}$ is the i^{th} element of $\mathbf{x}$ and $P^{[i,j]}$ is the i^{th}, j^{th} element of P. C_4 is approximated in the same manner.

2.7 Performance evaluation in object tracking

In order to evaluate performance of MOT algorithms metrics are needed that give some sort of score when comparing the estimated positions and extents of objects to the ground-truth data. In this work these metrics will be used to evaluate the performance of the tracking algorithms.

2.7.1 Trajectories

To evaluate performance of filtering algorithms that track finite sets of trajectories a mathematically sound metric is needed. This metric should consider things such as localization error, properly detected, missed and false targets and track switches and give a score based on these. Some different possible scenarios are depicted in Figure 2.2.

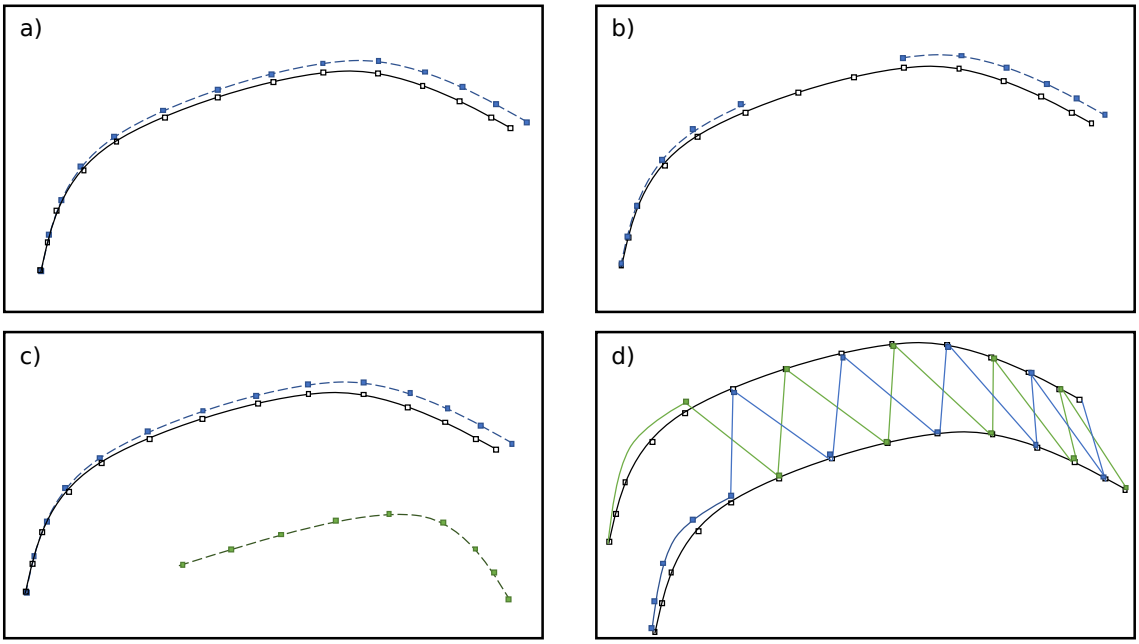


Figure 2.2: Colored trajectories denote estimates and black trajectories denote ground-truths. The different scenarios depict: a) Localization error b) Two trajectory estimates with two missed point estimates c) One correct estimate and one false track d) Track switching between two ground-truth trajectories and two estimated trajectories.

In point-object tracking some common metrics are the optimal sub-pattern assignment metric (OSPA) [44] and generalized optimal sub-pattern assignment metric (GOSPA) [45] that solves an assignment problem in where the set of estimated states $\mathcal{Y}$ at are assigned and compared to ground-truth states $\mathcal{X}$. GOSPA is defined as:

$$d_p^{(c,\alpha)}(\mathcal{X},\mathcal{Y}) \triangleq \left(\min_{\pi\in\Pi_{|\mathcal{Y}|}} \sum_{i=1}^{|\mathcal{X}|} d^{(c)}\left(x_i, y_{\pi(i)}\right)^p + \frac{c^p}{\alpha}(|\mathcal{Y}| - |\mathcal{X}|) \right)^{\frac{1}{p}} \tag{2.27}$$

Where

- $d(x,y)$ denotes a distance metric
- Π_n denotes the set of all permutations of $\{1,\ldots,n\}$
- $1 \leq p < \infty$ determines how much the metric should penalize outliers
- $c \geq 0$ is a cutoff-distance parameter
- $0 < \alpha \leq 2$ determines the error due to cardinality missmatch

In the GOSPA metric the standard choice is to set $\alpha = 2$.

The GOSPA metric was extended to handle sets of trajectories by Rahmathullah et al. in [19]. The metric for two sets of trajectories $\mathbf{X}$ and $\mathbf{Y}$ is defined as:

$$d_p^{(c,\gamma)}(\mathbf{X},\mathbf{Y}) \triangleq \min_{\substack{\pi^k\in\Pi\mathbf{x},\mathbf{Y} \\ k=1,\ldots,T}} \left(\sum_{k=1}^{T} d_k^{\mathbf{X},\mathbf{Y}}\left(\mathbf{X},\mathbf{Y},\pi^k\right)^p + \sum_{k=1}^{T-1} s_{\mathbf{X},\mathbf{Y}}\left(\pi^k,\pi^{k+1}\right)^p \right)^{\frac{1}{p}} \tag{2.28}$$

Where the p-th power cost for detected, missed and false targets at time step k is

$$d_k^{\mathbf{X},\mathbf{Y}}(\mathbf{X},\mathbf{Y},\pi_k)^p = \sum_{(i,j)\in\theta_k} d\left(\mathbf{x}_k^{(i)}, \mathbf{y}_k^{(j)}\right)^p + \frac{c^p}{2}\left(|\tau_k(\mathbf{X})| + |\tau_k(\mathbf{Y})| - 2\,|\theta_k|\right) \tag{2.29}$$

with

$$\theta_k = \left\{ \left(i, \pi_k^{(i)}\right) : \left|\mathbf{x}_k^{(i)}\right| = \left|\mathbf{y}_k^{(\pi_k^i)}\right| = 1, d\left(\mathbf{x}_k^{(i)}\mathbf{y}_k^{(\pi_k^i)}\right) < c \right\} \tag{2.30}$$

θ_k refers to the assignments between trajectories $\mathcal{X}^{(i)} \in \mathbf{X}$, $\mathcal{Y}^{(j)} \in \mathbf{Y}$ and $\mathbf{x}_k^{(i)} = \tau_k\left(\mathcal{X}^{(i)}\right)$, $\mathbf{y}_k^{(j)} = \tau_k\left(\mathcal{Y}^{(j)}\right)$. Thus θ_k is the a set of pairs of indexes that map the trajectories in $\mathbf{X}$ to $\mathbf{Y}$. To find these mappings it is common to use assignments algorithms like the Hungarian or Murtys algorithm. Missed and false targets are accounted for by the terms $(|\tau_k(\mathbf{X})| - |\theta_k|)$ and $(|\tau_k(\mathbf{Y})| - |\theta_k|)$ respectively. Also note that $\pi_k^{(i)} = 0$ implies that trajectory i in $\mathbf{X}$ is unassigned at time k.

The second term of the trajectory metric refers to a switching cost. This cost from time step k to $k+1$ is defined as:

$$s\left(\pi_k^{(i)}, \pi_{k+1}^{(i)}\right) = \begin{cases} 0 & \pi_k^{(i)} = \pi_{k+1}^{(i)} & \text{(No switch)} \\ 1 & \pi_k^{(i)} \neq \pi_{k+1}^{(i)},\ \pi_k^{(i)} \neq 0,\ \pi_{k+1}^{(i)} \neq 0 & \text{(Full switch)} \\ \frac{1}{2} & \text{otherwise} & \text{(Half switch)} \end{cases} \tag{2.31}$$

Where the parameter γ is the switching penalty factor. The switching cost is divided into three cases. No switch, half switch and full switch. If a trajectory $\mathcal{Y}^{(j)}$ is assigned to a ground truth $\mathcal{X}^{(i)}$ at two consecutive time steps k and $k+1$ then no switch has occurred. A half switch occurs when one ground truth trajectory is split into two or more estimated trajectories similar to case b) in Figure 2.2. A full switch occurs when two trajectories switch estimates between each other as in case d) in Figure 2.2. It is arguably more severe when this happens which is why it is penalized more heavily in the metric.

2.7.2 Extended objects

In extended object tracking both the position and extent of the objects should be evaluated. In this work the Gaussian Wasserstein [46] distance proposed by Yang *et al.* [20] is used alongside a discretized OSPA metric for extended targets proposed in the same paper. The p^{th} Wasserstein distance for between two general distributions is defined as:

$$W_p(g, f) := \left(\inf_{h \in H(g,f)} \int d(x,y)^p \cdot h(x,y) dx dy \right)^{1/p} \tag{2.32}$$

Where $d(\cdot,\cdot)$ is a metric on $\mathbb{R}^d$ and $H(g,f)$ denotes the set of all joint densities $h : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with marginals g and f defined as:

$$\int h(x,y)dy = g(x) \tag{2.33}$$

$$\int h(x,y)dx = f(y),\ x, y \in \mathbb{R}^d \tag{2.34}$$

Intuitively, the Wasserstein distance measures the amount of "mass" that is needed to transform one density into another. Generally there are no closed-form solutions for the Wasserstein distance [20, 44]. In the case where both distributions are Gaussian Equation (2.32) does have a closed-form solution that is given as:

$$d_{\mathrm{GW}}\left(\mathcal{N}_{\boldsymbol{x}}, \mathcal{N}_{\hat{\boldsymbol{x}}}\right)^2 = \|m_{\boldsymbol{x}} - m_{\hat{\boldsymbol{x}}}\|^2 + \mathrm{Tr}\left(\Sigma_{\boldsymbol{x}} + \Sigma_{\hat{\boldsymbol{x}}} - 2\sqrt{\sqrt{\Sigma_{\boldsymbol{x}}}\Sigma_{\hat{\boldsymbol{x}}}\sqrt{\Sigma_{\boldsymbol{x}}}}\right) \tag{2.35}$$

Where $\mathrm{Tr}(\cdot)$ denotes the trace operator, $m_{\boldsymbol{x}}$ and $m_{\hat{\boldsymbol{x}}}$ denotes the means of the two Gaussians and $\Sigma_{\boldsymbol{x}}$, $\Sigma_{\hat{\boldsymbol{x}}}$ denotes the covariances. $\sqrt{\Sigma}$ is the matrix Γ such that $\Sigma = \Gamma\Gamma$.

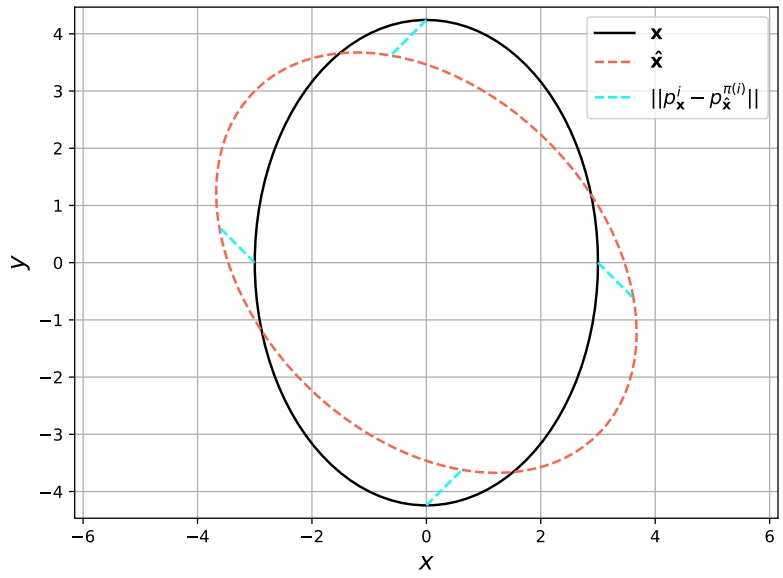
Since elliptical object extent estimates can be seen as a normal distribution with mean m and covariance Σ where m represents the objects center and Σ represents the extent the Wasserstein distance seems like a good choice in evaluating extent estimates when using the random matrix model.

An issue when using this metric for evaluating the distance is that the metric will be more biased towards the center of the ellipse since that is where most of the mass of the Gaussian distribution is centered. Therefore Yang *et al.* [20] also propose a sampling based performance matrix that is based on generating a uniform number of samples on the boundary of the ellipses and calculating the Wasserstein/OSPA distance as:

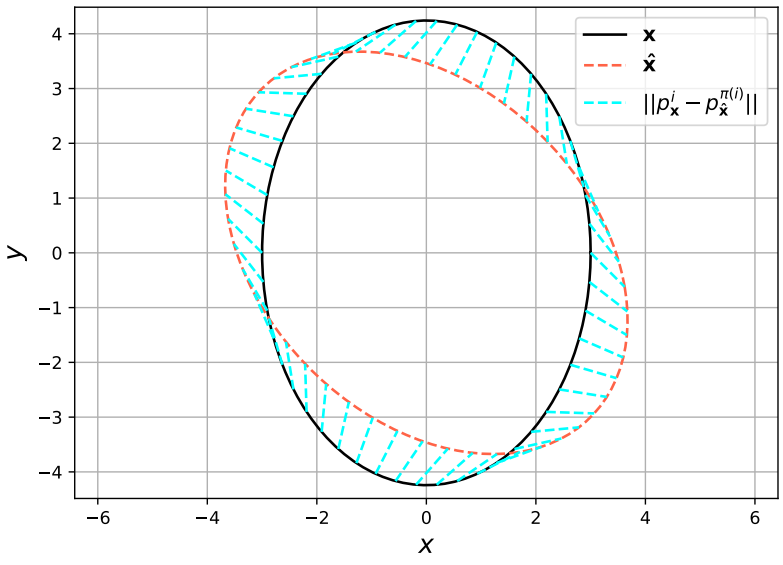
$$d_{\mathrm{OSPA}_n}(p_x, p_{\hat{x}}) = \min_{\pi \in \Pi} \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left\| p_x^{(i)} - p_{\hat{x}}^{(\pi(i))} \right\|^2} \tag{2.36}$$

where $p_x = \{p_x^1, .., p_x^n\}$, $p_{\hat{x}} = \{p_{\hat{x}}^1, .., p_{\hat{x}}^n\}$ are the uniformly distributed points on the ellipse boundaries and Π is the set of all permutations of $\{1, .., n\}$. Depictions of the sum of distances calculated by this method is shown in Figures 2.3a and 2.3b.

This method is much more computationally complex than the closed-form solution for the Gaussian Wasserstein distance given in Equation (2.35) since the number of permutations of a set of n elements is $n!$, thus making it computationally intractable for larger n. An assignment algorithm like the Hungarian [30] or Auctioneer [47] algorithm can however be used to compute π and make the problem more tractable for larger n.



(a) d_{OSPA_4}



(b) $d_{OSPA_{50}}$

Figure 2.3: Visual representations of d_{OSPA_4} and $d_{OSPA_{50}}$ depicted for the same two ellipses $\mathbf{x}$ and $\hat{\mathbf{x}}$.

3

Methods

In this chapter methods used to implement and test the different filters are presented. Details on filter implementations are given here as well as some details regarding the metric used to evaluate the filters.

In order to compare the filters, ground truth data is needed in order to evaluate the performance. Since the focus of this thesis is to implement and evaluate a GGIWTPHD filter, the data used does not completely conform to reality but will still highlight the differences in performance between the different filters.

3.1 Models

In this section the different models used in the filters are presented and explained. Firstly, the motion and measurement model used in the prediction and update step of the filters are defined. These models are also used to generate data which the filters are tested on, see Section 4.1. Secondly, some models for object birth are explained. These models are used to describe where it is likely that new objects appear in the sensors field of view.

3.1.1 Motion model

To describe how the extended object state evolves over time, i.e. the transition from ξ_k to ξ_{k+1}, a motion model is used.

Two assumptions are made regarding the dynamics in the surveillance area. Firstly, it is assumed that all objects follow the same motion model and secondly it is assumed that all objects move independently of each other.

Kinematic state
The kinematic state is assumed to evolve according to a coordinated turn (CT) motion model. The random vector $\mathbf{x}_k = [\mathbf{p}_k, v_k, \phi_k, \omega_k]^{\mathrm{T}} \in \mathbb{R}^5$ describes the kinematic state where, $\mathbf{p}_k \in \mathbb{R}^2$ is the position, v_k is the polar velocity, θ_k is the heading and ω_k is the turn rate.

The motion model is defined as

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k\right) + \mathbf{q}_k, \quad \mathbf{q}_k \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}\right) \tag{3.1}$$

where the state prediction $\mathbf{f}(\mathbf{x}_k)$ and process noise covariance $\mathbf{Q}$ are

$$\mathbf{f}\left(\mathbf{x}_k\right) = \mathbf{x}_k + \begin{bmatrix} T_s v_k \cos\phi_k \\ T_s v_k \sin\phi_k \\ 0 \\ T_s \omega_k \\ 0 \end{bmatrix} \tag{3.2a}$$

$$\mathbf{Q} = T_s \mathbf{G} \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} \mathbf{G}^\top, \quad \mathbf{G} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.2b}$$

Here, T_s is the sampling time, σ_v is the standard deviation for the velocity and σ_ω is the standard deviation for the turn rate.

The Jacobian of $\mathbf{f}$ with respect to $\mathbf{x}$ is:

$$F(\mathbf{x}_k) = \mathbf{J}_\mathbf{f}(\mathbf{x}_k) = \begin{bmatrix} 1 & 0 & T_s \cos\phi_k & -T_s v_k \sin\phi_k & 0 \\ 0 & 1 & T_s \sin\phi_k & T_s v_k \cos\phi_k & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Extension state
The random matrix transition density $p(X_{k+1}|\mathbf{x}_k, X_k)$ is conditioned on the kinematic state $\mathbf{x}_k$, which can be utilized when using coordinated turn models to predict rotations of the extension state. The predicted rotation of the extent state can thus be expressed as:

$$X_{k+1} = M\left(\mathbf{x}_k\right) X_k M\left(\mathbf{x}_k\right)^\top \tag{3.4a}$$

$$M\left(\mathbf{x}_k\right) = \begin{bmatrix} \cos\left(\omega_k T_s\right) & -\sin\left(\omega_k T_s\right) \\ \sin\left(\omega_k T_s\right) & \cos\left(\omega_k T_s\right) \end{bmatrix} \tag{3.4b}$$

Using this model, the shape is rotated $\omega_k T_s$ radians.

The transition density with this model is Wishart

$$p\left(X_{k+1}|\mathbf{x}_k, X_k\right) = \mathcal{W}\left(X_{k+1}; n, n^{-1} M\left(\mathbf{x}_k\right) X_k M\left(\mathbf{x}_k\right)^\top\right) \tag{3.5}$$

n is a scalar that represents the uncertainty of the transition where a higher value on n means higher certainty [42].

Measurement rate
It is assumed that the number of measurements generated from an object stays relatively constant, therefore the evolution of the measurement rate is:

$$\gamma_{k+1} = \gamma_k \tag{3.6}$$

Here the expected number of measurements is kept constant and the variance is increased by a multiplying with a factor $\eta_k > 1$ where the closer this factor is to 1, the more certain the time evolution. This can be seen in the prediction step of the GGIWPHD filter in Table 3.1.

As mentioned earlier, this assumption does not conform with reality since the size and position of the object should affect the number of measurements. But since this assumption simplifies the implementation a lot and it has been shown to not have that big of an impact on performance [35] in many applications, this way of modeling the evolution of the measurement rate is sufficient.

3.1.2 Measurement models

The measurements are assumed to be range-bearing measurements and a conversion to Cartesian coordinates is performed. This conversion is applied as follows:

$$\mathbf{z}_k^{(j)} = \left[r_k^{(j)} \cos\left(\varphi_k^{(j)}\right), \quad r_k^{(j)} \sin\left(\varphi_k^{(j)}\right) \right]^\top \tag{3.7}$$

where $r_k^{(j)}$ and $\varphi_k^{(j)}$ are the range and bearing measurements corresponding to a position $\mathbf{p}_k^{(j)}$. This means that a linear measurement model can be used in the update step of the filters.

The number of measurements generated from each object is modeled as Poisson distributed with rate γ_k [11].

$$P\left(N_{z,k}|\xi_k\right) = P\left(N_{z,k}|\gamma_k\right) = \mathcal{PS}\left(N_{z,k};\gamma_k\right) \tag{3.8}$$

The object oriented measurements are then modeled as being uniformly distributed over the surface of the object, and it is defined by the likelihood [11]:

$$p\left(\mathbf{z}_k^{(j)}|\xi_k\right) = p\left(\mathbf{z}_k^{(j)}|\mathbf{x}_k, X_k\right) = \mathcal{N}\left(\mathbf{z}_k^{(j)}; H_k\mathbf{x}_k, \rho X_k + R\left(\mathbf{p}_k\right)\right) \tag{3.9}$$

The position in the state vector is directly mapped to measurements with $H_k = \begin{bmatrix}\mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 3}\end{bmatrix}$ and ρ is a scaling parameter. $R(\mathbf{p})$ is given by a first order Taylor approximation of the polar to Cartesian conversion of the measurement noises and it models the non-constant across-range noise variance.

$$R(\mathbf{p}) = \mathbf{J}(\mathbf{p}) \operatorname{diag}\left(\left[\sigma_r^2, \sigma_\varphi^2\right]\right) \mathbf{J}(\mathbf{p})^{\mathbf{T}} \tag{3.10a}$$

$$\mathbf{J}(\mathbf{p}) = \begin{bmatrix} \cos(\varphi) & -r\sin(\varphi) \\ \sin(\varphi) & r\cos(\varphi) \end{bmatrix} \tag{3.10b}$$

3.1.3 Birth models

A birth model describes where in the surveillance area new objects are likely to appear. Depending on the situation, the knowledge of where new objects should appear will be different. Since data is generated manually in this thesis, full knowledge of the positions where objects are born as well as their initial velocity, heading and turn rate exist. This is however most often not the case in reality, therefore a few different approaches to the modeling of object birth have been tested.

Three different birth models have been tested for the following cases, where the models include

- Prior knowledge on full initial state.
- Prior knowledge on only positions in the initial state.
- No prior knowledge except size of surveillance area.

When no prior knowledge of the birth positions exist, the birth model is represented by a uniform distribution over the surveillance area. This could be considered the most realistic approach, since it is often not known exactly where objects appear.

From these three alternatives, the results presented in Chapter 4 will use a birth model where only the birth positions are known. This is a reasonable assumption since prior knowledge of an appearing objects velocity is often hard to obtain. The same birth model is used in all of the filters in this thesis to ensure that the comparison between them is fair.

3.2 GGIWPHD filter

The GGIWPHD filter is implemented as in [11] except for the prediction step which is more similiar to [43].

In the GGIWPHD filter the PHD intensity is approximated by a mixture of GGIW components as

$$D_{k|k}\left(\xi_k\right) = \sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)} \mathcal{GGIW}\left(\xi_k; \zeta_{k|k}^{(j)}\right) \tag{3.11}$$

where

- $J_{k|k}$ is the number of components
- $w_{k|k}^{(j)}$ is the weight of the j:th component
- $\zeta_{k|k}^{(j)}$ is the density parameter of the j:th component

The predicted PHD consists of two parts, one for new objects that appear in the surveillance area (birth) and one for objects that stay in the surveillance area from preivous timesteps (survive). This PHD is represented by a GGIW mixture as in

Table 3.1: Equations for the GGIWPHD filter prediction step

$$
\begin{aligned}
w_{k+1|k}^{(j)} &= P_S w_{k|k}^{(j)} \\
\alpha_{k+1|k}^{(j)} &= \frac{\alpha_{k|k}^{(j)}}{\eta_k}, \quad \beta_{k+1|k}^{(j)} = \frac{\beta_{k|k}^{(j)}}{\eta_k} \\
m_{k+1|k}^{(j)} &= \mathbf{f}_k(m_{k|k}^{(j)}), \quad P_{k+1|k}^{(j)} = F_k P_{k|k}^{(j)} F_k^{\mathsf{T}} + \mathbf{Q} \\
v_{k+1|k}^{(j)} &= d + 1 + \psi^{-1}(v_{k|k}^{(j)} - d - 1), \quad V_{k+1|k}^{(j)} = \psi^{-1} \left(1 - \frac{d+1}{s}\right) \left(1 - \frac{d+1}{s}\right) C_2 \\
\psi &= 1 + (v_{k+1|k}^{(j)} - 2d - 2) \left(\frac{1}{s} + \frac{1}{n} + \frac{d+1}{ns}\right) \\
s &= \frac{d+1}{d} \mathrm{Tr}\Big\{C_1 C_2 (C_1 C_2 - \mathbf{I}_d)^{-1}\Big\} \\
F_k &= \mathbf{J}_{\mathbf{f}_k}(\mathbf{x})\big|_{\mathbf{x}=m_{k|k}^{(j)}} \\
C_1 &= \mathbb{E}_{k|k}\left[\left(M_x V_{k|k}^{(j)} M_x^{\mathsf{T}}\right)^{-1}\right], \quad C_2 = \mathbb{E}_{k|k}\left[M_x V_{k|k}^{(j)} M_x^{\mathsf{T}}\right]
\end{aligned}
$$

Equation 3.11 and is defined as

$$D_{k+1|k}(\xi_{k+1}) = D_{k+1}^{b}(\xi_{k+1}) + D_{k+1|k}^{s}(\xi_{k+1}) \tag{3.12}$$

where $D_k^b(\xi_k)$ is the birth intensity and $D_{k+1|k}^s(\xi_{k+1})$ is the intensity for surviving existing targets which is defined as

$$D_{k+1|k}^{s}(\xi_{k+1}) = \sum_{j=1}^{J_{k|k}} w_{k+1|k}^{(j)} \mathcal{GGIW}\left(\xi_{k+1}; \zeta_{k+1|k}^{(j)}\right) \tag{3.13}$$

Objects that stay in surveillance area are modeled by the probability of survival P_S, which yields the weight update

$$w_{k+1|k}^{(j)} = P_S w_{k|k}^{(j)} \tag{3.14}$$

The density parameters $\zeta_{k+1|k}^{(j)}$ are updated according to the motion model (see Section 3.1.1 for further details). Table 3.1 shows details on the prediction step in the GGIWPHD filter.

It is noteworthy that the predicted PHD does not include so called spawned targets, which denotes new targets that appear close to already existing targets, e.g. a missile fired from an aircraft. This can easily be implemented but will not be considered in this thesis.

In order to compute the parameters v, V in the $\mathcal{IW}_d$ distribution the expected value of $M_{\mathbf{x}} V M_{\mathbf{x}}^{\mathsf{T}}$ is needed. In [43] this value is approximated by using the third-order

Taylor expansion as:

$$C_1 \approx \left(M_{\mathrm{x}} V M_{\mathrm{x}}^{\mathsf{T}}\right)^{-1} + \sum_{i=1}^{n_x}\sum_{j=1}^{n_x} \left.\frac{\mathrm{d}^2\left(M_{\mathrm{x}} V M_{\mathrm{x}}^{\mathsf{T}}\right)^{-1}}{\mathrm{d}\mathbf{x}^{[i]}\mathrm{d}\mathbf{x}^{[j]}}\right|_{\mathbf{x}=m} P^{[i,j]} \tag{3.15}$$

Where $\mathbf{x}^{[i]}$ is the i^{th} element of $\mathbf{x}$ and $P^{[i,j]}$ is the i^{th}, j^{th} element of P. C_2 is approximated in the same manner.

With a predicted PHD intensity, the next step is a measurement update and calculation of a posterior PHD which is defined as

$$D_{k|k}(\xi_k) = D^m_{k|k}(\xi_k) + D^b_{k|k}(\xi_k) + D^d_{k|k}(\xi_k) \tag{3.16}$$

This consists of three parts, $D^m_{k|k}$ which represents not detected previously existing objects, $D^b_{k|k}$ which represents new targets appearing at measurement cluster centers and finally $D^d_{k|k}$ which represents detected previously existing targets. $D^d_{k|k}$ is updated according to the measurement model (see Section 3.1.1). In the following part of this section, the three parts of the posterior PHD will be explained further.

The PHD for not detected previously existing objects is an approximate intensity resulting from gamma-mixture reduction [11] and is defined as

$$D^m_{k|k}(\xi_k) \approx \sum_{j=1}^{J_{k|k-1}} \tilde{w}^{(j)}_{k|k} \mathcal{GGIW}\left(\xi_k; \tilde{\zeta}^{(j)}_{k|k}\right) \tag{3.17}$$

where $\tilde{\zeta}^{(j)}_{k|k} = \left\{\alpha^{(j)}_{k|k}, \tilde{\beta}^{(j)}_{k|k}, m^{(j)}_{k|k-1}, P^{(j)}_{k|k-1}, v^{(j)}_{k|k-1}, V^{(j)}_{k|k-1}\right\}$ and $\tilde{\beta}^{(j)}_{k|k} = \beta^{(j)}_{k|k} + 1$. The weight update in this mixture is approximated as

$$\tilde{w}^{(j)}_{k|k} \approx (1 - P_D) + P_D \left(\frac{\beta^{(j)}_{k|k}}{\beta^{(j)}_{k|k} + 1}\right)^{\alpha^{(j)}_{k|k}} w^{(j)}_{k|k-1} \tag{3.18}$$

and the rest of the mixture components stay the same, except β which is increased by one.

The measurement update of the PHD intensity using the measurement set $\mathbf{Z}_k$ should consider all possible partitions $\mathcal{P}$ according to Bayes optimality, However, this is not computationally efficient and is therefore not done in this implementation. More details regarding clustering is provided in Section 3.5.2.

Each partition $\mathcal{P}$ contains a number of cells $\mathbf{W}$, and for each cell a new object is birthed at the center of a cell (i.e. a measurement cluster). This is represented by the following PHD

$$D^b_{k|k}(\xi_k) = \sum_{\mathcal{P} \angle \mathbf{Z}_k} \sum_{\mathbf{W} \in \mathcal{P}} w^{(b,\mathbf{W})}_{k|k} \mathcal{GGIW}\left(\xi_k; \zeta^{(b,\mathbf{W})}_{k|k}\right) \tag{3.19}$$

where the density parameters $\zeta^{(b,\mathbf{W})}_{k|k}$ are updated using predefined birth parameters $\zeta^{(b)}_k = \left\{\alpha^{(b)}_k, \tilde{\beta}^{(b)}_k, m^{(b)}_k, P^{(b)}_k, v^{(b)}_k, V^{(b)}_k\right\}$.

The last part of the measurement update is the detected previously existing objects, which is represented by the following PHD

$$D^d_{k|k}(\xi_k) = \sum_{\mathcal{P} \angle \mathbf{Z}_k} \sum_{\mathbf{W} \in \mathcal{P}} \sum_{j=1}^{J_{k|k-1}} w^{(j,\mathbf{W})}_{k|k} \mathcal{GGIW}\left(\xi_k; \zeta^{(j,\mathbf{W})}_{k|k}\right) \tag{3.20}$$

This is updated according to measurement model (see Section 3.1.2). A deeper explanation of the measurement update is given in [11] and details on the implementation can be seen in Table 3.2.

Equation 3.12 and 3.16 shows the prediction and update causes the number of hypotheses to grow exponentially. Therefore methods to reduce the number of hypothesis are used. In the GGIWPHD filter pruning, merging and capping is used to for hypothesis reduction, more on this in Section 3.4.

The last step of the filter recursion is extraction, or estimation, where extended object states are estimated from the posterior PHD. This is done by looking for the most likely object states, which are given by peaks in the PHD intensity. When using a distribution mixture to approximate the intensity, these peaks correspond to mixture components with large weights. Thus, in order to estimate a set of likely extended object states, a set of GGIW components is extracted for which $w^{(j)}_{k|k} > \bar{w}_e$ where $\bar{w}_e$ is a threshold. In short, this step provides

$$\hat{\mathbf{X}}_{k|k} = \left\{\hat{\xi}^{(i)}_{k|k}\right\}_{i:1}^{\hat{N}_{k|k}}, \quad \hat{\xi}^{(i)}_{k|k} = \left(\mathbb{E}\left[\gamma_k\right], \mathbb{E}\left[\mathbf{x}_k\right], \mathbb{E}\left[X_k\right]\right) \tag{3.21}$$

From these estimates, it is not clear which estimate correspond to which objects over time. In order to know this, additional information needs to be stored in each mixture component. Therefore a labeling scheme is used as in [11]. it is explained briefly as *"[...] we assign to each GGIW component a label $\ell^{(j)}_{k|k}$, whose initial value is 0. If the component is extracted and a label has not yet been assigned (i.e., $\ell^{(j)}_{k|k} = 0$), a unique positive integer is assigned to the label. The assigned labels are kept in the time update and measurement update steps."* With labeled components, trajectories are built by stacking extracted components with the same label from previous timesteps.

Table 3.2: Equations for the GGIWPHD filter update step

New objects (for intensity in Equation 3.19):

$$\alpha_{k|k}^{(b,\mathbf{W})} = \alpha_k^{(b)} + |\mathbf{W}|, \quad \beta_{k|k}^{(b,\mathbf{W})} = \beta_k^{(b)} + 1$$
$$m_{k|k}^{(b,\mathbf{W})} = \left[\left(\bar{z}_k^{\mathbf{W}}\right)^{\mathsf{T}}, \left(m_k^{(b)}\right)^{\mathsf{T}}\right]^{\mathsf{T}}, \quad P_{k|k}^{(b,\mathbf{W})} = \text{blkdiag}\left(Z_k^{\mathbf{W}}/|\mathbf{W}|, P_k^{(b)}\right)$$
$$v_{k|k}^{(b,\mathbf{W})} = v_{k|k-1}^{(b)} + |\mathbf{W}| - 1, \quad V_{k|k}^{(b,\mathbf{W})} = V_{k|k-1}^{(b)} + Z_k^{\mathbf{W}}$$

Detected previously existing objects (for intensity in Equation 3.20):

$$\alpha_{k|k}^{(j,\mathbf{W})} = \alpha_{k|k-1}^{(j)} + |\mathbf{W}|, \quad \beta_{k|k}^{(j,\mathbf{W})} = \beta_{k|k-1}^{(j)} + 1$$
$$m_{k|k}^{(j,\mathbf{W})} = m_{k|k-1}^{(j)} + K_{k|k-1}^{(j,\mathbf{W})}\varepsilon_{k|k-1}^{(j,\mathbf{W})}, \quad P_{k|k}^{(j,\mathbf{W})} = P_{k|k-1}^{(j)} - K_{k|k-1}^{(j,\mathbf{W})}H_k P_{k|k-1}^{(j)}$$
$$v_{k|k}^{(j,\mathbf{W})} = v_{k|k-1}^{(j)} + |\mathbf{W}|, \quad V_{k|k}^{(j,\mathbf{W})} = V_{k|k-1}^{(j)} + \hat{N}_{k|k-1}^{(j,\mathbf{W})} + \hat{Z}_{k|k-1}^{(j,\mathbf{W})}$$
$$\hat{X}_{k|k-1}^{(j)} = V_{k|k-1}^{(j)}\left(v_{k|k-1}^{(j)} - 2d - 2\right)^{-1}$$
$$\hat{R}_{k|k-1}^{(j,\mathbf{W})} = \rho\hat{X}_{k|k-1}^{(j)} + R\left(H_k m_{k|k-1}^{(j)}\right)$$
$$N_{k|k-1}^{(j,\mathbf{W})} = \varepsilon_{k|k-1}^{(j,\mathbf{W})}\left(\varepsilon_{k|k-1}^{(j,\mathbf{W})}\right)^{\mathsf{T}}, \quad \varepsilon_{k|k-1}^{(j,\mathbf{W})} = \bar{\mathbf{z}}_k^{\mathbf{W}} - H_k m_{k|k-1}^{(j)}$$
$$S_{k|k-1}^{(j,\mathbf{W})} = H_k P_{k|k-1}^{(j)} H_k^{\mathsf{T}} + \frac{\hat{R}_{k|k-1}^{(j,\mathbf{W})}}{|\mathbf{W}|}$$
$$K_{k|k-1}^{(j,\mathbf{W})} = P_{k|k-1}^{(j)} H_k^{\mathsf{T}}\left(S_{k|k-1}^{(j,\mathbf{W})}\right)^{-1}$$
$$\hat{Z}_{k|k-1}^{(j,\mathbf{W})} = \left(\hat{X}_{k|k-1}^{(j)}\right)^{1/2}\left(\hat{R}_{k|k-1}^{(j,\mathbf{W})}\right)^{-1/2} Z_k^{\mathbf{W}}\left(\hat{R}_{k|k-1}^{(j,\mathbf{W})}\right)^{-\mathsf{T}/2}\left(\hat{X}_{k|k-1}^{(j)}\right)^{\mathsf{T}/2}$$
$$\hat{N}_{k|k-1}^{(j,\mathbf{W})} = \left(\hat{X}_{k|k-1}^{(j)}\right)^{1/2}\left(S_{k|k-1}^{(j,\mathbf{W})}\right)^{-1/2} Z_k^{\mathbf{W}}\left(S_{k|k-1}^{(j,\mathbf{W})}\right)^{-\mathsf{T}/2}\left(\hat{X}_{k|k-1}^{(j)}\right)^{\mathsf{T}/2}$$

Weights:

$$w_{k|k}^{(b,\mathbf{W})} = \frac{\omega_{\mathcal{P}}\mathcal{L}_k^{(b,\mathbf{W})}w_k^{(b)}}{d_{\mathbf{W}}\beta_{FA,k}^{|\mathbf{W}|}V(\mathcal{A})}, \quad w_{k|k}^{(j,\mathbf{W})} = \frac{\omega_{\mathcal{P}}P_D\mathcal{L}_k^{(j,\mathbf{W})}w_{k|k-1}^{(j)}}{d_{\mathbf{W}}\beta_{FA,k}^{|\mathbf{W}|}}$$
$$d_{\mathbf{W}} = \delta_{|\mathbf{W}|,1} + \frac{\mathcal{L}_k^{(b,\mathbf{W})}w_k^{(b)}}{\beta_{FA,k}^{|\mathbf{W}|}V(\mathcal{A})} + \sum_{j=1}^{J_{k|k-1}}\frac{P_D\mathcal{L}_k^{(j,\mathbf{W})}w_{k|k-1}^{(j)}}{\beta_{FA,k}^{|\mathbf{W}|}}, \quad \omega_{\mathcal{P}} = \frac{\prod_{\mathbf{W}\in\mathcal{P}}d_{\mathbf{W}}}{\sum_{\mathcal{P}'\angle\mathbf{z}_k}\prod_{\mathbf{W}'\in\mathcal{P}'}d_{\mathbf{W}'}}$$
$$\mathcal{L}_k^{(b,\mathbf{W})} = \frac{|\mathbf{W}|^{-d/2}}{\pi^{|\mathbf{W}|(d-1)/2}}\frac{\Gamma_d\left(\frac{v_{k|k}^{(b,\mathbf{W})}}{2}\right)\left|V_k^{(b)}\right|^{\frac{v_k^{(b)}-d-1}{2}}}{\Gamma_d\left(\frac{v_k^{(b)}}{2}\right)\left|V_{k|k}^{(b,\mathbf{W})}\right|^{\frac{v_{k|k}(b,\mathbf{W})-d-1}{2}}}\frac{\Gamma\left(\alpha_{k|k}^{(b,|\mathbf{W}|)}\right)\left(\beta_k^{(b)}\right)^{\alpha_k^{(b)}}}{\Gamma\left(\alpha_k^{(b)}\right)\left(\beta_{k|k}^{(b,\mathbf{W})}\right)^{\alpha_{k|k}^{(b,|\mathbf{W}|)}}}$$
$$\mathcal{L}_k^{(j,\mathbf{W})} = \frac{(2\pi)^{\frac{|\mathbf{w}|d}{2}}2^{\frac{|\mathbf{W}|}{2}}}{|\mathbf{W}|^{\frac{d}{2}}}\frac{\left|V_{k|k-1}^{(j)}\right|^{\frac{v_{k|k-1}^{(j)}}{2}}}{\left|V_{k|k}^{(j,\mathbf{W})}\right|^{\frac{v_{k|k}(j,\mathbf{W})}{2}}}\frac{\Gamma_d\left(\frac{v_{k|k}^{(j,\mathbf{W})}}{2}\right)}{\Gamma_d\left(\frac{v_{k|k-1}^{(j)}}{2}\right)}\frac{\left|\hat{X}_{k|k-1}^{(j)}\right|^{\frac{|\mathbf{W}|}{2}}}{\left|\hat{R}_{k|k-1}^{(j,\mathbf{W})}\right|^{\frac{|\mathbf{W}|-1}{2}}\left|S_{k|k-1}^{(j,\mathbf{W})}\right|^{\frac{1}{2}}}$$
$$\times\frac{\Gamma\left(\alpha_{k|k}^{(j,|\mathbf{W}|)}\right)\left(\beta_{k|k-1}^{(j)}\right)^{\alpha_{k|k-1}^{(j)}}}{\Gamma\left(\alpha_{k|k-1}^{(j)}\right)\left(\beta_{k|k}^{(j,\mathbf{W})}\right)^{\alpha_{k|k}^{(j,|\mathbf{W}|)}}}$$

3.3 GGIWTPHD filter

In order to estimate trajectories using a GGIWPHD filter without labeling, a heuristic approach that combines the filter equations given in Section 3.2 with the work done by García-Fernández and Svensson in [17] is used. This filter will henceforth be referred to as the GGIWTPHD filter.

The method used to track sets of trajectories is to redefine the kinematic state $\mathbf{x}_k$ as a trajectory:

$$p(\mathbf{x}_k) = \mathcal{N}(x^{1:i_k};\ m_k,\ P_k) \tag{3.22}$$

Where $i_k = dim(m_k)/n_x$ denotes the lifespan of the trajectory. The mean $m_k \in \mathbb{R}^{i_k n_x}$ and covariance matrix $P_k \in \mathbb{R}^{i_k n_x \times i_k n_x}$ describes the mean and the covariance over the whole trajectory instead of just the current object estimate.

Using this redefinition of the objects kinematic state, the GGIWTPHD filter prediction for the kinematic state becomes:

$$m_{k+1|k}^{(j)} = \left[(m_k^{(j)})^\mathsf{T},\quad (\dot{F}^{(j)} m_k^{(j)})^\mathsf{T}\right]^\mathsf{T} \tag{3.23a}$$

$$\dot{F}^{(j)} = \left[0_{1,i_k^{(j)}-1},\quad 1\right] \otimes F^{(j)}(\tau(m_{k-1|k-1}^{(j)})) \tag{3.23b}$$

$$P_{k+1|k}^{(j)} = \begin{bmatrix} P_k^{(j)} & P_k^{(j)}(\dot{F}^{(j)})^\mathsf{T} \\ \dot{F}^{(j)} P_k^{(j)} & \dot{F}^{(j)} P_k^{(j)} (\dot{F}^{(j)})^\mathsf{T} + Q \end{bmatrix} \tag{3.23c}$$

where $\otimes$ refers to the Kronecker product and $0_{m,n}$ is a $m \times n$ zero matrix. The equations corresponding to the extent and measurement rate predictions are not calculated with respect to the trajectories and thus stay the same as in Table 3.1.

The GGIWTPHD filter update for the kinematic state is

$$\bar{\mathbf{z}}^{(j)} = \dot{H}^{(j)} m_{k+1|k}^{(j)}, \tag{3.24a}$$

$$\hat{R}_{k|k-1}^{(j)} = \rho \hat{X}_{k|k-1}^{(j)} + R(\dot{H}^{(j)} m_{k|k-1}^{(j)}) \tag{3.24b}$$

$$S^{(j)} = \dot{H}^{(j)} P_{k+1|k}^{(j)} (\dot{H}^{(j)})^\mathsf{T} + \frac{\hat{R}_{k|k-1}^{(j)}}{|\mathbf{W}|} \tag{3.24c}$$

$$\dot{H}^{(j)} = \left[0_{1,i_k^{(j)}-1},\quad 1\right] \otimes H \tag{3.24d}$$

$$m_{k+1|k+1}^{(j)} = m_{k+1|k}^{(j)} + P_{k+1|k}^{(j)} \dot{H}^\mathsf{T} (S^{(j)})^{-1} (\mathbf{z} - \bar{\mathbf{z}}^{(j)}) \tag{3.24e}$$

$$P_{k+1|k+1}^{(j)} = P_{k+1|k}^{(j)} - P_{k+1|k}^{(j)} \dot{H}^\mathsf{T} (S^{(j)})^{-1} \dot{H} P_{k+1|k}^{(j)} \tag{3.24f}$$

This means that the difference in the update equations for the GGIWPHD and GGIWTPHD filters only differs with regard to the kinematic state.

The rest of the parameters in the GGIW mixture are stored to be used in the smoothing of the extent state.

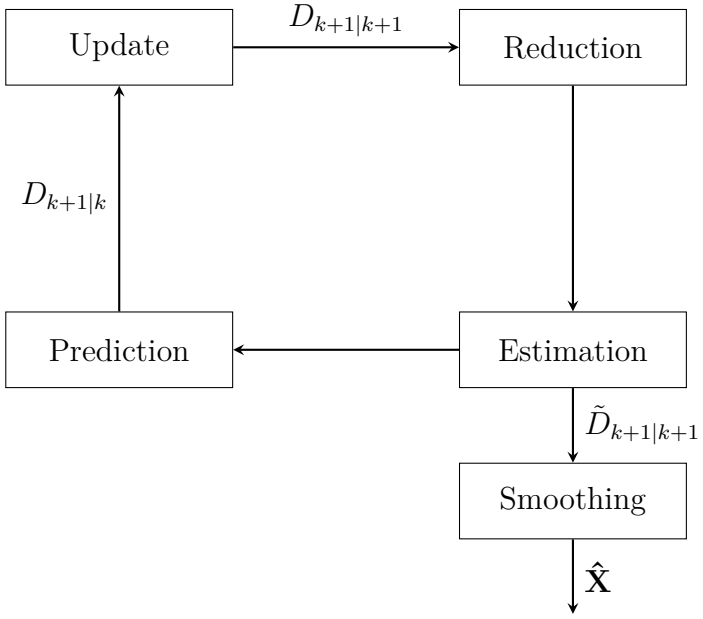


Figure 3.1: Flowchart of the GGIWTPHD filter recursion. $\tilde{D}$ denotes the reduced mixture that only contains mixture components with a weight larger than the extraction treshold $\bar{w}_e$. Smoothing is then performed on $\tilde{D}$ and the set of smoothed estimated trajectories $\hat{\mathbf{X}}$ that contain both the kinematic- and extent estimates are returned.

3.3.1 Smoothing of extent estimate

When smoothing the extent estimates the method proposed by Granström and Bramstång is used [43]. This requires that the predicted parameters of the mixture at each time step is stored as well as the updated parameters. The proposed smoothing procedure is consistent in a Bayesian setting and the smoothing equations are presented in Section 2.6.

3.4 Hypothesis reduction

As explained in Section 3.2, the number of hypothesis in a PHD filter grows very quickly over time. This has a large impact on the speed of the filter, therefore it is necessary to use methods that reduce the number of hypotheses [48]. In other terms, the number of components in the mixture that represents the PHD needs to be reduced. In order to do this, methods called pruning, merging and capping are used. In the trajectory PHD filter, absorption is used instead of merging.

The different reduction steps are as follows:

- **Pruning**
  The first stage to component reduction is often pruning. This removes the components that are least likely to give a good estimate, which is done by eliminating the components with a weight below a truncation threshold, T, as well as components where $\mathrm{E}[\gamma_k] = \alpha_{k|k}^{(j)}/\beta_{k|k}^{(j)} < 1$ since these components would on average yield less that one measurement per timestep and are therefore unlikely.

- **Merging**
  When there are many different mixture components that are very similar, it is reasonable to assume that these components are following the same target and can therefore be combined into one component with a larger weight. This means that for a collection of mixture components with a Mahalanobis distance below some threshold U between them, moment matching is performed and their weights are summed.

- **Capping**
  The last step is called capping in where, only the M components in the mixture with the largest weights are kept. This is used to put an upper bound on the number of components permitted in the mixture.

Pseudocode for these methods applied in the GGIWPHD filter can be seen in Appendix A.1. This is extended from the pseudocode presented in [49].

Absorption is a method very similar to merging, and is used instead of merging when tracking sets of trajectories. One of the main differences between merging and absorption is that in merging, moment matching is done, while absorption does not use moment matching. This is also why absorption is preferred when handling trajectory densities, since two trajectories can be similar in the current timestep but different in the past which would lead to incorrect past estimates during a merge [17]. Pseudocode for the reduction of mixtures containing sets of trajectories is presented in Appendix A.2.

3.5 Implementation notes

This section contains some specifics regarding the implementation of the filters, such as the chosen programming language and how clustering is performed.

3.5.1 Programming language

All code in this thesis is written in Python3 [50]. This is made possible by using packages such as NumPy [51] and SciPy [52].

3.5.2 Clustering method

The chosen approaches for filtering requires that clustering is performed to compute partitions of the measurement set $\mathbf{Z}_k$. Since the clustering algorithm only has access to the measurements at time k it is often difficult to know if a correct partitioning has been computed or not. An example of this can be seen in Figure 4.1. A common approach is to calculate many different partitions of the measurement set and weigh these partitions in order to find the most likely one. This is reflected in the GGIWPHD filtering recursions given in Table 3.2 where $\omega_{\mathcal{P}}$ is the weight of the partition $\mathcal{P}$.

In this thesis clustering is performed using the DBSCAN algorithm from scikit-learn [53]. DBSCAN performs density-based spatial clustering of samples with noise. It operates by finding samples of high density and expands clusters from them.

Clustering can have a major impact on the performance of a filtering algorithm since spatially close objects may easily be clustered together as a single object which will lead to missed object detections. If the clutter intensity is high it is also plausible that many false object tracks are initiated which is often not desirable.

4

Evaluation

In this chapter the methods for evaluating the different filters are presented together with results showcasing the performance of the filters. The generation of ground truth states and measurement sets is covered in Section 4.1 and the metric used is presented in Section 4.2. The parameters used in the filters are presented in Section 4.3.

Four different scenarios are constructed to evaluate different aspects of the filters. The construction of these scenarios together with Figures showing example measurements are shown in separate Sections together with the results from running the filters for each scenario.

4.1 Generating data

Using the motion model presented in Section 3.1, ground truth extended object states for a predefined number of time steps. Using these extended object states, object oriented measurements can be generated with a measurement model. To do this, a range-bearing measurement model is used. It is defined as:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{r}_k, \quad \mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \tag{4.1}$$

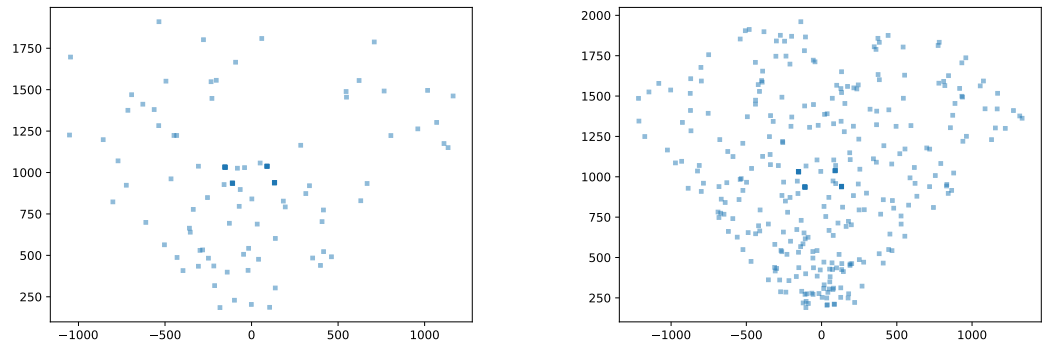
where the measurement function $h(\cdot)$ and measurement noise covariance $\mathbf{R}_k$ are:

$$h(\mathbf{x}_k) = \begin{bmatrix} \sqrt{(p_k^1)^2 + (p_k^2)^2} \\ \arctan\left(\frac{p_k^2}{p_k^1}\right) \end{bmatrix}, \quad \mathbf{R}_k = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\varphi^2 \end{bmatrix} \tag{4.2}$$

p_k^1 and p_k^2 are the positions of the object relative to the sensor in Cartesian coordinates. σ_r is the standard deviation of the range in meters and σ_φ is the standard deviation of the bearing in radians.

The clutter measurements are modeled as uniformly distributed over the surveillance area where the number of independent clutter measurements is Poisson distributed.

A set of initial states together with time of birth and death for each object as well as the total length of the sequence are defined. Using this information, ground truth



(a) Clutter rate $\lambda_c = 100$ **(b)** Clutter rate $\lambda_c = 300$

Figure 4.1: Two examples of generated measurements from a single timestep with object oriented measurement rate $\gamma = 8$.

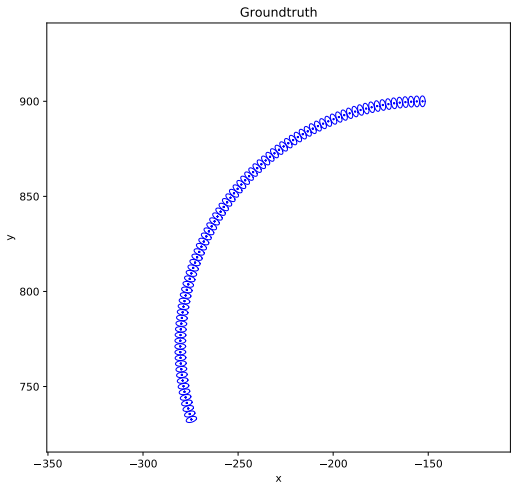
extended object states are generated by propagating the initial states through time according to the motion model.

The generated extended objects states are used to generate measurements. Objects only generate measurements if detected, therefore measurements originating from an object is generated with a probability of detection P_D. If an object is detected, it will generate a Poisson distributed number of measurements with a rate γ, spread uniformly according to the ground truth extent. These measurements are corrupted by noise according to the measurement covariance in the measurement model.

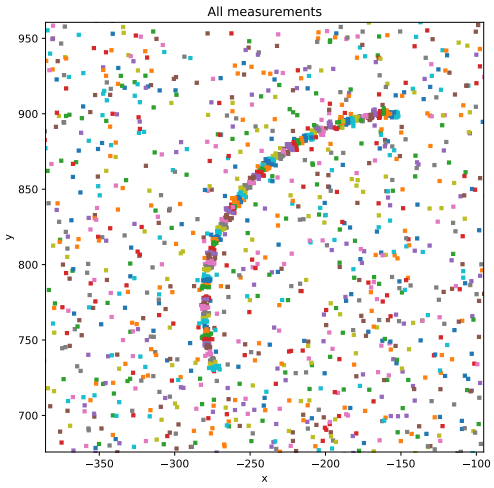
In addition to the object oriented measurements, clutter detections are also generated. The number of clutter measurements each timestep is Poisson distributed with a rate λ_c and are distributed uniformly[1] over the surveillance area. Examples of measurements from a single timestep with different clutter rate are shown in Figure 4.1. In this timestep, there are four objects present and the measurements generated from these can be seen as clusters around the center of the surveillance area.

Figure 4.2 depicts a generated ground truth trajectory with corresponding measurements.

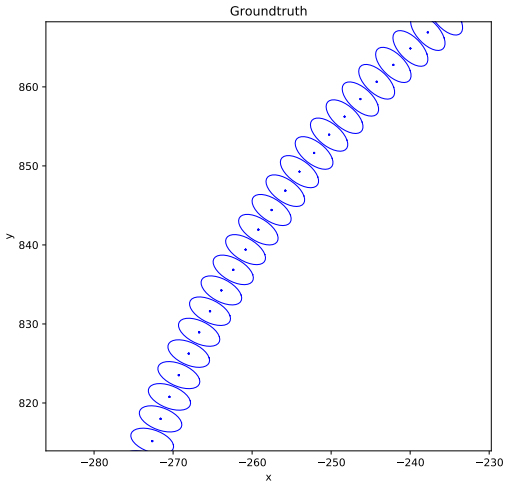
[1]Uniform over range and bearing in this case.



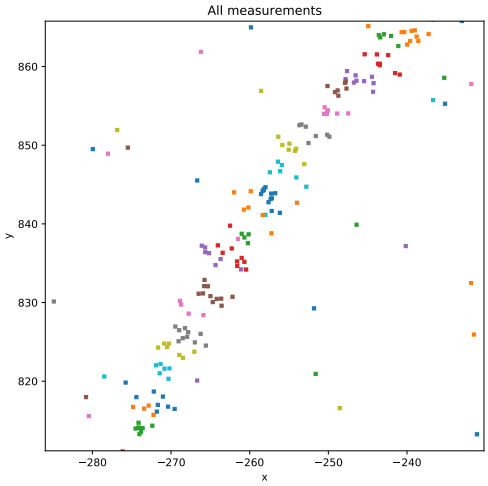
(a) Ground truth



(b) Measurements



(c) Ground truth zoomed in



(d) Measurements zoomed in

Figure 4.2: Generated ground truth extended object states and radar measurements from a sequence of length $K = 100$ with object oriented measurement rate $\gamma = 8$ and clutter rate $\lambda_c = 300$. The measurements are range-bearing converted to Cartesian coordinates and the different colors of measurements depict different timesteps.

4.2 Metric

The different filtering algorithms are evaluated by using the metrics defined in Section 2.7. The metric can be decomposed into four different terms as follows:

$$d^2(\mathbf{X}_k, \hat{\mathbf{X}}_k) = c_l^2(\mathbf{X}_k, \hat{\mathbf{X}}_k) + c_m^2(\mathbf{X}_k, \hat{\mathbf{X}}_k) + c_f^2(\mathbf{X}_k, \hat{\mathbf{X}}_k) + c_t^2(\mathbf{X}_k, \hat{\mathbf{X}}_k) \tag{4.3}$$

Where c_l is the Gaussian-Wasserstein distance and c_m, c_f and c_t are the missed, false and switch costs respectively. Using this decomposition allows for a more comprehensive overview of the performance of the different filters. $\mathbf{X}_k$ denotes the set of ground truth trajectories at time k and $\hat{\mathbf{X}}_k$ denotes the set of estimated alive trajectories at time k.

Since the generated measurement and ground-truth object states are random, Monte-Carlo simulation is used to obtain a more fair comparison between the algorithms. For each timestep k of the trajectory the root mean square (RMS) error given at a timestep is calculated as:

$$d(k) = \sqrt{\frac{1}{N_{mc}k} \sum_{i=1}^{N_{mc}} d^2\left(\mathbf{X}_k, \hat{\mathbf{X}}_k^{(i)}\right)} \tag{4.4}$$

Where N_{mc} denotes the number of Monte Carlo runs in the simulation.

The simulations are run using three different filters:

- GGIWTPHD with smoothing of the extent estimates as shown in Figure 3.1.
- GGIWTPHD where no smoothing is performed. This is included to highlight the difference in estimated extent error between the version that uses smoothing and the one that does not
- GGIWPHD that uses labels of the mixture components in order to form the trajectories.

4.3 Parameter settings

The three different filters are tested on different scenarios to compare their performance. In Table 4.1 the parameters used for all scenarios are given. Note that these parameters are identical in all three filters to provide a fair comparison.

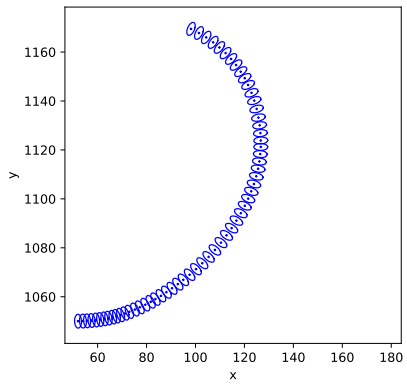
Table 4.1: Parameter settings for test scenarios.

Parameter		**Value**
Sampling time	T_s	1
Forgetting factor	η_k	2
Kinematics noise	σ_a, σ_ω	0.2, $0.2\frac{\pi}{180}$
Measurement noise	σ_r, σ_φ	1, $0.01\frac{\pi}{180}$
Extension uncertainty	n	120
Scaling parameter	ρ	0.75
Clutter rate	λ_c	100
Detection probability	P_D	0.99
Survival probability	P_S	0.99
Birth weight	$w_k^{(b)}$	0.03
Extraction threshold	$\bar{w}_e$	0.5
Pruning threshold	T	0.001
Merging/absorption threshold	U	5
Capping threshold	M	50

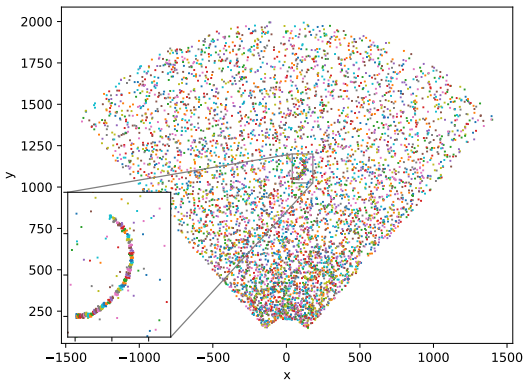
4.4 Scenario 1: Single object

The first scenario is composed by a single object that is born at time $k = 0$ with initial state $\mathbf{x}_0 = [50, 1050, 2, 0, \pi/120]$. The object then moves according to the motion model with the parameters in Table 4.1 until time $k = 60$ where it dies. Figure 4.3 shows the generated ground truth states and all measurements from the sequence.

An example of the estimates resulted from this scenario is seen in Figure 4.4 where the final trajectories built using both filters are presented. In time steps where the object is missed, the GGIWPHD filter is unable to fix this error and therefore some extent estimates are missing, the GGIWTPHD filter however is able to recover these misses.



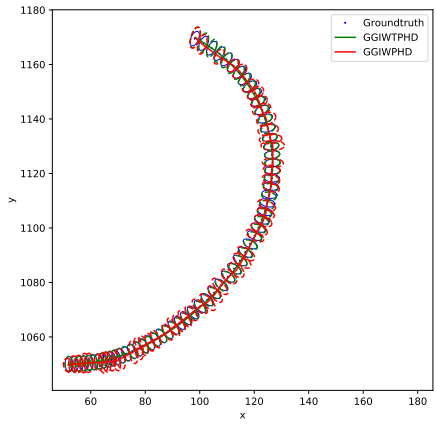
(a) Ground truth positions and extents



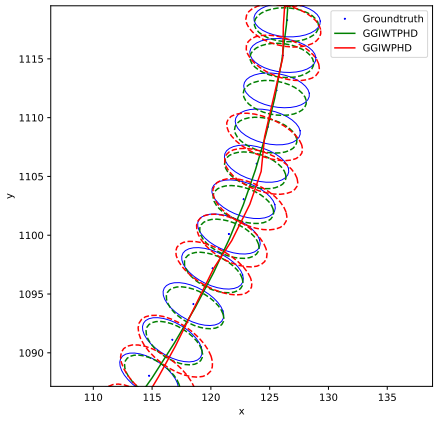
(b) Generated measurements. Note that the figure depicts all measurements over all time steps.

Figure 4.3: Generated data used for scenario 1. One target is born at time $k = 0$ and dies at $k = 60$. It is born in the lower left and moves in a curved trajectory.

4.4.1 Results

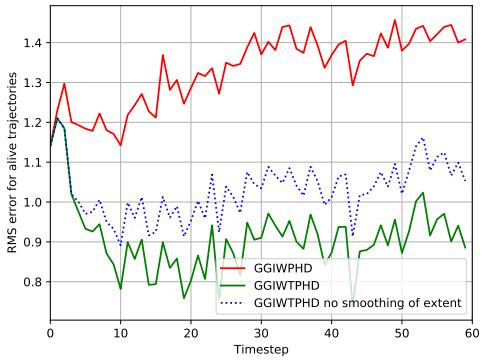


(a) Full trajectory.

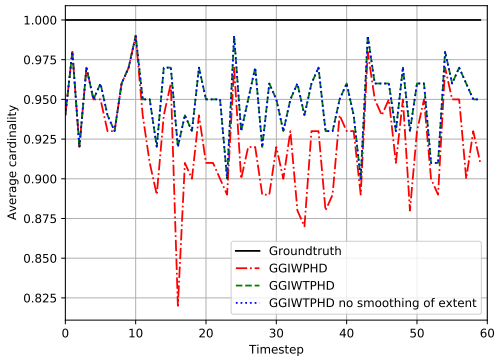


(b) Section of the full trajectory.

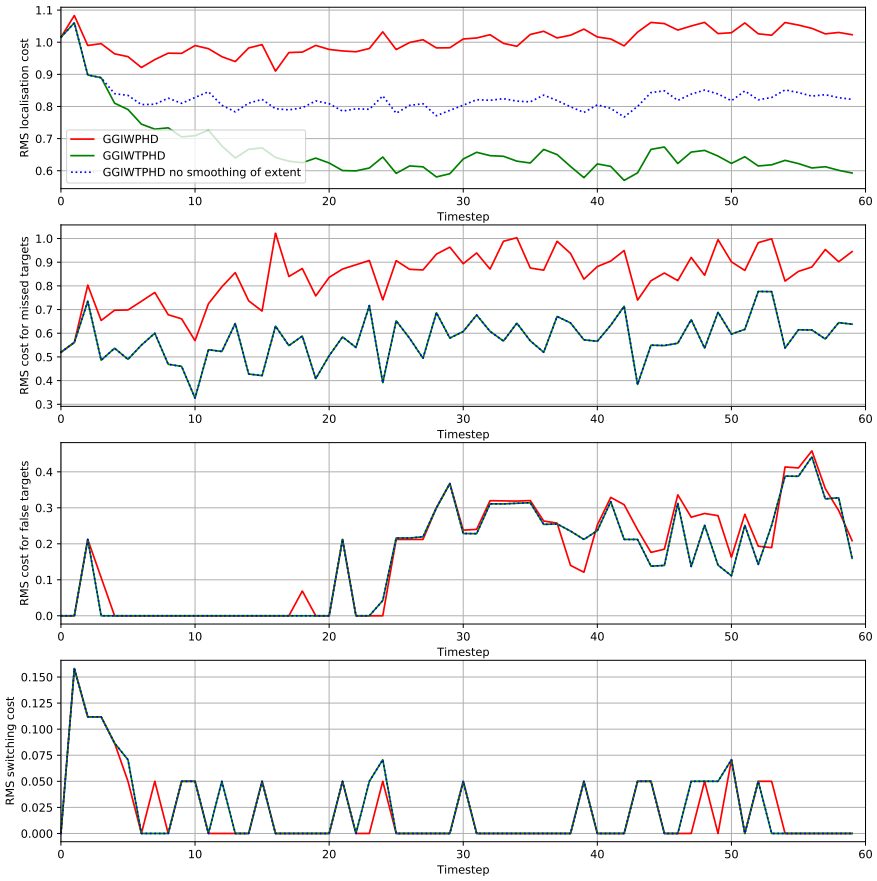
Figure 4.4: Example of the final trajectory estimates from running both filters in scenario 1. Ground truth position and extent is given by the blue dots and ellipses. Final trajectory estimates at the end of the sequence given by the full lines for trajectories and dashed lines for the extent.



(a) RMS error of the trajectory metric



(b) Average cardinality



(c) Decomposed RMS costs

Figure 4.5: Results comparing GGIWTPHD with smoothing of extent estimates, GGIWTPHD without smoothing of extent estimates and GGIWPHD using labeled components. These are results from a Monte Carlo simulation in scenario 1 of 100 runs.

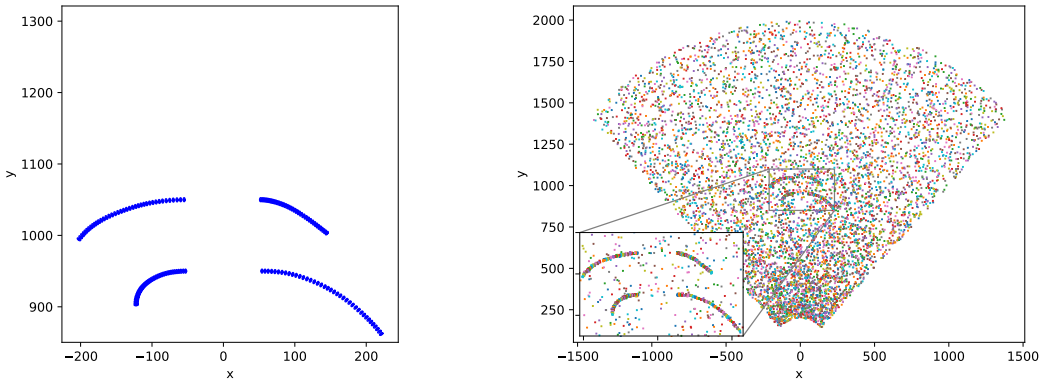
4.5 Scenario 2: Multiple objects

The second scenario is setup to demonstrate how the filters react to appearing and disappearing objects. It is composed by four objects that are born at different positions and timesteps, then move according to the motion model with parameters in Table 4.1 and finally die at different timesteps. Exact initial states and time of birth and death are given below.

1. Birth at $k = 0$, $\mathbf{x}_0 = [50, 1050, 2, 0, -\pi/120]$ and death at $k = 40$.
2. Birth at $k = 10$, $\mathbf{x}_0 = [-50, 950, -3, 0, \pi/135]$ and death at $k = 50$.
3. Birth at $k = 20$, $\mathbf{x}_0 = [-50, 1050, -5, 0, \pi/180]$ and death at $k = 60$.
4. Birth at $k = 30$, $\mathbf{x}_0 = [50, 950, x, 4, -\pi/225]$ and death at $k = 70$.

Figure 4.6 shows the generated ground truth states and all measurements from the sequence.

A portion of the estimates resulted from this scenario is seen in Figure 4.7 where the trajectories using both filters are presented.



(a) Ground truth positions and extents

(b) Generated measurements. Note that the figure depicts all measurements over all time steps.

Figure 4.6: Generated data used for scenario 2. Four targets are born at $k = [0, 10, 20, 30]$ and die at $k = [40, 50, 60, 70]$ respectively. The first is born at the top right, the second at the bottom left, the third at the top left and the fourth at the bottom right. All targets are born towards the center and move outward.

4.5.1 Results

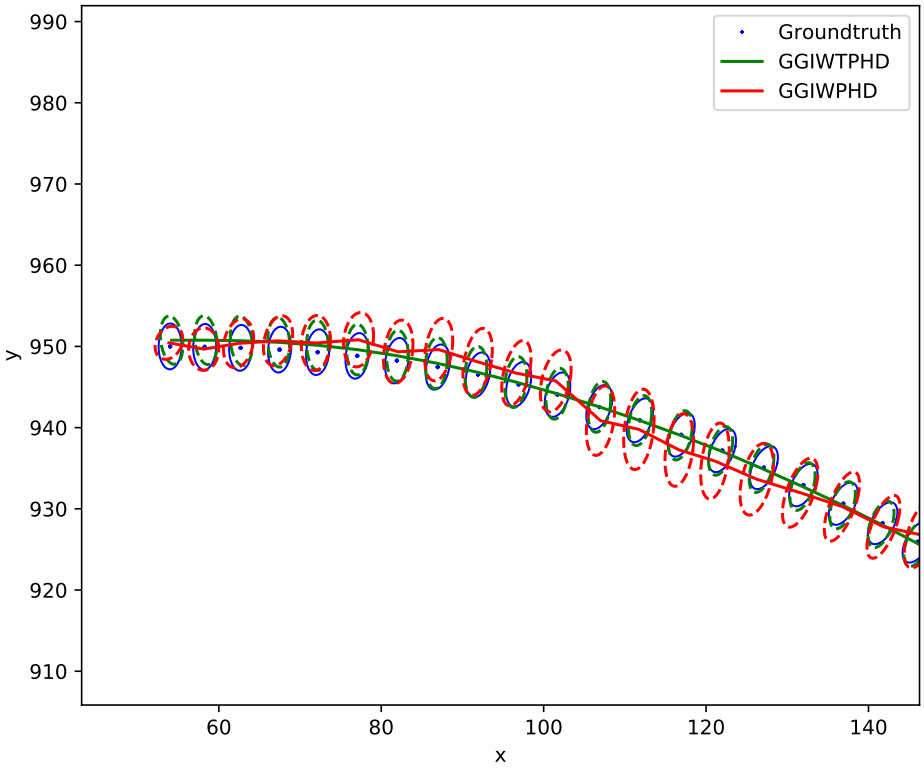
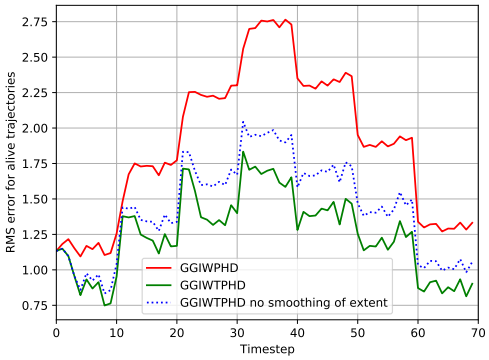
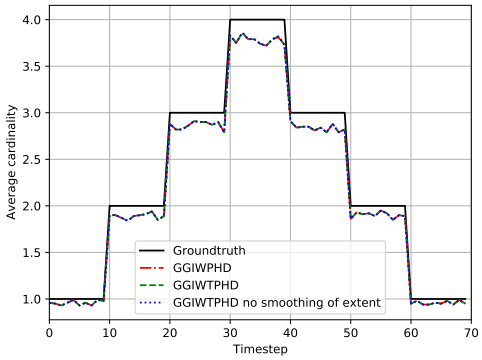


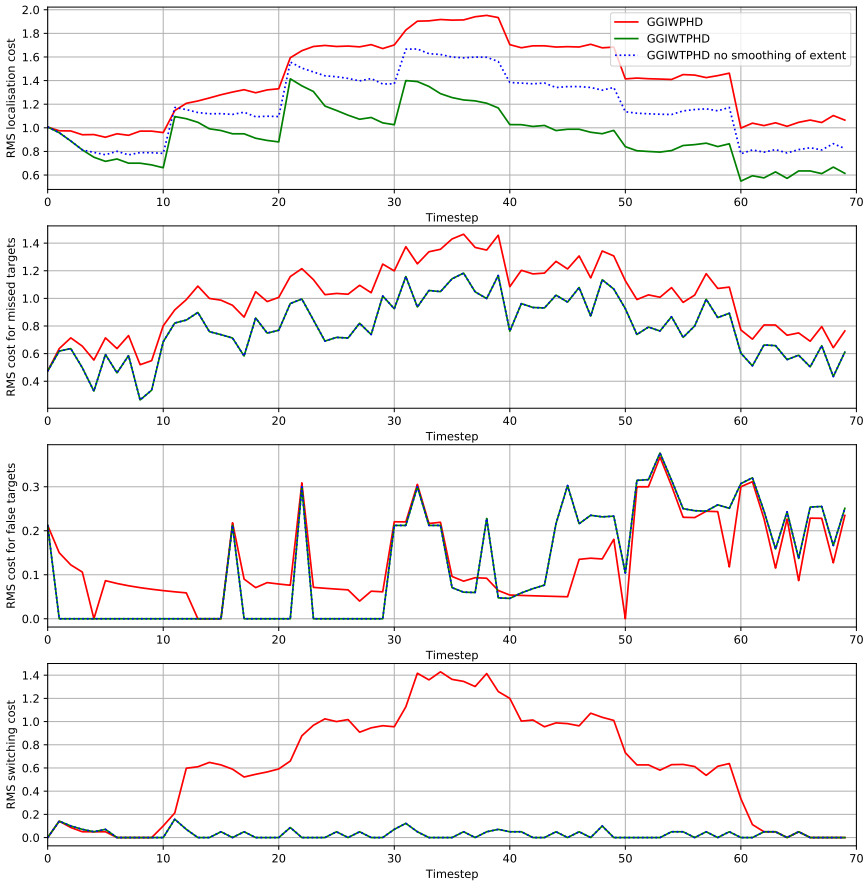
Figure 4.7: Estimates from the start of the lower right trajectory. Ground truth position and extent is given by blue dots and ellipses. Trajectory estimates are given by the full lines for trajectories and dashed lines for the extent.



(a) RMS error of the trajectory metric



(b) Average cardinality



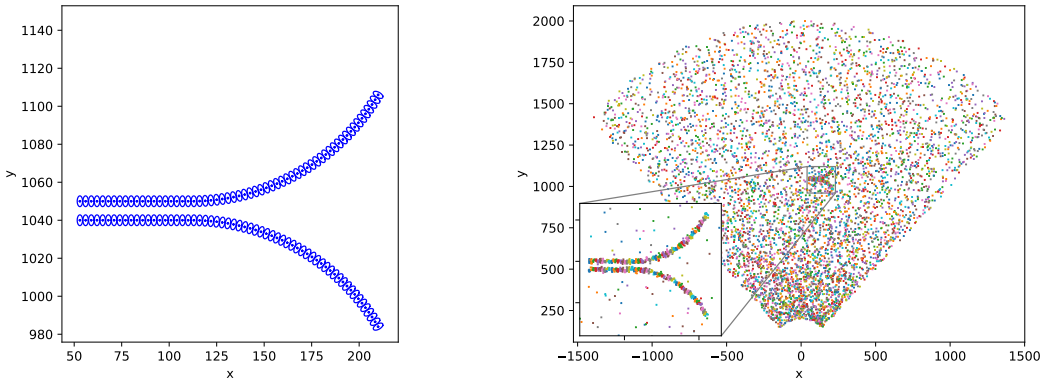
(c) Decomposed RMS costs

Figure 4.8: Results comparing GGIWTPHD with smoothing of extent estimates, GGIWTPHD without smoothing of extent estimates and GGIWPHD using labeled components. These are results from a Monte Carlo simulation in scenario 2 of 100 runs.

4.6 Scenario 3: Spatially close objects

The third scenario is built to show how the filters work when there are close objects that move similarly. It is composed by two objects that are born close to each other at $k = 0$ and move in parallel until $k = 20$ where they turn away from each other and die at $k = 60$. Figure 4.9 shows the generated ground truth states and all measurements from the sequence.

This scenario causes switching to occur, an example of a case where there is heavy switching is shown in Figure 4.10. The GGIWPHD filter is not able to distinguish between the two objects and therefore trajectories can not be built from these estimates. The GGIWTPHD filter also has trouble distinguishing between the two objects, but is still able to build reasonable trajectories. Another example of trajectory estimates from both filters can be seen in Figure 4.11, this plot also include extent estimates.

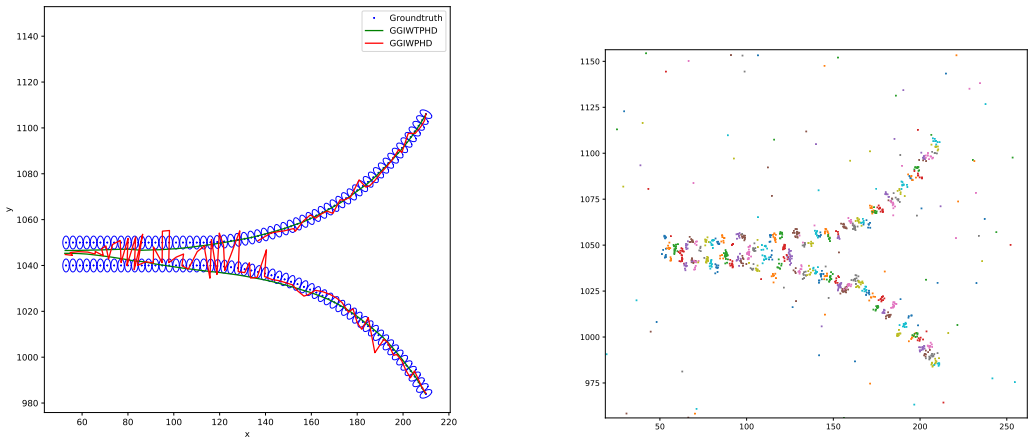


(a) Ground truth positions and extents

(b) Generated measurements. Note that the figure depicts all measurements over all time steps.

Figure 4.9: Generated data used for scenario 3. Two targets are born close to each other at time $k = 0$, then move in parallel and turn away from each other after some time. Both die at time $k = 60$.

4.6.1 Results



(a) Ground truth and estimated trajectories. **(b)** Measurements

Figure 4.10: Example of switches in scenario 3.

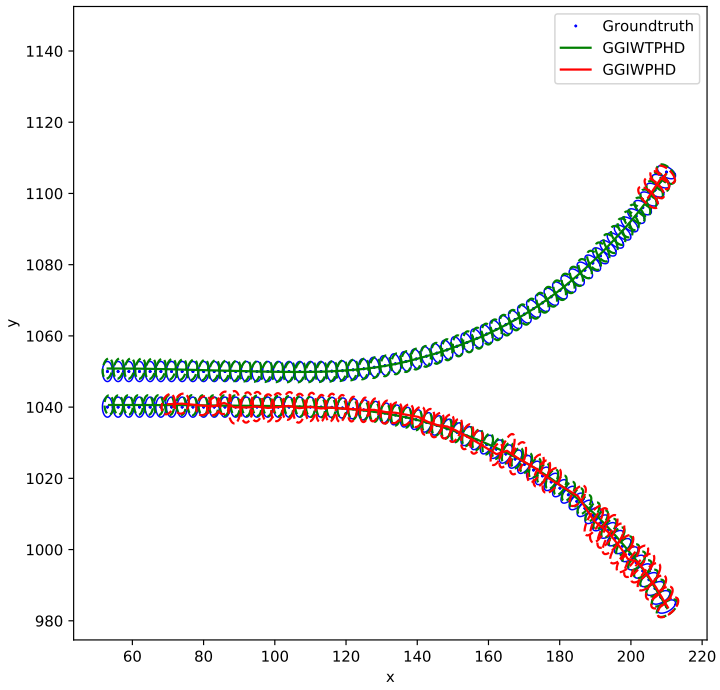
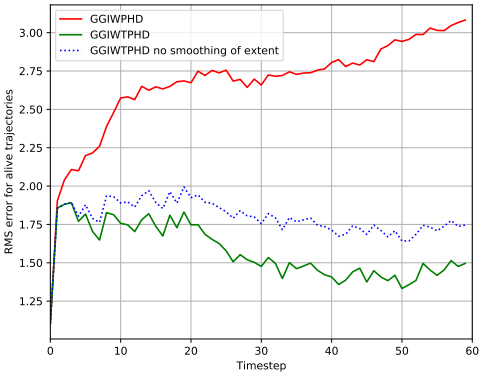
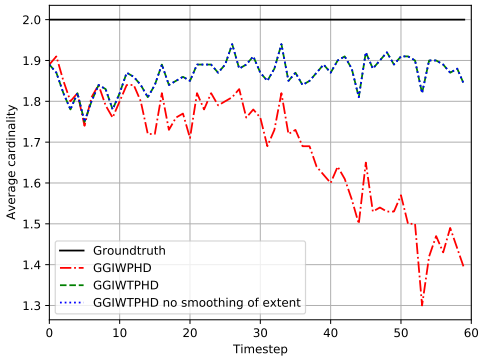


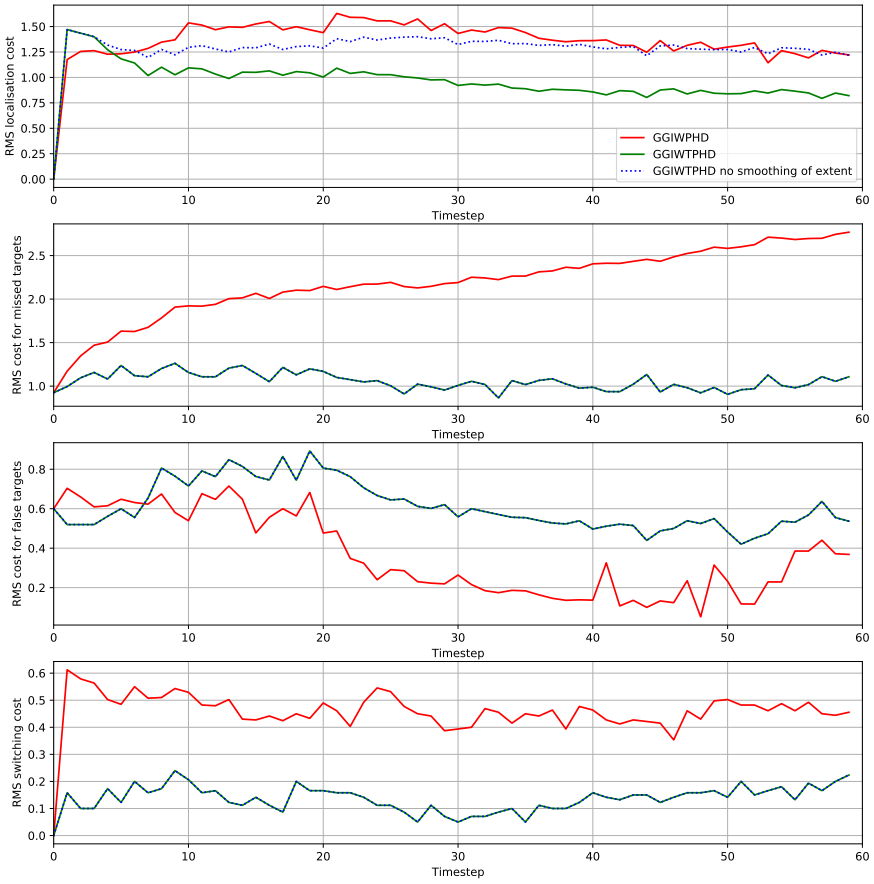
Figure 4.11: Example of the final trajectory estimates from running both filters in scenario 3. Ground truth position and extent is given by blue dots and ellipses. Trajectory estimates at the end of the sequence are given by the full lines for trajectories and dashed lines for the extent.



(a) RMS error of the trajectory metric



(b) Average cardinality



(c) Decomposed RMS costs

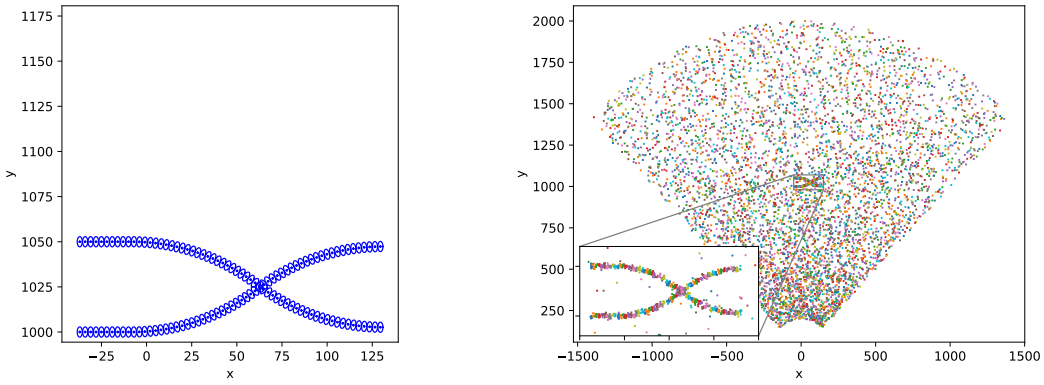
Figure 4.12: Results comparing GGIWTPHD with smoothing of extent estimates, GGIWTPHD without smoothing of extent estimates and GGIWPHD using labeled components. These are results from a Monte Carlo simulation in scenario 3 of 100 runs.

4.7 Scenario 4: Objects with intersecting trajectories

The fourth scenario demonstrates how the filters perform when there are objects that cross paths. It is composed of two objects that are born at $k = 0$ and move in parallel until $k = 10$ where they start turning towards each other and cross paths at approximately $k = 35$. Both objects die at $k = 60$. Figure 4.13 shows the generated ground truth states and all measurements from the sequence.

An example of the trajectory estimates from both filters in this scenario can be seen in Figure 4.14. In Figure 4.14 The GGIWTPHD filter manages to build trajectories and handles the crossing while the GGIWPHD filter does not. The GGIWPHD is able to build full trajectories in some cases, but in general it fails.

In Figure 4.15b it can be seen that both filters have trouble differentiating between the two objects at the crossing point, i.e. around $k = 35$.

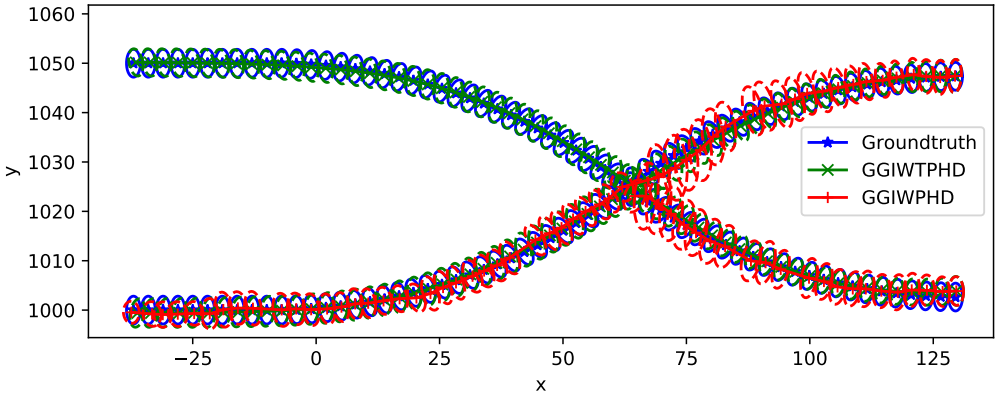


(a) Ground truth positions and extents

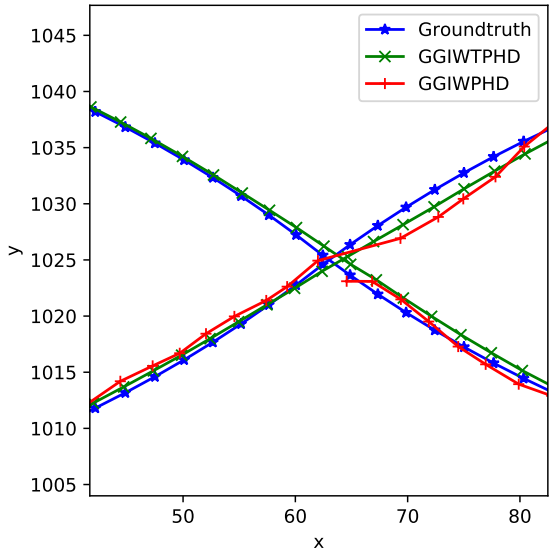
(b) Generated measurements. Note that the figure depicts all measurements over all time steps.

Figure 4.13: Generated data used for scenario 4. Two targets are born at time $k = 0$ and move in parallel until turning towards each other such that they cross paths after some time. Both die at time $k = 60$.

4.7.1 Results

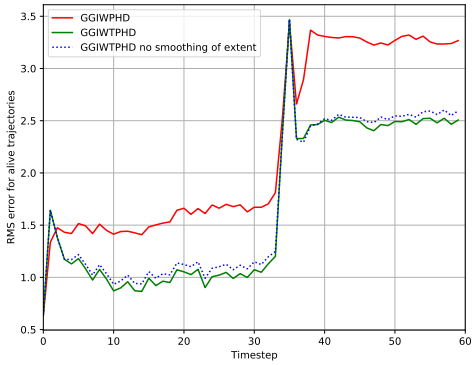


(a) Full trajectories including extent.

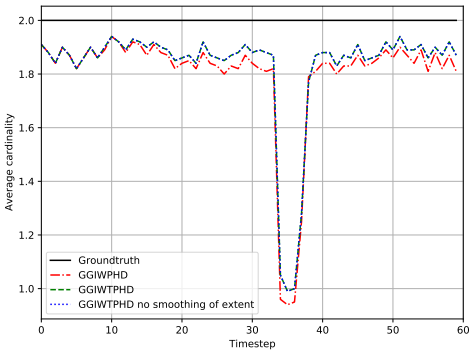


(b) Trajectories at crossing point without extent estimates.

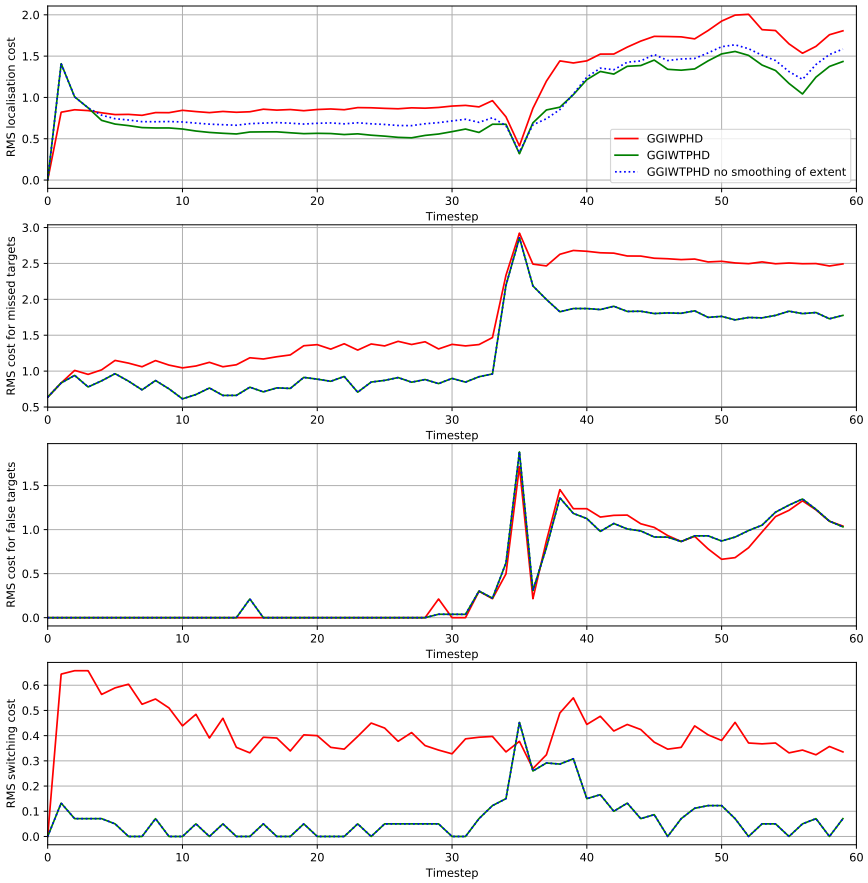
Figure 4.14: Example of the final trajectory estimates from running both filters in scenario 4. Ground truth position and extent is given by blue dots and ellipses. Final trajectory estimates at the end of the sequence are given by the full lines for trajectories and dashed lines for the extent.



(a) RMS error of the trajectory metric



(b) Average cardinality



(c) Decomposed RMS costs

Figure 4.15: Results comparing GGIWTPHD with smoothing of extent estimates, GGIWTPHD without smoothing of extent estimates and GGIWPHD using labeled components. These are results from a Monte Carlo simulation in scenario 4 of 100 runs.

5

Discussion

From the results presented in the previous chapter the conclusion is drawn that the GGIWTPHD filter outperforms the GGIWPHD filter with labeled components when tracking trajectories in all cases demonstrated. The most notable result is that the GGIWTPHD filter suffers much less from track switches and can thus estimate longer trajectories better than the GGIWPHD filter with labeled components. This section will present some further discussion of the results as well as a short discussion on ethics.

5.1 Filtering Performance

Evaluation of the trajectory metric gives a good indication regarding performance when trajectories are estimated but it does not give a metric on filtering performance. Filtering performance denotes real-time tracking performance given only the latest time step in a sequence. To demonstrate filtering performance the GOSPA metric defined in Section 2.7 is calculated at each time step k for $\tau_k(\mathbf{X}_k)$ where $\mathbf{X}_k$ denotes the current trajectory estimates. The results are shown in Figure 5.1. Performance of the GGIWTPHD and GGIWPHD filters are identical in these cases. This is deemed reasonable since a trajectory only infers on past estimates and does not impact the prediction and update steps for future estimates.

It is therefore important to make the distinction that if filtering performance is the main interest then the GGIWTPHD filter will not perform better than the GGIWPHD filter.

5.2 Future work

The filtering algorithms used for extended object tracking in this thesis rely on clustering to find partitions $\mathcal{P}$. This implies that the clustering algorithm used have a large impact on the performance of the filters, especially in the case of spatially close objects. A demonstration of different partitions computed from the same set of measurements can be seen in Figure 5.2.

Since the GGIWTPHD filter applies smoothing to each extent estimate at every

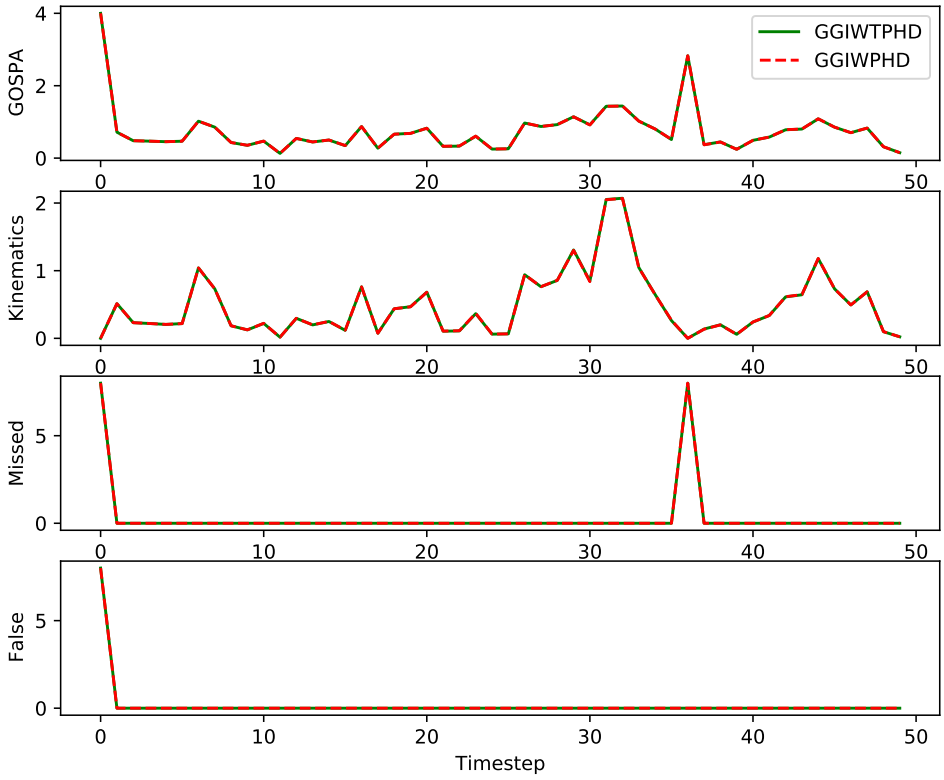


Figure 5.1: Filtering performance for the GGIWTPHD and GGIWPHD filters. This figure shows that the filtering performance for both filters is equal.

time step the computational complexity increases for longer trajectories. A topic for future research could be to estimate trajectories for the extent estimates using a similar approach that was used to calculate the kinematic trajectory. Smoothing would then be applied at each timestep due to the prediction and measurement updates considering the whole trajectory of the extent and not just the current timestep.

If two targets that are initially far apart move in close to each other it is likely that the clustering algorithm will interpret these targets as one single target. This will affect both cardinality and extent estimates since two small objects will be interpreted as one larger object by the filtering algorithms. One way to remedy this problem is to use information from prior states when performing clustering. One way could be to factor in the likelihood $p(\mathbf{Z}_k|\xi_k)$ into the clustering. These types of approaches have been proposed in the literature by Granström et al. [32, 33]. Due to time constraints this is not considered further in this thesis but would likely improve filtering performance in some scenarios.

Because of the nature of the GGIWTPHD filter recursions, the past means and covariances of the trajectory are not removed by marginalization. This means that

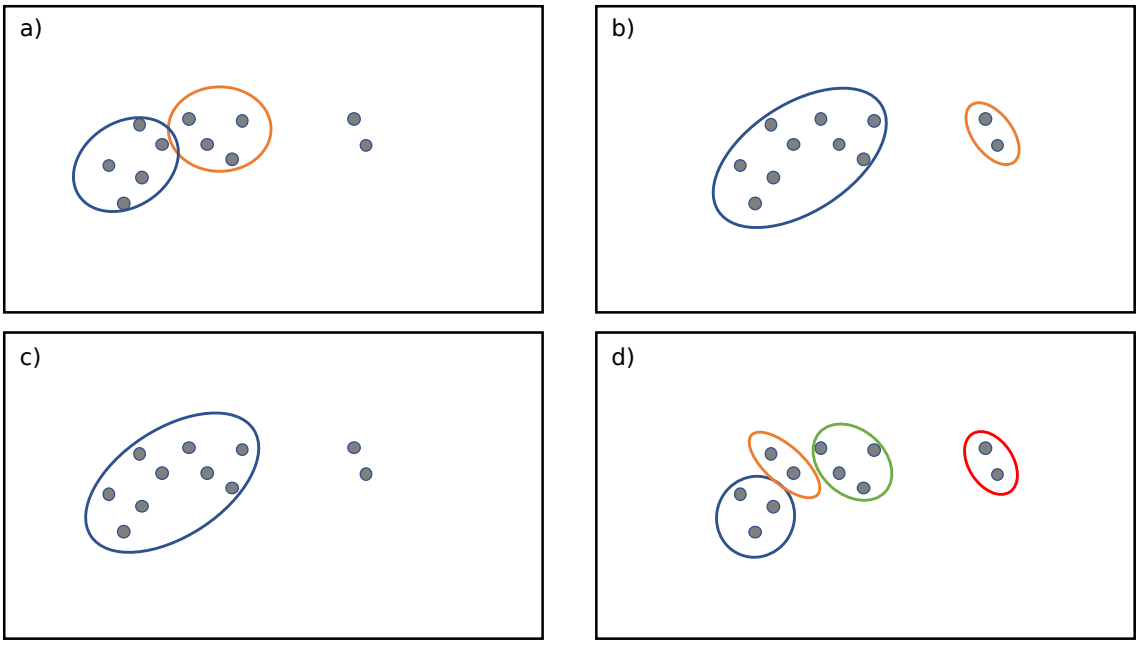


Figure 5.2: Different clusters that can be interpreted from a set of measurements. The gray dots indicate measurements and the ellipses show the computed partitions

the computational complexity increases linearly with the length of the trajectory. For very long trajectories and small sampling times this can become a problem if speed of the algorithm is an important factor. One way to remedy this proposed in [17] is to not consider the whole trajectory at each measurement update step in the filter but only the L last ones. This gives the L-scan version by approximating the covariance matrix of each trajectory as:

$$P_k^j \approx \text{diag}\left(\tilde{P}_{t_j^k}^j, \tilde{P}_{t_j^k+1}^j, \ldots, \tilde{P}_{k-L}^j, \tilde{P}_{k-L+1:k}^j\right) \tag{5.1}$$

where $\tilde{P}_{k-L+1:k}^j \in \mathbb{R}^{L \cdot n_x \times L \cdot n_x}$ represents the joint covariance of the last L time instants. This assumption can be reasonable since recent measurements are not likely to affect a trajectories past in any significant way past a few time steps.

When developing filtering algorithms that are to be used in real scenarios it is important that the measurements that the filters are processing represent the reality well. In this thesis only simulated scenarios are shown that have little motivation in real-world scenarios where the probability of detection may be lower and fewer measurements may be received. An expansion to the work presented in this thesis would be try test performance of the GGIWTPHD filter on real data from for example, an automotive radar.

5.3 Ethics

In this section, some of the ethical aspect that the thesis concerns will be addressed. When developing new technologies, consideration should be given to the disadvantages and advantages that may come from these technologies.

5.3.1 Societal aspects

When developing systems used for tracking it is important to consider the possible applications in which it can be used, in the case of this master thesis the algorithms developed can be used to localize and track extended objects like cars or pedestrians in an autonomous driving scenario. It must be considered that it could be used in applications that could lead to the harm of human life but is stressed that it is not the intention of the authors that the algorithms developed in this thesis are to be used in any applications that can be used to harm others.

5.3.2 Academic ethics

It is also important to consider the ethical aspect of correctness, since this is a master thesis project it will could be the basis of further research and must therefore be honest when it comes to presenting results and what has actually been achieved. It is therefore important not to selectively choose results which can give an inaccurate representation of the actual achievements of the thesis. This is why Monte Carlo simulation is used as it tests the algorithms in a more broad sense and computes a cross-section regarding performance. It is also important that academic integrity is upheld and that the work of others is properly cited and credit is given where it is due.

5.4 Conclusion

This thesis has presented a new algorithm for tracking sets of trajectories in extended object tracking which is named the GGIWTPHD filter. The filter is derived using a heuristic approach that combines the results presented in three different papers. The first concerns the construction of trajectories for the kinematic state [17]. The second implements a PHD filter using a Gamma Gaussian inverse Wishart PHD [11] and the third concerns Bayesian smoothing of extent estimates in the random matrix framework [43]. The filter recursions estimates the set of alive trajectories at each time step and then performs smoothing on the set of extent estimates at each time step.

A GGIWPHD filter is also implemented that constructs trajectories by labelling the GGIW components in the PHD. This filter is used for comparison when evaluating the performance of the GGIWTPHD filter.

Different simulation scenarios were set up and Monte Carlo simulations were run to evaluate performance of the GGIWTPHD filter with smoothed extent estimates,

GGIWTPHD filter without smoothed extent estimates and the GGIWPHD filter. It was shown that according to the defined metric, the GGIWTPHD filter with smoothing performs better in all presented scenarios. All algorithms suffer a decrease in performance when incorrect clusters are computed but it is also shown that even in this case, the GGIWTPHD filter outperforms the GGIWPHD filter.

Bibliography

[1] Merill I Skolnik. *Radar - Factors affecting radar performance*. 2019. URL: https://www.britannica.com/technology/radar/Factors-affecting-radar-performance (visited on 02/27/2020).

[2] R.V. Ostrovitianov and F.A. Basalov. *Statistical theory of extended radar targets*. Artech House, 1985. ISBN: 9780890061442. URL: https://books.google.se/books?id=97o2AQAAIAAJ.

[3] Samuel S Blackman. "Multiple hypothesis tracking for multiple target tracking". In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 (2004), pp. 5–18.

[4] Y Bar-Shalom, F Daum, and J Huang. "The probabilistic data association filter". In: *IEEE Control Systems Magazine* 29.6 (Dec. 2009), pp. 82–100. ISSN: 1941-000X. DOI: 10.1109/MCS.2009.934469.

[5] Roy L Streit and Tod E Luginbuhl. *Probabilistic multi-hypothesis tracking*. Tech. rep. NAVAL UNDERWATER SYSTEMS CENTER NEWPORT RI, 1995.

[6] Ronald P.S Mahler. "Advances in Statistical Multisource-Multitarget Information Fusion". In: *Advances in Statistical Multisource-Multitarget Information Fusion*. 1st ed. Norwood: Artech House, 2014. Chap. 13, pp. 379–403. ISBN: 9781608077991.

[7] Ronald P.S. Mahler. "Multitarget Bayes Filtering via First-Order Multitarget Moments". In: *IEEE Transactions on Aerospace and Electronic Systems* 39.4 (Oct. 2003), pp. 1152–1178. ISSN: 00189251. DOI: 10.1109/TAES.2003.1261119.

[8] Ba-Ngu Vo and Wing-Kin Ma. "The Gaussian Mixture Probability Hypothesis Density Filter". In: *Signal Processing, IEEE Transactions on* 54 (Dec. 2006), pp. 4091–4104. DOI: 10.1109/TSP.2006.881190.

[9] Karl Granström, Christian Lundquist, and Omut Orguner. "Extended target tracking using a gaussian-mixture PHD filter". In: *IEEE Transactions on Aerospace and Electronic Systems* 48.4 (2012), pp. 3268–3286. ISSN: 00189251. DOI: 10.1109/TAES.2012.6324703.

[10] Christian Lundquist, Karl Granström, and Umut Orguner. "An extended target CPHD filter and a gamma gaussian inverse wishart implementation". In: *IEEE Journal on Selected Topics in Signal Processing* 7.3 (2013), pp. 472–483. ISSN: 19324553. DOI: 10.1109/JSTSP.2013.2245632.

[11] Karl Granström et al. "Gamma Gaussian Inverse Wishart Probability Hypothesis Density for Extended Target Tracking Using X-Band Marine Radar Data". In: *IEEE Transactions on Geoscience and Remote Sensing* 53.12 (Dec. 2015), pp. 6617–6631. ISSN: 01962892. DOI: 10.1109/TGRS.2015.2444794.

[12] Johann Wolfgang Koch. "Bayesian approach to extended object and cluster tracking using random matrices". In: *IEEE Transactions on Aerospace and Electronic Systems* 44.3 (2008), pp. 1042–1059. ISSN: 00189251. DOI: 10.1109/TAES.2008.4655362.

[13] Karl Granström and Marcus Baum. "Extended Object Tracking: Introduction, Overview and Applications". In: *CoRR* abs/1604.00970 (2016). arXiv: 1604.00970. URL: http://arxiv.org/abs/1604.00970.

[14] D E Clark, K Panta, and B Vo. "The GM-PHD Filter Multiple Target Tracker". In: *2006 9th International Conference on Information Fusion*. July 2006, pp. 1–8. DOI: 10.1109/ICIF.2006.301809.

[15] H Zhang et al. "An improved GM-PHD tracker with track management for multiple target tracking". In: *2015 International Conference on Control, Automation and Information Sciences (ICCAIS)*. Oct. 2015, pp. 185–190. DOI: 10.1109/ICCAIS.2015.7338659.

[16] K Panta, D E Clark, and B Vo. "Data Association and Track Management for the Gaussian Mixture Probability Hypothesis Density Filter". In: *IEEE Transactions on Aerospace and Electronic Systems* 45.3 (July 2009), pp. 1003–1016. ISSN: 1557-9603. DOI: 10.1109/TAES.2009.5259179.

[17] Ángel F. García-Fernández and Lennart Svensson. "Trajectory PHD and CPHD filters". In: *IEEE Transactions on Signal Processing* 67.22 (Nov. 2018), pp. 5702–5714. DOI: 10.1109/TSP.2019.2943234. arXiv: 1811.08820. URL: http://arxiv.org/abs/1811.08820%20http://dx.doi.org/10.1109/TSP.2019.2943234.

[18] Ángel F. García-Fernández, Lennart Svensson, and Mark R. Morelande. "Multiple target tracking based on sets of trajectories". In: *IEEE Transactions on Aerospace and Electronic Systems* (May 2016), pp. 1–1. arXiv: 1605.08163. URL: http://arxiv.org/abs/1605.08163.

[19] Abu Sajana Rahmathullah, Ángel F. García-Fernández, and Lennart Svensson. "A metric on the space of finite sets of trajectories for evaluation of multi-target tracking algorithms". In: (May 2016). arXiv: 1605.01177. URL: http://arxiv.org/abs/1605.01177.

[20] Shishan Yang, Marcus Baum, and Karl Granstrom. "Metrics for performance evaluation of elliptic extended object tracking methods". In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. Vol. 0. Institute of Electrical and Electronics Engineers Inc., July 2016, pp. 523–528. ISBN: 9781467397087. DOI: 10.1109/MFI.2016.7849541.

[21] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, Jan. 2010. ISBN: 9781139344203. DOI: 10.1017/CBO9781139344203.

[22] R. E. Kalman. "A new approach to linear filtering and prediction problems". In: *Journal of Fluids Engineering, Transactions of the ASME* 82.1 (1960), pp. 35–45. ISSN: 1528901X. DOI: 10.1115/1.3662552.

[23] Ienkaran Arasaratnam and Simon Haykin. "Cubature kalman filters". In: *IEEE Transactions on Automatic Control* 54.6 (2009), pp. 1254–1269. ISSN: 00189286. DOI: 10.1109/TAC.2009.2019800.

[24] E. A. Wan and R. Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". In: *IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, AS-SPCC 2000*. Institute of Electrical and Electronics Engineers Inc., 2000, pp. 153–158. ISBN: 0780358007. DOI: 10.1109/ASSPCC.2000.882463.

[25] Karl Granström et al. "Random set methods: Estimation of multiple extended objects". In: *IEEE Robotics and Automation Magazine* 21.2 (2014), pp. 73–82. ISSN: 10709932. DOI: 10.1109/MRA.2013.2283185.

[26] S S Blackman and R Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House radar library. Artech House, 1999. ISBN: 9781580530064. URL: https://books.google.se/books?id=lTIfAQAAIAAJ.

[27] Ronald P.S Mahler. "PHD filters for nonstandard targets, I: Extended targets". In: *2009 12th International Conference on Information Fusion, FUSION 2009*. Seattle, WA, USA: IEEE, 2009, pp. 915–921. ISBN: 9780982443804.

[28] Gian-Carlo Rota. "The Number of Partitions of a Set". In: *The American Mathematical Monthly* 71.5 (May 1964), pp. 498–504. ISSN: 00029890. DOI: 10.2307/2312585.

[29] Katta G. Murty. "An Algorithm for Ranking all the Assignments in Order of Increasing Cost". In: *Operations Research* 16.3 (1968), pp. 682–687. ISSN: 0030364X, 15265463. URL: http://www.jstor.org/stable/168595.

[30] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval Research Logistics (NRL)* 52.1 (2005), pp. 7–21. DOI: 10.1002/nav.20053. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.20053. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20053.

[31] Songhwai Oh, Stuart Russell, and Shankar Sastry. "Markov chain Monte Carlo data association for multi-target tracking". In: *IEEE Transactions on Automatic Control* 54.3 (2009), pp. 481–497.

[32] Karl Granstrom et al. “Pedestrian tracking using Velodyne data-Stochastic optimization for extended object tracking”. In: *IEEE Intelligent Vehicles Symposium, Proceedings*. Institute of Electrical and Electronics Engineers Inc., July 2017, pp. 39–46. ISBN: 9781509048045. DOI: 10.1109/IVS.2017.7995696.

[33] Karl Granstrom et al. “Likelihood-Based Data Association for Extended Object Tracking Using Sampling Methods”. In: *IEEE Transactions on Intelligent Vehicles* 3.1 (Jan. 2018), pp. 30–45. ISSN: 2379-8858. DOI: 10.1109/tiv.2017.2788184.

[34] Kevin Gilholm et al. “Poisson models for extended target and group tracking”. In: *Signal and Data Processing of Small Targets 2005*. Vol. 5913. SPIE, Aug. 2005, 59130R. DOI: 10.1117/12.618730.

[35] K Granström and U Orguner. “Estimation and maintenance of measurement rates for multiple extended target tracking”. In: *2012 15th International Conference on Information Fusion*. Singapore: IEEE, July 2012, pp. 2170–2176.

[36] Karl Granström and Umut Orguner. “A phd filter for tracking multiple extended targets using random matrices”. In: *IEEE Transactions on Signal Processing* 60.11 (2012), pp. 5657–5671. ISSN: 1053587X. DOI: 10.1109/TSP.2012.2212888.

[37] L. Hammarstrand et al. “Extended Object Tracking using a Radar Resolution Model”. In: *IEEE Transactions on Aerospace and Electronic Systems* 48.3 (July 2012), pp. 2371–2386. ISSN: 1557-9603. DOI: 10.1109/TAES.2012.6237597.

[38] K. Granström, C. Lundquist, and U. Orguner. “Tracking rectangular and elliptical extended targets using laser measurements”. In: *14th International Conference on Information Fusion*. July 2011, pp. 1–8.

[39] Dengsheng Zhang and Guojun Lu. “Study and evaluation of different Fourier methods for image retrieval”. In: *Image and Vision Computing* 23.1 (Jan. 2005), pp. 33–49. ISSN: 02628856. DOI: 10.1016/j.imavis.2004.09.001.

[40] Marcus Baum and Uwe D. Hanebeck. “Extended Object Tracking with Random Hypersurface Models”. In: *IEEE Transactions on Aerospace and Electronic Systems* 50.1 (Apr. 2013), pp. 149–159. arXiv: 1304.5084. URL: http://arxiv.org/abs/1304.5084.

[41] Prasanta Chandra Mahalanobis. “On the generalized distance in statistics”. In: National Institute of Science of India. 1936.

[42] K. Granström and U. Orguner. “New prediction for extended targets with random matrices”. In: *IEEE Transactions on Aerospace and Electronic Systems* 50.2 (Apr. 2014), pp. 1577–1589. ISSN: 1557-9603. DOI: 10.1109/TAES.2014.120211.

[43] Karl Granström and Jakob Bramstång. “Bayesian Smoothing for the Extended Object Random Matrix Model”. In: *IEEE Transactions on Signal Processing* 67.14 (Jan. 2019), pp. 3732–3742. DOI: 10.1109/TSP.2019.2920471. arXiv: 1901.05301. URL: http://arxiv.org/abs/1901.05301%20http://dx.doi.org/10.1109/TSP.2019.2920471.

[44] Dominic Schuhmacher, Ba Tuong Vo, and Ba Ngu Vo. "A consistent metric for performance evaluation of multi-object filters". In: *IEEE Transactions on Signal Processing* 56.8 I (Aug. 2008), pp. 3447–3457. ISSN: 1053587X. DOI: 10.1109/TSP.2008.920469.

[45] Abu Sajana Rahmathullah, Angel F. Garcia-Fernandez, and Lennart Svensson. "Generalized optimal sub-pattern assignment metric". In: *20th International Conference on Information Fusion, Fusion 2017 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., Jan. 2017. ISBN: 9780996452700. DOI: 10.23919/ICIF.2017.8009645. arXiv: 1601.05585. URL: http://arxiv.org/abs/1601.05585%20http://dx.doi.org/10.23919/ICIF.2017.8009645.

[46] John R. Hoffman and Ronald P.S. Mahler. "Multitarget miss distance and its applications". In: *Proceedings of the 5th International Conference on Information Fusion, FUSION 2002*. Vol. 1. IEEE Computer Society, 2002, pp. 149–155. DOI: 10.1109/ICIF.2002.1021144.

[47] Dimitri P. Bertsekas. "An Auction Algorithm for Shortest Paths". In: *SIAM Journal on Optimization* 1.4 (Nov. 1991), pp. 425–447. ISSN: 1052-6234. DOI: 10.1137/0801026.

[48] K Granström and U Orguner. "On the reduction of Gaussian inverse Wishart mixtures". In: *2012 15th International Conference on Information Fusion*. July 2012, pp. 2162–2169.

[49] Karl Granström and Umut Orguner. *Implementation of the GIW-PHD filter Implementation of the GIW-PHD lter*. Tech. rep. 2012.

[50] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[51] Travis E Oliphant. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006.

[52] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: https://doi.org/10.1038/s41592-019-0686-2.

[53] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

A

Appendix 1

A.1 Pseudocode for GGIWPHD filter pruning, merging and capping

Table A.1: Pseudocode for GGIWPHD filter pruning, merging and capping

---

input: GGIW components $\left\{w_{k|k}^{(j)}, \xi_{k|k}^{(j)}\right\}_{j=1}^{J_{k|k}}$, a pruning threshold T, a merging threshold U and a maximum number of allowed components M.

init: Set $\ell \leftarrow 0$ and $I \leftarrow \left\{ i = 1, ..., J_{k|k} \middle| w_{k|k}^{(i)} > T \ \wedge \ \frac{\alpha_{k|k}^{(i)}}{\beta_{k|k}^{(i)}} > 1 \right\}$

repeat

$\ell \leftarrow \ell + 1$

$j \leftarrow \arg\max_{i \in I} w_{k|k}^{(i)}$

$L \leftarrow \left\{ i \in I \middle| \left(m_{k|k}^{(i)} - m_{k|k}^{(j)}\right)^{\mathsf{T}} \left(P_{k|k}^{(j)}\right)^{-1} \left(m_{k|k}^{(i)} - m_{k|k}^{(j)}\right) \leq U \right\}$

$\tilde{w}_{k|k}^{(\ell)} \leftarrow \sum_{i \in L} w_{k|k}^{(i)}$

$\tilde{\alpha}_{k|k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k|k}^{(\ell)}} \sum_{i \in L} w_{k|k}^{(i)} \alpha_{k|k}^{(i)}$

$\tilde{\beta}_{k|k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k|k}^{(\ell)}} \sum_{i \in L} w_{k|k}^{(i)} \beta_{k|k}^{(i)}$

$\tilde{m}_{k|k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k|k}^{(\ell)}} \sum_{i \in L} w_{k|k}^{(i)} m_{k|k}^{(i)}$

$\tilde{P}_{k|k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k|k}^{(\ell)}} \sum_{i \in L} w_{k|k}^{(i)} P_{k|k}^{(i)}$

$\tilde{\nu}_{k|k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k|k}^{(\ell)}} \sum_{i \in L} w_{k|k}^{(i)} \nu_{k|k}^{(i)}$

$\tilde{V}_{k|k}^{(\ell)} \leftarrow \frac{1}{\tilde{w}_{k|k}^{(\ell)}} \sum_{i \in L} w_{k|k}^{(i)} V_{k|k}^{(i)}$

$I \leftarrow I \setminus L$

until $I = \emptyset$

If $\ell > M$ then replace $\left\{\tilde{w}_{k|k}^{(j)}, \tilde{\alpha}_{k|k}^{(j)}, \tilde{\beta}_{k|k}^{(j)}, \tilde{m}_{k|k}^{(j)}, \tilde{P}_{k|k}^{(j)}, \tilde{\nu}_{k|k}^{(j)}, \tilde{V}_{k|k}^{(j)}\right\}_{j=1}^{\ell}$ by those of the M components with largest weights.

output: $\left\{\tilde{w}_{k|k}^{(j)}, \tilde{\xi}_{k|k}^{(j)}\right\}_{j=1}^{\ell}, \tilde{\xi}_{k|k}^{(j)} = \left(\tilde{\alpha}_{k|k}^{(j)}, \tilde{\beta}_{k|k}^{(j)}, \tilde{m}_{k|k}^{(j)}, \tilde{P}_{k|k}^{(j)}, \tilde{\nu}_{k|k}^{(j)}, \tilde{V}_{k|k}^{(j)}\right)$

A.2 Pseudocode for GGIWTPHD filter pruning, absorption and capping

Table A.2: Pseudocode for GGIWTPHD filter pruning, absorption and capping

input: GGIWT components $\left\{w_{k|k}^{(j)}, \xi_{k|k}^{(j)}\right\}_{j=1}^{J_{k|k}}$, a pruning threshold T, an absorption threshold U and a maximum number of allowed components M.

init: Set $\ell \leftarrow 0$ and $I \leftarrow \left\{i = 1, ..., J_{k|k} \middle| w_{k|k}^{(i)} > T \ \wedge \ \frac{\alpha_{k|k}^{(i)}}{\beta_{k|k}^{(i)}} > 1\right\}$

repeat

  $\ell \leftarrow \ell + 1$

  $j \leftarrow \arg\max_{i \in I} w_{k|k}^{(i)}$

  $L \leftarrow \left\{i \in I \middle| \left(m_{k|k}^{(i)} - m_{k|k}^{(j)}\right)^{\mathsf{T}} \left(P_{k|k}^{(j)}\right)^{-1} \left(m_{k|k}^{(i)} - m_{k|k}^{(j)}\right) \leq U\right\}$

  $\tilde{w}_{k|k}^{(\ell)} \leftarrow \sum_{i \in L} w_{k|k}^{(i)}$

  $\tilde{\xi}_{k|k}^{(\ell)} = \xi_{k|k}^{(j)}$

  $I \leftarrow I \setminus L$

until $I = \emptyset$

If $\ell > M$ then replace $\left\{\tilde{w}_{k|k}^{(j)}, \tilde{\alpha}_{k|k}^{(j)}, \tilde{\beta}_{k|k}^{(j)}, \tilde{m}_{k|k}^{(j)}, \tilde{P}_{k|k}^{(j)}, \tilde{\nu}_{k|k}^{(j)}, \tilde{V}_{k|k}^{(j)}\right\}_{j=1}^{\ell}$ by those of the M components with largest weights.

output: $\left\{\tilde{w}_{k|k}^{(j)}, \tilde{\xi}_{k|k}^{(j)}\right\}_{j=1}^{\ell}, \tilde{\xi}_{k|k}^{(j)} = \left(\tilde{\alpha}_{k|k}^{(j)}, \tilde{\beta}_{k|k}^{(j)}, \tilde{m}_{k|k}^{(j)}, \tilde{P}_{k|k}^{(j)}, \tilde{\nu}_{k|k}^{(j)}, \tilde{V}_{k|k}^{(j)}\right)$

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY