FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Improvement of multi-target tracker performance by on-demand external sensors
Student:	Bc. Anna Tesaříková
Supervisor:	doc. Ing. Kamil Dedecius, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2024/2025

Instructions

Abstract: The domain of multi-target tracking assumes that the models in use well correspond to reality. In particular, the measurement and clutter properties are assumed to be known, as well as the target detection and survival probabilities. However, practical scenarios often deviate from these assumptions. For instance, there may be unknown but spatially heterogeneous clutter distribution, the measurement noise covariances may differ across the field of view, and the detection and survival probabilities may be spatially inconsistent too. We can conjecture, that a deployment of a relatively cheap single target tracking device in problematic areas may provide a significant modeling performance improvement.

The goals are as follows:
1) Study the single-target tracking algorithms, e.g., the PDA filter.
2) Study the domain of multi-target tracking filters. Focus on one particular algorithm, e.g., the PHD filter.
3) Propose a method for improving the performance of the multi-target filter based on the location-specific information from the single-target filter. If possible, suggest an on-demand procedure allowing for a cheap adaptive improvement of the quality.
4) Provide an assessment of the proposed framework based on simulation examples.

References:
[1] B. N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density filter," IEEE

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 1 February 2024 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Transactions on Signal Processing, vol. 54, no. 11, pp. 4091–4104, 2006, doi: 10.1109/TSP.2006.881190.
[2] E. Brekke, Fundamentals of Sensor Fusion. NTNU, 2020.
[3] A. S. Rahmathullah, A. F. Garcia-Fernandez, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in 2017 20th International Conference on Information Fusion (Fusion), Xi'an, China: IEEE, Jul. 2017, pp. 1–8. doi: 10.23919/ICIF.2017.8009645.
[4] R. Mahler, Advances in Statistical Multisource-Multitarget Information Fusion. Artech house, 2014.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 1 February 2024 in Prague.

Master's thesis

IMPROVEMENT OF MULTI-TARGET TRACKER PERFORMANCE BY ON-DEMAND EXTERNAL SENSORS

Bc. Anna Tesaříková

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: doc. Ing. Kamil Dedecius, Ph.D.
May 2, 2024

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Tesaříková Anna. *Improvement of multi-target tracker performance by on-demand external sensors.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Contents

List of Figures

List of Tables

I would like to thank my supervisor, doc. Ing. Kamil Dedecius, Ph.D., for his guidance during writing this thesis and for introducing me to the interesting field of target tracking. I would also like to thank my cats, Václav and Zikmund, for their emotional support.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Czech Technical University in Prague has the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant of Section 60 (1) of the Act.

In Prague on May 2, 2024

Abstract

This thesis focuses on the problem of multi-target tracking, especially on the probability hypothesis density (PHD) filter, and on methods of improving its performance in regions where the sensor is affected by significant misdetections or high clutter rate. To this end, the information from on-demand external sensors is employed. We introduce three approaches to include additional information from single-target trackers – probabilistic data association (PDA) filters. We provide all required theoretical background for single-target tracking, multi-target tracking, and information fusion methods. We evaluate the performance of proposed approaches and demonstrate that they improve tracking results.

Keywords single-target tracking, multi-target tracking, probability hypothesis density filter, probabilistic data association filter, state estimation, random finite sets, information fusion

Abstrakt

Tato práce se zabývá problematikou sledování více cílů, především probability hypothesis density (PHD) filtrem, a metodami, jak zlepšit jeho výkonnost v oblastech, kde je senzor zatížen významným množstvím misdetekcí nebo vysokou intenzitou clutteru. K tomuto účelu jsou využívány informace z on-demand externích senzorů. Představujeme tři postupy, jak zahrnout dodatečné informace z algoritmů pro sledování jednoho cíle – probabilistic data association (PDA) filtrů. Poskytujeme veškeré potřebné teoretické základy pro metody sledování jednoho cíle, sledování více cílů a fúze informací. Vyhodnotíme výkonnost navrhovaných postupů a ukážeme, že zlepšují výsledky sledování.

Klíčová slova sledování jednoho cíle, sledování více cílů, probability hypothesis density filtr, probabilistic data association filtr, odhad stavů, konečné náhodné množiny, fúze informací

Acronyms

PDA	Probabilistic Data Association
PHD	Probability Hypothesis Density
GM-PHD	Gaussian Mixture PHD
KF	Kalman filter
CVM	Constant velocity model
RFS	Random finite set
CI	Covariance intersection
MM	Moment matching

Introduction

Target tracking is a research field that finds a lot of applications in numerous civilian and military fields such as border patrolling, convey protection, or aerial and underwater surveillance. It is is a process in which the state of single or multiple targets is estimated using measurements from sensors. The measurements are received at regular intervals and the states of the targets are determined at each time step. The state of a target may consist of, e.g., the target position, velocity, acceleration, and potentially other attributes.

An approach which is widely used to describe this problem is Bayesian filtering. The Bayesian filters are able to predict the current state of a target based on past measurements. These algorithms perform recursive prediction of the target state probability distribution by propagation through a state evolution model and update this distribution when new measurements are received. The fundamental algorithm for many tracking methods is the Kalman filter, which was introduced in 1960. In this thesis, the Kalman filter is described in detail.

A common problem in target tracking is the presence of clutter. Clutter is the unwanted echoes that occur when using electronic systems such as radars to obtain measurements. The clutter points are difficult to distinguish from actual measurements originating from targets and can be interpreted as false target measurements, that make the association among true targets and their measurements ambiguous. This naturally needs to be addressed by all target tracking algorithms, be it the basic ones, e.g., the probabilistic data association-based filters, or the advanced random finite sets-based filters. The single-target probabilistic data association (PDA) filter is presented in the following.

The multi-target tracking with unknown number of targets is widely recognized to be extremely difficult. It aims to estimate the number of targets as well as their states. The number of targets often varies in time due to tar-

gets appearing and disappearing and the whole task is more difficult due to the presence of clutter. This thesis describes the random finite sets formulation of this problem, especially the probability hypothesis density (PHD) filter.

The aim of this thesis is to improve the performance of the PHD filter by proposing methods to incorporate information from external sensors. Such sensors may have smaller field of view, but their resolution could be much better. They are able to track single targets only, which is a less computationally demanding task. We introduce three methods of integrating external information to the primary PHD filter and analyze their impact on its performance.

Chapter 1

Target tracking preliminaries

In this chapter, we present the state-space model, which is widely used to describe the evolution of a target state. We introduce the Kalman filter, which is one of the most important algorithms for sequential estimation.

1.1 State-space model

State-space model is a mathematical model of a physical system. It is characterized by state, which is usually marked as x_t, and output (also called measurement or observation), marked as z_t. The state contains all information of interest to the user about the current state of the system and cannot be measured directly, but can be estimated. The output can be observed. It is determined by the state x_t and can be used to estimate the state of the system. In general, both the state x_t and the output z_t can be uni- or multivariate and therefore consist of more than one element [1].

The state-space model can be either discrete – described for time steps $t = 1, 2, ...$, or continuous – described by derivatives. In the following, we will consider only the discrete model.

Usually, we describe the model by functions of state, which are known but stochastic mainly due to statistical noise,

$$x_t = f_t(x_{t-1}), \tag{1.1}$$
$$z_t = g_t(x_t), \tag{1.2}$$

where t is the time index. These functions may be time-varying or time-invariant.

If the system is linear, the equations can be written in a matrix form,

$$x_t = F_t x_{t-1} + w_t, \tag{1.3}$$
$$z_t = H_t x_t + \epsilon_t, \tag{1.4}$$

where w_t is the process noise, ϵ_t is the measurement noise, and F_t and H_t are matrices of corresponding dimensions. If the system is time-invariant, the matrices are constant, $F_t = F$, $H_t = H$. Apparently, the state equation (1.3) describes the evolution of the state from time $t-1$ to t. It is a Markov process that is not observable. The output z_t is then generated by the current state x_t. The state model (1.3)-(1.4) is therefore also known as the hidden Markov model [2].

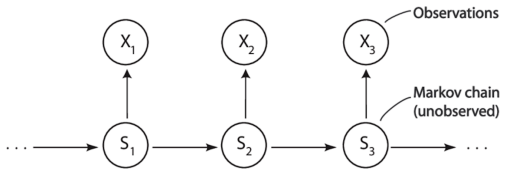

Figure 1.1 Hidden Markov model. (Source: www.dlab.berkeley.edu)

The process noise w_t and the measurement noise ϵ_t are both independent identically distributed and zero-centered. We often assume normality,

$$w_t \sim \mathcal{N}(0, Q_t), \tag{1.5}$$
$$\epsilon_t \sim \mathcal{N}(0, R_t). \tag{1.6}$$

The matrix Q_t is the process noise covariance and R_t is the measurement noise covariance. In this work, Q_t and R_t are assumed constant in time, $Q_t = Q$, $R_t = R$.

1.1.1 Example: constant velocity model

The constant velocity model is one of the most widely used state-space models. In its basic version, it describes the location of a target in two-dimensional space. The key assumption is that the true velocity of the target is known to be a constant.

The state vector is represented as

$$x_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ v_{1,t} \\ v_{2,t} \end{bmatrix} \tag{1.7}$$

and we keep track there of the longitude and latitude $x_{1,t}$, $x_{2,t}$ and the velocities $v_{1,t}$, $v_{2,t}$ in these directions. The location is modeled by the previous location at time $t-1$ and the velocities during the period Δt. The velocities are modeled as a random walk. The state evolution equations are

$$x_{1,t} = x_{1,t-1} + v_{1,t}\Delta t + w_{x_1,t}, \tag{1.8}$$
$$x_{2,t} = x_{2,t-1} + v_{2,t}\Delta t + w_{x_2,t}, \tag{1.9}$$
$$v_{1,t} = v_{1,t-1} + w_{v_1,t}, \tag{1.10}$$
$$v_{2,t} = v_{2,t-1} + w_{v_2,t}, \tag{1.11}$$

and can be written in a matrix form as

$$\underbrace{\begin{bmatrix} x_{1,t} \\ x_{2,t} \\ v_{1,t} \\ v_{2,t} \end{bmatrix}}_{x_t} = \underbrace{\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \\ v_{1,t-1} \\ v_{2,t-1} \end{bmatrix}}_{x_{t-1}} + \underbrace{\begin{bmatrix} w_{x_1,t} \\ w_{x_2,t} \\ w_{v_1,t} \\ w_{v_2,t} \end{bmatrix}}_{w_t}, \tag{1.12}$$

where

$$w_t \sim \mathcal{N}(0, Q), \tag{1.13}$$

and therefore

$$x_t \sim \mathcal{N}(Fx_{t-1}, Q). \tag{1.14}$$

A widely used form of the covariance matrix Q is

$$Q = q^2 \begin{bmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix}, \tag{1.15}$$

where q is a parameter which can be tuned.

In the same coordinate space, there are noisy measurements modeled as

$$z_{1,t} = x_{1,t} + \epsilon_{z_1,t}, \tag{1.16}$$
$$z_{2,t} = x_{2,t} + \epsilon_{z_2,t}, \tag{1.17}$$

and their matrix notation is

$$\underbrace{\begin{bmatrix} z_{1,t} \\ z_{2,t} \end{bmatrix}}_{z_t} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix}}_{x_t} + \underbrace{\begin{bmatrix} \epsilon_{z_1,t} \\ \epsilon_{z_2,t} \end{bmatrix}}_{\epsilon_t}, \tag{1.18}$$

where

$$\epsilon_t \sim \mathcal{N}(0, R), \tag{1.19}$$

and therefore

$$z_t \sim \mathcal{N}(H_t x_t, R). \tag{1.20}$$

Another examples of state-space models are the Brownian motion, removing the velocity components, and the constant acceleration model, adding the acceleration components [3].

1.2 Kalman filter

The Kalman filter is an algorithm that uses a series of measurements z_t observed over time to produce estimates of the state of the system x_t. Such estimates are usually more accurate than those based on a single measurement alone. The filter was primarily created by Rudolf E. Kálman [4], although similar algorithms occured eariler.

The filter aims to estimate the hidden states of x_t from the measurements z_t. It assumes the linear state-space model described above and the system of equations

$$x_t = F_t x_{t-1} + w_t, \tag{1.21}$$
$$z_t = H_t x_t + \epsilon_t. \tag{1.22}$$

It meets the following requirements:

- the average value of the state estimate is equal to the average value of the true state (it is not biased);
- the state estimate has the smallest possible variation;

as long as the process noise w_t and the measurement noise ϵ_t are Gaussian, the mean value of both of them is zero, and no correlation exists between them (they are independent random variables at any time step) [1].

From normality, we can see that

$$x_t \sim \mathcal{N}(F_t x_{t-1}, Q) \qquad \text{with the density } p(x_t|x_{t-1}), \tag{1.23}$$
$$z_t \sim \mathcal{N}(H_t x_t, R) \qquad \text{with the density } f(z_t|x_t), \tag{1.24}$$

and the prior distribution for x_t is also a normal distribution,

$$\pi(x_t|z_{0:t-1}) = \mathcal{N}(x_{t-1}, P_{t-1}). \tag{1.25}$$

The Kalman filter consists of two steps: prediction and update. The predict step uses the state estimate from the previous time step to produce the state estimate at the current time step, the prior state estimate. The update step combines the current measurement information with the prior state estimate and creates the posterior state estimate. These two steps typically alternate; if a measurement is unavailable, the update can be skipped and multiple predictions can be performed.

1.2.1 Prediction

In the prediction step, the model for state development (1.21) is used to create a state estimate for the current time step. The estimate is based on the posterior estimate from the previous step, which becomes the prior estimate at the current step.

We obtain the posterior distribution by combining the prior distribution with the evolution model. Using the Chapman-Kolmogorov formula, also known as the chain rule and marginalization, we see that

$$\pi(x_t|z_{0:t-1}) = \int p(x_t|x_{t-1})\pi(x_{t-1}|z_{0:t-1})dx_{t-1}. \tag{1.26}$$

As the factors on the right-hand side are Gaussian distributions, it can be shown that the result is again a Gaussian distribution $\mathcal{N}(x_{t|t-1}, P_{t|t-1})$. The predicted state estimate $x_{t|t-1}$ and the predicted estimate covariance $P_{t|t-1}$ are obtained by equations

$$x_{t|t-1} = F_t x_{t-1}, \tag{1.27}$$

$$P_{t|t-1} = F_t P_{t-1} F_t^\intercal + Q. \tag{1.28}$$

The predicted state estimate $x_{t|t-1}$ describes our knowledge about the state x_t based on its past value and its evolution model (1.21). The covariance matrix $P_{t|t-1}$ expresses our uncertainty connected with this knowledge. It is obvious that Equations (1.27) and (1.28) can be derived directly as a linear transform of the state variable described by Equation (1.25).

1.2.2 Update

The update step corrects the predicted state estimate by using a measurement z_t. It is a Bayesian update and the Bayes formula is used in the following way

$$\pi(x_t|z_{0:t}) \propto f(z_t|x_t)\pi(x_t|z_{0:t-1}). \tag{1.29}$$

The updated state estimate x_t and the updated estimate covariance P_t are expressed as follows,

$$x_t = x_{t|t-1} + K_t \tilde{z}_t, \tag{1.30}$$

$$P_t = (I - K_t H_t) P_{t|t-1}, \tag{1.31}$$

where $\tilde{z}_t$ is the innovation and K_t is the Kalman gain,

$$\tilde{z}_t = z_t - H_t x_{t|t-1}, \tag{1.32}$$
$$K_t = P_{t|t-1} H_t^\intercal S_t^{-1}, \tag{1.33}$$

and S_t is the innovation covariance,

$$S_t = H_t P_{t|t-1} H_t^\intercal + R. \tag{1.34}$$

These formulas are valid for the optimal Kalman gain which minimizes the mean-square error. For greater gain, more emphasis is placed on new measurements. The filter is more sensitive, but the filtration of noise is worse.

1.2.3 Remarks

The standard Kalman filter assumes the state-space model to be linear. But if any or both of the functions

$$x_t = f_t(x_{t-1}), \tag{1.35}$$
$$z_t = g_t(x_t), \tag{1.36}$$

are nonlinear, there are still ways to perform filtering. In case of weak nonlinearity, the extended Kalman filter (EKF) [5] can be used. The EKF performs linearization of the functions using derivatives and is not optimal, but usually works well and is a standard in the GPS navigation. In case of stronger nonlinearity, the unscented Kalman filter (UKF) [6] can be used. The UKF uses sigma points to estimate the state. Antoher option is to use Monte Carlo methods, e.g., the particle filter [7].

Chapter 2
Single-target tracking

In single-target tracking, we only focus on one target of interest and we use sensor data to estimate its state. In the previous chapter, we described the Kalman filter which is capable of tracking one target. However, a frequent problem in target tracking is cluttered environment and the resulting uncertainity about the target-measurement association.

The uncertainty, which measurement was produced by the tracked object (target) and which are false alarms (clutter), has attracted considerable focus in the engineering community in the past five decades, giving rise to a multitude of algorithms. Generally, two branches can be distinguished. First, the association-based, explicitly enumerating possible target-measurement associations. The nearest neighbor [8], multiple hypothesis tracking (MHT) filter [9], and the probabilistic data association (PDA) filter [10] belong to this branch. The second branch is based on random finite set (RFS) setting of the problem. The probability hypothesis density (PHD) filter [11], multi-Bernoulli mixture (MBM) filter [12], and Poisson multi-Bernoulli mixture (PMBM) filter [13] belong to the second branch.

In this chapter, we describe clutter and its mathematical models. We focus on the PDA filter and its ability to perform target tracking in cluttered environment.

2.1 Clutter

The term ”clutter” (or false alarms, false measurements) refers to all unwanted returns in electronic systems, for instance, radars. Similarly, unwanted return occurs in tracking of objects in video, or lidar-based surveillance. Its nature varies with application and radar parameters; what we consider clutter in one case may be a desired target in another. Clutter is usually returned from land,

sea, rain, or atmospheric turbulences. In radar systems, it can cause serious performance issues, because it could be confused with targets.

Clutter may have many varieties and many models or models with numerous parameters are needed to represent them. One of the most common models for clutter is a set of independent and identically distributed random variables with uniform spatial distribution. The probability density funciton of such clutter over the observation volume V is

$$p_0(z) = p(z|z \text{ is a false measurement}) = \frac{1}{V}. \tag{2.1}$$

The number of clutter points obeys a Poisson distribution,

$$P_{\text{FA}}(n) = e^{-\lambda V} \frac{(\lambda V)^n}{n!}, \tag{2.2}$$

where λ is the spatial density of the false alarms (number of false alarms per unit volume) [14].

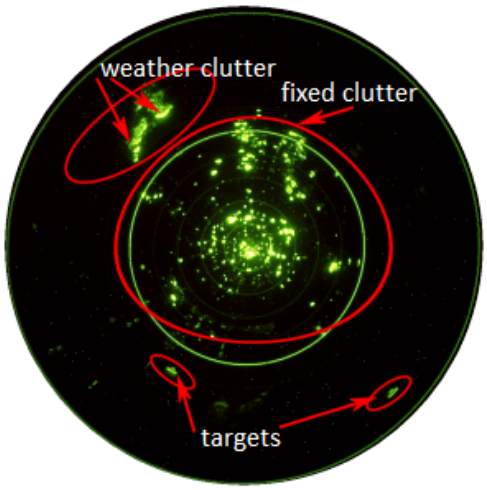

Figure 2.1 Example of radar clutter. (Source: www.radartutorial.eu)

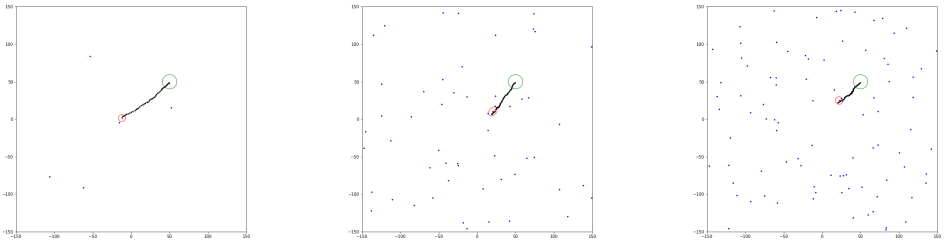

Figure 2.2 Clutter in simulation for $\lambda = 0.0001$, $\lambda = 0.0005$, and $\lambda = 0.001$.

2.2 PDA filter

The PDA filter is based on Kalman filter, or extended Kalman filter in case of nonlinear state or measurement equations (we leave this filter behind the scope of this thesis). At each time step, it calculates the association probabilities to the tracked target for each validated measurement. This process usually takes place when the measurements have origin uncertainty due to false alarms, clutter, or interfering targets. This section is mostly inspired by [10] and [15].

2.2.1 Assumptions

There are some assumptions which make it possible to create a state-estimation scheme that is almost as simple as the Kalman filter, but more effective in clutter.

1. Only one target of interest is present and its state x evolves in time by the equation

$$x_t = F_t x_{t-1} + w_t, \qquad w_t \sim \mathcal{N}(0, Q), \tag{2.3}$$

and the true, target-originated measurement is given by

$$z_t = H_t x_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, R), \tag{2.4}$$

where Q is the process noise covariance matrix and R is the measurement noise covariance matrix.

2. The track of the target has been initialized.

3. The target's past information through time $t-1$ is approximately summarized by the prior probability density function

$$p(x_{t-1}|Z_{0:t-1}) = \mathcal{N}(x_{t-1}; \hat{x}_{t-1}, P_{t-1}), \tag{2.5}$$

where $\hat{x}_{t-1}$ is the state estimate and $Z_{0:t-1}$ is the set of all past measurements.

4. At each time step, a measurement validation region is used for the selection of candidate measurements. The region is set up around the predicted measurements and the selected candidates are associated to the target of interest.

5. At most one of the validated measurements can be target originated.

6. The remaining measurements originate from false alarms or clutter. The clutter is modeled as a set of independent and identically distributed random variables with uniform spatial distribution; the number of clutter points obeys a Poisson distribution or a diffuse prior.

7. The target detections occur independently over time. The detection probability P_D is known.

2.2.2 Algorithm

A cycle of the PDA filter is similar to Kalman filter, but contains these additional features for update step:

1. Candidate measurements are selected using the measurement validation region.
2. An association probability is computed for each measurement. This probability is used as the weighting of this measurement in the combined innovation.
3. The final updated state covariance takes into account the measurement origin uncertainty.

2.2.3 Prediction

As in the standard Kalman filter, the state vector $\hat{x}$ and the covariance matrix P are predicted in time from $t-1$ to t,

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1}, \tag{2.6}$$
$$P_{t|t-1} = F_t P_{t-1} F_t^\intercal + Q. \tag{2.7}$$

Also the predicted measurement $\hat{z}_{t|t-1}$ and the innovation covariance matrix S_t are computed,

$$\hat{z}_{t|t-1} = H_t \hat{x}_{t|t-1}, \tag{2.8}$$
$$S_t = H_t P_{t|t-1} H_t^\intercal + R. \tag{2.9}$$

The matrix S_t describes the spread of the distribution, where the measurement is expected to occur, based on the uncertain knowledge about the state and the measurement noise properties.

2.2.4 Measurement validation

The validation region is the ellipsoidal gating region given by

$$\{z : (z - \hat{z}_{t|t-1})^\intercal S_t^{-1} (z - \hat{z}_{t|t-1}) \leq \gamma\}, \qquad \gamma > 0, \tag{2.10}$$

where the gate threshold γ corresponds to the gate probability P_G. The probability P_G is the probability that the gate contains the true measurement if detected. Only the measurements lying inside the gate are taken for the update step.

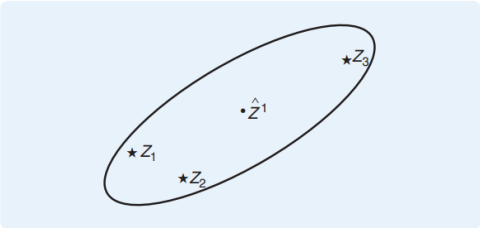

Figure 2.3 Example of validation region around a predicted measurement [15].

2.2.5 Data association

The association probability of the parametric PDA with the Poisson clutter model with spatial density λ for z_i being the correct measurement is given by

$$\beta_{i,t} = \begin{cases} \frac{\mathcal{L}_{i,t}}{1-P_D P_G+\sum_{j=1}^{m_t}\mathcal{L}_{j,t}} & i = 1, ..., m_t, \\ \frac{1-P_D P_G}{1-P_D P_G+\sum_{j=1}^{m_t}\mathcal{L}_{j,t}} & i = 0, \end{cases} \tag{2.11}$$

where $i = 0$ means that not a single measurement has been originated from the target, P_D is the target detection probability, P_G is the gate probability, and m_t is the number of measurements in the validation region. The likelihood ratio of the ith measurement originating from the target rather than from the clutter $\mathcal{L}_{i,t}$ is given by

$$\mathcal{L}_{i,t} = \frac{\mathcal{N}(z_{i,t}; \hat{z}_{t|t-1}, S_t) P_D}{\lambda}. \tag{2.12}$$

2.2.6 Update

The update state equation of the PDA filter is

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t \nu_t, \tag{2.13}$$

where K_t is the gain, same as in the standard Kalman filter, and ν_t is the combined innovation,

$$\nu_t = \sum_{i=1}^{m_t} \beta_{i,t} \nu_{i,t}. \tag{2.14}$$

The covariance of the updated state is given by

$$P_t = \beta_{0,t} P_{t|t-1} + (1 - \beta_{0,t}) P_t^C + \tilde{P}_t, \tag{2.15}$$

where P_t^C is the covariance of the state updated with the correct measurement and $\tilde{P}_t$ is the spread of the innovation term,

$$P_t^C = P_{t|t-1} + K_t S_t K_t^\intercal, \tag{2.16}$$

$$\tilde{P}_t = K_t \left(\sum_{i=1}^{m_t} \beta_{i,t} \nu_{i,t} \nu_{i,t}^\intercal - \nu_t \nu_t^\intercal \right) K_t^\intercal. \tag{2.17}$$

With probability $\beta_{0,t}$, there is no correct measurement for updating, so there is no update of the state estimate and the covariance from prediction step $P_{t|t-1}$ is used with weighting $\beta_{0,t}$. With probability $1-\beta_{0,t}$, there is an available correct measurement and the updated covariance P_t^C is used with weighting $1-\beta_{0,t}$. The term $\tilde{P}_t$ increases the covariance of the updated state, since we do not know which measurement is correct. It can be written as a merge of multiple posterior distributions which are generated from the predicted distribution by each measurement in the validation region and weighted by probabilities $\beta_{i,t}$.

Chapter 3

Multi-target tracking

In the previous chapter, we discussed target tracking in an environment where only one target of interest is present. In this chapter, we discuss tracking of multiple targets. In a multiple-target environment, we use sensor data to estimate not only the states of the targets, but also the number of targets which varies in time due to targets appearing and disappearing. Moreover, the observations usually contain clutter, which cannot be simply distinguished from target-originated measurements.

The main problem in multi-target tracking is to decide which measurements originate from which targets or whether they are false alarms. Due to its combinational nature, the data association problem creates the most of the computational load of multi-target tracking algorithms. Many traditional approaches are based on explicit target-measurement associations, such as the joint probabilistic data association (JPDA) filter [10] and the joint integrated probabilistic data assocaition (JIPDA) filter [16]. Alternative formulations of the problem avoid explicit associations. In this chapter, we discuss the random finite sets (RFS) approach, especially the PHD filter. This chapter is mainly inspired by [17] and [18].

3.1 Derivation of PHD filter

In this section, we present a formulation of multi-target filtering in the random finite set approach. We derive all the important concepts of the PHD filter.

3.1.1 Multiple target scenario

Let M_t be the number of targets at time t. At time $t-1$, the target states are $x_{1,t-1}, ..., x_{M_{t-1},t-1} \in \mathcal{X}$, where $\mathcal{X}$ is the set of all states. At the next time step, there are three possibilities of what may happen to a target:

1. The target may survive and evolve to its new state. We model this situation using a probability $p_{S,t}(x_{t-1})$, which is the probability of the target survival. The probability density of a transition to state x_t at time t is $f_{t|t-1}(x_t|x_{t-1})$.
2. The target may die. This happens with probability $1 - p_{S,t}(x_{t-1})$.
3. A new target may appear. This can be either by a spontaneous birth (independent of any existing target) or by spawning from another target. Modeling of target birth may vary in different applications of the PHD filter, but in case of this thesis, we consider that a target can only appear in a location which we know in advance.

The result of this process is M_t new targets with states $x_{1,t}, ..., x_{M_t,t} \in \mathcal{X}$. In the RFS multi-target model formulation, their order has no significance.

At time t, N_t measurements $z_{1,t}, ..., z_{N_t,t} \in \mathcal{Z}$, where $\mathcal{Z}$ stands for the set of all measurements, are returned by the sensor. In general, only some of these measurements are target-originated and cannot be distinguished from false alarms. Even if all measurements are target-originated and all targets were detected by the sensor, we still have no information about which measurement comes from which target. Due to the measurements' origin uncertainty, the order in which they appear is not important. The aim od multi-target tracking is to jointly estimate the number of targets and the states of these targets from such observations.

Since there is no significant ordering of target states and measurements, they can be represented as finite sets,

$$X_t = \{x_{1,t}, ..., x_{M_t,t}\} \in \mathcal{F}(\mathcal{X}), \tag{3.1}$$
$$Z_t = \{z_{1,t}, ..., z_{N_t,t}\} \in \mathcal{F}(\mathcal{Z}), \tag{3.2}$$

where $\mathcal{F}(\mathcal{X})$ and $\mathcal{F}(\mathcal{Z})$ are the respective collections of all finite subsets of $\mathcal{X}$ and $\mathcal{Z}$.

3.1.2 Random finite set formulation

In the random finite set formulation, the target set X_t and the measurement set Z_t are treated as a multi-target state and a multi-target observation. The multi-target tracking problem can then be stated as a filtering problem with multi-target state space $\mathcal{F}(\mathcal{X})$ and observation space $\mathcal{F}(\mathcal{Z})$. The uncertainty in multi-target system is expressed by modeling X_t and Z_t as random finite sets.

A random finite set is a random variable that takes values as unordered sets with a finite number of elements. The number of elements is random as well as the elements themselves. It can be described by a discrete distribution

which characterizes the cardinality (number of elements) and a family of joint distributions which specifies the distribution of the elements depending on the cardinality [19].

In the RFS model for the time evolution of the multi-target state, the X_t is created by merging three RFSs: the set of surviving targets from previous time step, the set of spawn targets, and the set of newborn targets.

The set of surviving targets is formed by computing a RFS $S_{t|t-1}(x_{t-1})$ for each $x_{t-1} \in X_{t-1}$ at time $t-1$. This RFS can be either $\{x_t\}$ if the target survives, or $\emptyset$ if the target dies. The computation of the set of spawn targets $B_{t|t-1}(x_{t-1})$ is analogous. The set of newborn targets Γ_t is created by adding a target to all birthplaces.

The multi-target state X_t is then given by

$$X_t = \left[\bigcup_{\zeta \in X_{t-1}} S_{t|t-1}(\zeta) \right] \cup \left[\bigcup_{\zeta \in X_{t-1}} B_{t|t-1}(\zeta) \right] \cup \Gamma_t. \tag{3.3}$$

It is assumed that the RFSs from (3.3) are independent of each other.

The RFS model of measurements is similar. It takes into account detection uncertainity and clutter. A target $x_t \in X_t$ is either detected with probability $p_{D,t}(x_t)$, or misdetedted with probability $1 - p_{D,t}(x_t)$. The probability density of receiving an observation z_t from state x_t is $g_t(z_t|x_t)$. For each state x_t, the RFS $\Theta_t(x_t)$ is computed. This RFS can be either $\{z_t\}$ if the target is detected, or $\emptyset$ if the target is misdetected. In addition to the target-originated measurements, the sensor may also receive a set of clutter K_t.

Given the multi-target state X_t, the multi-target measurement Z_t can be written as

$$Z_t = K_t \cup \left[\bigcup_{x \in X_t} \Theta_t(x) \right]. \tag{3.4}$$

The RFSs from (3.4) are also assumed to be independent of each other.

3.1.3 Multi-target Bayes filter

The optimal multi-target Bayes filter propagates the multi-target posterior density $p_t(.|Z_{1:t})$ in time via the recursion

$$p_{t|t-1}(X_t|Z_{1:t-1}) = \int f_{t|t-1}(X_t|X)p_{t-1}(X|Z_{1:t-1})\mu_s(dX), \qquad \text{(prediction)} \tag{3.5}$$

$$p_t(X_t|Z_{1:t}) = \frac{g_t(Z_t|X_t)p_{t|t-1}(X_t|Z_{1:t-1})}{\int g_t(Z_t|X)p_{t|t-1}(X|Z_{1:t-1})\mu_s(dX)}, \qquad \text{(update)} \tag{3.6}$$

where $f_{t|t-1}(.|.)$ is the multi-target transition density, $g_t(.|.)$ is the multi-target likelihood, and μ_s is an appropriate reference measure on $\mathcal{F}(\mathcal{X})$.

3.2 PHD filter

Due to the computational complexity of recursion (3.5) and (3.6), which involves multiple integrals on the space $\mathcal{F}(\mathcal{X})$, an approximation of the optimal multi-target Bayes filter is needed. The PHD filter provides a sub-optimal strategy and propagates the posterior intensity, a first-order statistical moment of the posterior multi-target state, in time instead of the full multi-target posterior density.

3.2.1 Intensity

For a RFS X on $\mathcal{X}$ with probability distribution P, the intensity (also known as the probability hypothesis density) is a nonnegative function v on $\mathcal{X}$ such that for each region $S \subseteq \mathcal{X}$

$$\int |X \cap S| P(dX) = \int_S v(x)dx, \tag{3.7}$$

where $|.|$ stands for the cardinality of the set. This means that when v is integrated over any region S, it gives the expected number of targets from X that are in S. The estimates for elements of X can be created using the local maxima of v, which are points in $\mathcal{X}$ where the local concentration of expected number of elements is the highest. One of the possible approaches to estimate the number of targets $\hat{N}$ in X is to choose the intensity peaks that exceed a predetermined threshold.

An important class of RFSs are the Poisson RFSs which can be completely described by their intensities. A RFS X is Poisson if its cardinality distribution is Poisson with mean $\hat{N}$ and for any finite cardinality, its elements are independently identically distributed. Because the intensities of Poisson RFSs fully characterize them, they can be used as prior and posterior distributions.

3.2.2 PHD recursion

For the PHD filter, the following assumptions are made:

1. Each target evolves and generates observations independently of one another.
2. Clutter is Poisson and independent of target-originated measurements.
3. The predicted multi-target RFS is Poisson.

Under these assumptions, the PHD recursion, using the prediction and update step, is defined as

$$v_{t|t-1}(x) = \int p_{S,t}(\zeta) f_{t|t-1}(x|\zeta) v_{t-1}(\zeta) d\zeta + \int \beta_{t|t-1}(x|\zeta) v_{t-1}(\zeta) d\zeta + \gamma_t(x), \tag{3.8}$$

$$v_t(x) = (1 - p_{D,t}(x)) v_{t|t-1}(x) + \sum_{z \in Z_t} \frac{p_{D,t}(x) g_t(z|x) v_{t|t-1}(x)}{\kappa_t(z) + \int p_{D,t}(\xi) g_t(z|\xi) v_{t|t-1}(\xi)}, \tag{3.9}$$

where v_t and $v_{t|t-1}$ are the respective intensities associated with the multi-target posterior density p_t and predicted density $p_{t|t-1}$, and

- $\gamma_t(.)$ is the intensity of the birth RFS Γ_t;
- $\beta_{t|t-1}(.|\zeta)$ is the intensity of the RFS $B_{t|t-1}(\zeta)$ spawned by a target with previous state ζ;
- $p_{S,t}(\zeta)$ is the probability of survival of a target with previous state ζ;
- $p_{D,t}(x)$ is the probability of detection of a target with state x;
- $\kappa_t(.)$ is the intensity of the clutter RFS K_t.

From (3.8) and (3.9) can be seen that the PHD filter avoids the combinatorial computations which result from the unknown association of measurements with corresponding targets. The PHD recursion is also less computationally intensive than (3.5) and (3.6) since it operates on the single-target state space $\mathcal{X}$ instead of $\mathcal{F}(\mathcal{X})$. In general, the PHD recursion unfortunately does not have closed-form solutions, and numerical integration suffers from the "curse of dimensionality".

3.3 GM-PHD

For a certain class of multi-target models, the closed-form solution of the PHD recursion can be found. These models are the linear Gaussian multi-target models. For these models, an efficient multi-target tracking algorithm can be developed.

3.3.1 Linear Gaussian multi-target model

In addition to the PHD recursion assumptions in Section 3.2.2, the linear Gaussian multi-target model requires the following three assumptions on the birth, death, detection, and evolution of targets:

1. Each target follows a linear Gaussian dynamical model and the sensor has a linear Gaussian measurement model,

$$f_{t|t-1}(x|\zeta) = \mathcal{N}(x; F_{t-1}\zeta, Q_{t-1}), \tag{3.10}$$

$$g_t(z|x) = \mathcal{N}(z; H_t x, R_t), \tag{3.11}$$

   where $\mathcal{N}(.; m, P)$ is a Gaussian density with mean m and covariance P, F_{t-1} is the state transition matrix, H_t is the observation matrix, and Q_{t-1} and R_t are the process noise and measurement noise covariances.

2. The probabilities of survival and detection are state independent,

$$p_{S,t}(x) = p_{S,t}, \tag{3.12}$$

$$p_{D,t}(x) = p_{D,t}. \tag{3.13}$$

3. The intensities of the birth and spawn RFSs are modeled as Gaussian mixtures of the form

$$\gamma_t(x) = \sum_{i=1}^{J_{\gamma,t}} w_{\gamma,t}^{(i)} \mathcal{N}(x; m_{\gamma,t}^{(i)}, P_{\gamma,t}^{(i)}), \tag{3.14}$$

$$\beta_{t|t-1}(x|\zeta) = \sum_{i=1}^{J_{\beta,t}} w_{\beta,t}^{(i)} \mathcal{N}(x; F_{\beta,t-1}^{(i)}\zeta + d_{\beta,t-1}^{(i)}, Q_{\beta,t-1}^{(i)}), \tag{3.15}$$

   where $J_{\gamma,t}$, $w_{\gamma,t}^{(i)}$, $m_{\gamma,t}^{(i)}$, $P_{\gamma,t}^{(i)}$, $i = 1, ..., J_{\gamma,t}$ are model parameters which determine the shape of the birth intensity, and similarly $J_{\beta,t}$, $w_{\beta,t}^{(i)}$, $F_{\beta,t-1}^{(i)}$, $d_{\beta,t-1}^{(i)}$, $Q_{\beta,t-1}^{(i)}$, $i = 1, ..., J_{\gamma,t}$ determine the shape of the spawning intensity of a target with previous state ζ.

3.3.2 Prediction

Suppose that we have a posterior intensity v_{t-1} from previous time step $t-1$. This posterior intensity is a Gaussian mixture,

$$v_{t-1}(x) = \sum_{i=1}^{J_{t-1}} w_{t-1}^{(i)} \mathcal{N}(x; m_{t-1}^{(i)}, P_{t-1}^{(i)}). \tag{3.16}$$

The predicted intensity at time t is then also a Gaussian mixture and is given by

$$v_{t|t-1}(x) = v_{S,t|t-1}(x) + v_{\beta,t|t-1}(x) + \gamma_t(x), \tag{3.17}$$

where the birth intensity γ_t is described in (3.14). The intensity $v_{S,t|t-1}$ is the intensity of the surviving targets and is given by

$$v_{S,t|t-1}(x) = p_{S,t} \sum_{j=1}^{J_{t-1}} w_{t-1}^{(j)} \mathcal{N}(x; m_{S,t|t-1}^{(j)}, P_{S,t|t-1}^{(j)}), \tag{3.18}$$

where

$$m_{S,t|t-1}^{(j)} = F_{t-1} m_{t-1}^{(j)}, \tag{3.19}$$

$$P_{S,t|t-1}^{(j)} = Q_{t-1} + F_{t-1} P_{t-1}^{(j)} F_{t-1}^{\intercal}. \tag{3.20}$$

The intensity $v_{\beta,t|t-1}$ is the spawn intensity and is computed as

$$v_{\beta,t|t-1}(x) = \sum_{j=1}^{J_{t-1}} \sum_{l=1}^{J_{\beta,t}} w_{t-1}^{(j)} w_{\beta,t}^{(l)} \mathcal{N}(x; m_{\beta,t|t-1}^{(j,l)}, P_{\beta,t|t-1}^{(j,l)}), \tag{3.21}$$

where

$$m_{\beta,t|t-1}^{(j,l)} = F_{\beta,t-1}^{(l)} m_{t-1}^{(j)} + d_{\beta,t-1}^{(l)}, \tag{3.22}$$

$$P_{\beta,t|t-1}^{(j,l)} = Q_{\beta,t-1}^{(l)} + F_{\beta,t-1}^{(l)} P_{\beta,t-1}^{(j)} (F_{\beta,t-1}^{(l)})^{\intercal}. \tag{3.23}$$

The mean of the predicted number of targets is

$$\hat{N}_{t|t-1} = \hat{N}_{t-1} \left(p_{S,t} + \sum_{j=1}^{J_{\beta,t}} w_{\beta,t}^{(j)} \right) + \sum_{j=1}^{J_{\gamma,t}} w_{\gamma,t}^{(j)}. \tag{3.24}$$

3.3.3 Update

From the prediction step, we have the predicted intensity $v_{t|t-1}$. The posterior intensity v_t at time t is a Gaussian mixture given by

$$v_t(x) = (1 - p_{D,t}) v_{t|t-1}(x) + \sum_{z \in Z_t} v_{D,t}(x; z). \tag{3.25}$$

The equation consists of a misdetection term $(1 - p_{D,t}) v_{t|t-1}(x)$ and $|Z_t|$ detection terms $v_{D,t}(x; z)$. The intensity $v_{D,t}$ is the intensity of one detected measurement $z \in Z_t$ and is computed as

$$v_{D,t}(x; z) = \sum_{j=1}^{J_{t|t-1}} w_t^{(j)}(z) \mathcal{N}(x; m_{t|t}^{(j)}(z), P_{t|t}^{(j)}), \tag{3.26}$$

where

$$w_t^{(j)}(z) = \frac{p_{D,t} w_{t|t-1}^{(j)} q_t^{(j)}(z)}{\kappa_t(z) + p_{D,t} \sum_{l=1}^{J_{t|t-1}} w_{t|t-1}(l) q_t^{(l)}(z)}, \tag{3.27}$$

$$m_{t|t}^{(j)}(z) = m_{t|t-1}^{(j)} + K_t^{(j)}(z - H_t m_{t|t-1}^{(j)}), \tag{3.28}$$

$$P_{t|t}^{(j)} = (I - K_t^{(j)} H_t) P_{t|t-1}^{(j)}, \tag{3.29}$$

$$K_t^{(j)} = P_{t|t-1}^{(j)} H_t^\intercal (H_t P_{t|t-1}^{(j)} H_t^\intercal + R_t)^{-1}. \tag{3.30}$$

The mean of the updated number of targets is

$$\hat{N}_t = \hat{N}_{t|t-1}(1 - p_{D,t}) + \sum_{z \in Z_t} \sum_{j=1}^{J_{t|t-1}} w_t^{(j)}(z). \tag{3.31}$$

It is worth noting that the recursions for means and covariances of $v_{S,t|t-1}$ and $v_{\beta,t|t-1}$ are Kalman predictions as well as the recursions for means and covariances of $v_{D,t}$ are Kalman updates.

3.3.4 Pruning

The number of Gaussian components in the posterior intensities, as well as the computation time, increases quickly and some pruning procedure is therefore needed. The number of Gaussian components can be reduced by discarding those components with weights lower than some predetermined threshold, or by keeping only a certain number of components with highest weights. Some of the Gaussian components are also close together enough to be merged into a single component.

3.3.5 State estimation

After computing the posterior intensity v_t, the multi-target state estimates have to be extracted. In the Gaussian mixture representation, the extraction is straightforward, since the means of the Gaussian components are the local maxima of v_t. One idea is to select the $\hat{N}_t$ highest peaks of v_t, but this may result in state estimates corresponding to Gaussian components with low weights, because the height of each peak depends on both the weight and covariance. A better alternative is to select the components that have weights higher than some predetermined threshold.

Algorithm 1 Pseudocode for the prediction step of the GM-PHD filter

given $\left\{w_{t-1}^{(i)}, m_{t-1}^{(i)}, P_{t-1}^{(i)}\right\}_{i=1}^{J_{t-1}}$.

step 1. (prediction for newborn targets)
$i = 0$.
for $j = 1, ..., J_{\gamma,t}$:
$i := i + 1$.
$w_{t|t-1}^{(i)} = w_{\gamma,t}^{(j)}$.
$m_{t|t-1}^{(i)} = m_{\gamma,t}^{(j)}$.
$P_{t|t-1}^{(i)} = P_{\gamma,t}^{(j)}$.
end
step 2. (prediction for spawned targets)
for $j = 1, ..., J_{\beta,t}$:
for $l = 1, ..., J_{t-1}$:
$i := i + 1$.
$w_{t|t-1}^{(i)} = w_{t-1}^{(l)} w_{\beta,t}^{(j)}$.
$m_{t|t-1}^{(i)} = d_{\beta,t-1}^{(j)} + F_{\beta,t-1}^{(j)} m_{t-1}^{(l)}$.
$P_{t|t-1}^{(i)} = Q_{\beta,t-1}^{(j)} + F_{\beta,t-1}^{(j)} P_{t-1}^{(l)} (F_{\beta,t-1}^{(j)})^{\intercal}$.
end
end
step 3. (prediction for existing targets)
for $j = 1, ..., J_{t-1}$:
$i := i + 1$.
$w_{t|t-1}^{(i)} = p_{S,t} w_{t-1}^{(j)}$.
$m_{t|t-1}^{(i)} = F_{t-1} m_{t-1}^{(j)}$.
$P_{t|t-1}^{(i)} = Q_{t-1} + F_{t-1} P_{t-1}^{(j)} F_{t-1}^{\intercal}$.
end
$J_{t|t-1} = i$.

output $\left\{w_{t|t-1}^{(i)}, m_{t|t-1}^{(i)}, P_{t|t-1}^{(i)}\right\}_{i=1}^{J_{t|t-1}}$.

Algorithm 2 Pseudocode for the update step of the GM-PHD filter

given $\left\{w_{t|t-1}^{(i)}, m_{t|t-1}^{(i)}, P_{t|t-1}^{(i)}\right\}_{i=1}^{J_{t|t-1}}$, and the measurement set Z_t.

step 1. (construction of PHD update components)
for $j = 1, ..., J_{t|t-1}$:
$\eta_{t|t-1}^{(j)} = H_t m_{t|t-1}^{(j)}$.
$S_t^{(j)} = R_t + H_t P_{t|t-1}^{(j)} H_t^\intercal$.
$K_t^{(j)} = P_{t|t-1}^{(j)} H_t^\intercal (S_t^{(j)})^{-1}$.
$P_{t|t} = (I - K_t^{(j)} H_t) P_{t|t-1}^{(j)}$.
end
step 2. (update)
for $j = 1, ..., J_{t|t-1}$:
$w_t^{(j)} = (1 - p_{D,t}) w_{t|t-1}^{(j)}$.
$m_t(j) = m_{t|t-1}^{(j)}$.
$P_t^{(j)} = P_{t|t-1}^{(j)}$.
end
$l := 0$.
for each $z \in Z_t$:
$l := l + 1$.
for $j = 1, ..., J_{t|t-1}$:
$w_t^{(lJ_{t|t-1}+j)} = p_{D,t} w_{t|t-1}^{(j)} \mathcal{N}(z; \eta_{t|t-1}^{(j)}, S_t^{(j)})$.
$m_t^{(lJ_{t|t-1}+j)} = m_{t|t-1}^{(j)} + K_t^{(j)}(z - \eta_{t|t-1}^{(j)})$.
$P_t^{(lJ_{t|t-1}+j)} = P_{t|t}^{(j)}$.
end
for $j = 1, ..., J_{t|t-1}$:
$w_t^{(lJ_{t|t-1}+j)} := \frac{w_t^{(lJ_{t|t-1}+j)}}{\kappa_t(z) + \sum_{i=1}^{J_{t|t-1}} w_t^{(lJ_{t|t-1}+i)}}$.
end
end
$J_t = lJ_{t|t-1} + J_{t|t-1}$.

output $\left\{w_t^{(i)}, m_t^{(i)}, P_t^{(i)}\right\}_{i=1}^{J_t}$.

Algorithm 3 Pruning for the GM-PHD filter

given $\left\{w_t^{(i)}, m_t^{(i)}, P_t^{(i)}\right\}_{i=1}^{J_t}$, a truncation threshold T, a merging threshold U, and a maximum allowable number of Gaussian terms J_{max}.

$l := 0$.
$I = \{i = 1, ..., J_t \mid w_t^{(i)} > T\}$.
repeat
  $l := l + 1$.
  $j := \arg\max_{i \in I} w_t^{(i)}$.
  $L := \left\{i \in I \mid (m_t^{(i)} - m_t^{(j)})^{\intercal} (P_t^{(i)})^{-1} (m_t^{(i)} - m_t^{(j)}) \leq U\right\}$.
  $\tilde{w}_t^{(l)} = \sum_{i \in L} w_t^{(i)}$.
  $\tilde{m}_t^{(l)} = \frac{1}{\tilde{w}_t^{(l)}} \sum_{i \in L} w_t^{(i)} m_t^{(i)}$.
  $\tilde{P}_t^{(l)} = \frac{1}{\tilde{w}_t^{(l)}} \sum_{i \in L} w_t^{(i)} (P_t^{(i)} + (\tilde{m}_t^{(l)} - m_t^{(i)})(\tilde{m}_t^{(l)} - m_t^{(i)})^{\intercal})$.
  $I := I \setminus L$.
until $I = \emptyset$.
if $l > J_{max}$ **then** replace $\left\{\tilde{w}_t^{(i)}, \tilde{m}_t^{(i)}, \tilde{P}_t^{(i)}\right\}_{i=1}^{l}$ by those of the J_{max} Gaussians with largest weights.

output $\left\{\tilde{w}_t^{(i)}, \tilde{m}_t^{(i)}, \tilde{P}_t^{(i)}\right\}_{i=1}^{l}$ as pruned Gaussian components.

Algorithm 4 Multi-target state extraction for GM-PHD filter

given $\left\{w_t^{(i)}, m_t^{(i)}, P_t^{(i)}\right\}_{i=1}^{J_t}$.

$\hat{X}_t := \emptyset$.
for $i = 1, ..., J_t$:
  if $w_t^{(i)} > 0.5$,
    for $j = 1, ..., \text{round}(w_t^{(i)})$:
      $\hat{X}_t := \left[\hat{X}_t, m_t^{(i)}\right]$.
    end
  end
end

output $\hat{X}_t$ as the multi-target state estimate.

Chapter 4

Information fusion

In general, information fusion is a term that describes the combination of data from multiple sources to produce more consistent and useful information and reduce redundancy and uncertainty. It has many subsets such as data fusion or image fusion. In this chapter, we discuss another subset called sensor fusion.

In many civilian and military fields, a large number of various sensors is widely used. Processing the information collected by them separately and in isolation would discard all the internal connection between the information of each sensor and may eventually lead to the loss of key information. The current research therefore focuses on the advantages of various sensors, on sensor networks, and on methods of combining information collected by multiple sensors.

A sensor network consists of a number of spatially distributed and interconnected sensor nodes. These nodes are able to process information and communicate with each other [20]. An important application of sensor networks is collaborative target tracking, which is the subject of this thesis.

The multi-sensor fusion makes full use of the information collected by each sensor, analyzes this information to obtain a consistent and more accurate description of the measured object, and creates the corresponding decision and estimation. The aim of multi-sensor fusion is to improve the performance of the total sensor system and eliminate the limitations of individual sensors [21].

The multi-sensor fusion can be centralized or distributed. Centralized fusion is optimal and easier to implement, but may bring significant communication overhead and has a single point of failure which makes it vulnerable. Distributed fusion is more robust to failure and more flexible and scalable, but unless the network is synchronous and tree-connected, it is difficult to perform the fusion in a consistent manner [22].

This thesis focuses on the use of on-demand external sensors to improve the performance of a multi-target tracking algorithm. These external sensors provide data to PDA filters, which can only track one individual target. Therefore, we need to combine only the state estimates of a single target. In this chapter, we introduce the methods of state estimate fusion.

4.1 Normal mixture model

In many areas of applied statistics, data for which two or more normal distributions are mixed appear frequently. A state estimate of a target in a multi-sensor system can be also seen as a mixture of several normal distributions given by individual sensors.

4.1.1 Finite mixture model

At first, let us look at the finite mixture model for univariate random variable Y. It is assumed that the distribution of Y is a mixture of K univariate normal distributions. The density of this distribution is given by

$$p(y|\boldsymbol{\vartheta}) = \omega_1 f_{\mathcal{N}}(y; \mu_1, \sigma_1^2) + ... + \omega_K f_{\mathcal{N}}(y; \mu_K, \sigma_K^2), \tag{4.1}$$

where $f_{\mathcal{N}}(y; \mu_k, \sigma_k^2)$ is the density of a univariate normal distribution. The model parameters are $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K, \omega_1, ..., \omega_K)$, where $\boldsymbol{\theta}_k = (\mu_k, \sigma_k^2)$ and $\omega_1, ..., \omega_K$ are weights of each normal distributions. The weights satisfy

$$0 < \omega_k < 1 \tag{4.2}$$

and

$$\sum_{k=1}^{K} \omega_k = 1, \tag{4.3}$$

but are arbitrary otherwise.

The model for a mixture of multivariate normal distributions is similar. The distribution of a multivariate random variable $\boldsymbol{Y}$ is assumed to be a mixture of K multivariate normal distributions and its density can be described as

$$p(\boldsymbol{y}|\boldsymbol{\vartheta}) = \omega_1 f_{\mathcal{N}}(\boldsymbol{y}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + ... + \omega_K f_{\mathcal{N}}(\boldsymbol{y}; \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K), \tag{4.4}$$

where $f_{\mathcal{N}}(\boldsymbol{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the density of a multivariate normal distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ [23].

4.1.2 Mean and covariance of a multivariate mixture

For a multivariate mixture, the overall center of the distribution of $\boldsymbol{Y}$ is defined by the expected value

$$\mathrm{E}(\boldsymbol{Y}|\boldsymbol{\vartheta}) = \sum_{k=1}^{K} \omega_k \boldsymbol{\mu}_k. \tag{4.5}$$

The covariance matrix of $\boldsymbol{Y}$ is then given by

$$\mathrm{Var}(\boldsymbol{Y}|\boldsymbol{\vartheta}) = \sum_{k=1}^{K} \omega_k \boldsymbol{\Sigma}_k + \sum_{k=1}^{K} \omega_k (\boldsymbol{\mu}_k - \mathrm{E}(\boldsymbol{Y}|\boldsymbol{\vartheta}))(\boldsymbol{\mu}_k - \mathrm{E}(\boldsymbol{Y}|\boldsymbol{\vartheta})^\intercal. \tag{4.6}$$

The total covariance $\mathrm{Var}(\boldsymbol{Y}|\boldsymbol{\vartheta})$ takes into account two sources of variability. These sources are the within-group heterogeneity and the between-group heterogeneity [23].

4.2 Mixture reduction

The normal mixture can be reduced to a single Gaussian. This corresponds to the fusion of multiple state estimates from different sensors into one state estimate. In this section, two methods are described: the moment matching method and the covariance intersection method.

4.2.1 Moment matching

In the moment matching method, the mixture is replaced by a single Gaussian with mean and covariance given by Equations (4.5) and (4.6). In case of target tracking, it means that the state estimate $\hat{x}_t$ at time t is given by

$$\hat{x}_t = \sum_{i=1}^{N} \omega_i \hat{x}_{i,t} \tag{4.7}$$

and the covariance matrix P_t is computed as

$$P_t = \sum_{i=1}^{N} \omega_i P_{i,t} + \sum_{i=1}^{N} \omega_i (\hat{x}_{i,t} - \hat{x}_t)(\hat{x}_{i,t} - \hat{x}_t)^\intercal, \tag{4.8}$$

where N is the number of sensors which have observed the given target and $\omega_1, ..., \omega_N$ are the weights of each sensors, $0 < \omega_i < 1$ and $\sum_{i=1}^{N} \omega_i = 1$.

4.2.2 Covariance intersection

Covariance intersection (CI) [24] is a popular algorithm of data fusion which fuses the local estimates of individual sensors for arbitrary cross-correlations. The unknown cross-correlations of the local estimates are often problematic and ignoring these correlations may lead to overconfident covariance matrices of fused estimates. The CI algorithm can be easily employed as a part of a sequential estimation scheme.

Let us consider a sensor network of N sensors. At each time step t, each sensor i computes a local estimate $\hat{x}_{i,t}$ of an unknown state vector x_t and a corresponding covariance matrix $P_{i,t}$. The local estimates $\hat{x}_{i,t}$ are assumed unbiased and can be correlated with unknown cross-correlations.

The aim of the CI algorithm is to fuse the local estimates $\hat{x}_{i,t}$ and their local covariances $P_{i,t}$ into a global estimate $\hat{x}_t$ and its global covariance P_t. At first, P_t is calculated as

$$P_t = \left(\sum_{i=1}^{N} \omega_i P_{i,t}^{-1} \right)^{-1} \tag{4.9}$$

and $\hat{x}_t$ is then given by

$$\hat{x}_t = P_t \sum_{i=1}^{N} \omega_i P_{i,t}^{-1} \hat{x}_{i,t}. \tag{4.10}$$

The weights satisfy $0 < \omega_i < 1$ and $\sum_{i=1}^{N} \omega_i = 1$, but are arbitrary otherwise [24]. There are more ways to choose them.

The simplest way to determine weights for the CI algorithm is to assign equal weights to all sensors. This method is sufficient for practical use. However, the optimal weights can be found by minimizing the performance index $\mathrm{tr}\{P_t\}$ such that

$$\min \mathrm{tr}\{P_t\} = \min \mathrm{tr} \left\{ \left(\sum_{i=1}^{N} \omega_i P_{i,t}^{-1} \right)^{-1} \right\}, \tag{4.11}$$

where the weights satisfy the constraints described above and $\mathrm{tr}\{.\}$ denotes the trace of a matrix. It is a nonlinear optimization problem and the resulting weights are different at every time step. If the number of sensors N is larger, optimizing the weights is complex and more computationally demanding [25].

Another option is to minimize the determinant of the fused covariance or to use the Metropolis weights, which are simple to compute, well-suited for

distributed implementation, and take into account the network topology [26].

The advantage of the CI algorithm is that the computation of cross covariances is completely avoided. The method also creates a consistent fused estimate, and a non-divergent filter is therefore obtained. The accuracy of the fused estimate outperforms each local one [25].

Chapter 5
Experiments

In this chapter, we describe the methods of sensor fusion and their impact on the performance of a multi-target tracking algorithm, in this case, a GM-PHD filter. We present three methods: the temporary birthplaces approach, the moment matching approach, and the covariance intersection approach. We introduce several test scenarios, measure the estimation performance of these methods, and compare it to the estimation performance of a plain GM-PHD filter.

5.1 Sensor fusion methods

In the following experiments, we assume this situation: A primary sensor with a large field of view performs a surveillance task, namely detection and tracking of moving targets. For this purpose, it employs the Gaussian mixture probability density (GM-PHD) filter. Quite naturally, the price paid for the large field of view is the sensor's sensitivity to clutter, lower spatial resolution etc.

In order to resolve these issues, a deployment of secondary sensors in problematic areas is supposed. These sensors could be much cheaper, which immediately imposes some advantages and disadvantages. First, they could be much smaller in size, which would allow their easier deployment. Also, due to their smaller field of view, it could be expected that their resolution could be much better, and the intensity of clutter could be much lower than in the case of the primary sensor. They may have, e.g., smaller measurement covariance R. However, the expected drawback would be their lower computational and memory performance. We simulate this by running the basic single-target PDA filter in them.

The fields of view of secondary sensors, which cover the problematic areas,

lie within the field of view of the primary sensor and are non-overlapping. The state estimates from the additional PDA filters are provided to the PHD filter to perform fusion. The sensor fusion is therefore centralized.

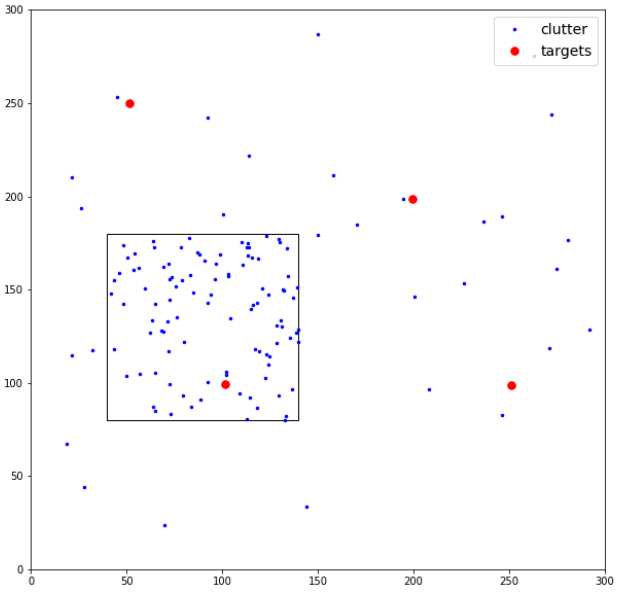

Figure 5.1 Example of a field of view of a primary sensor with a problematic, high-cluttered area.

A sensor is enabled only when a target appears in its field of view. The weight w of such target must be higher than a predetermined threshold, in this case, $w > 0.8$ to prevent enabling sensors due to false targets. If a target leaves the field of view of a sensor, the sensor is disabled. A sensor is also disabled if the estimate covariance of corresponding PDA filter is too large, which means that the filter probably tracks false alarms instead of target.

A cycle of the sensor fusion algorithm is as follows:

1. Combine step, which combines estimates from previous time step (if there are any) and also includes enabling sensors if there is a target in their field of view and disabling them if there is no target in their field of view.
2. Prediction step of the PHD filter and prediction steps of the PDA filters of enabled sensors.
3. Update step of the PHD filter and update steps of the PDA filters of enabled sensors.

5.1.1 Temporary birthplaces approach

The first method is the temporary birthplaces approach. This method does not perform any combination of PDA and PHD estimates. Instead, it passes the estimates from PDA filters to PHD filter as birthplaces. These birthplaces are temporary – they are only present in the PHD filter for one time step. The PHD filter treats them like any other of its birthplaces. They become part of the prediction step (the prediction for newborn targets), but they are removed afterwards.

To avoid an excessive increase in the number of targets and thus an increase in computational complexity, it is determined that there is lower survival probability in the problematic areas. All targets from PHD filter that appear in these areas are set to a lower probability of survival than targets outside these areas. If a target leaves the problematic area, its previous probability of survival is set again.

The weight w of targets born at temporary birthplaces is set to $w = 0.7$. The algorithm of this method is described by the pseudocode in Algorithm 5. For simplicity, pseudocodes of individual methods do not contain any time indices.

5.1.2 Moment matching approach

The second method is the moment matching approach presented in Section 4.2.1. This method performs fusion of the state estimate from PDA filter with the state estimate of the nearest target from PHD filter.

At first, the target from PHD filter which is the closest to the target from PDA filter is selected. This is performed using the Mahalanobis distance. The nearest target from PHD filter is the one whose state estimate m_{PHD} has the shortest Mahalanobis distance to the PDA state estimate m_{PDA} with respect to PDA estimate covariance P_{PDA}. The formula of the Mahalanobis distance is

$$d_M(m_{\text{PHD}}, m_{\text{PDA}}, P_{\text{PDA}}) = \sqrt{(m_{\text{PHD}} - m_{\text{PDA}})^\intercal (P_{\text{PDA}})^{-1} (m_{\text{PHD}} - m_{\text{PDA}})}. \tag{5.1}$$

The selected PHD target should preferably have weight $w > 0.5$. If no such target is present in the field of view of the sensor, a target with $w \leq 0.5$ is selected.

The PDA target and the nearest PHD target are then combined by moment matching. The fused state estimate m_{new} and the fused estimate covariance P_{new} with respect to ω as the fusion weight of the additional sensor are given

by

$$m_{\text{new}} = \omega m_{\text{PDA}} + (1-\omega) m_{\text{PHD}}, \tag{5.2}$$

$$\begin{aligned} P_{\text{new}} = {} & \omega P_{\text{PDA}} + (1-\omega) P_{\text{PHD}} + \omega (m_{\text{PDA}} - m_{\text{new}})(m_{\text{PDA}} - m_{\text{new}})^{\intercal} \\ & + (1-\omega)(m_{\text{PHD}} - m_{\text{new}})(m_{\text{PHD}} - m_{\text{new}})^{\intercal}. \end{aligned} \tag{5.3}$$

The state estimate and covariance of the nearest PHD target are then replaced by m_{new} and P_{new}. The new weight of this target is given by

$$w_{\text{new}} = \omega w_{\text{PDA}} + (1-\omega) w_{\text{PHD}}. \tag{5.4}$$

The weight w_{PHD} of the PHD target is known, the weight w_{PDA} of the PDA target needs to be specified. In this work, it is set to fixed value of 0.95. A pseudocode for this method is shown in Algorithm 6.

5.1.3 Covariance intersection approach

The third method is the covariance intersection approach presented in Section 4.2.2. This method is similar to the previous method, but instead of moment matching, the covariance intersection is used to fuse the state estimates together.

As in the previous method, the target from PHD filter that is the closest to the PDA target is selected using the Mahalanobis distance. Targets with $w > 0.5$ are prefered. The state estimates of the PDA and PHD targets are then combined as follows with respect to the fusion weight of the additional sensor ω,

$$P_{\text{new}} = \left(\omega (P_{\text{PDA}})^{-1} + (1-\omega)(P_{\text{PHD}})^{-1}\right)^{-1}, \tag{5.5}$$

$$m_{\text{new}} = P_{\text{new}} \left(\omega (P_{\text{PDA}})^{-1} m_{\text{PDA}} + (1-\omega)(P_{\text{PHD}})^{-1} m_{\text{PHD}}\right). \tag{5.6}$$

The state estimate and covariance of the nearest PHD target are then replaced by m_{new} and P_{new}. The new weight of the PHD target is also computed as in the previous method,

$$w_{\text{new}} = \omega w_{\text{PDA}} + (1-\omega) w_{\text{PHD}}, \tag{5.7}$$

with the weight w_{PDA} set to fixed value of 0.95. The algorithm of this method is described by Algorithm 7.

Algorithm 5 Temporary birthplaces approach

given T as a set of PHD targets, m_{PDA} and P_{PDA} from additional PDA filters, p_S as the survival probability, and $p_{S,\text{add}}$ as the survival probability in areas of interest.

for each $t \in T$:
  if t is in the field of view of a sensor s:
    if sensor s is not enabled **and** $w^{(t)} > 0.8$:
      enable sensor s
      $p_S^{(t)} = p_{S,\text{add}}$
      $m_{\text{PDA}}^{(s)} = m^{(t)}$
      $P_{\text{PDA}}^{(s)} = P^{(t)}$
    else if sensor s is enabled:
      $p_S^{(t)} = p_{S,\text{add}}$
  else if t is not in the field of view of any sensor **and** $p_S^{(t)} \neq p_S$:
    $p_S^{(t)} = p_S$
for each s **in** enabled sensors:
  if the target tracked by sensor s is in its field of view:
    add $m_{\text{PDA}}^{(s)}$ and $P_{\text{PDA}}^{(s)}$ to PHD filter as a temporary birthplace with weight 0.7
  else if the target tracked by sensor s disappeared from its field of view:

    disable sensor s

output a set of temporary birthplaces for PHD filter.

Algorithm 6 Moment matching approach

given T as a set of PHD targets, m_{PDA} and P_{PDA} from additional PDA filters, and additional sensor fusion weight ω.

for each $t \in T$:
  if t is in the field of view of a sensor s:
    if sensor s is not enabled **and** $w^{(t)} > 0.8$:
      enable sensor s
      $m_{\text{PDA}}^{(s)} = m^{(t)}$
      $P_{\text{PDA}}^{(s)} = P^{(t)}$
for each s **in** enabled sensors:
  if the target tracked by sensor s is in its field of view:
    find PHD target t such that $w^{(t)} > 0.5$ and $m^{(t)}$ is the closest to $m_{\text{PDA}}^{(s)}$
    if there is no such target:
      find the closest target with $w^{(t)} \leq 0.5$
    $m_{\text{new}} = \omega m_{\text{PDA}}^{(s)} + (1-\omega)m^{(t)}$
    $P_{\text{new}} = \omega P_{\text{PDA}}^{(s)} + (1-\omega)P^{(t)}$
    $P_{\text{new}} = P_{\text{new}} + \omega(m_{\text{PDA}}^{(s)} - m_{\text{new}})(m_{\text{PDA}}^{(s)} - m_{\text{new}})^{\intercal}$
    $P_{\text{new}} = P_{\text{new}} + (1-\omega)(m^{(t)} - m_{\text{new}})(m^{(t)} - m_{\text{new}})^{\intercal}$
    $w_{\text{new}} = \omega \cdot 0.95 + (1-\omega)w^{(t)}$
    $m^{(t)} = m_{\text{new}}$
    $P^{(t)} = P_{\text{new}}$
    $w^{(t)} = w_{\text{new}}$
  else if the target tracked by sensor s disappeared from its field of view:
    disable sensor s

output a new set of PHD targets.

Algorithm 7 Covariance intersection approach

given T as a set of PHD targets, m_{PDA} and P_{PDA} from additional PDA filters, and additional sensor fusion weight ω.

for each $t \in T$:
  if t is in the field of view of a sensor s:
    if sensor s is not enabled **and** $w^{(t)} > 0.8$:
      enable sensor s
      $m_{\text{PDA}}^{(s)} = m^{(t)}$
      $P_{\text{PDA}}^{(s)} = P^{(t)}$
for each s **in** enabled sensors:
  if the target tracked by sensor s is in its field of view:
    find PHD target t such that $w^{(t)} > 0.5$ and $m^{(t)}$ is the closest to $m_{\text{PDA}}^{(s)}$
    if there is no such target:
      find the closest target with $w^{(t)} \leq 0.5$
    $P_{\text{new}} = \left(\omega (P_{\text{PDA}}^{(s)})^{-1} + (1-\omega)(P^{(t)})^{-1}\right)^{-1}$
    $m_{\text{new}} = P_{\text{new}} \left(\omega (P_{\text{PDA}}^{(s)})^{-1} m_{\text{PDA}}^{(s)} + (1-\omega)(P^{(t)})^{-1} m^{(t)}\right)$
    $w_{\text{new}} = \omega \cdot 0.95 + (1-\omega) w^{(t)}$
    $m^{(t)} = m_{\text{new}}$
    $P^{(t)} = P_{\text{new}}$
    $w^{(t)} = w_{\text{new}}$
  else if the target tracked by sensor s disappeared from its field of view:
    disable sensor s

output a new set of PHD targets.

5.2 Evaluation

For evaluation of the experiments, the generalized optimal sub-pattern assignment (GOSPA) metric[1] is used. It is a metric that evaluates multi-target tracking algorithms and penalizes localization errors for detected targets as well as missed and false targets. It is also able to handle the random finite sets of targets.

Let $c > 0$, $0 < \alpha \leq 2$, and $1 \leq p < \infty$. Assume that $d(x, y)$ denotes a metric for any $x, y \in \mathbb{R}^N$ and $d^{(c)}(x, y) = \min(d(x, y), c)$ is its cut-off metric. Let Π_n be the set of all permutations of $\{1, ..., n\}$ for any $n \in \mathbb{N}$ and any element $\pi \in \Pi_n$ be a sequence $(\pi(1), ..., \pi(n))$. Assume that $X = \{x_1, ..., x_{|X|}\}$

[1] In this work, implementation of GOSPA from `https://github.com/ewilthil/gospapy/tree/master` is used.

and $Y = \{y_1, ..., y_{|Y|}\}$ are finite subsets of $\mathbb{R}^N$. For $|X| \leq |Y|$, the GOSPA metric is defined as

$$d_p^{(c,\alpha)}(X,Y) = \left(\min_{\pi \in \Pi_{|Y|}} \sum_{i=1}^{|X|} d^{(c)}(x_i, y_{\pi(i)})^p + \frac{c^p}{\alpha}(|Y| - |X|) \right)^{\frac{1}{p}}. \tag{5.8}$$

If $|X| > |Y|$, then $d_p^{(c,\alpha)}(X,Y) = d_p^{(c,\alpha)}(Y,X)$.

The parameter p determines the penalization of outliers. Larger values of p mean that the outliers are penalized more. The parameter c is the maximum allowable localization error and, together with α, it also determines the error due to cardinality mismatch. The most appropriate choice of α for evaluation of multi-target tracking algorithms is $\alpha = 2$ [27].

In this work, the GOSPA parameters are set as follows: $c = 8$, $p = 1$, and $\alpha = 2$.

5.3 Parameters

In this section, we describe the parameters common for all experiments. The examples in the following are based on the CVM model known from Section 1.1.1. We can think of these examples as tracking airplanes in 2D space using a radar instrument and additional on-demand radars which can track these airplanes in problematic areas. The radar measuring period is one second. The transition matrix F and the process noise covariance matrix Q of this state-space model are

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad Q = q^2 \begin{bmatrix} \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^3}{3} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t \end{bmatrix}, \tag{5.9}$$

where $\Delta t = 1$ is the time step and q is the parameter of the process noise matrix, which is set to $q = 1.2$.

The measurement matrix H and the measurement noise covariance matrix R have the form

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \qquad R = r^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{5.10}$$

where r is the parameter of the measurement noise matrix which shows how inaccurate the sensor measurements are. We consider it as $r = 4$ for the main

radar and $r = 2$ for the more precious additional radars.

The area where the targets are tracked has square shape with side length 300. The birthplaces of new targets, which we can think of as airports where the airplanes take off, are located at coordinates $[100, 100]^\intercal$ and $[200, 200]^\intercal$. These airports have the same matrix P and weight w which they pass to the newborn targets. The parameters of the two airports are as follows,

$$w1 = w2 = 0.15, \quad m1 = \begin{bmatrix} 100 \\ 100 \\ 0 \\ 0 \end{bmatrix}, m2 = \begin{bmatrix} 200 \\ 200 \\ 0 \\ 0 \end{bmatrix},$$
$$P1 = P2 = \begin{bmatrix} 20 & 0 & 20 & 0 \\ 0 & 20 & 0 & 20 \\ 20 & 0 & 40 & 0 \\ 0 & 20 & 0 & 40 \end{bmatrix}. \tag{5.11}$$

These parameters create a newborn PHD target.

The parameters for the PHD pruning procedure also have to be set. The truncation threshold T, the merging threshold U, and the maximum number of components that can pass to the next step J_{max} are

$$T = 0.00001, \qquad U = 4, \qquad J_{max} = 100. \tag{5.12}$$

The survival probability p_S of PHD targets, the gate probability p_D of PDA filter and the detection probability p_D of both of them are

$$p_D = 0.95, \qquad p_S = 0.99, \qquad p_G = 0.99. \tag{5.13}$$

There are also some common parameters for the sensor fusion methods. The fusion weight of the additional sensors is

$$\omega = 0.5. \tag{5.14}$$

If a new weight of a selected PHD target is computed, the weight w_{PDA} of the PDA target is set for this computation to

$$w_{\text{PDA}} = 0.95. \tag{5.15}$$

For the temporary birthplaces approach, the survival probability inside the problematic areas is set to

$$p_{S,\text{add}} = 0.2. \tag{5.16}$$

5.4 Example 1: Homogeneous clutter case

The purpose of this simulation example is to demonstrate the behavior of the proposed methods in the case of homogeneous clutter. That is, the spatial distribution of clutter does not vary in the observed area. The aim is to show that the collaboration does not negatively impact the performance of the primary estimator (the PHD filter).

In this example, four targets appear over time in the observed area and follow independent paths. The simulation takes 100 time steps and the targets are born at time steps $t_1 = 0$, $t_2 = 25$, $t_3 = 50$, and $t_4 = 75$. Their initial states are

$$x_1 = [200, 200, -1, -1]^\intercal, \tag{5.17}$$
$$x_2 = [100, 100, 1, -1]^\intercal, \tag{5.18}$$
$$x_3 = [200, 200, 0, -1]^\intercal, \tag{5.19}$$
$$x_4 = [100, 100, 1, -1]^\intercal. \tag{5.20}$$

A secondary sensor is added. The field of view of this sensor has a square shape with side length 170 and top left corner coordinates $[20, 280]$.

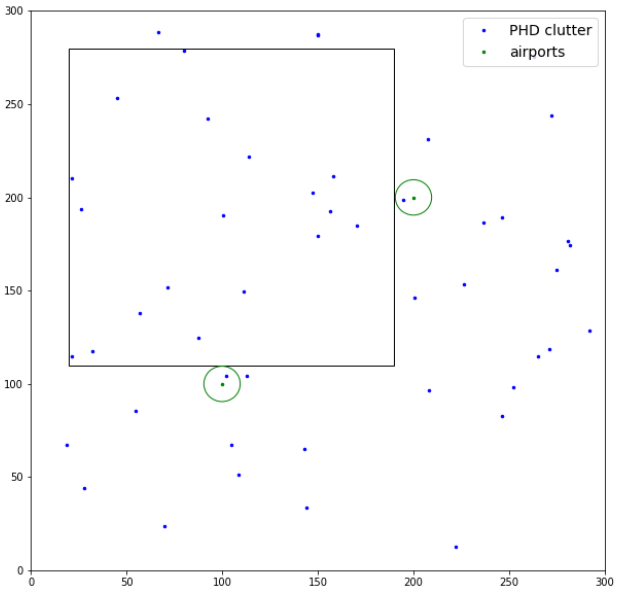

Figure 5.2 The field of view of the primary and secondary sensor in simulation examples.

Four algorithms are tested: no fusion, temporary birthplaces approach, moment matching approach, and covariance intersection approach. The performance of these algorithms is measured using the GOSPA metric. The algorithms are tested for several different values of main sensor clutter intensity λ. The clutter intensity of the secondary sensor equals the clutter intensity of the primary sensor, $\lambda_{\text{PDA}} = \lambda$. For each λ, 50 different trajectories are generated and the average GOSPA error of each trajectory is computed.

method	$\lambda = 0.0001$	$\lambda = 0.0003$	$\lambda = 0.0005$	$\lambda = 0.0007$	$\lambda = 0.001$
PHD avg	**12.37996**	**12.49728**	**12.56554**	**12.39241**	**12.36013**
PHD best	11.18835	11.40151	11.79767	11.29602	11.15049
PHD worst	13.37372	13.95824	13.97562	13.46601	13.78913
TB avg	**12.49509**	**12.52925**	**12.57335**	**12.39107**	**12.17103**
TB best	11.05388	11.54402	11.77318	11.54739	11.17907
TB worst	13.36078	13.74126	13.85764	13.47850	13.21648
MM avg	**12.82672**	**12.76087**	**12.69413**	**12.50860**	**12.31817**
MM best	10.98839	11.54012	11.79713	11.49984	11.14135
MM worst	14.09090	13.85145	14.24221	13.81632	13.62628
CI avg	**12.00795**	**12.11625**	**12.22240**	**12.11454**	**12.07289**
CI best	10.88331	10.79391	11.46600	11.38945	11.05966
CI worst	12.88551	13.32223	13.75112	13.07624	13.82783

Table 5.1 Average, best and worst GOSPA combined error over 50 runs for different values of clutter intensity λ.

method	$\lambda = 0.0001$	$\lambda = 0.0003$	$\lambda = 0.0005$	$\lambda = 0.0007$	$\lambda = 0.001$
PHD avg	**8.83596**	**8.01968**	**6.87834**	**5.64841**	**3.69053**
PHD best	8.12384	6.44711	5.64590	4.38034	1.81063
PHD worst	9.79592	9.07137	8.29809	7.49087	5.09031
TB avg	**8.80149**	**7.83485**	**6.63975**	**5.45827**	**3.44223**
TB best	7.98180	6.29241	4.98308	4.20496	1.76744
TB worst	10.04078	8.94906	8.01318	7.30834	5.12209
MM avg	**8.38192**	**7.39447**	**6.02453**	**4.95500**	**3.26457**
MM best	6.31308	5.41593	3.61692	3.59416	1.91977
MM worst	9.54316	8.61340	7.70057	7.03696	4.75050
CI avg	**6.33995**	**5.62985**	**5.00640**	**4.29694**	**3.15929**
CI best	4.17184	3.57690	2.70867	1.80497	1.38392
CI worst	9.54316	7.82429	7.41209	5.96105	4.59233

Table 5.2 Average, best and worst GOSPA localization error over 50 runs for different values of clutter intensity λ.

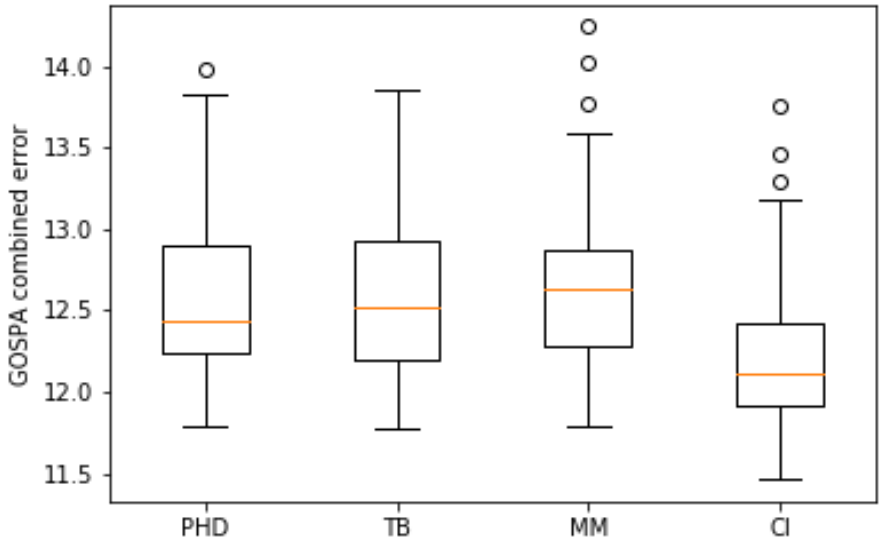

Figure 5.3 Visualization of GOSPA combined error for $\lambda = 0.0005$. Each box contains 50 samples.

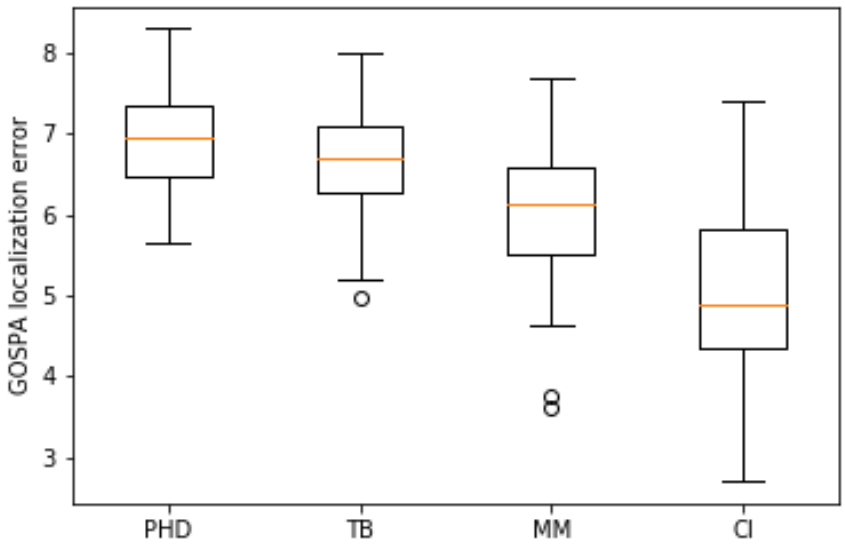

Figure 5.4 Visualization of GOSPA localization error for $\lambda = 0.0005$. Each box contains 50 samples.

The results as well as the tested values of λ are shown in Table 5.1. The computed values of GOSPA localization error, which is a component of the GOSPA combined error, are shown in Table 5.2. As an example, Figure 5.3 and Figure 5.4 present graphs of the distribution of the GOSPA combined error and GOSPA localization error for $\lambda = 0.0005$.

The results show that the fusion methods can reduce the localization error. However, the combined error remains the same or even increases. This is mainly due to the other components of GOSPA combined error, the missed targets error and false targets error. Why this happens is a subject for further research.

The decrease in localization error is most significant for the covariance intersection approach. The second lowest error occurs with the moment matching method. These are the methods that directly combine the estimates of PDA and PHD filter for a selected target. The temporary birthplaces approach, which performs no direct combination and only passes the estimates from the PDA filter to the PHD filter as airports, shows less significant decrease in localization error.

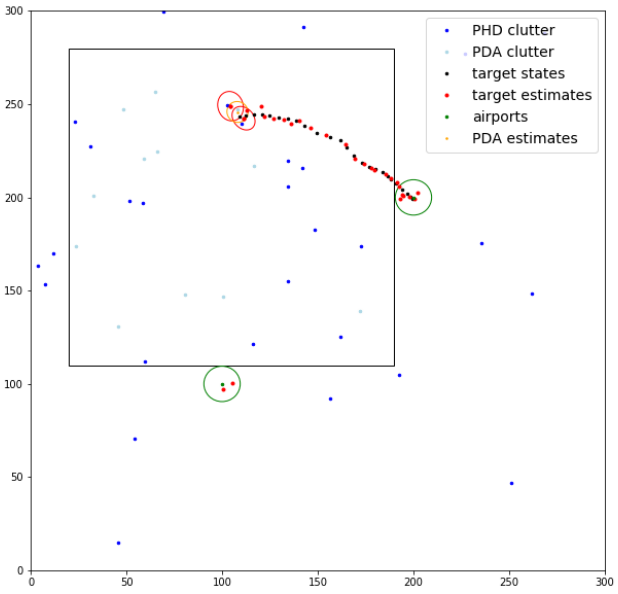

Figure 5.5 Example: real trajectories and target estimates with clutter ($\lambda = 0.0003$), simulated for 25 time steps.

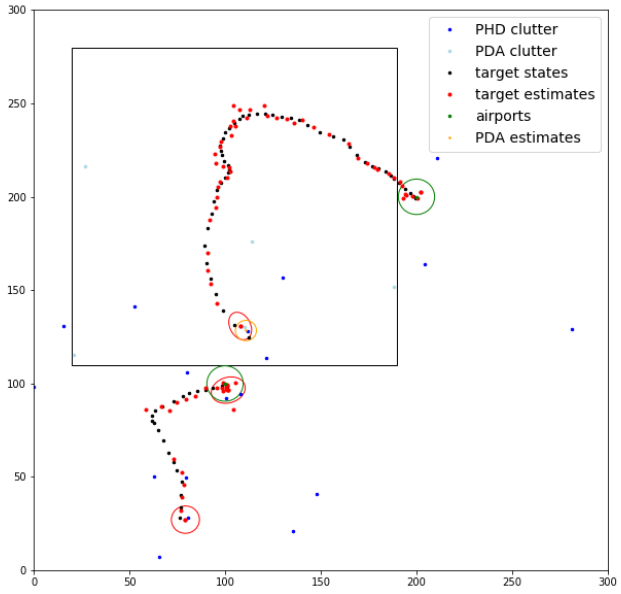

Figure 5.6 Example: real trajectories and target estimates with clutter ($\lambda = 0.0003$), simulated for 50 time steps.

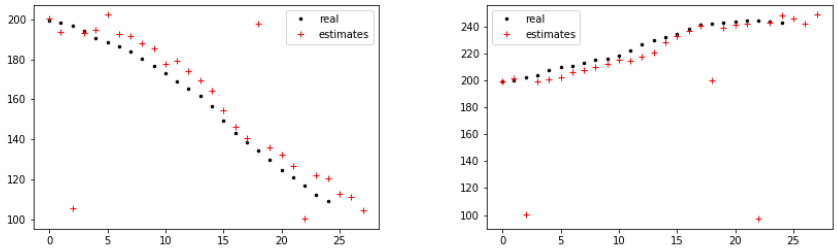

Figure 5.7 Longitude and latitude of a target, simulated for 25 time steps.

5.5 Example 2: Inhomogeneous clutter case

The second example studies the case where the field of view of the main PHD tracker contains areas with dense clutter. If a target enters these areas, the PHD tracker performance may be significantly impaired. The aim of this simulation example is to demonstrate that a collaboration with more precious yet simpler PDA filter may positively affect the PHD tracker performance.

As well as in the previous example, the simulation example takes 100 steps and four targets appear in the observed area at time steps $t_1 = 0$, $t_2 = 25$, $t_3 = 50$, and $t_4 = 75$. Their initial states are

$$x_1 = [200, 200, -1, -1]^\intercal, \tag{5.21}$$
$$x_2 = [100, 100, 1, -1]^\intercal, \tag{5.22}$$
$$x_3 = [200, 200, 0, -1]^\intercal, \tag{5.23}$$
$$x_4 = [100, 100, 1, -1]^\intercal. \tag{5.24}$$

A secondary sensor with the same field of view as in the previous example is added.

The inhomogeneous spatial distribution of clutter is simulated as follows: the clutter outside the problematic areas has intensity λ_1 and the clutter in the problematic areas has higher intensity λ_2. This can be interpreted as a situation in which the problematic areas are, e.g., covered by clouds or disturbed by electromagnetic interference. In this example, there is one area with higher clutter intensity which equals the field of view of the secondary sensor.

No fusion, temporary birthplaces approach, moment matching approach, and covariance intersection approach are tested in such cluttered environment. The algorithms are tested for several different combinations of clutter intensities λ_1 and λ_2. The clutter intensity of the secondary sensor equals the higher one of those intensities, $\lambda_{\mathrm{PDA}} = \lambda_2$. For each combination, 50 different trajectories are generated.

The resulting values of GOSPA combined error and GOSPA localization error are shown in Table 5.3 and Table 5.4. Figure 5.8 and Figure 5.9 show graphs of the distribution of GOSPA combined error and GOSPA localization error for $\lambda_1 = 0.0005$ and $\lambda_2 = 0.001$.

The results are similar to the results of the first example. The fusion methods can reduce the GOSPA localization error, while the GOSPA combined error remains the same due to missed targets error and false targets error. An exception is the case of $\lambda_2 = 0.002$, which is rather extreme case. The combined error is significantly lower for the temporary birthplaces approach and higher for the moment matching approach.

method	$\lambda_1 = 0.0001$, $\lambda_2 = 0.0005$	$\lambda_1 = 0.0005$, $\lambda_2 = 0.001$	$\lambda_1 = 0.0005$, $\lambda_2 = 0.002$
PHD avg	**12.17661**	**11.62210**	**16.24350**
PHD best	11.26360	10.55281	13.30828
PHD worst	13.73334	12.94708	20.00331
TB avg	**11.95555**	**11.39246**	**11.89194**
TB best	10.83213	10.64005	11.27865
TB worst	12.89080	12.08872	12.93776
MM avg	**12.78197**	**11.91106**	**20.17275**
MM best	11.57491	10.77584	17.01672
MM worst	14.39841	13.24457	23.23291
CI avg	**11.79937**	**11.43072**	**16.98416**
CI best	10.89226	10.66442	14.41285
CI worst	12.71226	12.68882	20.51757

Table 5.3 Average, best and worst GOSPA combined error over 50 runs for different values of clutter intensity λ_1 and clutter intensity in areas of interest λ_2, $\lambda_{\text{PDA}} = \lambda_2$.

method	$\lambda_1 = 0.0001$, $\lambda_2 = 0.0005$	$\lambda_1 = 0.0005$, $\lambda_2 = 0.001$	$\lambda_1 = 0.0005$, $\lambda_2 = 0.002$
PHD avg	**7.85581**	**4.16450**	**1.21470**
PHD best	6.50696	2.62815	0.05853
PHD worst	9.09088	5.56847	3.70625
TB avg	**7.63315**	**3.84846**	**0.54954**
TB best	6.50701	1.66217	0.02807
TB worst	8.93432	4.97811	1.92582
MM avg	**6.76677**	**3.50546**	**1.01675**
MM best	4.22046	1.61154	0.07443
MM worst	8.24031	5.15858	2.86749
CI avg	**4.64577**	**2.92912**	**0.81376**
CI best	2.27873	1.26196	0.06048
CI worst	6.66079	4.59547	1.96115

Table 5.4 Average, best and worst GOSPA localization error over 50 runs for different values of clutter intensity λ_1 and clutter intensity in areas of interest λ_2, $\lambda_{\text{PDA}} = \lambda_2$.

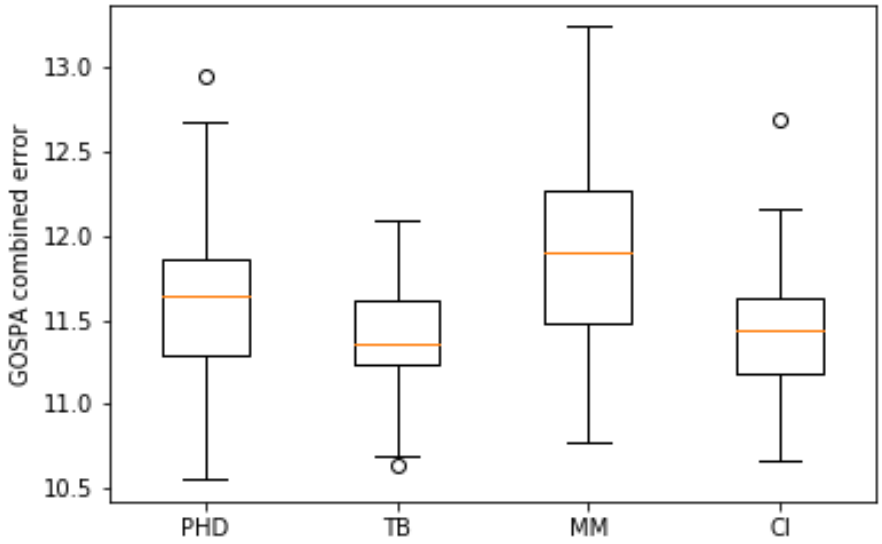

Figure 5.8 Visualization of GOSPA combined error for $\lambda_1 = 0.0005$ and $\lambda_2 = 0.001$, $\lambda_{\text{PDA}} = \lambda_2$. Each box contains 50 samples.

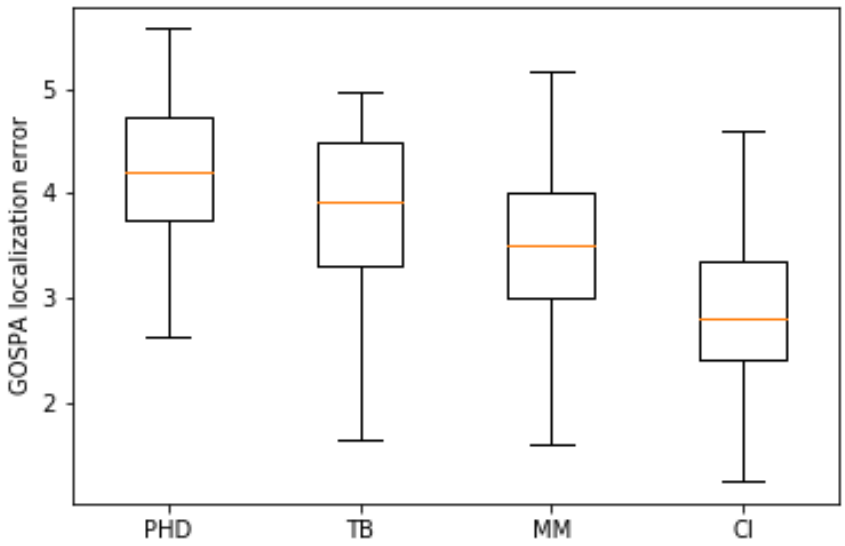

Figure 5.9 Visualization of GOSPA localization error for $\lambda_1 = 0.0005$ and $\lambda_2 = 0.001$, $\lambda_{\text{PDA}} = \lambda_2$. Each box contains 50 samples.

As in the first example, the decrease in localization error is most significant for the covariance intersection approach, except for $\lambda_2 = 0.002$ where the most successful method is the temporary birthplaces approach. The clutter intensity of 0.002 is rather high for the PHD filter and in this case it may be advantageous to perform no direct combination of PHD and PDA estimates and just pass the estimates from the more precious PDA filter to the PHD filter.

Another experiment was performed for the clutter intensity of the secondary sensor that equals the lower one of the two clutter intensities, $\lambda_{\text{PDA}} = \lambda_1$. This corresponds to a situation where the sensor has a more accurate view of the area of interest. Compared to the previous experiment, the results show no significant improvement. This may be due to small validation region of the PDA filter. Such region contains none or only a few clutter points and the clutter has therefore minimal effect on performance.

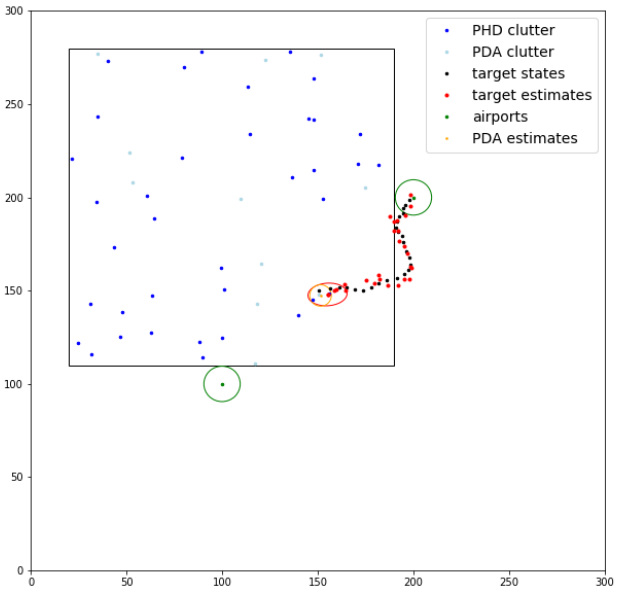

Figure 5.10 Example: real trajectories and target estimates with clutter ($\lambda_1 = 0.0005$, $\lambda_2 = 0.001$), simulated for 25 time steps.

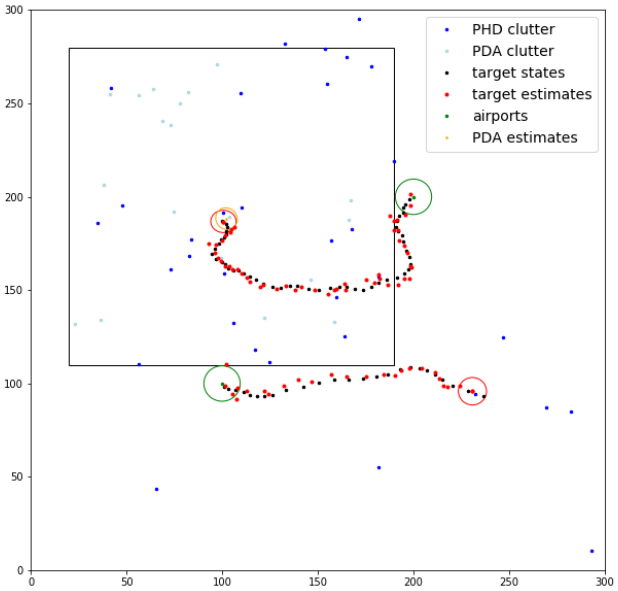

Figure 5.11 Example: real trajectories and target estimates with clutter ($\lambda_1 = 0.0005$, $\lambda_2 = 0.001$), simulated for 50 time steps.

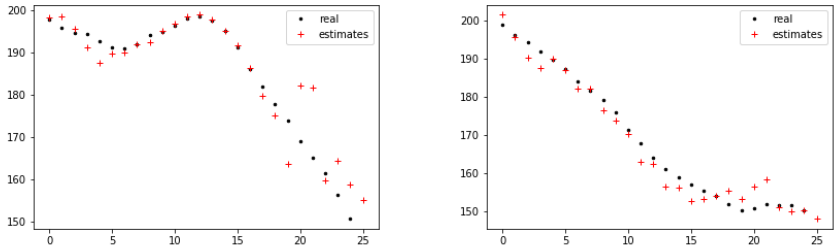

Figure 5.12 Longitude and latitude of a target, simulated for 25 time steps.

5.6 Future work

An important subject for further research is to find the causes of the increase in missed targets error and false targets error in the sensor fusion methods. This increase is more significant in the moment matching approach and the covariance intersection approach, which do not directly change the number of PHD targets.

One of the complications of the sensor fusion methods may be enabling the additional sensors by false targets. In the current state, the only criterion for enabling a sensor is the weight of the target that appears in its field of view. The PDA filter also takes its state estimate and covariance directly from this target, which may be problematic if this target is a false target. A more spohisticated method of PDA track initiation could be used to prevent these problems.

Another option is to focus on the parameters of the sensor fusion methods. There are numerous parameters that can be tuned, such as the fusion weight ω, the weight assigned to the PDA target w_{PDA}, or the weight of a PHD target born at a temporary birthplace. Similarly, we can look at the parameters of the PHD filter.

An interesting problem may be using the proposed sensor fusion methods on real data. This would bring more parameters to tune, such as the process noise covariance matrix Q or the measurement noise covariance matrix R.

Chapter 6

Conclusion

In this work, we presented one of the most popular approaches to multi-target filtering, the Gaussian mixture probability hypothesis density (GM-PHD) filter. We explained the single-target probabilistic data association (PDA) filter and proposed three methods to improve the performance of the PHD filter by using location-specific information from the PDA filter. These methods are the temporary birthplaces approach, the moment matching approach, and the covariance intersection approach. The temporary birthplaces approach uses the state estimate and its covariance from the PDA filter to create airports for the PHD filter which become part of the following prediction step and are removed afterwards. The moment matching approach and the covariance intersction approach search for a target from the PHD filter which is nearest to the target from the PDA filter and fuse their state estimates together using the moment matching method or the covariance intersection method.

The performance of these methods was evaluated using the GOSPA metric and compared to the performance of a plain GM-PHD filter. Two main simulation examples were introduced. First, the homogeneous clutter case, where the spatial distribution of clutter did not vary in the observed area. The purpose of this example was to demonstrate that the collaboration with PDA filter does not negatively impact the performance of the primary PHD tracker. Second, the inhomogeneous clutter case, where the field of view of the primary tracker contained an area with dense clutter. The aim of this simulation was to show that the collaboration may positively affect the performance of the PHD estimator. The methods were tested for several different values of clutter intensity in the first case and for several combinations of such values in the second case.

We demonstrated that the proposed methods can improve the performance of the GM-PHD filter. All methods have decreased the GOSPA localization error. In most cases, the most significant decrease was brought by the co-

variance intersection approach. In the case of high clutter intensity, the most successful algorithm was the temporary birthplaces approach, which performs no direct fusion of target state estimates. However, the GOSPA combined error has remained the same due to increase in missed targets and false targets error. The next step in further research may be to examine the causes of this increase.

Bibliography

[1] D. Simon, "Kalman filtering," *Embedded systems programming*, vol. 14, no. 6, pp. 72–79, 2001.

[2] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[3] M. Kumar and S. Mondal, "Recent developments on target tracking problems: A review," *Ocean Engineering*, vol. 236, p. 109 558, 2021.

[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[5] G. Einicke and L. White, "Robust extended Kalman filtering," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2596–2599, 1999. DOI: 10.1109/78.782219.

[6] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, IEEE, 2000, pp. 153–158.

[7] P. M. Djuric, J. H. Kotecha, J. Zhang, *et al.*, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[8] X. R. Li and Y. Bar-Shalom, "Tracking in clutter with nearest neighbor filters: Analysis and performance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 3, pp. 995–1010, 1996.

[9] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979. DOI: 10.1109/TAC.1979.1102177.

[10] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009. DOI: 10.1109/MCS.2009.934469.

[11] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003. DOI: 10.1109/TAES.2003.1261119.

[12] B. Vo, B. Vo, and A Cantoni, "On multi-Bernoulli approximations to the Bayes multi-target filter," in *Proc. Int. Symp. Inf. Fusion*, 2007, pp. 71–81.

[13] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: Direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1883–1901, 2018.

[14] X. Zhang, P. Willett, and Y. Bar-Shalom, "Uniform versus nonuniform sampling when tracking in clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 2, pp. 388–400, 2006. DOI: 10.1109/TAES.2006.1642559.

[15] T. Thomas and S Sreeja, "Comparison of nearest neighbor and probabilistic data association filters for target tracking in cluttered environment," in *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*, IEEE, 2021, pp. 272–277.

[16] E. F. Brekke, A. G. Hem, and L.-C. N. Tokle, "Multitarget tracking with multiple models and visibility: Derivation and verification on maritime radar data," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 4, pp. 1272–1287, 2021. DOI: 10.1109/JOE.2021.3081174.

[17] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[18] Z. Prihoda, A. Jamgochian, B. Moore, and B. Lange, "Probability hypothesis density filter implementation and application," 2019.

[19] B. T. Vo, "Random finite sets in multi-object filtering," 2008.

[20] W. Yi, G. Li, and G. Battistelli, "Distributed multi-sensor fusion of PHD filters with different sensor fields of view," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5204–5218, 2020.

[21] P. Lu and F. Dai, "An overview of multi-sensor information fusion," in *2021 6th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, vol. 6, 2021, pp. 5–9. DOI: 10.1109/ICIIBMS52876.2021.9651656.

[22] M. Ueney, D. E. Clark, and S. J. Julier, "Distributed fusion of PHD filters via exponential mixture densities," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 521–531, 2013.

[23] S. Frühwirth-Schnatter, *Finite mixture and Markov switching models*. Springer, 2006.

[24] O. Hlinka, O. Slučiak, F. Hlawatsch, and M. Rupp, "Distributed data fusion using iterative covariance intersection," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1861–1865. DOI: 10.1109/ICASSP.2014.6853921.

[25] Z. Deng, P. Zhang, W. Qi, J. Liu, and Y. Gao, "Sequential covariance intersection fusion Kalman filter," *Information Sciences*, vol. 189, pp. 293–309, 2012.

[26] L. Xiao, S. Boyd, and S. Lall, "Distributed average consensus with time-varying Metropolis weights," *Automatica*, vol. 1, pp. 1–4, 2006.

[27] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *2017 20th International Conference on Information Fusion (Fusion)*, 2017, pp. 1–8. DOI: 10.23919/ICIF.2017.8009645.

Enclosed medium contents

- `readme.txt` ..... the file with media contents description
- `src`
  - `impl` ..... source codes of simulations
  - `thesis` ..... LaTeX source codes of the thesis
- `thesis.pdf` ..... the thesis text in PDF format