Multisensor data fusion for autonomous driving and advanced driver-assistance systems

Louis Guerlin

UNIVERSITÉ GRENOBLE ALPES

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE ALPES

Spécialité : **Signal, Image, Parole, Télécom**

Arrêté ministériel : 25 mai 2016

Présentée par **Louis GUERLIN**

Thèse dirigée par **Michèle ROMBAUT** et

codirigée par **Benjamin PANNETIER** et **Maxime DEROME**

préparée au sein du **Laboratoire Grenoble Images Parole Signal Automatique (GIPSA-Lab)**

dans **l'école doctorale d'Électronique, d'Électrotechnique, d'Automatique et de Traitement du Signal (EEATS)**

Approches bayésiennes pour le suivi d'objets étendus appliquées à la lutte anti-drone et au véhicule autonome

Thèse soutenue publiquement le 25 mars 2022,

devant le jury composé de:

Président :	**Guillaume GINOLHAC**	Professeur des universités, Université Savoie Mont Blanc
Rapporteur :	**Daniel CLARK**	Maître de conférence HDR, Telecom SudParis
Directrice :	**Michèle ROMBAUT**	Professeure des universités, Université Grenoble Alpes
Co-directeur :	**Benjamin PANNETIER**	Ingénieur HDR, CS Group
Examinateurs :	**Olivier MICHEL**	Professeur des universités, Université Grenoble Alpes
	Frédéric DAMBREVILLE	Ingénieur HDR, ONERA
Invités :	**Franck LI**	Ingénieur PHD, Renault
	Maxime DEROME	Ingénieur PHD, Renault

UNIVERSITÉ DE GRENOBLE ALPES

ÉCOLE DOCTORALE EEATS

École Doctorale d'Électronique, d'Électrotechnique, d'Automatique et de Traitement du Signal

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université de Grenoble Alpes

Mention : Signal, Image, Parole, Télécom

Présentée et soutenue par

Louis GUERLIN

Approches bayésiennes pour le suivi d'objets étendus appliquées à la lutte anti-drone et au véhicule autonome

Thèse dirigée par Michèle ROMBAUT

préparée au Laboratoire Grenoble Images Parole Signal Automatique

soutenue le 25 mars 2022

Jury :

Président :	Guillaume GINOLHAC	-	Professeur des universités, Université Savoie Mont Blanc
Rapporteur :	Daniel CLARK	-	Maître de conférence, Telecom SudParis
Directrice :	Michèle ROMBAUT	-	Professeure des universités, Université Grenoble Alpes
Co-directeur :	Benjamin PANNETIER	-	Ingénieur HDR, CS Group
Examinateurs :	Olivier MICHEL	-	Professeur des universités, Université Grenoble Alpes
	Frédéric DAMBREVILLE	-	Ingénieur HDR, ONERA
Invités :	Franck LI	-	Ingénieur PHD, Renault
	Maxime DEROME	-	Ingénieur PHD, Renault

Acknowledgements

Je remercie tout d'abord l'ensemble de mon jury pour leur attention lors de la lecture de ce manuscrit. Plus particulièrement, merci à mes rapporteurs, Daniel Clark et Guillaume Ginolhac, pour leurs commentaires de ce manuscrit. Mes remerciements vont également à Olivier Michel, Frédéric Dambreville et Franck Li, dont les remarques et les questions pertinentes ont permis d'améliorer ce travail.

À mes encadrants de thèse, Michèle Rombaut, Benjamin Pannetier et Maxime Derome, un simple merci ne saurait exprimer la gratitude que j'ai à votre égard. Vous m'avez épaulé pendant ces trois années, dont deux furent compliquées. Je suis heureux d'avoir pu compter sur vos qualités humaines et votre intégrité scientifique. Michèle et Benjamin, je vous remercie spécialement d'avoir gardé un calme olympien pendant la rédaction.

Merci aussi à l'équipe Fusion de Renault et à Maxime et Guillermo qui m'en ont ouvert les portes. Je remercie spécialement Maxime, Franck, Jean, Federico et Bruno pour leur curiosité et leur enthousiasme à parler de recherche autour d'un tableau pendant les pauses café. Bien entendu, Iris, Simon, Katherine, Omer et Nicole vous n'êtes pas en reste, merci pour les pauses café en extérieur. Également merci pour les soirées jeux de société et les after-work en votre compagnie!

Mes remerciements s'adressent aussi à mes collègues de l'ONERA. Vous êtes nombreux et je ne pourrais tous vous nommer ici, mais je repars avec une anecdote ou un souvenir de chacun de vous. Je remercie en particulier mes co-doctorants pour les jeux de société et les after-work qui furent un moyen efficace de relâcher la pression. Donc merci à la vieille garde pour les parties de tamalou: Marcela, Rodolphe, Rodrigo, Pierre, Soufiane, Guillaume, Alexis, Guillaume, Javiera, Benjamin, Rémy, Gaston, Philippe et Laurane. Je n'oublie pas les jeunes et vifs nouveaux doctorants Maxime, Nathan, Marius, Thomas, Quentin et ceux que j'oublie: bon courage à vous pour les années qui arrivent et les parties de tarots qui s'annoncent! Anthelme, merci d'avoir placé ton Fab-Lab sur le chemin de la fontaine à eau: c'était un bon moyen de me divertir, même si je soupçonne qu'il s'agisse d'un piège à doctorant.

Je termine ces remerciements par mes proches. Vous avez toujours fait preuve d'un soutien indéfectible pour moi, malgré les difficultés que j'ai rencontré pendant ces trois années. Je vous aime.

Contents

List of Figures

List of Tables

Acronyms

MOT	Multi Object Tracking
GNNSF	Global Nearest Neighbours Standard Filter
MHT	Multi Hypotheses Tracking
JPDAF	Joint Probability Association Filter
RFS	Random Finite Sets
FISST	FInite Set STatistics
PHD	Probability Hypothesis Density
GM-PHD	Gaussian Mixture Poisson Hypothesis Density
GLMB	Generalized Labeled Multi-Bernoulli
PMBM	Poisson Multi-Bernoulli Mixture
PMB	Poisson Multi-Bernoulli
EO	Extended Object
SCS	Star Convex Shape
RM	Random Matrix
GP	Gaussian Process
GIW	Gaussian Inverse Wishart
GGIW	Gamma Gaussian Inverse Wishart
GO-PMB	Group Object Poisson Multi-Bernoulli
GO-PHD	Group Object Probability Hypothesis density

List of Publications

This work resulted in the following contributions:

[Gue+20] Louis Guerlin et al. "Study on group target tracking to counter swarms of drones." In: *Signal Processing, Sensor/Information Fusion, and Target Recognition XXIX*. Vol. 11423. International Society for Optics and Photonics. 2020, p. 1142304.

[Gue+21] Louis Guerlin et al. "UAS Swarm detection and tracking with passive-radar." In: *2021 International Conference on Radar*. CIE. 2021.

[Gue+22] Louis Guerlin et al. "Détection et suivi multi-objets étendus par des approches Bayésiennes." In: *GRETSI'22, Accepted*. GRETSI. 2022.

Introduction

This thesis is interested in multiple object tracking using small or large datasets for the estimation of dynamic objects trajectories in an open environment. This work is a collaboration between the research teams of Renault Group, the ONERA, and the Gipsa-lab. It stands at the crossroad of two applications: the estimation of road users' kinematic parameters for Renault Group and the prevention of drone intrusions in restricted aerial space for the ONERA. For both applications, the main objective is to estimate the kinematic parameters of dynamic objects traveling in an area covered by heterogeneous sensors.

An object is a dynamic entity perceptible by one or more sensors. Several parameters characterize a dynamic object: for instance, its position, size, speed, or acceleration. They are partially observable using sensors that return noisy measurements. Hence, a filtering step is essential to reduce the measurement error and estimate non-observable parameters. For instance, if a sensor measures the successive positions of an object, its speed and acceleration can be deduced using an appropriate filter; Bayesian statistics enable the derivation of optimal filters that minimize the average mean square error under specific hypotheses.

Only rare applications focus on a single object. Often, multiple objects need to be estimated simultaneously, leading to an association problem between the measurements and the objects. The field of Multi-Object Tracking (MOT) aims to tackle this problem. Moreover, MOT algorithms need to address the sensors' flaws: they can miss objects or detect false alarms in addition to the measurement noise. Modeling heuristics or Bayesian statistics can address these issues.

A significant problem in MOT arises with extended objects. An object is extended when its size exceeds the sensor resolution and it returns multiple measurements. The number of measurements and their spatial organization relate to the shape and size of this extended object. Hence, in addition to the kinematic parameters, the extent of the object can be estimated. Once again, Bayesian statistics can help build an extent model whenever small datasets are available. However, if large datasets are available, the extent estimation process can be directly learned from the data using deep learning. The distinction between small and large datasets is essential. The available state of the art depends more and more on the quantity of data available.

This work proposes an original approach relying on the extensive use of Bayesian inference for the MOT and extended objects problems. Both the automotive and the groups of drones tracking applications rely on the Random Finite Sets (RFS) modeling of the multi-object system. However, the solutions differ since the groups of drones tracking application provides small datasets, contrary to the automotive one. Hence, to track drones we adopt a promising exhaustive Bayesian modeling, relying on the Random Matrix model to estimate extended objects. The automotive application benefits from large datasets. Thus, the extended objects can be detected using deep learning. From a certain point of view, the Bayesian modeling of the extent is equivalent to learning the data features with a deep learning algorithm on a

large dataset, except a human does it on a small dataset.

The first chapter dives into the Multi-Object Tracking problem, highlighting how some classical filters solve the MOT problem. These filters consider only single objects for the Bayesian inference, which is sub-optimal. Hence, more recent methods that consider the whole multi-object system as a random variable are introduced in the second chapter: RFS-based filters. The third chapter focuses on the extended object tracking problem: several solutions are presented, especially the random matrix model for small datasets. The two following chapters concern the groups of drones tracking and automotive perception applications. Finally, the manuscript is synthesized in a concluding chapter that proposes perspectives opened by this work.

Chapter 1

Bayesian statistics for the Multi-Object Tracking problem

Contents

1.1 Tracking objects

Tracking an object in an environment is a crucial perception task with various applications. For instance, in autonomous driving applications, it is essential to track the speed and position of surrounding road users to avoid any collision. Another example is the tracking of airborne drones to avoid illegal intrusions into restricted areas.

These two applications rely on sensors that generate noisy detections. In the figure 1.1, a radar sensor returns detections within its field of view at regular time intervals. The green dots represent the ground truth. At timestamp k, one object is missed, and several measurements are not associated with any object: they are called false alarms. In addition, at timestamp $k+1$, one object disappears while a new one appears in the field of view. This simple example covers most of the problems encountered with all sensors used for object detection. The main objective of Multi-Object Tracking (MOT) is to tackle these problems. The solution is to filter the measurements associated with each object throughout time, using *a priori* information about the object and the sensor.

This chapter introduces the MOT problem and some of its most classical solutions. We start with the single object Bayes filter used in the Global Nearest Neighbor Standard Filter

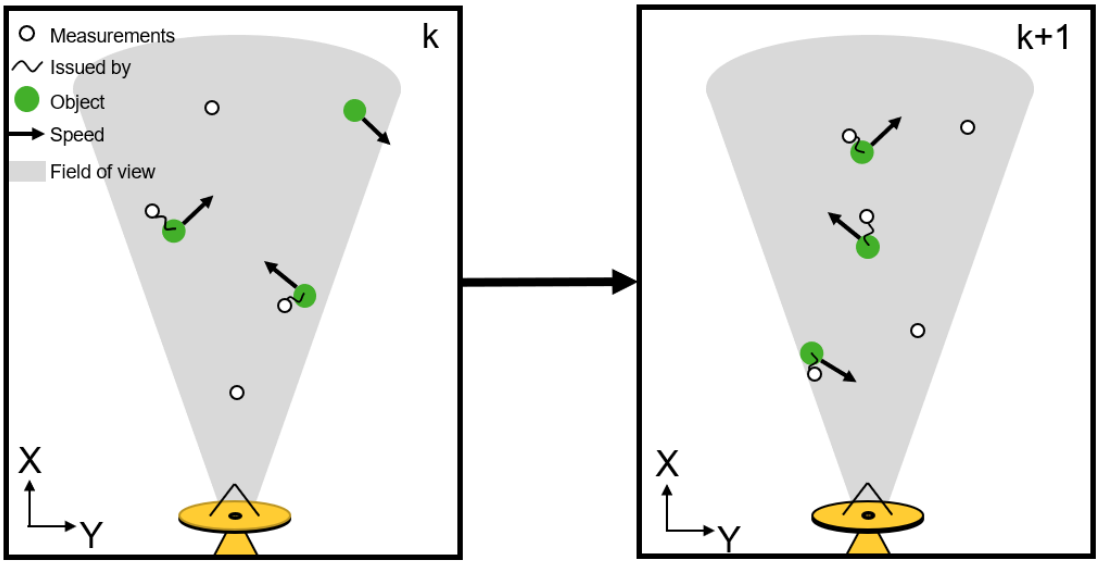


Figure 1.1: A radar sensor returns measurements issued by objects. This simple example highlights most of the problems encountered in MOT.

(GNNSF) to address the MOT problem. Then, building on the GNNSF filter, the Multi-Hypothesis Tracking (MHT) filter is presented.

1.2 Single object tracking

The figure 1.2 presents a single object evolving in the field of view of a radar sensor. Using only the measurements to follow the object is an option, but the tracking result depends on the measurement's uncertainty, leading to a high mean square error. A minimal mean square error can be achieved using Bayesian inference: it relies on prior information provided by a measurement model and an evolution model. Such filter is called a Kalman filter and is name after Kalman who was one of its primary developers [Kal60].

1.2.1 Single object measurement model

A single object at time step k is represented by a state vector $\mathbf{x}_k$. This state vector is partially observable by a sensor whose measurements are subject to noise. $\mathbf{z}_{\{1:k\}} = \{\mathbf{z}_1, ..., \mathbf{z}_k\}$ denotes the sequence of k successive measurements taken with a time interval d_t. Each measurement $\mathbf{z}_k$ can be linked to the state vector $\mathbf{x}_k$ with the linear measurement model

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k \tag{1.1}$$

where $\mathbf{H}_k$ is the observation matrix, and $\mathbf{w}_k$ is the measurement noise. $\mathbf{w}_k$ is an additive multivariate white Gaussian noise with noise covariance $\mathbf{R}_k$. A multivariate Gaussian probability

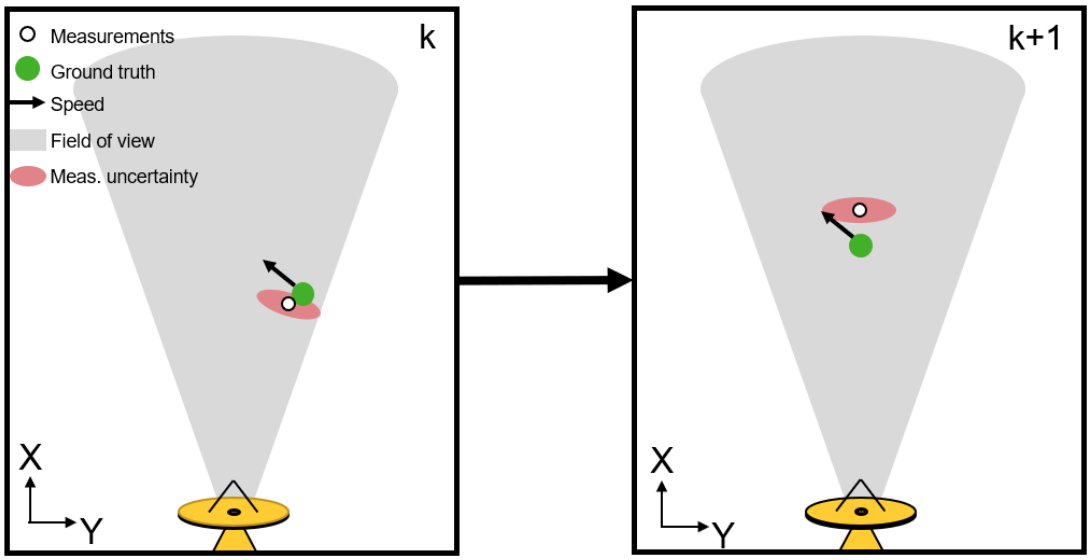


Figure 1.2: A radar sensor returning measurements issued by one single object.

density of a vector variable $\mathbf{x}$ is defined as $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})$, with

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}) \\ &= \frac{1}{\sqrt{(2\pi)^{d_\mathbf{x}}|\mathbf{P}|}} e^{-\frac{1}{2}(\hat{\mathbf{x}}-x)^T \mathbf{P}^{-1}(\hat{\mathbf{x}}-x)} \end{aligned} \tag{1.2}$$

Here $\hat{\mathbf{x}}$ stands for the expectation of $\mathbf{x}$, and $\mathbf{P}$ is the error covariance matrix. $d_\mathbf{x}$ is the number of state dimensions of $\mathbf{x}$.

According to the measurement model (1.1), the measurement likelihood is given by

$$\begin{aligned} p(\mathbf{z}_k|\mathbf{x}_k) &= p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{R}_k) \\ &= \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k) \end{aligned} \tag{1.3}$$

with $\mathbf{x}_k$ the state vector of the object we want to estimate.

The measurement model (1.3) assumes that $\mathbf{x}_k$ is known, whereas only a prior estimation of $\mathbf{x}_k$ is available: $\mathbf{x}_k$ is a random vector subject to Gaussian noise. Only its prior estimate is accessible, it is given by $\mathbf{x}_{k|k-1} = (\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$, an estimate that is fully represented by its expected value $\hat{\mathbf{x}}_{k|k-1}$, and its covariance matrix $\mathbf{P}_{k|k-1}$:

$$\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{v}_k \tag{1.4}$$

where $\mathbf{v}_k$ is an additive multivariate white Gaussian noise from the probability density with noise covariance $\mathbf{P}_{k|k-1}$. The prior estimate $\mathbf{x}_{k|k-1}$ is built from previous measurements or a chosen initialization state, and its probability density is denoted

$$p(\mathbf{x}_k|\mathbf{z}_{\{1:k-1\}}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{1.5}$$

also called the prior probability density. The notation $p(\mathbf{x}_k|\mathbf{z}_{\{1:k-1\}})$ means that the estimation of $\mathbf{x}_k$ benefits from all the information given by all the measurements between timestamp 1 and k. In addition, it is assumed that the tracked object itself is subject to a Markov evolution process, meaning that all the necessary information about $\mathbf{x}_k$ are included in the last measurement and the prior estimate of $\mathbf{x}_k$.

The minimum mean square error estimator for this problem is the Kalman filter [Kal60], it merges the *a priori* information of (1.5) with the measurement likelihood of (1.3) using Bayes equation. It results in the estimation of the posterior probability density $p(\mathbf{x}_k|\mathbf{z}_{\{1:k\}})$, such as:

$$p(\mathbf{x}_k|\mathbf{z}_{\{1:k\}}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{\{1:k-1\}})}{\int p(\mathbf{z}_k|\mathbf{u})\ p(\mathbf{u}|\mathbf{z}_{\{1:k-1\}})d\mathbf{u}} \tag{1.6}$$

Here the denominator, the marginal probability of the measurement $\mathbf{z}_k$, is intractable in practice. In fact, it is an integral over all the possible prior states of $\mathbf{x}_k$. Since it is equivalent to a multiplicative factor, it is often ignored, keeping only the numerator for the computations, such as:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{\{1:k\}}) &\propto p(\mathbf{z}_k|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{\{1:k-1\}}) \\ &= \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k)\ \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})\ \mathcal{N}(\mathbf{z}_k; \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}, \mathbf{S}) \\ &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})\ \mathcal{L}_{marg} \\ &\propto \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \end{aligned} \tag{1.7}$$

This equation lets the heart of Bayesian inference appear, computing the posterior probability density using the prior and the measurement likelihood. Here, $\mathcal{L}_{marg}$ is the marginal likelihood of measurement $\mathbf{z}_k$, given the set of prior parameters and the observation model, in the case of the single object tracking application. It is not relevant for single object tracking. The resulting density $p(\mathbf{x}_k|\mathbf{z}_{\{1:k\}})$ is a Gaussian density. Its expectation $\hat{\mathbf{x}}_{k|k}$ and error covariance matrix $\mathbf{P}_{k|k}$ are

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{P}_{k|k} &= (\mathbf{I}_{d_\mathbf{x}} - \mathbf{K}\mathbf{H}_k)\mathbf{P}_{k|k-1} \\ \mathbf{K} &= \mathbf{P}\mathbf{H}_k^T\mathbf{S}^{-1} \\ \mathbf{S} &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \end{aligned} \tag{1.8}$$

where $\mathbf{I}_{d_\mathbf{x}}$ is the $d_\mathbf{x}$ dimensional identity matrix. The explicit formula for the posterior density is

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{z}_{\{1:k\}}) &= p(\mathbf{x}_k|\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \\ &= \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \end{aligned} \tag{1.9}$$

Once the posterior probability density $p(\mathbf{x}_k|\mathbf{z}_{\{1:k\}})$ is computed, it cannot be compared to the next measurement $\mathbf{z}_{k+1}$ as is. An intermediate step that considers the object's motion model is mandatory to filter the next measurement. Thus, a new prior state needs to be estimated the posterior probability density induces a motion bias in the estimation process. The forecasting of the future state with an evolution model alleviates this problem.

1.2.2 Single object evolution model

This model is assumed to be a Markov process, meaning that all information about the object state is summarized in the last posterior probability density and the last measurement received. Of course, the evolution model is not perfect; hence it is subject to uncertainties. With a classic linear evolution model, the evolution process density is Gaussian:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{u}_k \tag{1.10}$$

with $\mathbf{F}_k$ the motion model matrix and $\mathbf{u}_k$ a Gaussian white noise sample, with covariance $\mathbf{Q}_k$. Hence, this model can be

$$f(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{F}_k \mathbf{x}_k, \mathbf{Q}_k) \tag{1.11}$$

However, $\mathbf{x}_k$ is not accessible as is, only a noisy estimation is available: posterior estimate $\mathbf{x}_{k|k}$. Based on the posterior and the motion model, the Chapman Kolmogorov equation allows computing a new prior state density, such as

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{\{1:k\}}) = \int f(\mathbf{x}_{k+1}|\mathbf{x}_k)\ p(\mathbf{x}_k|\mathbf{z}_{\{1:k\}})\ d\mathbf{x}_k \tag{1.12}$$

Integrating (1.12) over $\mathbf{x}_k$ gives the prior probability density, and for a linear Gaussian evolution model and the posterior probability density (1.9), the computations leads to

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= \mathbf{F}_k \hat{\mathbf{x}}_{k|k} \\ \mathbf{P}_{k+1|k} &= \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k \end{aligned} \tag{1.13}$$

Once the measurement model and the Markov evolution model are set, the recursive use of Chapman Kolmogorov (1.12) and Bayes (1.6) equations allows the estimation of the state vector of the tracked object. In the case described above, the prior density and the posterior density belong to the same Gaussian probability density family: it is the linear Gaussian case. This model results in the recursive block diagram schematic of the filter pictured in the figure 1.3.

Having the same type of prior and posterior probability density is highly desirable in the case of Bayesian inference: it allows having a recursive filter. When it is the case, the filter is closed under Chapman Kolmogorov and Bayes equations [VV13], [GFS19]. Prior probability densities leading to a closed formulation of the Bayes filter are called conjugate priors. With any conjugate prior density, the block diagram shown in the figure 1.3 is valid. Notable examples are: the Gaussian density is a conjugate prior to the Gaussian density, the Gamma density is a conjugate prior to the Poisson density, and Inverse Wishart density is a conjugate prior to the multivariate normal density [AOG15].

1.2.3 Application example

With a 2 dimensional space, a kinematic state vector $\mathbf{x}_k = [p_x, v_x, p_y, v_y]^T$, and a sensor returning the measurements $\mathbf{z}_k = [p_x, p_y]^T$, (p_x, p_y) denotes the Cartesian coordinates along

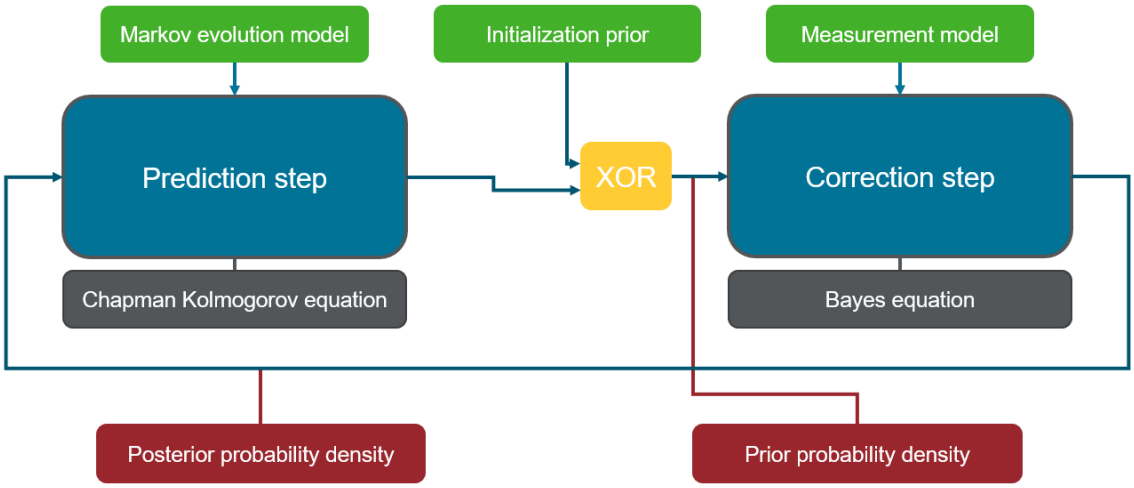


Figure 1.3: Block diagram of a recursive Bayes filter.

x and y axis, and (v_x, v_y) the speed along x axis and y axis. Hence, the observation matrix is

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1.14}$$

The sensor noise is assumed independent on each axis of the sensor

$$\mathbf{R}_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{1.15}$$

with σ_x and σ_y the measurement standard deviation along each axis.

The motion model is a constant velocity model, and $\mathbf{F}_k$ is given by

$$\mathbf{F}_k = \begin{bmatrix} 1 & d_t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.16}$$

where d_t is the period between two measurements. This model error covariance is

$$\mathbf{Q}_k = \sigma_p^2 \begin{bmatrix} \frac{d_t^3}{3} & \frac{d_t^2}{2} & 0 & 0 \\ \frac{d_t^2}{2} & d_t & 0 & 0 \\ 0 & 0 & \frac{d_t^3}{3} & \frac{d_t^2}{2} \\ 0 & 0 & \frac{d_t^2}{2} & d_t \end{bmatrix} \tag{1.17}$$

with σ_p the prediction standard deviation. This error covariance comes from the Brownian motion noise, its proof can be found in appendix C.

In order to initialize the filter, a first guess on the state vector $\mathbf{x}_k$ at $k = 1$ is mandatory, it is denoted $\mathbf{x}_{1|0}$. This is the initialization prior density $\mathcal{N}(\mathbf{x}_1; \hat{\mathbf{x}}_{1|0}, \mathbf{P}_{1|0})$. The better this first

estimate is, the faster the filter converges: the confidence in $\hat{\mathbf{x}}_{1|0}$ is given by the covariance matrix $\mathbf{P}_{1|0}$.

The main parameters are the evolution and measurement models standard deviations: σ_p, σ_x, and σ_y. Most of the tuning happens using these parameters. Another way to tune this filter is to use different evolution and measurement models depending on the problem. Some problems may require non-linear or other types of prior and posterior densities; Bayesian statistics offer a very powerful formalism that can fit them as well.

Since the Kalman filter is an optimal filter, as it minimizes the mean square root error, it is state of the art for single object tracking under linear Gaussian assumptions. It is a solid backbone for the Multi-Object Tracking (MOT) filter. In addition, it can be adapted to non-linear problems when the sensor is polar, or the motion model is non-linear. For instance, a problem can be linearized around the expected mean and the covariance matrix using a Taylor series expansion. This approach is known as the Extended Kalman filter [SSM62], [McE66], [DT97]. Other linearization techniques exist, such as the Unscented Kalman filter [JU04], but the backbone remains the Bayesian inference.

The previous problem formulation must be changed when dealing with multiple objects. It is now assumed that several objects are tracked at each time step k, and they each return at most one measurement. The notations evolve accordingly, the single object $\mathbf{x}_k$ becomes the set of n_k objects $\{\mathbf{x}_k^i\}_{i=1}^{n_k}$. Also, the measurement $\mathbf{z}_k$ received becomes the set of m_k measurements $\{\mathbf{z}_k\}_{j=1}^{m_k}$. Due to these changes, new problems arise in addition to the object's state vector estimation.

1.3 Global Nearest Neighbor Standard Filter (GNNSF)

In the figure 1.1, the typical MOT problem is illustrated with false alarms, missed objects, death, and birth of objects. The goal is to filter these phenomena and the measurement uncertainty. When dealing with MOT, one of the most critical information is knowing which object has generated which measurement. It is called the association problem. Once this question is answered, each object state vector can be estimated with a single object Kalman filter. Hence, a straightforward solution is to separate the association problem from the state estimation problem. The Global Nearest Neighbor Standard Filter (GNNSF) adopts this methodology and is a common technique, as shown by Bar-Shalom *et al.* in [BSF88] and [BSL95].

The GNNSF aims at propagating the posterior probability densities of each tracked object using the best association between the measurements and the detections. In a first approximation, all the objects are detected, and there are no false alarms, so $n_k = m_k$. The figure 1.4 illustrates this case, one radar returns three measurements corresponding to three objects. Let $\{\mathbf{x}_{k|k-1}^i\}_{i=1}^{n_k} = \{(w_{k|k-1}^i, \hat{\mathbf{x}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i)\}_{i=1}^{n_k}$ denote the set of prior parameters describing the set of tracked objects. Here, each object $\mathbf{x}_{k|k-1}^i$ is described by its weight $w_{k|k-1}^i$, its expected value $\hat{\mathbf{x}}_{k|k-1}^i$, and its error covariance matrix $\mathbf{P}_{k|k-1}^i$.

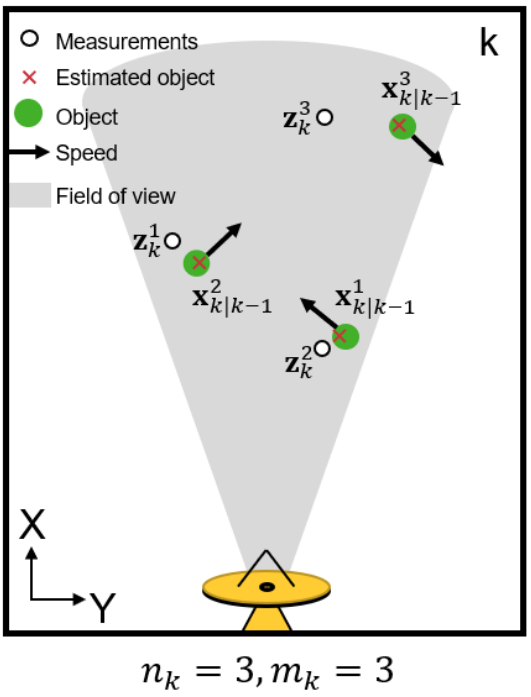


Figure 1.4: A radar sensor returning measurements issued by three objects.

When receiving the set of measurements $\{\mathbf{z}_k^j\}_{j=1}^{m_k}$, the first step is to compute the marginal likelihood of each measurement $\mathbf{z}_k^j$, given each object $\mathbf{x}_{k|k-1}^i$ prior parameters, for all $(i,j) \in [\![1, n_k]\!] \times [\![1, m_k]\!]$. Hence, the measurement likelihood is identical to the single object likelihood (1.3).

With the same Gaussian prior and the same measurement likelihood as equations (1.5) and (1.3), the association likelihood needs to be evaluated. This role is fulfilled by the marginal probability of measurement $\mathbf{z}_k$ given the prior parameters. Adapting the equation (1.7), to the MOT case, the marginal probability density $\mathcal{L}_{marg}$ is

$$\begin{aligned} p(\mathbf{z}_k^j|\mathbf{x}_k^i) &= \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}^i, \mathbf{S}) \\ &= c^{i,j} \end{aligned} \tag{1.18}$$

with the notation $c^{i,j}$ standing for the association cost. Here $\mathbf{S}$ is computed as in equation (1.8).

The marginal likelihood $c^{i,j}$ is then stored in a cost matrix where the lines represent the tracked objects, and the columns represent the measurements

$$C = \begin{bmatrix} c^{1,1} & c^{1,2} & \dots & c^{1,m_k} \\ c^{1,1} & c^{1,2} & \dots & c^{1,m_k} \\ \dots & \dots & \dots & \dots \\ c^{n_k,1} & c^{n_k,2} & \dots & c^{n_k,m_k} \end{bmatrix} \tag{1.19}$$

The goal is to find the maximum association cost for this matrix. Finding the n_k best associations between the measurements and the tracked objects is equivalent to solving

$$\Pi^* = \underset{\{(i^*,j^*)\}_{i^*=1}^{n_k}}{\operatorname{argmax}} \prod_{i^*=1}^{n_k} c^{i^*,j^*} \tag{1.20}$$

where $\Pi^* = \{(i^*, j^*)\}_{i^*=1}^{n_k}$ defines the best association map, or global association hypothesis. It is an assignment problem that can be solved with the Munkres algorithm [Mun57]: it is a deterministic algorithm that has a complexity $\mathcal{O}(n^3)$, for a $n \times n$ square association matrix [JV87]. The figure 1.5 illustrates this procedure for the example shown in the figure 1.4, adding the missed detection hypothesis to the possible association outcomes.

Once the association map is computed, all single objects Kalman filters are updated with their corresponding measurements. The resulting posterior density is noted $p(\mathbf{x}_{k|k}^1, .., \mathbf{x}_{k|k}^{n_k}, \Pi^* | \mathbf{z}_k^1, ..., \mathbf{z}_k^{m_k})$, with

$$\begin{aligned} w_{k|k}^i &= c^{(i^*,j^*)} \\ \hat{\mathbf{x}}_{k|k}^i &= \hat{\mathbf{x}}_{k|k-1}^{i^*} + \mathbf{K}(\mathbf{z}_k^{j^*} - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}^{i^*}) \\ \mathbf{P}_{k|k}^i &= (\mathbf{I}_{d_\mathbf{x}} - \mathbf{K}\mathbf{H}_k)\mathbf{P}_{k|k-1}^{i^*} \end{aligned} \tag{1.21}$$

However, in most cases, the number of measurements m_k is different from that of tracked objects n_k. The modeling should account for three cases:

- the measurement $\mathbf{z}_k^j$ is generated by object $\mathbf{x}_k^i$,
- the measurement $\mathbf{z}_k^j$ is a false alarm, and an object did not generate it,
- the object $\mathbf{x}_k^i$ is not detected and has generated no measurement.

When an object i is not detected, the prior density $\{(w_{k|k-1}^i, \hat{\mathbf{x}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i)\}$ should be kept as a posterior density, accounting for the non-detection by decreasing the prior density weight $w_{k|k-1}^i$. The corrected weight is a non-detection likelihood, and it is equal to the missed detection cost, denoted $c^{i,0}$. For instance, in the figure 1.5, it is not evident that measurement $\mathbf{z}_k^3$ was issued by the object $\mathbf{x}_k^3$: maybe the object was missed, and the measurement is a false alarm.

The calculation of the missed detection cost depends on a detection probability $p_D(\mathbf{x}_{k|k-1})$, and in a first approximation it is set to a constant value p_D. Using p_D, the association likelihood becomes

$$c^{i,j} = \begin{cases} (1 - p_D) w_{k|k-1}^i & \text{if } j = 0 \\ p_D w_{k|k-1}^i \mathcal{N}(\mathbf{z}_k^j; \hat{\mathbf{x}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i) & \text{if } j \geq 1 \end{cases} \tag{1.22}$$

and the association matrix becomes

$$C = \begin{bmatrix} c^{1,0} & 0 & ... & 0 & c^{1,1} & ... & c^{1,m_k} \\ 0 & c^{2,0} & ... & 0 & c^{2,1} & ... & c^{2,m_k} \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & c^{n_k,0} & c^{n_k,1} & ... & c^{n_k,m_k} \end{bmatrix} \tag{1.23}$$

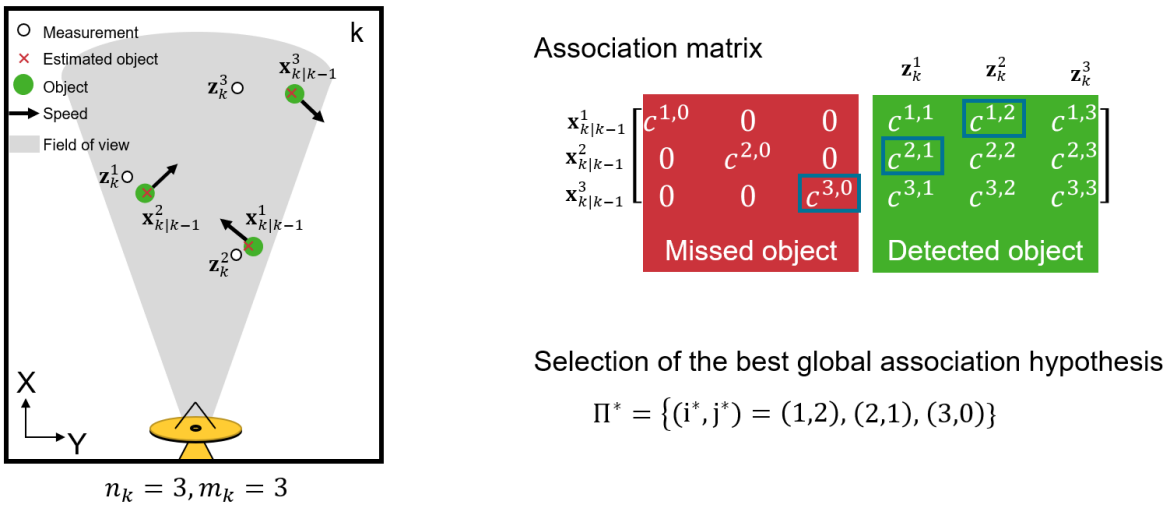


Figure 1.5: Creation of the association matrix and selection of the best global association hypothesis taking into account the missed detection hypothesis.

where the first half is a diagonal matrix for the non-detection hypothesis or missed hypothesis.

Once the best global association hypothesis Π^* is computed using the Munkres algorithm, the posterior density is computed according to the selected object-measurement couples. For a detected object, the update is similar to equation (1.21). The update of a missed object, *i.e.* an object whose association hypothesis (i^*, j^*) is equal to $(i^*, 0)$, is given by

$$\begin{aligned} w_{k|k}^{i} &= c^{i^*,0} \\ \hat{\mathbf{x}}_{k|k}^{i} &= \hat{\mathbf{x}}_{k|k-1}^{i^*} \\ \mathbf{P}_{k|k}^{i} &= \mathbf{P}_{k|k-1}^{i^*} \end{aligned} \tag{1.24}$$

that corresponds to the validation of the prior probability density.

As for the false alarm problem, a specific new object initialization procedure can filter them out. For instance, new objects can be initialized using several successive measurements, thus avoiding the initialization from an isolated false alarm.

This formulation of the GNNSF initializes new objects but never terminates any objects. It may cause a combinatorial explosion after a few recursions. Most GNNSF filters implement a merging and pruning algorithm to tackle this problem.

1.3.1 Merging and pruning

The idea is to merge the objects that are closer than threshold T_{merge} and prune the objects whose weight is smaller than a threshold T_{prune}. This algorithm takes place after the correction step.

The merging step relies on the Mahalanobis distance to discriminate near and far objects. The Mahalanobis distance between the object $\mathbf{x}_{k|k}^{i_1}$ and $\mathbf{x}_{k|k}^{i_2}$ is given by

$$d(\mathbf{x}_{k|k}^{i_1}, \mathbf{x}_{k|k}^{i_2}) = (\hat{\mathbf{x}}_{k|k}^{i_1} - \hat{\mathbf{x}}_{k|k}^{i_2})^T {\mathbf{P}_{k|k}^{i_1}}^{-1} (\hat{\mathbf{x}}_{k|k}^{i_1} - \hat{\mathbf{x}}_{k|k}^{i_2}) \tag{1.25}$$

where $\mathbf{P}_{k|k}^{i_1}$ is the covariance of object $\mathbf{x}_{k|k}^{i_1}$, and $\hat{\mathbf{x}}_{k|k}^{i_1}$ its expectation. If multiple objects are closer than a threshold T_{merge}, they are merged using a weighted average. Then these merged objects are pruned according to a threshold T_{thresh}. The full algorithm is described in the table 1.

Algorithm 1 Merging and pruning algorithm

Input: $\{(w_{k|k}^i, \hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i)\}_{i=1}^{n_{k|k}}$

Set $l = 0$
Initialize $I = \{1, ..., n_{k|k}\}$ ▷ Initialize a list of indexes
while $I \neq \emptyset$ **do** ▷ While there are indexes
  Set $l = l + 1$
  Set $j = \underset{j \in I}{\operatorname{argmax}}\ w_{k|k}^j$ ▷ Take the object with the highest weight
  Compute $L = \left\{ i \in I \mid d(\mathbf{x}_{k|k}^j, \mathbf{x}_{k|k}^i) \leq T_{merge} \right\}$ ▷ Find the objects that are close
  Compute $w^l = \sum_{i \in L} w_{k|k}^i$ ▷ Merge the close objects
  Compute $\hat{\mathbf{x}}^l = \frac{1}{w^l} \sum_{i \in L} w_{k|k}^i \hat{\mathbf{x}}_{k|k}^i$
  Compute $\mathbf{P}^l = \frac{1}{w^l} \sum_{i \in L} w_{k|k}^i \mathbf{P}_{k|k}^i$
  Set $I = I \backslash L$ ▷ Discard the merged objects
end while
Set $i = 0$
for $j = 1, \ldots, l$ **do**
  if $w^j \geq T_{thresh}$ **then** ▷ Keep the merged objects with a high weight
    Set $i = i + 1$
    Set $w_{k|k}^i = w^j$
    Set $\hat{\mathbf{x}}_{k|k}^i = \hat{\mathbf{x}}^j$
    Set $\mathbf{P}_{k|k}^i = \mathbf{P}^j$
  end if
end for
$n_{k|k} = i$

Output: $\{(w_{k|k}^i, \hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i)\}_{i=1}^{n_{k|k}}$

Using this merging and pruning strategy with the GNNSF algorithm limits the risk of combinatorial explosion. However, if the association map Π^* is erroneous, it cannot be corrected *a posteriori*. The Multi-Hypothesis Tracking algorithm propagates more than one global association hypothesis to avoid this problem.

1.4 Multi-Hypothesis Tracking

The most straightforward implementation of the Multi-Hypothesis Tracking algorithm has the same structure as the GNNSF but keeps n^H global association hypotheses after the association matrix resolution [BSL95]. The figure 1.6 illustrates this algorithm. In this figure, two global association hypotheses are kept: Π^{1^*} and Π^{2^*}. It should be noted that the estimated object, as well as the cardinality, take one more index to denote the hypothesis h: contrary to the GNNSF, where $\mathbf{x}_k^i$ is the i^{th} tracked object, now it is $\mathbf{x}_k^{h,i}$. At each correction step, a maximum number of n^H association matrices 1.23 need to be computed, then n^H association maps Π^{h^*}, $h \in [\![1, n^H]\!]$ is computed using the Murty assignment algorithm [Mur68]. Each global association hypothesis h corresponds to a number n_k^h of tracked objects.

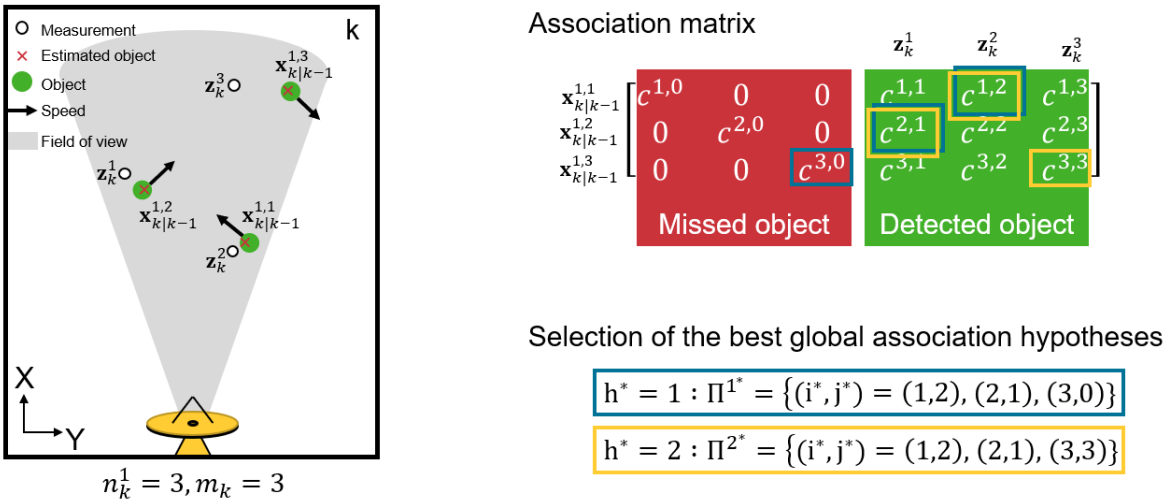


Figure 1.6: Creation of one association matrix and keeping a maximum of $n^H = 2$ global association hypotheses.

Suppose n^H association maps are selected for each global association hypothesis. In that case, $n^H \times n^H$ association maps result from the association process, and the number of hypotheses grows exponentially as the MHT recursion continues. The figure 1.7 shows this combinatorial explosion with $n^H = 2$.

n^H global association hypotheses must be selected to avoid the combinatorial explosion: the least plausible hypotheses are pruned after each correction step. The pruning process requires the definition of a global association weight. Generally, it is defined as the product of the likelihoods of association

$$\begin{aligned} W^h &= \prod_{(i^*,j^*)\in\Pi^{h^*}} c^{i^*,j^*} \\ &= \prod_{i=1}^{n_k^h} w_{k|k}^{h,i} \end{aligned} \tag{1.26}$$

for the hypothesis h, that consists of n_k^h tracked objects and the global association hypothesis

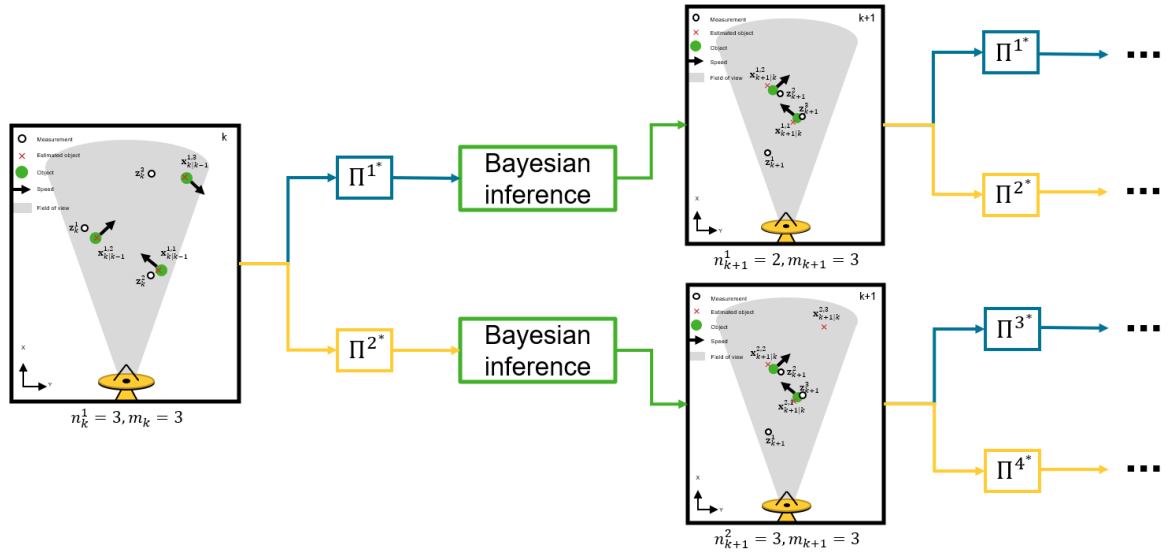


Figure 1.7: Explosion of the number of hypotheses for $n^H = 2$.

$\Pi^{h^*} = \{(i^*, j^*)\}_{i^*=1}^{n_k^h}$. The association likelihoods $w_{k|k}^{h,i}$ are computed like in equations (1.21) and (1.24).

Once the n^H best hypotheses are selected, the global hypothesis likelihood, or association map likelihoods, are normalized such as the marginal global hypothesis probability is given by

$$p(\Pi^{h^*}) = \frac{W^h}{\sum_{h'=1}^{n^H} W^{h'}} \tag{1.27}$$

The MHT algorithm is very similar to the GNNSF, with an additional loop to account for the n^H different global association hypotheses. The main advantage of this filter is that it is very close to an optimal solution, especially with a large n^H.

However, it is also much more computationally expensive than the GNNSF. Another disadvantage is the modeling of the sensor's false alarm and detection phenomenon: they are not explicitly modeled. It could be avoided if the problem was considered from a Multi-Object Tracking instead of a multiple single object tracking perspective. Indeed, the MHT and GNNSF consider the association and state estimation problems separately.

1.5 Conclusion

Other well-studied solutions to the MOT problem exist. For instance, the Joint Probability Data Association Filter (JPDAF) is a very efficient filter that limits its computational complexity by leveraging the moment matching approach. Instead of solving an association matrix, the contribution of each measurement is taken into account for each tracked object.

This algorithm is not discussed here for the sake of brevity and because it was not implemented during our work, contrary to the Random Finite Set approaches discussed in the following section.

The GNNSF, MHT, or JPDAF algorithms propose hybrid models, using both the Bayesian formalism for single object state estimation and empirical reasoning for the measurement and evolution models. While this leads to robust filters, it also raises some paradoxes in practice. Indeed, the association map is assumed to be known in the prior, while estimated simultaneously [Mah07]. In addition, the sets of tracked objects and measurements are processed like ordered sets, while they are not. Building on these observations [Mah07] proposes a fully-fledged methodology to derive genuine Bayesian Multi-Object Tracking filters.

Chapter 2

Modeling the Multi-Object Tracking problem with Random Finite Sets

Contents

2.1 The Random Finite Sets (RFS)

Instead of filtering each state vector individually with Bayesian inference, the Random Finite Set theory aims at estimating a set of state vectors all at once. Since the object of the study is a set, the measurement likelihood and evolution model are not limited to single objects as in the MHT or GNNSF algorithms. Let $X_k = \{\mathbf{x}_k^1, ..., \mathbf{x}_k^{n_k}\}$ be a set of n_k state vectors. Depending on the time step k, the composition of X_k evolves:

- new objects can appear,
- tracked objects can disappear, or die,
- most of the objects survives and are still tracked.

In that regard, the cardinality of this set behaves like a random variable. In that sense, X_k is a Random Finite Set (RFS): it is a set whose cardinality behaves like a random variable and is finite. Hence, the RFS X_k can be modeled by a prior probability density $p(X_k|Z_{\{1:k-1\}})$. This formulation of the problem was first proposed by Mahler in [Mah07].

The same reasoning can be applied to the set of m_k measurements $Z_k = \{\mathbf{z}^1, ..., \mathbf{z}^{m_k}\}$ at time k. It is the RFS of measurements that is subject to random effects

- most of the measurements corresponds to detected objects,
- some measurements correspond to false alarms,
- some tracked objects are not detected.

The cardinality of the set of measurements Z_k behaves like a random variable. As such, it can be modeled by a measurement likelihood $p(Z_k|X_k)$, with X_k the set of n_k state vectors.

With the measurement likelihood and the prior probability density of RFS Z_k and X_k, Bayes equation can be used to get the posterior probability density:

$$p(X_k|Z_{\{1:k\}}) = \frac{p(Z_k|X_k)\ p(X_k|Z_{\{1:k-1\}})}{\int p(Z_k|U)\ p(U|Z_{\{1:k-1\}})\delta U} \tag{2.1}$$

This formulation is similar to the equation (1.6), but it relies on RFS instead of vectors. The same observation can be done about Chapman Kolmogorov equation (1.12) compared to its RFS formulation.

The evolution model $f(X_{k+1}|X_k)$ can be defined for the RFS X_k, with assumptions regarding the number of new objects appearing at the next time step, the number of disappearing objects, and the number of surviving tracked objects. Using the posterior probability density computed in (2.1), the prior probability density is given by

$$p(X_{k+1}|Z_{\{1:k\}}) = \int f(X_{k+1}|X_k)\ p(X_k|Z_{\{1:k\}})\ \delta X_k \tag{2.2}$$

which is, once again, similar to equation (1.12).

These hypothesis are usual in the field of MOT. For instance, they can be found in the work of Mahler [Mah07], the work of Vo *et al.* [VM06], and the contributions of Clark *et al.* [CVV07].

Even though there are some similarities between the state vector and the RFS formulations of the Bayesian inference problem, the comparison stops here since the RFS probability density is different from that of a state vector, as highlighted by Mahler in [Mah03]. In addition, (2.2) relies on a set integration, which might seems odd: this notation is explained in the next section.

The RFS theory offers a mathematical toolbox to solve Bayes and Chapman Kolmogorov equations using set-theoretic probability densities, set evolution models, and set measurement likelihoods. In the following sections, a more formal introduction to RFS and its statistical toolbox, Finite Set STatistics (FISST), is proposed, followed by the presentation and implementation details of three state-of-the-art algorithms we have tested during our work. These algorithms were developed by Clark *et al.* in [CPV06], Vo *et al.* in [VVH16], and Williams in [Wil15b].

2.2 Formal definition

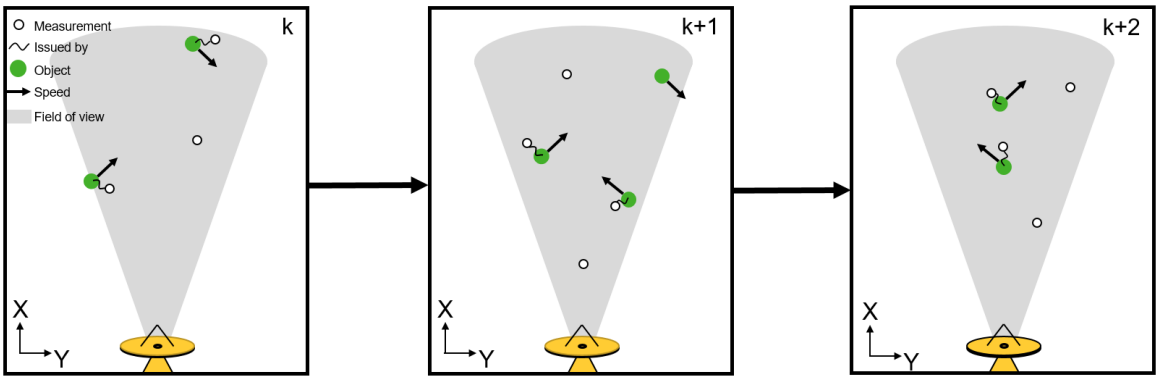


Figure 2.1: Example of three RFS whose cardinalities behave like discrete random variables.

For this section, let X be a set of vectors $\{\mathbf{x}^1, ..., \mathbf{x}^n\}$ that takes their values in S, a subset of the euclidean vector space $\mathbb{X}$. For instance, $X = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$ could be a set of 3 objects whose state vectors are only consisting of a range. Hence the state vectors $\{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$ could take their values in $S = [0, 100] \subset \mathbb{R}$, if their range varies between 0 and 100 meters.

A more formal definition of an RFS is:

Definition 2.1 (Random Finite Set)**.** A Random Finite Set (RFS) X is a random variable that takes its values from the collection of all finite subsets of n vectors, $n \in \mathbb{N}$, of the euclidean vector space $\mathbb{X}$.

Remark 2.1.1. A realization of the RFS X can be $\{\mathbf{x}^1, ..., \mathbf{x}^n\}$ or $\emptyset$. The empty set is always included in the possible realizations.

Remark 2.1.2. For MOT, the state space $\mathbb{X}$ is $\mathbb{R}^{d_\mathbf{x}}$, where $d_\mathbf{x}$ is the number of dimensions of the vector space. For instance, if the speed and position are estimated along one spatial dimension, $d_\mathbf{x} = 2$.

Remark 2.1.3. For each realization, $\{\mathbf{x}^1, ..., \mathbf{x}^n\}$ of an RFS X, the cardinality n is the random variable to estimate with Bayesian inference. Any discrete probability density can model it.

Remark 2.1.4. An RFS X can be decomposed into subsets. This makes the RFS theory a powerful tool to simply describe complex phenomena arising in MOT.

The figure 2.1 is an example of three RFS, taken at three timestamps. The RFS of measurements Z can be decomposed into the RFS of false alarms Z^{FA} and the RFS of detected objects $Z^D(X)$: $Z = Z^D(X) \uplus Z^{FA}$ with $\uplus$ the union of disjoints sets. In this simple example, the cardinality of each set behaves like a random variable.

An RFS is subject to random effects; it can be characterized by a probability density that depends on its cardinality.

Definition 2.2 (Probability density of an RFS)**.** The probability density of an RFS depends on its cardinality, as it is its random variable. For an RFS X, it can be expressed as

$$p(X) = \begin{cases} p(\emptyset) & \text{if } X = \emptyset \\ p(\{\mathbf{x}^1\}) & \text{if } X = \{\mathbf{x}^1\} \\ p(\{\mathbf{x}^1, \mathbf{x}^2\}) & \text{if } X = \{\mathbf{x}^1, \mathbf{x}^2\},\ \mathbf{x}^1 \neq \mathbf{x}^2 \\ ... & ... \end{cases} \tag{2.3}$$

Remark 2.2.1. Sets are unordered, $\{\mathbf{x}^1, \mathbf{x}^2\} = \{\mathbf{x}^2, \mathbf{x}^1\}$.

Remark 2.2.2. Following the previous remark, the probability density over X can also be expressed with vector notations

$$p(X) = \begin{cases} p(\emptyset) & \text{if } X = \emptyset \\ 1!\ p(\mathbf{x}^1) & \text{if } X = \{\mathbf{x}^1\} \\ 2!\ p(\mathbf{x}^1, \mathbf{x}^2) & \text{if } X = \{\mathbf{x}^1, \mathbf{x}^2\}, \text{ and } \mathbf{x}^1 \neq \mathbf{x}^2 \\ ... & ... \\ n!\ p(\mathbf{x}^1, ..., \mathbf{x}^n) & \text{if } X = \{\mathbf{x}^1, ..., \mathbf{x}^n\}, \text{ and } \mathbf{x}^i \neq \mathbf{x}^j\ \forall\ i \neq j \end{cases} \tag{2.4}$$

where the factorial notation distributes the probability over all $n!$ ordered arrangements of the set $\{\mathbf{x}^1, ..., \mathbf{x}^n\}$.

Remark 2.2.3. If X represents a set of tracked objects, its probability density is also called a multi-object probability density.

To get a better understanding of the RFS probability density, a first example could be the RFS X containing exactly one random vector such as

$$p(X) = \begin{cases} 0 & \text{if } X = \emptyset \\ p(\mathbf{x}) & \text{if } X = \{\mathbf{x}\} \\ 0 & \text{if } X = \{\mathbf{x}^1, ..., \mathbf{x}^n\}, \text{ and } |X| \geq 2 \end{cases} \tag{2.5}$$

with the notation $|\cdot|$ denoting the cardinality of a set, meaning the number of strictly different elements. Here $p(\mathbf{x})$ is a standard vector probability density, such as the multivariate Gaussian probability density introduced in the section 1.2.

As a second example, let X represent the RFS of independent and identically distributed random vectors. The probability density of X depends on the cardinality distribution of X, the discrete density $p(N = n)$, and the joint probability density of the vectors composing X. For instance, with the set $\{\mathbf{x}^1, ..., \mathbf{x}^n\}$,

$$p(\{\mathbf{x}^1, ..., \mathbf{x}^n\}) = n!\ p(N = n) \prod_{i=1}^{n} p(\mathbf{x}^i) \tag{2.6}$$

Indeed, because of the independence hypothesis,

$$p(\mathbf{x}^1, ..., \mathbf{x}^n) = \prod_{i=1}^{n} p(\mathbf{x}^i) \tag{2.7}$$

In order to prove that equations (2.5) and (2.6) are indeed two probability densities, their integral should be equal to one, such as $\int_S p(X)\delta X = 1$, where S is the space of the state vectors constituting X.

Definition 2.3 (Set Integral)**.** The set integral is defined such as

$$\begin{aligned} \int_S p(X)\delta X &\triangleq \sum_{n=0}^{\infty} \frac{1}{n!} \int_{\underbrace{S \times ... \times S}_{n}} p(\{\mathbf{x}^1, ..., \mathbf{x}^n\}) d\mathbf{x}^1 \dots d\mathbf{x}^n \\ &= p(\emptyset) + \int_S p(\{\mathbf{x}\}) d\mathbf{x} + \frac{1}{2} \int_{S \times S} p(\{\mathbf{x}^1, \mathbf{x}^2\}) d\mathbf{x}^1 d\mathbf{x}^2 + \dots \end{aligned} \tag{2.8}$$

where $p(X)$ is a finite set function, and S is the integration region of the underlying Euclidean space $\mathbb{X}$, where the vectors of X are taking their values.

Remark 2.3.1. $\mathbb{X}$ is $\mathbb{R}^{d_{\mathbf{x}}}$ for the MOT application, where $d_{\mathbf{x}}$ is the number of state dimensions of the state vectors composing constituting a set X.

Remark 2.3.2. If $p(X)$ is a probability density, then its integral over $S = \mathbb{R}^{d_{\mathbf{x}}}$ is 1.

Regarding the last remark, the integration of (2.5) and (2.6) are good examples. The set integral of $p(X)$, defined in (2.5), is

$$\begin{aligned} \int_{S=\mathbb{R}^{d_{\mathbf{x}}}} p(X)\delta X &= \int_S p(\{\mathbf{x}\}) d\mathbf{x} \\ &= \int_S p(\mathbf{x}) d\mathbf{x} \\ &= 1 \end{aligned} \tag{2.9}$$

since $p(\mathbf{x})$ is the state probability density of a random vector $\mathbf{z} \in \mathbb{R}^{d_\mathbf{x}}$. Thus it is a probability density.

For the RFS X of independent random vectors $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$, the equation (2.6), its integration over the entire underlying space $\mathbb{R}^{d_\mathbf{x}}$ is

$$\begin{aligned}
\int_{S=\mathbb{R}^{d_\mathbf{x}}} p(X)\delta X &= p(\emptyset) + \sum_{n=1}^{\infty} \frac{1}{n!} \int_{\underbrace{S \times ... \times S}_{n}} p(\{\mathbf{x}^1, ..., \mathbf{x}^n\}) d\mathbf{x}^1 \dots d\mathbf{x}^n \\
&= 0!\ p(N=0) + \sum_{n=1}^{\infty} \frac{1}{n!} \int_{S \times ...} n!\ p(N=n) \prod_{i=1}^{n} \left[p(\mathbf{x}^i)\right]\ d\mathbf{x}^1 \dots d\mathbf{x}^n \\
&= p(N=0) + \sum_{n=1}^{\infty} p(N=n) \prod_{i=1}^{n} \left[\int_S p(\mathbf{x}^i) d\mathbf{x}^i\right] \\
&= p(N=0) + \sum_{n=1}^{\infty} p(N=n) \\
&= 1
\end{aligned} \tag{2.10}$$

since $p(N=n)$ is a discrete probability density. Hence $p(X)$ is also a probability density.

The probability density associated with independent and identically distributed events is versatile. Depending on the choice of $p(N=n)$, this density can model a wide variety of phenomena. For instance, choosing $p(N=n)$ to be a discrete Poisson probability density leads to the RFS version of the Poisson probability density [Mah07].

The RFS-based filters we have implemented rely on three types of multi-object probability densities; they are presented below.

2.3 Multi-object probability densities

2.3.1 Poisson multi-object probability density

A multi-object Poisson probability density describes the intensity of a process that occurs homogeneously or heterogeneously in an area. It can be the number of objects per square meter that lays in the field of view of a sensor. Its proper definition is:

Definition 2.4 (Poisson density)**.** An RFS X follows a multi-object Poisson density $p_p(X)$ when its members are identically, independently distributed, and its cardinality is subject to a discrete Poisson probability density $p_\lambda(\cdot)$. For a sample $\{\mathbf{x}^1, ..., \mathbf{x}^n\}$ of the RFS $\mathbf{X}$, with $\mathbf{x}^i \neq \mathbf{x}^j,\ \forall\ 1 \leq i < j \leq n$, the multi-object probability density is

$$\begin{aligned}
p_p(X) &= n!\ p(N=n|\lambda) \times \prod_{i=1}^{n} p(\mathbf{x}^i) \\
&= e^{-\lambda} \lambda^n \prod_{i=1}^{n} p(\mathbf{x}^i)
\end{aligned} \tag{2.11}$$

where $p(\mathbf{x}^i)$ is the state probability density associated to each vector $\mathbf{x}^i$. $p(N = n|\lambda)$ is Poisson probability density with expected value λ.

Remark 2.4.1. An equivalent formulation can be obtained when defining the intensity as $I(\mathbf{x}^i) = \lambda p(\mathbf{x}^i)$: this notation is preferred because $I(\mathbf{x}^i)$ encapsulate all there is to know about a Poisson multi-object density, it is its expectation.

Remark 2.4.2. The intensity can also be interpreted as the density of objects per unit of space. Hence, the integral of the intensity $I(\mathbf{x})$, over the observation area S, gives the expected cardinality of the RFS. Thus, $\int_S I(\mathbf{x})d\mathbf{x} = \lambda$, the expected cardinality of the studied RFS.

A simple example of Poisson RFS is the set of false alarms from a Cartesian sensor, Z^{FA}. The false alarms can be modeled as homogeneously, independently, and identically distributed events occurring over the observation area S. If the sensor generates an average of N^{FA} false alarms per scan, then $I(\mathbf{x}) = \frac{N^{FA}}{S}$, a constant intensity since the clutter is considered homogeneous.

2.3.2 Multi-Bernoulli density

Bernoulli densities are helpful to model single objects with their existence probability. They are defined as:

Definition 2.5 (Bernoulli density)**.** An RFS X is subject to a Bernoulli density $p_b(X)$ when its cardinality is either 0 or 1 with the existence probability r,

$$p_b(X) = \begin{cases} 1 - r & \text{if } X = \{\emptyset\} \\ r\ p(\mathbf{x}) & \text{if } X = \{\mathbf{x}\} \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

where $p(\mathbf{x})$ is the state probability density of the singleton $\mathbf{x}$, or single object.

Remark 2.5.1. Since r can be interpreted as an existence probability, this type of RFS is interesting to model the existence of an object in the observation area S.

MOT is interested in multiple objects: hence multiple Bernoulli densities are necessary to describe a set of objects. The union of multiple Bernoulli densities is called the Multi-Bernoulli (MB) density. The Multi-Bernoulli density is very useful to model the set of tracked objects since each might have a different state probability density, and some might be non-existent.

Definition 2.6 (Multi-Bernoulli density)**.** An RFS is subject to a multi-object Multi-Bernoulli density $p_{mb}(X)$ when it can be decomposed into subsets, each subject to a single Bernoulli density, equation (2.12). A Multi-Bernoulli (MB) consisting of N Bernoulli RFS has a probability density function

$$\begin{aligned} p_{mb}(X = \{\mathbf{x}^1, ..., \mathbf{x}^n\}) &= \sum_{X^1 \uplus ... \uplus X^N = X} \prod_{i=1}^{N} p_b^i(X_i) \\ &= \prod_{i=1}^{N}(1 - r^i) \left(\sum_{1 \le i_1 < i_n \le N} \prod_{i_j = i_1}^{i_n} \frac{r^{i_j} p^{i_j}(\mathbf{x}^j)}{1 - r^{i_j}} \right) \end{aligned} \tag{2.13}$$

where X_k is a sample of X, $\uplus$ denotes the union of disjoints sets, $n \leq N$ and the summation is taken over permutations of n elements within N Bernoulli densities, $p_b^i(X_i)$. Here $p^i(\mathbf{x})$ is the state probability density associated with the ith Bernoulli density.

Remark 2.6.1. This probability density is just the product of each Bernoulli probability densities: considering that a set is unordered, this product is taken over all permutations. [Wil15b] proposes such formulation

Remark 2.6.2. As in the Bernoulli probability density, $p(X = \emptyset) = \prod_{i=1}^{N}(1 - r^i)$. This is the product of the non existence probabilities.

Remark 2.6.3. $p(X = \{\mathbf{x}^1, ...\mathbf{x}^n\}) = 0$ whenever $n > N$, since each single Bernoulli density concerns at most a singleton.

2.3.3 Multi-Bernoulli Mixture density

One Multi-Bernoulli density corresponds to one global association hypothesis between measurements and tracked objects. However, keeping more than one global association hypothesis can be a good solution to prevent wrong associations. To do so, multiple Multi-Bernoulli densities are required, leading to the Multi-Bernoulli Mixture (MBM) density.

Definition 2.7 (Multi-Bernoulli Mixture density)**.** The Multi-Bernoulli Mixture density $p_{mbm}(X)$ is a linear combination of several MB densities, defined in equation (2.13). It is expressed as

$$p_{mbm}(X) = \sum_{h=1}^{n^H} \sum_{X^1 \uplus ... \uplus X^{n^h} = X} \prod_{i=1}^{n^h} w^{h,i}\ p_b^{h,i}(X_i) \tag{2.14}$$

where each component h of the n^H terms of the Multi-Bernoulli Mixture represents one global association hypothesis. $w^{h,i}$ and $p_b^{h,i}$ denotes the weight and the Bernoulli component associated with the set X^i when the global association hypothesis h is considered.

Remark 2.7.1. $\prod_{i=1}^{n^h} w^{h,i} = W^h$ is the weight of global hypothesis h, and $\sum_{h=1}^{n^H} W^h = 1$. Thus when $n^H = 1$, the MBM density is just an MB density.

Remark 2.7.2. Here a global association hypothesis has the same signification as in the MHT filter introduced in the section 1.4.

Depending on the modeling assumptions, these densities can be used for different RFS. For instance, the Probability Hypothesis Density (PHD) filter relies only on Poisson RFS, resulting in a simple yet powerful solution to the MOT problem: its Gaussian Mixture (GM) approximation, the GMPHD filter, is a good entry point to the RFS approach [VM06]. The derivation of an RFS filter such as the GMPHD relies on the FInite Set STatistics (FISST) methodology.

2.4 Finite Set Statistics: A recipe for RFS filters

This methodology proposes a systematic approach to derive MOT filters with RFS. This chapter focuses more on the modeling steps and the choice of the prior and the posterior densities. The FISST mathematical toolbox presentation and usage is not in the scope of this thesis, the existing material in [Mah03], [Mah07], and [Mah07] covers this topic extensively.

To derive an RFS filter [Mah13] proposes a five steps methodology using FInite Set STatistics (FISST),

1. the measurement model of the sensor and the evolution model for the objects are constructed using assumptions regarding the problem at hand and RFS densities,
2. these models are converted to a Markov evolution density and a measurement likelihood, using mass belief functions,
3. the optimal Bayesian MOT filter is built, using the measurement likelihood and the evolution model,
4. using the FISST mathematical toolbox, the Bayesian MOT filter is converted to a probability generating functional,
5. the prior and posterior densities are chosen, according to simplifying assumptions.

All the filters introduced below use the same standard evolution and measurement models, with different simplifying assumptions regarding the choice of the probability densities of the RFS.

2.4.1 Standard Evolution Model Assumptions

The standard evolution model proposed [Mah03] assumes that

Assumption 2.1 (Markov evolution process)**.** Single objects are subject to a Markov evolution process, the evolution of an object from state $\mathbf{x}_k$ to state $\mathbf{x}_{k+1}$ is described by the Markov transition density $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$.

Assumption 2.2 (Surviving objects)**.** A single object $\mathbf{x}_k$ will survive at time step $k+1$ with probability $p_S(\mathbf{x}_k)$, and it will die with probability $1-p_S(\mathbf{x}_k)$. A surviving object is an object that still exists at the next step, even if the sensor does not detect it. For all tracked objects, this leads to the construction of a surviving object set $X^s_{k+1|k}$.

Assumption 2.3 (Birth object)**.** Spontaneous births might happen during the evolution process and constitute the RFS X^b.

Assumption 2.4 (Birth, death and evolution independence)**.** Birth, death, and evolution of tracked objects are assumed conditionally independent.

With these assumptions presented, the standard evolution model for an RFS based MOT filter is

$$X_{k+1|k} = X^s_{k+1|k} \uplus X^b \tag{2.15}$$

Here $\uplus$ denotes the disjoint union: a tracked object cannot survive and be born at the same time. In [Mah07], the standard evolution model is completed with an RFS describing spawning objects, but in practice, it is rarely considered because of the additional complexity; the interested reader can refer to [LSH13] for one of the few examples.

2.4.2 Standard Measurement Model Assumptions

The standard measurement model hypotheses are

Assumption 2.5 (False alarms)**.** The number of false alarms is a random variable that is subject to a Poisson density. Its expectation is a constant denoted λ_{FA}, and the false alarms spatial distribution is uniform. In addition, they are assumed to be independent. The resulting RFS Z^{FA} is subject to a Poisson RFS of constant intensity I^{FA}, such as $\int_S I^{FA} d\mathbf{x} = \lambda_{FA}$.

Assumption 2.6 (Detected objects)**.** Each object generates at most one measurement. In other words, the object $\mathbf{x}_k$ is either detected with probability $p_D(\mathbf{x}_k)$, or it is missed with probability $1 - p_D(\mathbf{x}_k)$. The detections are independent of the false alarms. The resulting set of detections is denoted $Z^D(X_{k|k-1})$.

These assumptions lead to the standard measurement model

$$Z_{k|k} = Z^D(X_{k|k-1}) \uplus Z^{FA} \tag{2.16}$$

where $\uplus$ denotes the disjoint union since a measurement cannot be an object detection and a false alarm at the same time.

These models are then derived using assumptions regarding the probability densities that model each RFS. For instance, in the Gaussian Mixture Probability Hypothesis Density filter, only Poisson densities are used for the modeling step.

2.5 The Gaussian Mixture Probability Hypothesis Density filter

The first filter introduced for MOT using the RFS theory is the Probability Hypothesis Density filter [VM06]. It relies exclusively on Poisson RFS to describe the Multi-Object Tracking problem. As presented in the section 2.3.1, the only parameter to estimate in a Poisson density is its intensity, denoted $I(\mathbf{x})$. Many variants of this algorithm exist, but the most efficient ones rely on a Gaussian Mixture (GM) for the intensity of non-homogeneous processes. A Gaussian Mixture is a linear combination of n Gaussian densities, with weights $w^i \in [0; 1]$,

$$\sum_{i=1}^{n} w^i \, \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^i, \mathbf{P}^i) \tag{2.17}$$

For instance, the tracked object posterior intensity $I_{k|k}(\mathbf{x})$ that represents $n_{k|k}$ objects, is given by the GM triplet $(w^i_{k|k}, \hat{\mathbf{x}}^i_{k|k}, \mathbf{P}^i_{k|k})$ for $i \in [\![1, n_{k|k}]\!]$; it is respectively the weight $w^i_{k|k}$, the expected state vector $\hat{\mathbf{x}}^i_{k|k}$, and the error covariance matrix $\mathbf{P}^i_{k|k}$ of the GM component. The indexing of the GM triplet is moved to the superscript position to avoid overloading the time step indication in subscript. Since $I_{k|k}(\mathbf{x})$ is an intensity, its integral should equal the expected number of tracked objects:

$$\begin{aligned}\int_S I(\mathbf{x})d\mathbf{x} &= \int_S \left(\sum_{i=1}^{n_{k|k}} w^i_{k|k}\, \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^i_{k|k}, \mathbf{P}^i_{k|k}) \right) d\mathbf{x} \\ &= \sum_{i=1}^{n_{k|k}} w^i_{k|k} \left(\int_S \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^i_{k|k}, \mathbf{P}^i_{k|k}) d\mathbf{x} \right) \\ &= \sum_{i=1}^{n_{k|k}} w^i_{k|k}\end{aligned} \tag{2.18}$$

When all components have a weight equal to $w^i_{k|k} = 1$, all the tracked objects exist, and the integral of the intensity gives the number of objects.

In addition to the standard measurement and evolution model assumptions, the GM-PHD filter adds the following hypotheses:

2.5.1 Evolution Model Assumptions

Assumption 2.7 (Birth objects)**.** It is assumed that the RFS of birth objects $X^b = \{\mathbf{x}^1, ..., \mathbf{x}^{n^b}\}$ is subject to a non homogeneous Poisson density, with intensity $I^b(\mathbf{x})$. This intensity is a Gaussian Mixture (GM), as given in (2.17). $I^b(\mathbf{x})$ is described by the set of parameters $\{(w^i_b, \hat{\mathbf{x}}^i_b, \mathbf{P}^i_b)\}_{i=1}^{n^b}$.

Assumption 2.8 (Surviving objects)**.** Surviving tracked objects $X^s_{k+1|k} = \{\mathbf{x}^1_{k+1|k}, ..., \mathbf{x}^{n^s}_{k+1|k}\}$ are also subject to a Poisson density, and the intensity is a non homogeneous Gaussian Mixture (GM), $I^s_{k+1|k}(\mathbf{x})$. Each object constituting the RFS of tracked objects is subject to a linear Gaussian evolution model as described in 1.2. The survival probability is constant, $p_S(\mathbf{x}) = p_S$.

Assumption 2.9 (Prior probability density)**.** The prior probability density $p(X_{k+1}|Z_{\{1:k\}})$, computed with the help of the posterior probability density $p(X_k|Z_{\{1:k\}})$ and the evolution model $f(X_{k+1}|X_k)$, is a Poisson probability density with a Gaussian Mixture intensity $I_{k+1|k}(\mathbf{x}) = I^s_{k+1|k}(\mathbf{x}) + I^b(\mathbf{x}) = \sum_{i=1}^{n_{k+1|k}} w^i_{k+1|k}\, \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^i_{k+1|k}, \mathbf{P}^i_{k+1|k})$.

2.5.2 Measurement Model Assumptions

Assumption 2.10 (Detected objects)**.** Objects are detected with a constant probability density $p_D(x) = p_D$. Their single target measurement likelihood is linear Gaussian when they are detected, as introduced in 1.2.

Assumption 2.11 (Posterior probability density)**.** The prior probability density $p(X_k|Z_{\{1:k\}})$, computed with the help of the prior probability density $p(X_k|Z_{\{1:k-1\}})$ and the measurement likelihood $p(Z_k|X_{k|k-1})$, is a Poisson probability density with a Gaussian Mixture intensity $I_{k|k}(\mathbf{x}) = \sum_{i=1}^{n_{k|k}} w_{k|k}^i \, \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i)$.

Using these assumptions and the FInite Set STatistics (FISST) mathematical toolbox, the GM-PHD filter is derived by solving the set-theoretic Chapman Kolmogorov (2.2) and Bayes (2.1) equations. The resulting prediction and correction recursion are detailed below.

2.5.3 Prediction with Chapman Kolmogorov equation

The prediction step aims at computing the prior density, which is the same as estimating the intensity $I_{k+1|k}(\mathbf{x})$, with components $(w_{k+1|k}^i, \hat{\mathbf{x}}_{k+1|k}^i, \mathbf{P}_{k+1|k}^i)$ for each of the $i \in [\![1, n_{k+1|k}]\!]$.

Based on the posterior Poisson probability density, $p(X_k|Z_{\{1:k\}})$, and the evolution model $f(X_{k+1}|X_k)$, the prediction step can be carried out with the Chapman Kolmogorov equation (2.2).

2.5.3.1 Surviving objects prediction equations

First, we predict the future state of the tracked objects. It is the same as predicting the future state of the Poisson intensity $I_{k|k}(\mathbf{x})$ with parameters $\{(w_{k|k}^i, \hat{\mathbf{x}}_{k|k}^i, \mathbf{P}_{k|k}^i)\}_{i=1}^{n_{k|k}}$.

$$\begin{aligned} w_{k+1|k}^i &= p_S \; w_{k|k}^i \\ \hat{\mathbf{x}}_{k+1|k}^i &= \mathbf{F}_k \hat{\mathbf{x}}_{k|k}^i \\ \mathbf{P}_{k+1|k}^i &= \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k \end{aligned} \tag{2.19}$$

where $i \in [\![1, n_{k|k}]\!]$, and the evolution matrix $\mathbf{F}_k$ and the prediction error covariance matrix $\mathbf{Q}_k$ are taken from section 1.2.

2.5.3.2 New objects prediction equations

We also need to predict the birth of new objects. It is the role of the birth Poisson intensity $I^b(\mathbf{x})$, that is represented by its parameters $(w_b^i, \hat{\mathbf{x}}_b^i, \mathbf{P}_b^i)$ for each of the $i \in [\![1, n^b]\!]$ new object component.

$$\begin{aligned} i' &= i + n_{k|k} \\ w_{k+1|k}^{i'} &= w_b^{i'} \\ \hat{\mathbf{x}}_{k+1|k}^{i'} &= \hat{\mathbf{x}}_b^{i'} \\ \mathbf{P}_{k+1|k}^{i'} &= \mathbf{P}_b^{i'} \end{aligned} \tag{2.20}$$

where the index $i' \in [\![n_{k|k}, n^b]\!]$ denotes that the birth objects are concatenated after the surviving objects.

The pseudo-code 2 gives the details of the GM-PHD prediction step.

Algorithm 2 GM-PHD prediction algorithm

Input: $(w^i_{k|k}, \hat{\mathbf{x}}^i_{k|k}, \mathbf{P}^i_{k|k}),\ \forall\ i \in [\![1, n_{k|k}]\!]$
$(w^i_b, \hat{\mathbf{x}}^i_b, \mathbf{P}^i_b),\ \forall\ i \in [\![1, n^b]\!]$
$\mathbf{F}_k,\ \mathbf{Q}_k$

for $i = 1, \ldots, n_{k|k}$ **do** ▷ Predict the state of the surviving objects
  $w^i_{k+1|k} = p_S w^i_{k|k}$
  $\hat{\mathbf{x}}^i_{k+1|k} = \mathbf{F}_k \hat{\mathbf{x}}^i_{k|k}$
  $\mathbf{P}^i_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k$
end for
for $i = 1, \ldots, n^b$ **do** ▷ Concatenate the new objects
  $i' = i + n_{k|k}$
  $w^{i'}_{k+1|k} = w^i_b$
  $\hat{\mathbf{x}}^{i'}_{k+1|k} = \hat{\mathbf{x}}^i_b$
  $\mathbf{P}^{i'}_{k+1|k} = \mathbf{P}^i_b$
end for
$n_{k+1|k} = n_{k|k} + n^b$ ▷ Update the number of objects

Output: $(w^i_{k+1|k}, \hat{\mathbf{x}}^i_{k+1|k}, \mathbf{P}^i_{k+1|k}),\ \forall\ i \in [\![1, n_{k+1|k}]\!]$

2.5.4 Correction with Bayes equation.

For the correction part, the filter aims at estimating the posterior probability density, that is represented by the Gaussian Mixture triplet $(w^i_{k|k}, \hat{\mathbf{x}}^i_{k|k}, \mathbf{P}^i_{k|k})$ for each of the component $i \in [\![1, n_{k|k}]\!]$. This Gaussian mixture is the intensity $I_{k|k}(\mathbf{x})$ corresponding to tracked objects.

The correction step is computed with the help of the measurement set $Z_k = \{\mathbf{z}_1, ..., \mathbf{z}_{m_k}\}$, and the measurement likelihood $p(Z_k|X_k)$. In addition, the prior probability density $p(X_k|Z_{\{1:k-1\}})$ is necessary, it is represented by its intensity $I_{k|k-1}(x)$, with components $(w^i_{k|k-1}, \hat{\mathbf{x}}^i_{k|k-1}, \mathbf{P}^i_{k|k-1})$.

The correction step can be decomposed into two parts: the detection and missed hypotheses.

2.5.4.1 Missed hypotheses

The missed hypotheses correspond to the validation of the prior probability density, in case there are no measurements. They rely on the detection probability p_D. This step is straight-

forward:

$$\begin{aligned} w^i_{k|k} &= w^i_{k|k-1}(1-p_D) \\ \hat{\mathbf{x}}^i_{k|k} &= \hat{\mathbf{x}}^i_{k|k-1} \\ \mathbf{P}^i_{k|k} &= \mathbf{P}^i_{k|k-1} \end{aligned} \tag{2.21}$$

These correspond to the first $[\![1, n_{k|k-1}]\!]$ components of the posterior intensity $I_{k|k}(\mathbf{x})$.

Algorithm 3 GM-PHD correction algorithm

Input: $(w^i_{k|k-1}, \hat{\mathbf{x}}^i_{k|k-1}, \mathbf{P}^i_{k|k-1}),\ \forall\ i \in [\![1, n_{k|k-1}]\!]$
$Z_k = \{\mathbf{z}_1, ..., \mathbf{z}_{m_k}\}$
$\mathbf{H}_k,\ \mathbf{R}_k$

for $i = 1, \ldots, n_{k|k-1}$ **do**
  $w^i_{k|k} = w^i_{k|k-1}(1-p_D)$
  $\hat{\mathbf{x}}^i_{k|k} = \hat{\mathbf{x}}^i_{k|k-1}$
  $\mathbf{P}^i_{k|k} = \mathbf{P}^i_{k|k-1}$
end for
for $i = 1, \ldots, n_{k-1|k}$ **do**
  for $j = 1, \ldots, m_k$ **do**
    $i' = n_{k|k-1} + (j-1)\ n_{k|k-1} + i$
    $\mathbf{S} = \mathbf{H}_k \mathbf{P}^i_{k|k-1} \mathbf{H}_k^T + \mathbf{R}^j_k$
    $\mathbf{K} = \mathbf{P}^i_{k|k-1} \mathbf{H}_k^T \mathbf{S}^{-1}$
    $w^{i'}_{k|k} = p_D\ w^i_{k|k-1}\ \mathcal{N}(\mathbf{z}_j; \mathbf{H}_k \hat{\mathbf{x}}^i_{k|k-1}, \mathbf{S})$
    $\hat{\mathbf{x}}^{i'}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_j - \mathbf{H}_k \hat{\mathbf{x}}^i_{k|k-1})$
    $\mathbf{P}^{i'}_{k|k} = (\mathbf{I}_{d_\mathbf{x}} - \mathbf{K}\mathbf{H}_k)\mathbf{P}^i_{k|k-1}$
  end for
end for
$n_{k|k} = m_k\ n_{k|k-1} + n_{k|k-1}$

Output: $(w^i_{k|k}, \hat{\mathbf{x}}^i_{k|k}, \mathbf{P}^i_{k|k}),\ \forall\ i \in [\![1, n_{k|k}]\!]$

2.5.4.2 Detection hypotheses

The detection hypotheses are computed for all the possible associations between the measurements $\mathbf{z}^j_k \in Z_k$ and the components of the prior intensity $I_{k|k-1}(\mathbf{x})$. With $n_{k|k-1}$ prior parameters triplets $\{(w^i_{k|k-1}, \hat{\mathbf{x}}^i_{k|k-1}, \mathbf{P}^i_{k|k-1})\}_{i=1}^{n_{k|k-1}}$, there are $m_k\ n_{k|k-1}$ detection hypotheses

to compute:

$$
\begin{aligned}
i' &= i + (j-1)\ n_{k|k-1} + n_{k|k-1} \\
w_{k|k}^{i'} &= p_D\ w_{k|k-1}^{i}\ \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}^{i}, \mathbf{S}) \\
\hat{\mathbf{x}}_{k|k}^{i'} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k^j - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}^{i}) \\
\mathbf{P}_{k|k}^{i'} &= (\mathbf{I}_{d_\mathbf{x}} - \mathbf{K}\mathbf{H}_k)\mathbf{P}_{k|k-1}^{i} \\
\mathbf{K} &= \mathbf{P}_{k|k-1}^{i}\mathbf{H}_k^T\mathbf{S}^{-1} \\
\mathbf{S} &= \mathbf{H}_k\mathbf{P}_{k|k-1}^{i}\mathbf{H}_k^T + \mathbf{R}_k^j
\end{aligned}
\tag{2.22}
$$

where the observation matrix $\mathbf{H}_k$ depends on the problem at hand, and the measurement error covariance matrix $\mathbf{R}_k^j$ depends on the measurement $\mathbf{z}_k^j$. Here, the linear Gaussian model of the first chapter is applied. It should be noted that the indexing $i' = i + (j-1)\ n_{k|k-1} + n_{k|k-1}$ allows the concatenation of each detection hypotheses after the missed hypotheses.

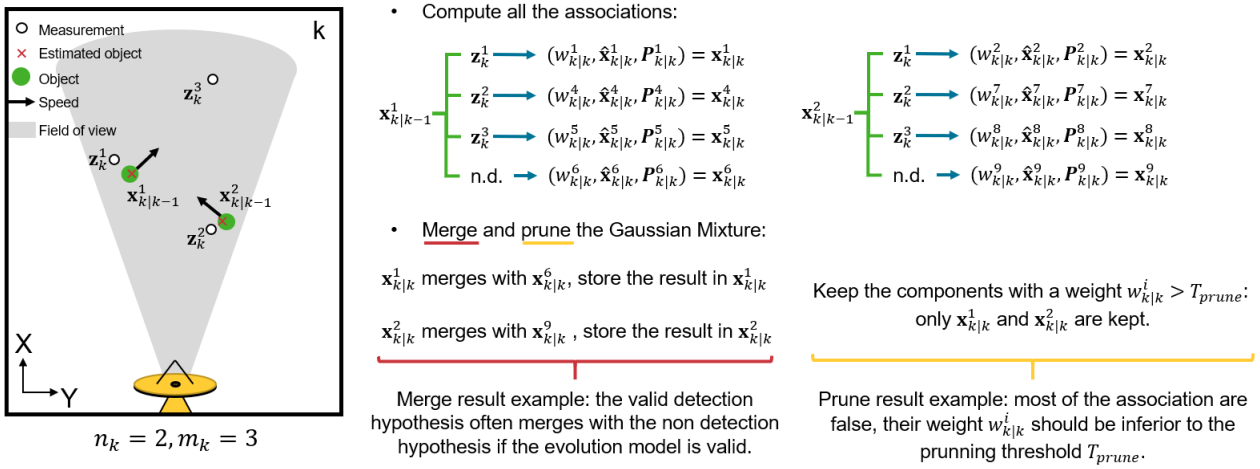


Figure 2.2: GM-PHD filter, for $n_{k|k-1} = 2$, and $m_k = 3$.

After the correction, the intensity $I_{k|k}(\mathbf{x})$ consist of $n_{k|k-1}m_k$ detection hypotheses and $n_{k|k-1}$ missed hypotheses. Hence, $n_{k|k} = (n_{k|k-1} + 1)m_k$ after the correction step. This is illustrated in figure 2.2: with $n_{k|k-1} = 3$ components for the prior intensity,and $m_k = 3$ measurements, $n_{k|k} = 12$ components are estimated in the corrected intensity $I_{k|k}(\mathbf{x})$. Hence, there is a risk of combinatorial explosion, and a pruning and merging procedure helps to limit this risk [VM06]. The combinatorial explosion can also be tackled with a gating algorithm.

The pseudo-code 3 gives the details of the correction algorithm.

The GM-PHD filter's main advantage is its low complexity, even with a time-varying number of targets. In addition, the implementation is straightforward. However, using a pruning and merging strategy is mandatory to avoid combinatorial explosion. The number of tracked objects estimated might have a high variance for low detection probability sensors. In addition, the labeling is not integrated into the filter, although Clark *et al.* proposed a solution in [CPV06].

Because of its advantages, many implementations of the GM-PHD filter exist. We developed a solution to the extended object tracking problem using this filter for the MOT problematic in [Gue+21]. This work is further developed in the chapter 4.

Another application example can be found in [Mic+17]. In this article, the GM-PHD filter is applied to autonomous driving using multiple heterogeneous sensors, which is one of the applications targeted in our work. However, the lack of label is a limitation in an autonomous driving application. This thesis considers the alternate modeling proposed by the Generalized Labeled Multi-Bernoulli (GLMB) filter to alleviate this problem.

2.6 The Generalized Labeled Multi-Bernoulli filter

When tracking a set of objects, each object is identified with a unique label. With the densities introduced in 2.3.3, a straightforward solution is to use a labeled MBM: this is the option chosen by [VV13], [VVH16] with the Generalized Labelled Multi-Bernoulli Filter (GLMB). The main idea is to augment the state space $\mathbb{X}$ of the RFS with a unique label space $\mathbb{L}$. This means that a RFS $\mathbf{X}$ subject to a MBM can be associated to a unique label sequence $L = \{(l^1, ..., l^n)\}$, such as $X = \{(\mathbf{x}^1, l^1), ..., (\mathbf{x}^n, l^n)\}$, where $l^i \neq l^j$ whenever $i \neq j$. The labeled MBM density p_{lmbm} is given by

$$p_{lmbm}(X) = \sum_{h=1}^{n^H} \left(\sum_{X^1 \uplus ... \uplus X^{n^h} = X} \left(\prod_{i=1}^{n^h} w^{h,i} p_{lb}^{h,i}(X^i) \right) \right) \tag{2.23}$$

where $p_{lb}^{h,i}(X_i)$ is a labeled Bernoulli density

$$p_{lb}^{h,i}(X_i = (\mathbf{x}^i, l^i)) = r^{h,i}.p^{h,i}(\mathbf{x}^i).\delta_{l^{h,i}}(l^i) \tag{2.24}$$

with $\delta_{l^{h,i}}(l^i)$ representing the Kronecker delta that takes the value of 1 when $l^{h,i} = l^i$ and 0 otherwise. Here h denotes a global association hypothesis. As a reminder, it is a version of the successive association maps. In the density (2.23), there are n^H global association hypothesis.

The density (2.23) is very close to the definition 2.7 of the MBM density, except for the labeling. In the case of the GLMB filter described in [VV13], the existence probability of each labeled Bernoulli density is set to $r^{h,i} = 1$. In that aspect, it can be seen as a deterministic filter: either an object exists or it does not. The resulting density is called a δ-GLMB density, it is used to model the tracked objects. As for the MBM density, a GLMB density with one global association hypothesis is an LMB density.

For instance, if the δ-GLMB density is chosen to model a posterior density representing multiple objects, then for each object $i \in [\![1, n_{k|k}^h]\!]$ of each global association hypothesis $h \in [\![1, n_{k|k}^H]\!]$, the set of variables to estimate is $(w_{k|k}^{h,i}, l_{k|k}^{h,i}, r_{k|k}^{h,i}, p_{k|k}^{h,i}(\mathbf{x}))$. In addition, if the state probability density is Gaussian, then the set of variables to estimate becomes $(w_{k|k}^{h,i}, l_{k|k}^{h,i}, r_{k|k}^{h,i}, \hat{\mathbf{x}}_{k|k}^{h,i}, \mathbf{P}_{k|k}^{h,i})$.

The GLMB filter needs additional assumptions to the standard evolution and measurement model presented in 2.4; they are presented below.

2.6.1 Evolution Model Assumptions

Assumption 2.12 (Birth objects)**.** The RFS of birth objects X^b is modeled by an LMB density consisting in n^b Bernoulli densities. Each single birth object is subject to a linear Gaussian model. Hence, the set of variables to estimate for each birth object $(\mathbf{x}_b^i, l_b^i) \in X^b$ is $(l_b^i, r_b^i, \hat{\mathbf{x}}_b^i, \mathbf{P}_b^i)$.

Assumption 2.13 (Surviving objects)**.** The set of surviving objects RFS X^s is subject to a δ-GLMB density. Each object follows a linear Gaussian model and either survives with a constant probability $p_S(\mathbf{x}^i, l^i) = p_S$ or dies with constant probability $1 - p_S$. For each association hypothesis $h \in [\![1, n^h]\!]$, and each tracked object $i \in [\![1, n^h]\!]$, the probability density parameters to estimate for each surviving object $(\mathbf{x}_s^{h,i}, l_s^{h,i})$ is $(w_s^{h,i}, l_s^{h,i}, r_s^{h,i}, \hat{\mathbf{x}}_s^{h,i}, \mathbf{P}_s^{h,i})$

Assumption 2.14 (Prior probability density)**.** The prior probability density is a δ-GLMB density, with n^H global association weights, $\{w^{h,i}\}_{i=1}^{n^h}$, n^h is the cardinality of hypothesis h. Tracked objects are subject to a linear Gaussian model. For each global association hypothesis $h \in [\![1, n^h]\!]$, and each tracked object $i \in [\![1, n^h]\!]$, the set of variables $(w_{k+1|k}^{h,i}, l_{k+1|k}^{h,i}, r_{k+1|k}^{h,i}, \hat{\mathbf{x}}_{k+1|k}^{h,i}, \mathbf{P}_{k+1|k}^{h,i})$ needs to be estimated.

2.6.2 Measurement Model Assumptions

For the measurement model, the usual measurement model hypotheses hold here:

Assumption 2.15 (Detected objects)**.** Tracked objects are detected with a constant probability $p_D(\mathbf{x}^i, l^i) = p_D$. Thus, the missed detection probability is $1 - p_D$ for any tracked object. The measurement likelihood of detected objects is linear Gaussian.

Assumption 2.16 (Posterior probability density)**.** The posterior probability density is a δ-GLMB. Hence, the same set of variables than for the prior needs to be estimated for each global association hypothesis $h \in [\![1, n^h]\!]$, and each tracked object $i \in [\![1, n^h]\!]$, $(w_{k|k}^{h,i}, l_{k|k}^{h,i}, r_{k|k}^{h,i}, \hat{\mathbf{x}}_{k|k}^{h,i}, \mathbf{P}_{k|k}^{h,i})$.

2.6.3 Joint prediction and correction strategy.

The chosen implementation relies on an efficient joint prediction and update strategy that limits the combinatorial complexity of the algorithm [VV13]. The main idea is to predict the birth of new objects while updating already tracked objects with a measurement set Z. It is illustrated in the figure 2.3: a set of measurements $Z = \{\mathbf{z}^1, \mathbf{z}^2\}$ needs to be associated with each global hypothesis of the tracked object density, here a δ-GLMB. Taking, for instance, global hypothesis h, with the objects set $X = \{\mathbf{x}^1, \mathbf{x}^2\}$, and considering the newly born objects

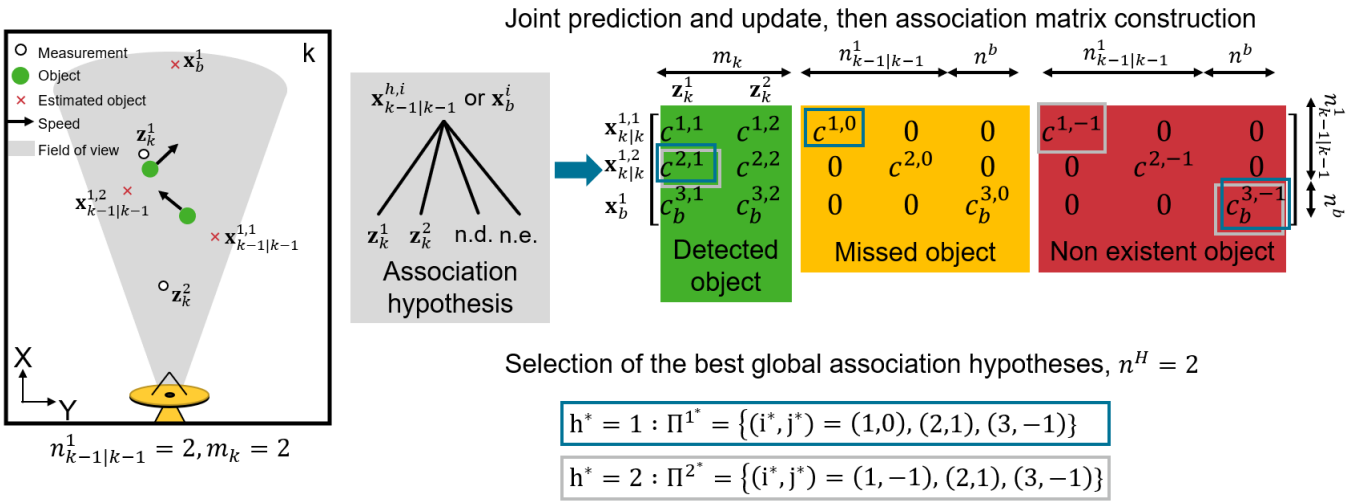


Figure 2.3: Two global associations, blue and red, were created from the term j of the two tracked objects and the newly born objects densities, $X \uplus X^b$, and the measurement set Z. n.e. stands for *non-existent*, and n.d. *not detected*.

LMB density consisting in one birth, $X^b = \{\mathbf{x}^1_b\}$, two new global associations and their weights are computed here. This process is repeated for the n^H terms of the δ-GLMB density, resulting in the creation of n^H association matrices.

The figure 2.3 shows the structure of the association matrix. For the global association history h, the first $n^h_{k-1|k-1}$ lines correspond to already tracked objects, and the n^b following lines correspond to birth objects.

2.6.3.1 Likelihood of already tracked objects

In the association matrix of figure 2.3, the cost $c^{i,j}$ for an already tracked object is computed such as:

$$c^{i,j} = \begin{cases} p_S \; p_D \; \mathcal{N}(\mathbf{z}^j_k - \hat{\mathbf{x}}^{h,i}_{k|k-1}; 0, \mathbf{S}) & \text{if } i \in [\![1, n^h]\!],\; j \in [\![1, m_k]\!] \\ p_S \; (1 - p_D) & \text{if } i \in [\![1, n^h]\!],\; j = 0 \\ 1 - p_S & \text{if } i \in [\![1, n^h]\!],\; j = -1 \end{cases} \tag{2.25}$$

where $j = 0$, and $j = -1$ stand respectively for the non-detection (n.d.), and non existence (n.e.) cases. $\mathcal{N}(\mathbf{z}^j_k - \hat{\mathbf{x}}^{h,i}_{k|k-1}; 0, \mathbf{S})$ is the likelihood of associating the measurement $\mathbf{z}^j_k$ with the tracked object $(\mathbf{x}^{h,i}_{k|k-1}, l^{h,i}_{k|k-1})$; $\mathbf{S}$ is computed as in equation (1.8) of the first chapter.

2.6.3.2 Likelihood of birth objects

The construction of the cost $c_b^{i,j}$ is similar for the new object $(\mathbf{x}_b^i, l_b^i)$

$$c_b^{i,j} = \begin{cases} r_b^{i,j}\ p_D\ \mathcal{N}(\mathbf{z}_k^j - \hat{\mathbf{x}}_b^{i,j}; 0, \mathbf{S}) & \text{if } i \in [\![n^h+1, n^h+n^b]\!],\ j \in [\![1, m_k]\!] \\ r_b^{i,j}\ (1-p_D) & \text{if } i \in [\![n^h+1, n^h+n^b]\!],\ j = 0 \\ 1 - r_b^{i,j} & \text{if } i \in [\![n^h+1, n^h+n^b]\!],\ j = -1 \end{cases} \tag{2.26}$$

with the birth existence probability $r_b^{i,j}$, and the measurement likelihood $\mathcal{N}(\mathbf{z}_k^j - \mathbf{x}_b^{i,j}; 0, \mathbf{S})$ is Gaussian, with $\mathbf{S}$ computed as in (1.8). Once again, $j \leq 0$ stands for n.d. or n.e. cases. The indexing of these costs follows the example set in the figure 2.3. It should be noted that the indexing of i starts at $n^h + 1$ since the n^h first lines of the association matrix are for the detected objects.

2.6.3.3 Selection of the best global associations

The best global association hypotheses are extracted from the association matrix using the estimator

$$\Pi^{h^*} = \underset{\{(i^*,j^*)\}_{i^*=1}^{n^h}}{\operatorname{argmax}} \prod_{i^*=1}^{n^h} c^{i^*,j^*} \tag{2.27}$$

where $\Pi^* = \{(i^*, j^*)\}_{i^*=1}^{n^h}$ defines the best association map, or global association hypothesis. It is equivalent to keeping the terms of the δ-GLMB with the highest weights, denoted W_1 and W_2 in 2.3.

The estimator (2.27) has proven to give good results on other datasets [GF+18]. Here it is coupled with Murty's algorithm to find and rank a maximum of n^H hypotheses. This is done for each global association hypothesis $h \in [\![1, n^H]\!]$, of the original δ-GLMB density. The result is a predicted and updated density consisting of $n^H \times n^H$ global association hypothesis. This density is truncated to the n^H best hypotheses to keep the solution computationally tractable.

Once Π^{h^*} is obtained, all the parameters of the global hypothesis h^* are computed. Two cases arise: either an existing object is updated, or a new object is created.

In the case where $i^* \in [\![1, n^h]\!]$, then an existing object is updated with the following

equations

$$
\begin{aligned}
l^{h,i^*}_{k+1|k+1} &= l^{h,i^*}_{k|k} \\
r^{h,i^*}_{k+1|k+1} &= 1 \\
w^{h,i^*}_{k+1|k+1} &= \begin{cases} c_{i^*,j^*} & \text{if } j^* \in [\![1, m_k]\!] \\ c_{i^*,0} & \text{if } j^* = m_k + i^* \end{cases} \\
\hat{\mathbf{x}}^{h,i^*}_{k+1|k+1} &= \begin{cases} \mathbf{F}_k \hat{\mathbf{x}}^{h,i^*}_{k|k} + \mathbf{K}(\mathbf{z}^{j^*}_k - \mathbf{H}_k \left[\mathbf{F}_k \hat{\mathbf{x}}^{h,i^*}_{k|k}\right]) & \text{if } j^* \in [\![1, m_k]\!] \\ \mathbf{F}_k \hat{\mathbf{x}}^{h,i^*}_{k|k} & \text{if } j^* = m_k + i^* \end{cases} \\
\mathbf{P}^{h,i^*}_{k+1|k+1} &= \begin{cases} (\mathbf{I}_{d_\mathbf{x}} - \mathbf{K}\mathbf{H}_k)(\mathbf{F}_k \mathbf{P}^{h,i^*}_{k|k} \mathbf{F}^T + \mathbf{Q}_k) & \text{if } j^* \in [\![1, m_k]\!] \\ \mathbf{F}_k \mathbf{P}^{h,i^*}_{k|k} \mathbf{F}^T + \mathbf{Q}_k & \text{if } j^* = m_k + i^* \end{cases}
\end{aligned}
\tag{2.28}
$$

where the first case corresponds to the detection case and the second to the missed case. All the parameters are from the linear Gaussian model of the first chapter. The joint prediction and update procedure appears clearly in these equations, for instance when computing $\hat{\mathbf{x}}^{h,i}_{k+1|k+1}$ directly from $\hat{\mathbf{x}}^{h,i}_{k|k}$.

If $i^* \in [\![n^h + 1, n^h + n^b]\!]$, then a new object is created, with the parameters:

$$
\begin{aligned}
l^{h,i^*}_{k+1|k+1} &= l^{i^*-n^h}_b \\
r^{h,i^*}_{k+1|k+1} &= 1 \\
w^{h,i^*}_{k+1|k+1} &= \begin{cases} c^{i^*,j^*}_b & \text{if } j^* \in [\![1, m_k]\!] \\ c^{i^*,0}_b & \text{if } i \in [\![n^h + 1, n^h + n^b]\!], \text{ and } j^* = m_k + i^* \end{cases} \\
\hat{\mathbf{x}}^{h,i^*}_{k+1|k+1} &= \begin{cases} \hat{\mathbf{x}}^{i^*-n^h}_b + \mathbf{K}(\mathbf{z}^{j^*}_k - \mathbf{H}_k \hat{\mathbf{x}}^{i^*}_b) & \text{if } j^* \in [\![1, m_k]\!] \\ \hat{\mathbf{x}}^{i^*-n^h}_b & \text{if } j^* = m_k + i^* \end{cases} \\
\mathbf{P}^{h,i^*}_{k+1|k+1} &= \begin{cases} (\mathbf{I}_{d_\mathbf{x}} - \mathbf{K}\mathbf{H}_k)\mathbf{P}^{i^*-n^h}_b & \text{if } j^* \in [\![1, m_k]\!] \\ \mathbf{P}^{i^*-n^h}_b & \text{if } j^* = m_k + i^* \end{cases}
\end{aligned}
\tag{2.29}
$$

where the first case corresponds to detecting a new object, and the second case corresponds to missing a new object.

In equation (2.28), and equation (2.29), only the cases with a predicted and corrected existence probability $r^{h,i}_{k+1|k+1}$ equal to 1 are shown since otherwise the updated object does not exist and is deleted.

The resulting filter is the joint prediction and update GLMB. It originates from the δ-GLMB filter of [VV13], but it is much more efficient thanks to the integration of the joint prediction and update procedure. Its pseudo-code is presented in table 4.

The main advantage of the GLMB filter is the integration of the labeling directly in the modeling step. Its complexity is $\mathcal{O}(n^H + 2(n^h + n^b)^4)$, it can be lowered to $\mathcal{O}(n^H + 2(n^h + n^b)^3)$ with the help of Gibbs sampling instead of Murty's algorithm to compute the global association hypothesis, leading to an efficient implementation when compared to the δ-GLMB filter of [VVH16].

Algorithm 4 GLMB joint prediction and update algorithm

Input: $\{\{(w_{k|k}^{h,i}, r_{k|k}^{h,i}, l^{h,i}, \hat{\mathbf{x}}_{k|k}^{h,i}, \mathbf{P}_{k|k}^{h,i})\}_{i=1}^{n^h}\}_{h=1}^{n^H}$
$\{(\hat{\mathbf{x}}_b^i, \mathbf{P}_b^i, r_b^i)\}_{i=1}^{n^b}$
$Z_k = \{\mathbf{z}_j\}_{j=1}^{m_k}$
$\mathbf{F}_k,\ \mathbf{Q}_k,\ \mathbf{H}_k,\ \mathbf{R}_k$

for $h = 1, \ldots, n^H$ **do** ▷ For each global association hypothesis h,
  for $i = 1, \ldots, n^h$ **do** ▷ compute the association matrix.
    $c_{i,-1} = 1 - p_S$ ▷ Start with each tracked object,
    $c^{i,0} = p_S\ (1 - p_D)$
    for $j = 1, \ldots, m_k$ **do**
      $c^{i,j} = p_S\ p_D\ \mathcal{N}(\mathbf{z}_k^j - \hat{\mathbf{x}}_{k|k-1}^{h,i}; 0, \mathbf{S})$
    end for
  end for
  for $i = 1, \ldots, n^b$ **do** ▷ then proceed with each new object.
    $c_b^{i,-1} = 1 - r_b^i$
    $c_b^{i,0} = r_b^i\ (1 - p_D)$
    for $j = 1, \ldots, m_k$ **do**
      $c_b^{i,j} = r_b^i\ p_D\ \mathcal{N}(\mathbf{z}_k^j - \hat{\mathbf{x}}_b^i; 0, \mathbf{S})$
    end for
  end for
  $\{\Pi^{h^*}\}_{h^*=1}^{n^H} = \text{Murty}(C, n^H)$ ▷ Compute the n^H best associations of this matrix.
  for $h^* = 1, \ldots, n^H$ **do** ▷ Create the new global association hypotheses
    $n^{h^*} = \text{card}(\{(i^*, j^*)|\ i* \in [\![1, n^h + n^b]\!],\ j* \leq (n^h + n^b + m_k)\}$
    for $i = 1, \ldots, n^{h^*}$ **do** ▷ Update the objects, using equation (2.29) or (2.28)
      Compute $(w_{k+1|k+1}^{h^*,i}, r_{k+1|k+1}^{h^*,i}, l^{h^*,i}, \hat{\mathbf{x}}_{k+1|k+1}^{h^*,i}, \mathbf{P}_{k+1|k+1}^{h^*,i})$
    end for
    $W^{h^*} = \prod_{i=1}^{n^{h^*}} w_{k+1|k+1}^{h^*,i}$ ▷ Compute the weight of this new global hypothesis.
  end for
end for

Output: $\{\{(w_{k+1|k+1}^{h,i}, r_{k+1|k+1}^{h,i}, l^{h,i}, \hat{\mathbf{x}}_{k+1|k+1}^{h,i}, \mathbf{P}_{k+1|k+1}^{h,i})\}_{i=1}^{n^h}\}_{h=1}^{(n^H)^2}$

However, the main drawback is the binary aspect of the existence probability $r_{k|k}^{h,i}$. Indeed, with a "soft" existence probability $r_{k|k}^{h,i} \in [0,1]$, one Bernoulli density describes the existence and non-existence case, while with a binary existence probability, two Bernoulli densities are necessary: one set to 1 and the other set to 0. Another disadvantage is the size of the association matrix: it is larger than for the Poisson Multi-Bernoulli Mixture filter discussed below.

The latest development around the GLMB filter introduces Gibbs sampling as well as a specific gating algorithm for the association matrix resolution [VVH16]. In [Mah19], Mahler proposes a multi-sensor integration, with dynamically moving, appearing, and disappearing sensors. Mahler acknowledges that the GLMB filter is an efficient and exact formulation of the Bayes Multi-Object Tracker in this article.

Our contribution for this filter is an implementation for MOT in an autonomous driving application [Gue+22]. We proposed to track seven types of objects using the GLMB filter, and we compared the results with other state-of-the-art solutions. We used the open-source nuScenes dataset to benchmark our implementation, [Cae+20]. This contribution is discussed more thoroughly in the chapter 5.

2.7 The Poisson Multi-Bernouli Mixture filter

The Poisson Multi-Bernoulli Mixture filter (PMBM) is a hybrid filter that relies on a Poisson RFS X^u to model presumably existing but not yet detected objects and a Multi-Bernoulli Mixture RFS X^d for detected and tracked objects [Wil15b]. The idea is to benefit from the efficient inference of Poisson RFS for undetected but predicted objects, including new objects, while using the descriptive power of the MBM RFS for detected and tracked objects with high existence probability. These RFS are supposed to be independent and included in the tracked object RFS X. The resulting multi-object probability density is denoted:

$$p_{pmbm}(X) = \sum_{X^u \uplus X^d = X} p_p(X^u) . p_{mbm}(X^d) \tag{2.30}$$

where the summation is taken over the disjoints union of the set X. As previously, the MBM allows to keep several global association hypotheses. Contrary to the GLMB filter, there are no explicit label in this density, and the existence probability $r^{h,i}$ associated to each tracked object is not deterministic: $r^{h,i} \in [0,1]$, and not $r^{h,i} \in \{0,1\}$ like in the GLMB filter.

At any time step, the MBM part of the multi-object system is determined, for each global hypothesis $1 \leq h \leq n^H$, by a set of state probability densities $\{p^{h,i}\}_{i=1}^{n^h}$, a set of existence probabilities $\{r^{h,i}\}_{i=1}^{n^h}$ and a set of global association weights $\{W^{h,i}\}_{i=1}^{n^H}$. Hence, for each global association hypothesis $h \in [\![1, n^H]\!]$, each tracked object $i \in [\![1, n^H]\!]$ is estimated with the couple of parameters $(r^{h,i}, p^{h,i})$. Regarding the Poisson probability density, only its intensity $I^u(x^u)$ needs to be estimated. For each undetected object $x^{u,i} \in X^u$ the couple of parameters $(w^{u,i}, p^{u,i})$ needs to be estimated. With a linear Gaussian model, the probability densities $p^{h,i}(\cdot)$ and $p^{u,i}(\cdot)$ are determined by their mean and covariance matrix, respectively $(\hat{\mathbf{x}}^{h,i}, \mathbf{P}^{h,i})$

and $(\hat{\mathbf{x}}^{u,i}, \mathbf{P}^{u,i})$. The resulting PMBM density can be estimated using Bayesian inference with the help of specific modeling assumptions [Wil15b].

2.7.1 Evolution Model Assumptions

Assumption 2.17 (Birth objects)**.** The RFS of birth objects $\mathbf{X^b}$ is modeled by a Poisson density with a non homogeneous intensity $I_b(x^b) = \prod_{i=1}^{n^b} w^{b,i}\mathcal{N}(x^b; \hat{\mathbf{x}}^{b,i}, P^{b,i})$; the superscript b stands for birth. Each single birth object is subject to a linear Gaussian model, and the set of variables to estimate for each birth object $x^{b,i} \in X^b$ is $(w^{b,i}, \hat{\mathbf{x}}^{b,i}, P^{b,i})$. As soon as they are initialized, birth objects are considered as unobserved objects.

Assumption 2.18 (Surviving undetected objects)**.** The RFS of undetected objects X^u evolves according a linear model, and each undetected object $\mathbf{x}^{u,i} \in X^u$ has a constant survival probability $p_S(\mathbf{x}^{u,i}) = p_S$. They are modeled by a Poisson density with a non homogeneous intensity $I^u(\mathbf{x}^u) = \prod_{i=1}^{n^u} w^{u,i}\mathcal{N}(\mathbf{x}^u; \hat{\mathbf{x}}^{u,i}, \mathbf{P}^{u,i})$; the superscript u stands for "undetected".

Assumption 2.19 (Surviving tracked objects)**.** The RFS of detected and surviving objects X^d is modeled by a Multi-Bernoulli Mixture density, with a linear Gaussian state model. The resulting set of parameters to estimate for each surviving object of a global association hypothesis $h \in [\![1, n^H]\!]$ is $\mathbf{x}^{h,i} \in X^d$ is $(r^{h,i}, \hat{\mathbf{x}}^{h,i}, \mathbf{P}^{h,i})$.

Assumption 2.20 (Prior probability density)**.** The predicted prior probability density consists of the birth objects' Poisson density, the surviving unobserved objects' Poisson density, and the surviving tracked objects' MBM density. The resulting density is the PMBM density of equation (2.30).

2.7.2 Measurement Model Assumptions

Assumption 2.21 (Unobserved objects)**.** Since unobserved objects are subjects to a Poisson density, their update procedure is similar to the one of the PHD filter. Each unobserved object gives rise to a new Bernoulli component for each measurement.

Assumption 2.22 (Detected objects)**.** The objects have a constant detection probability $p_D(x) = p_D$. Thus, the missed detection probability is given by $1 - p_D$ for any object. The measurement likelihood of detected objects is linear Gaussian.

Assumption 2.23 (Posterior probability density)**.** The posterior probability density is a PMBM density, as the prior density. Hence, the same set of variables as for the prior needs to be estimated.

With the help of these modeling assumptions, the PMBM density is propagated through the prediction and update equations. The Poisson density corresponds to unobserved objects, therefore it is very lightweight and straightforward to estimate recursively [GF+18]. Regarding the estimation of the MBM density for tracked objects, the prediction is straightforward too, resulting in the prediction of the existence probability and state parameters for each object.

2.7.3 Prediction with Chapman Kolmogorov equation

With the aforementioned hypotheses, [Wil15b] has derived a prediction procedure that is a combination of a Poisson and MBM prediction equations. Using the posterior probability density of the PMBM correction step, the goal is to compute the prior probability density. It is represented by the Poisson prior parameter $I^u_{k+1|k}(\mathbf{x}^u)$ for undetected objects, and the MBM prior parameters $(r^{h,i}_{k+1|k}, \hat{\mathbf{x}}^{h,i}_{k+1|k}, \mathbf{P}^{h,i}_{k+1|k})$ for each of the tracked objects $\mathbf{x}^{h,i}_{k+1|k}$, $i \in [\![1, n^h_{k+1|k}]\!]$, $h \in [\![1, n^H$.

2.7.3.1 Surviving objects prediction equations

With the help of the notations defined in the assumptions, the Multi-Bernoulli density prediction equations are

$$\begin{aligned} r^{h,i}_{k+1|k} &= p_S r^{h,i}_{k|k} \\ \hat{\mathbf{x}}^{h,i}_{k+1|k} &= \mathbf{F}_k \hat{\mathbf{x}}^{h,i}_{k|k} \\ \mathbf{P}^{h,i}_{k+1|k} &= \mathbf{F}_k \mathbf{P}^{h,i}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k \\ n^h_{k+1|k} &= n^h_{k|k} \end{aligned} \tag{2.31}$$

for each tracked object $i \in [\![1, n^h_{k+1|k}]\!]$. Here $\mathbf{F}_k$ and $\mathbf{Q}_k$ denote the evolution matrix and prediction error covariance matrix.

2.7.3.2 New and unobserved objects prediction equations

As for the Poisson intensity of tracked and unobserved objects:

$$\begin{aligned} w^{u,i}_{k+1|k} &= p_S w^{u,i}_{k|k} \\ \hat{\mathbf{x}}^{u,i}_{k+1|k} &= \mathbf{F}_k \hat{\mathbf{x}}^{u,i}_{k|k} \\ \mathbf{P}^{u,i}_{k+1|k} &= \mathbf{F}_k \mathbf{P}^{u,i}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k \end{aligned} \tag{2.32}$$

And the birth intensity is incorporated to the prior intensity of tracked and unobserved objects:

$$\begin{aligned} w^{u,i+n^u_{k|k}}_{k+1|k} &= w^i_b \\ \hat{\mathbf{x}}^{u,i+n^u_{k|k}}_{k+1|k} &= \hat{\mathbf{x}}^i_b \\ \mathbf{P}^{u,i+n^u_{k|k}}_{k+1|k} &= \mathbf{P}^i_b \\ n^u_{k+1|k} &= n^u_{k|k} + n^b \end{aligned} \tag{2.33}$$

for each of the n^b birth object $\mathbf{x}^i_b \in X^b$.

The pseudo-code associated to the prediction step is given in algorithm 5: it is just the straightforward application of equations (2.31), (2.32), and (2.33).

Algorithm 5 PMBM prediction algorithm

Input: $\{(r^{h,i}_{k|k}, \hat{\mathbf{x}}^{h,i}_{k|k}, \mathbf{P}^{h,i}_{k|k})\}_{i=1}^{n^h_{k|k}}$,
$\{(w^i_b, \hat{\mathbf{x}}^i_b, \mathbf{P}^i_b)\}_{i=1}^{n^b}$,
$\{(w^{u,i}_{k|k}, \hat{\mathbf{x}}^{u,i}_{k|k}, \mathbf{P}^{u,i}_{k|k})\}_{i=1}^{n^u_{k|k}}$,
$\mathbf{F}_k,\ \mathbf{Q}_k$

for $i = 1, \dots, n_{k|k}$ **do**
  $r^{h,i}_{k+1|k} = p_S r^{h,i}_{k|k}$
  $\hat{\mathbf{x}}^{h,i}_{k+1|k} = \mathbf{F}_k \hat{\mathbf{x}}^{h,i}_{k|k}$
  $\mathbf{P}^{h,i}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k$
end for
$n_{k+1|k} = n_{k|k}$
for $i = 1, \dots, n^b$ **do**
  $w^{u,i}_{k+1|k} = p_S w^{u,i}_{k|k}$
  $\hat{\mathbf{x}}^{u,i}_{k+1|k} = \mathbf{F}_k \hat{\mathbf{x}}^{u,i}_{k|k}$
  $\mathbf{P}^{u,i}_{k+1|k} = \mathbf{F}_k \mathbf{P}^{u,i}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k$
end for
for $i = 1, \dots, n^b$ **do**
  $i' = i + n^u_{k|k}$
  $w^{u,i'}_{k+1|k} = w^i_b$
  $\mathbf{x}^{u,i'}_{k+1|k} = \mathbf{x}^i_b$
  $\mathbf{P}^{u,i'}_{k+1|k} = \mathbf{P}^i_b$
end for
$n^u_{k+1|k} = n^u_{k|k} + n^b$

Output: $\{(r^{h,i}_{k+1|k}, \hat{\mathbf{x}}^{h,i}_{k+1|k}, \mathbf{P}^{h,i}_{k+1|k})\}_{i=1}^{n^h_{k+1|k}}$,
$\{(w^{u,i}_{k+1|k}, \hat{\mathbf{x}}^{u,i}_{k+1|k}, \mathbf{P}^{u,i}_{k+1|k})\}_{i=1}^{n^u_{k+1|k}}$

2.7.4 Correction with Bayes equation

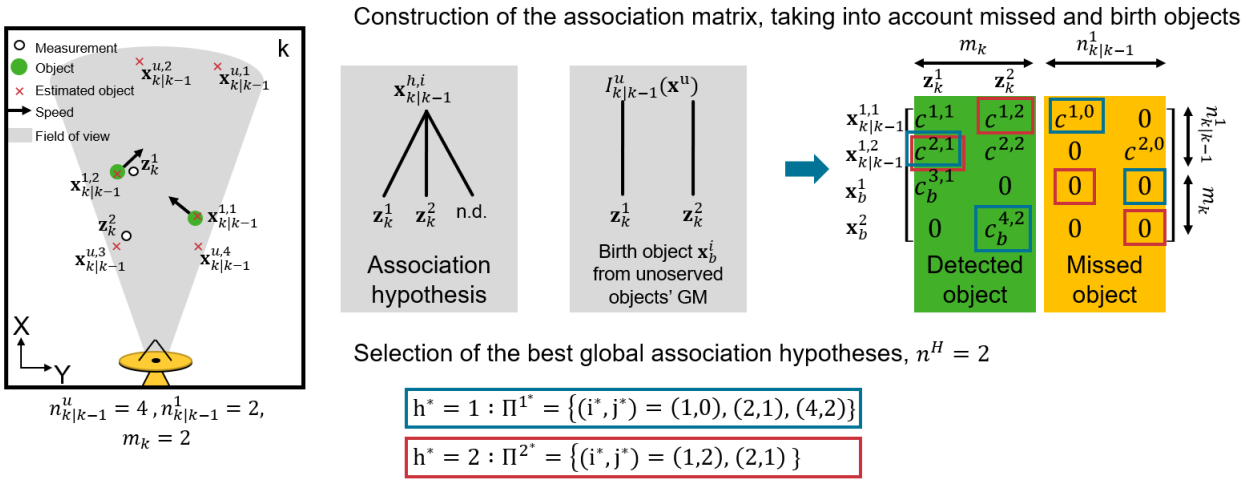


Figure 2.4: PMBM filter association matrix, for $h = 1$, $n^h_{k|k-1} = 2$, $n^u_{k|k-1} = 4$ and $m_k = 2$.

The correction step necessitates the creation of an association matrix, as shown in 2.4. In this example, two measurements are received, $Z = \{\mathbf{z}_1, \mathbf{z}_2\}$, and only one component h of the MBM density of tracked objects is considered. The Poisson density of intensity $I^u(\mathbf{x})$ corresponding to undetected objects generates one new object hypothesis per measurement. Two new global association hypotheses are highlighted in red and blue. With the measurement model assumptions of the PMBM filter, the non-existence (n.e.) hypothesis for each target does not need to be explicitly computed, thanks to the use of non-deterministic existence probabilities. This results in a more lightweight association matrix when compared to the joint GLMB filter.

2.7.4.1 Tracked objects likelihood of association

The figure 2.4 shows that three hypotheses for each tracked and detected object are computed during the update step: the detection hypothesis, the missed hypothesis, and the new object hypothesis. Thus, for each global association hypothesis $h \in [\![1, n^H]\!]$, the missed and detection costs are computed, such as

$$c^{i,j} = \begin{cases} p_D \; r^{h,i}_{k|k-1} \; \mathcal{N}(\mathbf{z}^j_k - \hat{\mathbf{x}}^{h,i}_{k|k-1}; 0, \mathbf{S}) & \text{if } i \in [\![1, n^h_{k|k-1}]\!],\ j \in [\![1, m_k]\!] \\ 1 - r^{h,i}_{k|k-1} + r^{h,i}_{k|k-1}(1 - p_D) & \text{if } i \in [\![1, n^h_{k|k-1}]\!],\ j = 0 \end{cases} \tag{2.34}$$

with index $j = 0$ denoting the missed hypothesis, and $\mathbf{S}$ computed as in (1.8).

2.7.4.2 New objects likelihood of association

The new object cost is computed using the whole undetected object Poisson intensity $I^u_{k|k-1}(\mathbf{x}^u)$ for each measurement:

$$c_b^{i,j} = \sum_{l=1}^{n^u_{k|k-1}} \left(w^{u,l}_{k|k-1}\, p_D\, \mathcal{N}(\mathbf{z}^j_k - \hat{\mathbf{x}}^{u,l}_{k|k-1}; 0, \mathbf{S}) \right) + I^{FA} \quad \text{for } i,j \in [\![1 + n^h_{k|k-1}, n^b + n^h_{k|k-1}]\!] \times [\![1, m_k]\!] \tag{2.35}$$

2.7.4.3 Selection of the best global associations

Once the association matrix is built for the global hypothesis $h \in [\![1, n^H]\!]$, the best global association hypothesis are extracted using the estimator (2.27) of the GLMB filter,

$$\Pi^{h^*} = \underset{\{(i^*,j^*)\}_{i^*=1}^{n^h}}{\operatorname{argmax}} \prod_{i^*=1}^{n^h} c^{i^*,j^*}$$

The best global association hypotheses Π^{h^*} are ranked using Murty's algorithm [Mur68]: in the example of 2.4 two new global association hypotheses for component h are kept. This process is repeated for all the $1 \leq h \leq n^H$ components of the MBM density, and as previously, the updated MBM density is truncated to contain at most n^H best global association hypotheses.

Once $\Pi^{h^*} = \{(i*, j*)\}_{i^*=1}^{n^h}$ is obtained, all the parameters of the global hypothesis h are computed.

If $i* \in [\![1, n^h_{k|k-1}]\!]$, then an already detected and tracked object is selected by the association map (i^*, j^*). The updated set of parameters are

$$\begin{aligned}
\mathbf{K} &= \mathbf{P}^{h,i^*}_{k|k-1}\mathbf{H}_k^T\mathbf{S}^{-1} \\
\mathbf{S} &= \mathbf{H}_k\mathbf{P}^{h,i^*}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \\
r^{h^*,i^*}_{k|k} &= \begin{cases} 1 & \text{if } j^* \in [\![1, m_k]\!] \\ \frac{r^{h,i^*}_{k|k-1}(1-p_D)}{1-r^{h,i^*}_{k|k-1}+r^{h,i^*}_{k|k-1}(1-p_D)} & \text{if } j^* = 0 \end{cases} \\
\hat{\mathbf{x}}^{h^*,i^*}_{k|k} &= \begin{cases} \hat{\mathbf{x}}^{h,i^*}_{k|k-1} + \mathbf{K}(\mathbf{z}^{j^*}_k - \mathbf{H}_k\hat{\mathbf{x}}^{h,i^*}_{k|k-1}) & \text{if } j^* \in [\![1, m_k]\!] \\ \hat{\mathbf{x}}^{h,i^*}_{k|k-1} & \text{if } j^* = 0 \end{cases} \\
\mathbf{P}^{h^*,i^*}_{k|k} &= \begin{cases} (\mathbf{I}_{d_\mathbf{x}} - \mathbf{K}\mathbf{H}_k)(\mathbf{P}^{h,i^*}_{k|k-1}) & \text{if } j^* \in [\![1, m_k]\!] \\ \mathbf{P}^{h,i^*}_{k|k-1} & \text{if } j^* = 0 \end{cases}
\end{aligned} \tag{2.36}$$

Otherwise, when $i^* \in [\![n^h_{k|k-1}+1, n^h_{k|k-1}+n^b]\!]$, and $j^* \in [\![1, m_k]\!]$, the association map points to the creation of a new Bernoulli component. This new tracked object is created comparing

all the undetected hypotheses of X^u to each measurement $\mathbf{z}_k^{j^*}$ such as

$$\begin{aligned}
r_{k|k}^{h^*,i^*} &= \frac{\sum_{l=1}^{n_{k|k-1}^u} \left(w_{k|k-1}^{u,l}\ p_D\ (\mathbf{z}_k^{j^*} - \hat{\mathbf{x}}_{k|k-1}^{u,l}; 0, \mathbf{S}) \right)}{\sum_{l=1}^{n_{k|k-1}^u} \left(w_{k|k-1}^{u,l}\ p_D\ \mathcal{N}(\mathbf{z}_k^{j^*} - \hat{\mathbf{x}}_{k|k-1}^{u,l}; 0, \mathbf{S}) \right) + I^{FA}} \\
\hat{\mathbf{x}}_{k|k}^{h^*,i^*} &= \frac{\sum_{l=1}^{n_{k|k-1}^u} c_b^{l,j^*} \mathbf{y}^l}{\sum_{l=1}^{n_{k|k-1}^u} c_b^{l,j^*}} \\
\mathbf{P}_{k|k}^{h^*,i^*} &= \frac{\sum_{l=1}^{n_{k|k-1}^u} c_b^{l,j^*} \mathbf{P}^l + (\hat{\mathbf{x}}_{k|k}^{h^*,i^*} - \mathbf{y}^l)(\hat{\mathbf{x}}_{k|k}^{h^*,i^*} - \mathbf{y}^l)^T}{\sum_{l=1}^{n_{k|k-1}^u} c_b^{l,j^*}} \\
\mathbf{S}^l &= \mathbf{H}_k \mathbf{P}_{k|k-1}^{u,l} \mathbf{H}_k^T + \mathbf{R}_k^{j^*} \\
\mathbf{K}^l &= \mathbf{P}_{k|k-1}^{u,l} \mathbf{H}_k^T \mathbf{S}^{-1} \\
\mathbf{P}^l &= \mathbf{P}_{k|k-1}^{u,l} - \mathbf{K}^l \mathbf{H}_k \mathbf{P}_{k|k-1}^{u,l} \\
\mathbf{y}^l &= \hat{\mathbf{x}}_{k|k-1}^{u,l} - \mathbf{K}^l (\mathbf{z}_k^{j^*} - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}^{u,l})
\end{aligned} \tag{2.37}$$

As for $n_{k|k}^h$, it is equal to the cardinality of Π^{*h} after the update.

Finally, the Poisson density $I^u(\mathbf{x})$ corresponding to undetected yet tracked objects needs to be updated. Since these objects are assumed unobserved, the update procedure corresponds to the validation of the predicted intensity:

$$\begin{aligned}
w_{k|k}^{u,i} &= (1 - p_D)\ w_{k|k-1}^{u,i} \\
\hat{\mathbf{x}}_{k|k}^{u,i} &= \hat{\mathbf{x}}_{k|k-1}^{u,i} \\
\mathbf{P}_{k|k}^{u,i} &= \mathbf{P}_{k|k-1}^{u,i} \\
n_{k|k}^u &= n_{k|k-1}^u
\end{aligned} \tag{2.38}$$

where $i \in [\![1, n_{k|k-1}^u]\!]$. The algorithm 6 gives the full correction procedure.

The resulting filter is quite efficient because it benefits from the Poisson density to solve a smaller association matrix compared to the GLMB filter. Moreover, the MBM part benefits from existence probabilities that are non-deterministic compared to the deterministic δ-GLMB density.

In the literature, this filter has drawn a lot of attention because of the hybrid approach it proposes for new object initialization and the non-deterministic choice for the existence probability $r^{h,i}$ that differentiates it from the GLMB filter. Amongst the latest development, [Xia+19] proposes a multi-scan implementation of the PMBM, which is refined in [GF+20] with the two Trajectories Poisson Multi-Bernoulli (TPMB) filter. Amongst the contributions, the sets of trajectories are a very interesting extension to the FISST mathematical toolbox.

Our contribution aims another application, we have implemented this filter for group objects in [Gue+20]. This extension of the original work of [Wil15b] necessitates the presentation of the extended object tracking problem in the next chapter.

Algorithm 6 PMBM correction algorithm

Input: $\{(r^{h,i}_{k|k-1}, \hat{\mathbf{x}}^{h,i}_{k|k-1}, \mathbf{P}^{h,i}_{k|k-1})\}_{i=1}^{n^h_{k|k-1}}$,
$\{(w^{u,i}_{k|k-1}, \hat{\mathbf{x}}^{u,i}_{k|k-1}, \mathbf{P}^{u,i}_{k|k-1})\}_{i=1}^{n^u_{k|k-1}}$,
$\{\mathbf{z}^j_k, \mathbf{R}^j_k\}_{j=1}^{m_k}$

for $h = 1, \dots, n^H$ **do**
  for $i = 1, \dots, n^h_{k|k-1}$ **do** ▷ Creation of the association matrix
    for $j = 0, \dots, m_k$ **do** ▷ Compute the detection costs
      Compute $c^{i,j}$ according to (2.34)
    end for
    for $j = 1, \dots, m_k$ **do** ▷ Compute new objects' birth costs
      Compute $c_b^{i+n^h_{k|k},j}$ according to (2.35)
    end for
  end for
  Compute $\{\Pi^{h^*}\}_{h^*=1}^{n^H} = \text{Murty}(C, n^H)$ ▷ Extract the n^H best associations
  for $h^* = 1, \dots, n_H$ **do** ▷ Create the n_H new global association hypotheses
    for $(i^*, j^*) \in \Pi^{h^*}$ **do**
      if $i^* < n^h_{k|k-1}$ **then**
        Compute $(r^{h^*,i^*}_{k|k}, \hat{\mathbf{x}}^{h^*,i^*}_{k|k}, \mathbf{P}^{h^*,i^*}_{k|k})$ using (2.36)
      else
        Compute $(r^{h^*,i^*}_{k|k}, \hat{\mathbf{x}}^{h^*,i^*}_{k|k}, \mathbf{P}^{h^*,i^*}_{k|k})$ using (2.37)
      end if
    end for
    $n^{h^*}_{k|k} = card(\Pi^{h^*})$ ▷ Update the cardinality of the new hypothesis
  end for
end for
for $i = 1, \dots, n^u_{k|k-1}$ **do** ▷ Update the undetected objects intensity
  Compute $(w^{u,i}_{k|k}, \hat{\mathbf{x}}^{u,i}_{k|k}, \mathbf{P}^{u,i}_{k|k})$ using (2.38)
  $n^u_{k|k} = n^u_{k|k-1}$
end for

Output: $\{(r^{h,i}_{k|k}, \hat{\mathbf{x}}^{h,i}_{k|k}, \mathbf{P}^{h,i}_{k|k})\}_{i=1}^{n^h_{k|k}}$,
$\{(w^{u,i}_{k|k}, \hat{\mathbf{x}}^{u,i}_{k|k}, \mathbf{P}^{u,i}_{k|k})\}_{i=1}^{n^u_{k|k}}$

2.8 Conclusion

In this chapter, we have solved the MOT problem for isolated objects, objects that return at most one measurement, using only Bayesian inference with the Random Finite Sets theory. It proposes to use Bayesian inference on Random Finite Sets (RFS) instead of single state vectors. With the help of the FISST mathematical toolbox, set operations such as set derivation or integration are possible, meaning that the derivation of a Bayesian filter for sets is achievable. Contrary to the previous chapter, the objects association and estimation problems are jointly solved with RFS-based filters, leading to optimal solutions for the MOT problem.

Building on the standard measurement and evolution models for RFS, we have proposed the study of three filters: the PHD filter, the GLMB filter, and the PMBM filter. They represent the state of the art of RFS-based MOT. We implemented and tested each one of them for this thesis in [Gue+21], [Gue+22], and [Gue+20].

The main problem we did not address in this first chapter is the question of extended objects, *i.e.*, objects that return more than one measurement. Indeed, one of the assumptions of the standard measurement model is that an object returns at most one measurement per acquisition. This assumption is too restrictive for the applications aimed in this work. The purpose of the next chapter is to address this problem. This will lead to the multiple extended object tracking applications of chapters 4 and 5.

Chapter 3

State models for extended objects

Contents

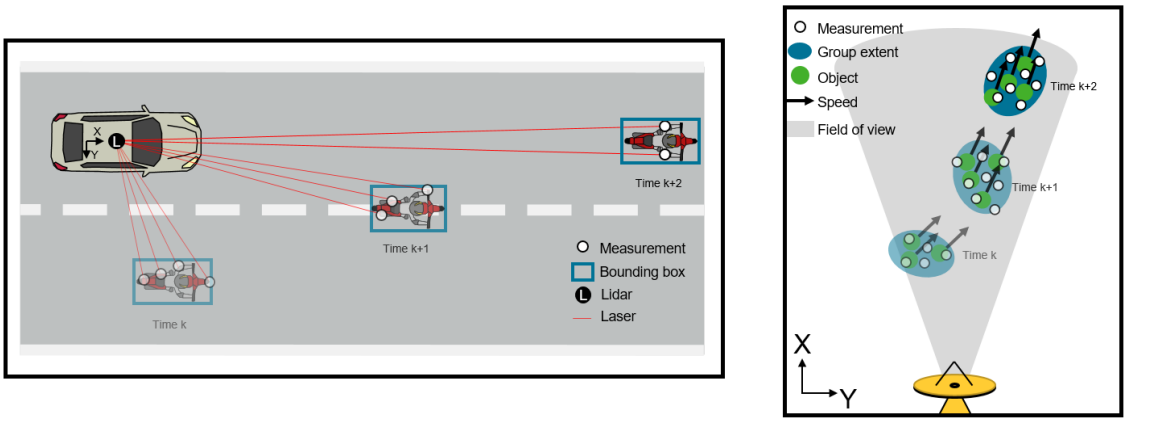
3.1 Detecting and estimating extended objects

The previous chapter presented the issue of Multi-Object Tracking (MOT) with a restrictive measurement model: tracked objects only returned at most one measurement. It is the case when tracked objects are far from the sensor or their size does not exceed the cell resolution of the sensor. The cell resolution of a sensor is the minimal size of an object to be detected. In this chapter, several solutions to this problem are presented, then we will focus on the Random Matrix (RM) model and the modeling of group objects.

3.1.1 Definitions and available solutions

In practice, several applications question this assumption: the objects might return more than one measurement per sensor scan. These objects are called Extended Objects (EO). This work considers two types of EOs: rigid EOs and non-rigid EOs.

First, rigid Extended Objects (EOs) return multiple measurements per sensor scan, while their shape remains the same through time. Rigid EOs are encountered both in automotive



a) A rigid extended object in an automotive application

b) A group object, or non-rigid extended object in a drone tracking application

Figure 3.1: The two types of extended objects encountered in two applications targeted by this work.

and drone tracking applications. For instance, as pictured in the figure 3.1, a lidar often returns numerous data points on each vehicle present in a scene. Similarly, a drone larger than the cell resolution of a radar sensor can also return several data points. Since a vehicle and a drone are rigid objects, their shape does not evolve throughout time. Most of the objects encountered in the automotive world have a fixed size and shape; they are rigid.

On the contrary, non-rigid EOs returns several measurements per acquisition, while their shape evolves through time. They might arise because of the apparent shape-changing characteristic of an object through time, such as an articulated bus. The other type of non-rigid EOs consists of multiple objects that evolve as a group, such as swarms of drones. Hence, a group object is a particular case of non-rigid EO.

Group objects are non-rigid EOs composed of multiple objects evolving in close formation and sharing the same kinematic parameters. Here "close" is defined with respect to the problem scale and the sensor resolution. Tracking the whole group instead of each object is a good practice to reduce the size of the association matrix and lower the complexity of the algorithms discussed in chapters 1 and 2.

Mostly rigid EOs are encountered in the automotive application, and the classes of interest are cars, bicycles, pedestrians, trucks, buses, motorcycles, and trailers. Thus there is a high diversity of shapes and sizes. However, the ground truth associated with the raw data is often composed of bounding boxes for all the classes, *i.e.*, all the objects are represented by straight rectangular prisms, as illustrated in the figure 3.1. Hence, for the automotive application, an EO has a right rectangular prism extent, whether it is a pedestrian or a car.

As pictured in the figure 3.1, groups of drones are primarily non-rigid EOs. Even a single drone can be seen as a non-rigid EO because of its rotors. Since the shape of a group is somewhat a fuzzy concept, we are restricting it to a smooth convex shape that circles the

objects within it. Hence, the extent's shape to estimate is not a bounding box but rather an elliptical shape, highlighting a first difference with the automotive application.

The applications targeted in this manuscript differ on two points. First, the EOs have different extents shapes. Furthermore, the quantity of data available to validate the algorithms is different between both applications. Indeed, the automotive application provides large datasets, while only a few small datasets are available for the drone tracking application. This is a significant difference since the current state of the art for detecting EOs consists of deep learning algorithms that necessitate large datasets for the learning step. While it is possible to use deep learning for the automotive application, it is not for the groups of drones tracking application.

The usage of deep learning for detecting EOs and estimating their shape will be discussed in the chapter 5. This chapter focuses on the case where only a few data are available, which prevents us from using deep learning algorithms. To bridge the gap between the quantity of data available and the expected EOs shape, we will rely on a modeling-based approach for the groups of drones tracking application. Amongst the modeling-based approaches, the state of the art consists of two propositions: the Random Matrix (RM) and the Star Convex Shape (SCS) models [GBR16].

In this chapter, we focus on the modeling of extended objects. The deep learning approach is not discussed here. Two models are of interest to us: the Random Matrix (RM) approach and the Star Convex Shape (SCS) approach. In the end, we have dropped the SCS model for the reasons highlighted in the next section.

3.1.2 The Star Convex Shape approach

Proposed by [BH09], the star convex shape model aims at estimating the shape of an EO using a surface-based sensor such as a lidar. The first proposal relied on a random hypersurface model. For each measurement $\{\mathbf{z}_k^j\}_{j=1}^{m_k}$ of the extended object, it consists in a parametric representation of the extent's contour with a radial function $f(\theta_k^j)$ and an orientation vector $\mathbf{p}(\theta_k^j)$ such as

$$\mathbf{z}_k^j = \mathbf{x}_k^c + \mathbf{p}(\theta_k^j) f(\theta_k^j) + \mathbf{e}_k^j \tag{3.1}$$

where $\mathbf{e}_k^j$ is the measurement error, subject to an unbiased Gaussian noise, and $\mathbf{x}_k^c$ is the object's center. The orientation vector $\mathbf{p}(\theta_k^j)$ is given by

$$\mathbf{p}(\theta_k^j) = \begin{bmatrix} cos(\theta_k^j) \\ sin(\theta_k^j) \end{bmatrix} \tag{3.2}$$

and the radial function $f(\theta_k^j)$ gives the radius between the center of the object and the contour shape. The parametrization can be seen in the figure 3.2.

If the measurements arise from the interior of the extended object, the measurement model (3.1) can be adapted using a scaling coefficient $s_k^j \in [0, 1]$ such as

$$\mathbf{z}_k^j = \mathbf{x}_k^c + s_k^j \ \mathbf{p}(\theta_k^j) f(\theta_k^j) + \mathbf{e}_k^j \tag{3.3}$$

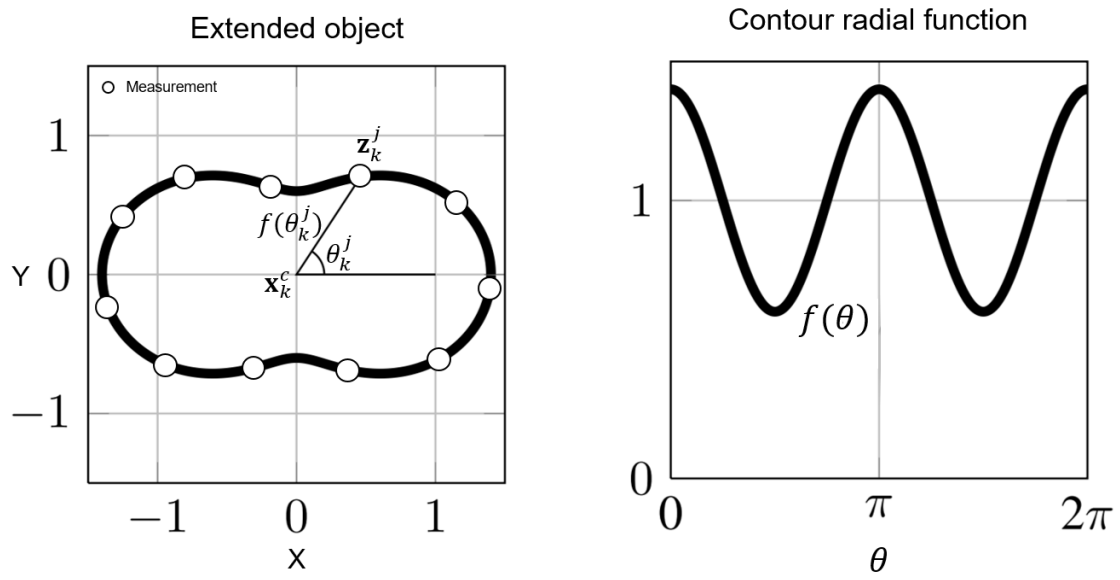


Figure 3.2: The Star Convex Shape model parametrization. This figure is based on the work of [WÖ15].

Baum *et al.* first proposed this model using a Fourier decomposition for the radial contour function [BH09], and it was further refined in [WÖ15]. Indeed, the Fourier decomposition formulation of the radial function is a shape-fitting technique that does not integrate well into a stochastic approach. More precisely, the uncertainty of non-visible faces of an object's extension cannot be taken into account: the Fourier expansion is purely deterministic. To solve this problem, [WÖ15] uses a Gaussian Process (GP) decomposition for the shape contour function $f(\theta_k^j)$.

The use of Gaussian Processes (GPs) has two advantages: the integration of the extent's measurement uncertainty with Gaussian densities, and the use of a spatial domain contour function instead of a frequency domain one, enabling the learning of fine details of the contour. This formulation is more flexible than its Fourier counterpart, and because of the uncertainty offered by GPs, the shapes can be estimated more accurately.

However, the extent estimation with a random hypersurface model using a GP parametrization of the contour has limitations. First, even if the extent estimation is accurate, it involves a computationally expensive Bayesian correction step for the GPs: one correction per measurement. Each GP component consists of an angular sector in which a measurement can be generated, so there is a fixed amount of GP components describing an extended object. For autonomous driving in a dense urban environment, the number of extended objects to track can be high, with hundreds of measurements per object, leading to heavy computations. In addition, this model is only adapted to rigid EOs with time stable contour details. For group objects, a fine representation is not mandatory, and they have a fast-changing shape which can be problematic for the estimation process.

Last but not least, the random hypersurface model assumes that the EOs are well resolved, meaning the sensor returns many measurements per acquisition. A lidar or a high-definition radar are two examples of dense sensors. In some of our applications, this assumption is not compliant with the available data, forcing us to consider another model. For instance, in the case of drone tracking, the tracked drones are often referred to as “under-resolved", returning at most one or two measurements per acquisition. This is also true for autonomous driving with a lidar: an object far from the sensor returns a small number of measurements.

Other contour estimation techniques exist for the SCS approach, but they do not solve the two main problems of this approach. The often low number of measurements returned by the objects in our applications and the fast-changing shape of group objects lead us to consider the Random Matrix approach.

3.1.3 The Random Matrix approach

The Random Matrix model is based on an elliptic approximation of the EO's shape. It assumes that the measurements are generated by the whole object and not only the surface of its extent. It was first developed in [Koc08] for aerospatial radar data, it was further extended in [FFK11], [LL12a], and [LL12b]. This model's main advantage is its lightness compared to the random hyperspace approach: it necessitates the estimation of two additional parameters during the Bayesian inference. In addition, an evolution model can be implemented for non-rigid EOs, including group objects, as discussed in [Koc08], [GBR16]. Consequently, we used the RM approach in our works about group objects in [Gue+22], and [Gue+20].

The main disadvantage of the Random Matrix approach is the elliptic shape assumption. In the case of an autonomous driving environment, the elliptic shape seems inappropriate, especially for four-wheeled vehicles with an approximately rectangular shape, as pictured in figure 3.1.a. At first, we propose to set aside this limitation and focus on elliptic extents. This problem will be addressed in the chapter 5. This chapter introduces the Random Matrix (RM) model and its extension for group objects, especially drones and swarms of drones.

A group is a set of rigid EOs evolving together and returning several measurements per acquisition. In a first approximation, a rigid EO is tracked with a radar sensor. This scenario is illustrated in figure 3.1.b. This section aims to efficiently estimate the EO's position and extent with the Random Matrix model.

The Random Matrix model is a relatively inexpensive stochastic approach to EO tracking [Koc08]; [GBR16]. In addition, it relies on Bayesian inference for the estimation process, meaning it can be integrated seamlessly into the MOT filters described in 2: this motivated our implementations in [Gue+21] and [Gue+20].

This chapter presents four different RM correction and prediction recursions, starting with the original formulation and adapting it to the modeling of group objects.

3.2 Rigid extended objects estimation with Random Matrices (RM)

3.2.1 Correction step

The correction step's goal is to compute the posterior probability density of the EO using a measurement likelihood and a prior probability density.

An extended object (EO) consists of a state vector $\mathbf{x}_k$, and an extent matrix $\mathbf{X}_k$ at timestep k. They can be estimated separately, assuming that the extent is independent of the state vector. Hence, the joint posterior density $p(\mathbf{x}_k, \mathbf{X}_k | Z_{\{1:k\}})$ can be expressed as

$$p(\mathbf{x}_k, \mathbf{X}_k | Z_{\{1:k\}}) = p(\mathbf{x}_k | \mathbf{X}_k, Z_{\{1:k\}}) p(\mathbf{X}_k | Z_{\{1:k\}}) \tag{3.4}$$

where $Z_{\{1:k\}}$ is composed of k consecutive sensor swipes, and the swipe Z_k at time k consists of m_k measurements: $Z_k = \{\mathbf{z}_k^j\}_{j=1}^{m_k}$. In a first approximation, the number of measurement m_k is assumed to be constant, it is not a random variable.

It is important to emphasize the decomposition offered by equation (3.4): the first density stands for the state vector while the second is the random matrix representation of the extent. This matrix is a $d \times d$ positive semidefinite covariance matrix, where d is the number of spatial dimensions of $\mathbf{x}_k$. These two probability densities can be deduced from the measurement likelihood. A linear Gaussian measurement model is assumed. The goal is to find a prior probability density that is closed under Bayes, meaning that the prior and posterior probability densities should be the same, as described by Koch in [Koc08]. Hence, the measurement likelihood must be computed first.

The measurement likelihood is built with the following consideration: for one extended object returning m_k measurement at each time step, the mean and the scattering matrix of the associated point cloud are

$$\begin{aligned} \overline{\mathbf{z}}_k &= \frac{1}{m_k} \sum_{j=1}^{m_k} \mathbf{z}_k^j \\ \mathbf{Z}_k &= \sum_{j=1}^{m_k} (\mathbf{z}_k^j - \overline{\mathbf{z}}_k)(\mathbf{z}_k^j - \overline{\mathbf{z}}_k)^T \end{aligned} \tag{3.5}$$

Since each measurement is subject to Gaussian noise, the measurement likelihood can be

noted[Koc08]

$$\begin{aligned} p(Z_k|m_k,\mathbf{x}_k,\mathbf{X}_k) &= \prod_{j=1}^{m_k}\mathcal{N}(\mathbf{z}_k^j;\mathbf{H}_k\mathbf{x}_k,\mathbf{X}_k) \\ &= \frac{1}{(2\pi)^{m_k d/2}\left|\mathbf{X}_k\right|^{m_k/2}}\; e^{-\frac{1}{2}\sum_{j=1}^{m_k}(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)^T X^{-1}(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)} \\ &= \frac{1}{(2\pi)^{m_k d/2}\left|\mathbf{X}_k\right|^{m_k/2}}\; e^{-\frac{1}{2}\sum_{j=1}^{m_k} tr((\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)^T X^{-1}(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k))} \\ &= \frac{1}{(2\pi)^{m_k d/2}\left|\mathbf{X}_k\right|^{m_k/2}}\; e^{-\frac{1}{2}\sum_{j=1}^{m_k} tr((\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)^T\mathbf{X}_k^{-1})} \\ &= \frac{1}{(2\pi)^{m_k d/2}\left|\mathbf{X}_k\right|^{m_k/2}}\; etr\left(-\frac{1}{2}\left(\sum_{j=1}^{m_k}(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)^T\right)\mathbf{X}_k^{-1}\right) \end{aligned} \tag{3.6}$$

where $(\mathbf{H}_k)$ is the observation matrix, m_k is the number of measurements, and d is the number of spatial dimensions of the state vector $\mathbf{x}_k$. The function $etr(\cdot)$ stands for the exponential of the trace. An interesting formulation of this measurements likelihood can be obtained using the variable change

$$\begin{aligned} \sum_{j=1}^{m_k}(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)^T &= \left(\sum_{j=1}^{m_k}(\mathbf{z}_k^j-\overline{\mathbf{z}}_k)(\mathbf{z}_k^j-\overline{\mathbf{z}}_k)^T\right) \\ &\quad + m_k(\overline{\mathbf{z}}_k-\mathbf{H}_k\mathbf{x}_k)(\overline{\mathbf{z}}_k-\mathbf{H}_k\mathbf{x}_k)^T \\ &= \mathbf{Z}_k + m_k(\overline{\mathbf{z}}_k-\mathbf{H}_k\mathbf{x}_k)(\overline{\mathbf{z}}_k-\mathbf{H}_k\mathbf{x}_k)^T \end{aligned} \tag{3.7}$$

which is possible thanks to the trick $(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)=(\mathbf{z}_k^j-\overline{\mathbf{z}}_k+\overline{\mathbf{z}}_k-\mathbf{H}_k\mathbf{x}_k)$.

Substituting equation (3.7) in equation (3.6), we obtain

$$\begin{aligned} p(Z_k|m_k,\mathbf{x}_k,\mathbf{X}_k) &= \frac{1}{(2\pi)^{m_k d/2}\left|\mathbf{X}_k\right|^{m_k/2}}\; etr\left(-\frac{1}{2}\left(\sum_{j=1}^{m_k}(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)^T\right)\mathbf{X}_k^{-1}\right) \\ &= \frac{1}{(2\pi)^{m_k d/2}\left|\mathbf{X}_k\right|^{m_k/2}}\; etr\left(-\frac{1}{2}\mathbf{Z}_k\mathbf{X}^{-1}\right) \\ &\quad etr\left(-\frac{1}{2}m_k(\overline{\mathbf{z}}_k-\mathbf{H}_k\mathbf{x}_k)(\overline{\mathbf{z}}_k-\mathbf{H}_k\mathbf{x}_k)^T\mathbf{X}_k^{-1}\right) \\ &= \frac{1}{(m_k)^{d/2}(2\pi)^{(m_k-1)d/2}\left|\mathbf{X}_k\right|^{(m_k-1)/2}}\; etr\left(-\frac{1}{2}\mathbf{Z}_k\mathbf{X}_k^{-1}\right)\mathcal{N}\left(\overline{\mathbf{z}}_k;\mathbf{H}_k\mathbf{x}_k,\frac{\mathbf{X}_k}{m_k}\right) \end{aligned} \tag{3.8}$$

The result is a measurement likelihood that can be separated into two probability densities: a state vector probability density and a scattering matrix probability density. It is the same type of decomposition as in (3.4). It can be noted that the scattering matrix probability

density is proportional to a Wishart probability density

$$\begin{aligned}\mathcal{W}(\mathbf{Z}_k; m_k - 1, \mathbf{X}_k) &\propto \frac{1}{(m_k)^{d/2}(2\pi)^{(m_k-1)d/2}\,|\mathbf{X}_k|^{(m_k-1)/2}}\; etr\left(-\frac{1}{2}\mathbf{Z}_k\mathbf{X}_k^{-1}\right) \\ &= \mathcal{L}_{scat}\end{aligned} \tag{3.9}$$

where $\mathcal{L}_{scat}$ stands for the scattering matrix likelihood. Here $\mathcal{W}(\cdot)$ is the Wishart probability density. It can be found in the appendix A.3.

The measurement likelihood (3.8) takes the form of a Gaussian density for the average measurement $\overline{\mathbf{z}}_k^j$, and a scattering matrix likelihood $\mathcal{L}_{scat}$ that resembles a Wishart density. To get a correction step that is closed under the Bayes equation, we must use a prior that is conjugate to the measurement likelihood. The conjugacy means that the prior and posterior probability densities have the same structure.

The prior probability density of the extended object, $p_{k|k-1}(\mathbf{x}_k, \mathbf{X_k}|Z_{\{1:k-1\}})$, is assumed to be separable in an extent matrix density and a state vector density as in the posterior equation (3.4)

$$p(\mathbf{x}_k, \mathbf{X}_k|Z_{1:k-1}) = p(\mathbf{x}_k, |\mathbf{X}_k, Z_{1:k-1})p(\mathbf{X}_k|Z_{1:k-1}) \tag{3.10}$$

To get a closed filter, the state prior $p(\mathbf{x}_k, |\mathbf{X}_k, Z_{1:k-1})$ must be conjugate to the Gaussian average measurement likelihood $\mathcal{N}(\overline{\mathbf{z}}_k; \mathbf{H}_k\mathbf{x}_k, \frac{\mathbf{X}_k}{m_k})$, and the extent prior $p(\mathbf{X}_k|Z_{1:k-1})$ must be conjugate to the scattering matrix likelihood $\mathcal{L}_{scat}$.

Since $\mathcal{L}_{scat}$ is proportional to a Wishart density, and because its conjugate prior probability density is an Inverse Wishart, the full prior is a Gaussian Inverse Wishart density. It relies on a Gaussian for the state vector and an Inverse Wishart for the extent. The Gaussian Inverse Wishart (GIW) prior is given by

$$p(\mathbf{x}_k, \mathbf{X}_k|Z_{\{1:k-1\}}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, p_{k|k-1} \otimes \mathbf{X}_k)\mathcal{IW}(\mathbf{X}_k; \nu_{k|k-1}, \mathbf{V}_{k|k-1}) \tag{3.11}$$

where $(\hat{\mathbf{x}}_{k|k-1}, \mathbf{p}_{k|k-1} \otimes \mathbf{X}_k)$ are the prior parameters for the state vector probability density, and $(\nu_{k|k-1}, \mathbf{V}_{k|k-1})$ are the prior parameters for the extent probability density. The Inverse Wishart probability density is described in appendix A.4.

The covariance matrix of the prior state vector (3.11) depends on the extent $\mathbf{X}_k$ and the prior covariance matrix $\mathbf{p}_{k|k-1}$. $\mathbf{p}_{k|k-1}$ should not be mistaken with $\mathbf{P}_{k|k-1}$. Indeed, the symbol $\otimes$ describes the Kronecker product, such as

$$\begin{aligned}\mathbf{A} \otimes \mathbf{B} &= \begin{pmatrix} a^1 & a^2 \\ a^3 & a^4 \end{pmatrix} \otimes \mathbf{B} \\ &= \begin{pmatrix} a^1\mathbf{B} & a^2\mathbf{B} \\ a^3\mathbf{B} & a^4\mathbf{B} \end{pmatrix}\end{aligned} \tag{3.12}$$

Hence, the Kronecker product of two matrices results in a greater matrix. For instance, the Kronecker product between a $M \times M$ matrix and a $N \times N$ matrix results in a $MN \times MN$ matrix.

The usage of the Kronecker product in (3.11) implies that on each spatial dimension, the RM model relies on the same state covariance matrix $\mathbf{p}_{k|k-1}$, that is scaled with the extent $\mathbf{X}_k$. Thus, $\mathbf{p}_{k|k-1}$ is of dimension $(d_{\mathbf{x}_k}/d) \times (d_{\mathbf{x}_k}/d)$, with $d_{\mathbf{x}_k}$ the number of dimensions of $\mathbf{x}_k$ and d the number of spatial dimensions of $\mathbf{x}_k$. In other words, $\mathbf{P}_{k|k-1} = \mathbf{p}_{k|k-1} \otimes \mathbf{X}_k$. Having the prior's covariance dependent on the extent prevents the usage of any non linear model, such as models involving a turn rate. In addition, this signifies that the kinematic state vector uncertainty is always oriented with the extent.

Using Bayes equation with the measurement likelihood (3.8) and the prior probability density (3.11), the posterior probability density can be computed:

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{X}_k | Z_{\{1:k\}}) \propto \mathcal{N}(\overline{\mathbf{z}}_k; \mathbf{H}_k \mathbf{x}_k, \frac{\mathbf{X}_k}{m_k})\, \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{p}_{k|k-1} \otimes \mathbf{X}_k) \\ \times \mathcal{L}_{scat}\, \mathcal{IW}(\mathbf{X}_k; \nu_{k|k-1}, \mathbf{V}_{k|k-1}) \end{aligned} \tag{3.13}$$

This Bayes equation can be split into a state vector update, corresponding to the product of the state vector densities, and an extent update, the product of the matrix densities.

The product of the state vector prior with the average measurement likelihood corresponds to a simple Kalman filter update. Using the Gaussian product formula [Koc08]:

$$\mathcal{N}(\overline{\mathbf{z}}_k; \mathbf{H}_k \mathbf{x}_k, \frac{\mathbf{X}_k}{m_k})\, \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{p}_{k|k-1} \otimes \mathbf{X}_k) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{p}_{k|k} \otimes \mathbf{X}_k)\, \mathcal{N}\left(\overline{\mathbf{z}}_k; \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}, \mathbf{S}\mathbf{X}_k\right) \tag{3.14}$$

This result is similar to the equation (1.7) of the chapter 1.1. The posterior probability density parameters are

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K} \otimes \mathbf{I}_d(\overline{\mathbf{z}}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{p}_{k|k} &= \mathbf{p}_{k|k-1} - \mathbf{K}\mathbf{S}\mathbf{K}^T \\ \mathbf{S} &= \mathbf{h}_k \mathbf{p}_{k|k-1} \mathbf{h}_k^T + \frac{1}{m_k} \\ \mathbf{K} &= \mathbf{p}_{k|k-1} \mathbf{h}_k \mathbf{S}^{-1} \end{aligned} \tag{3.15}$$

where $\mathbf{I}_d$ is the d dimensional identity matrix, and $\mathbf{h}_k$ is the observation matrix along one dimension, such as $\mathbf{H}_k = \mathbf{h}_k \otimes \mathbf{I}_d$. Here $\mathbf{K}$ is computed along one dimension. Thus, the Kronecker product $\mathbf{K} \otimes \mathbf{I}_d$ is a matrix with the right dimensions. Also, it should be noted that $\mathbf{S}$ is a scalar, thanks to the matrix product $\mathbf{h}_k \mathbf{p}_{k|k-1} \mathbf{h}_k^T$.

Using the result of equation (3.14), the posterior (3.13) becomes

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{X}_k | Z_{\{1,k\}}) \propto \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{p}_{k|k} \otimes \mathbf{X}_k)\, \mathcal{N}(\overline{\mathbf{z}}_k; \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}, \mathbf{S}\mathbf{X}_k) \\ \times \mathcal{L}_{scat}\, \mathcal{IW}(\mathbf{X}_k; \nu_{k|k-1}, \mathbf{V}_{k|k-1}) \end{aligned} \tag{3.16}$$

With this equation, there is yet to derive the posterior probability density of the extent. It can be carried out using the marginal likelihood of the average measurement, the extent

prior probability density and the scattering matrix likelihood:

$$
\begin{aligned}
&\mathcal{N}(\overline{\mathbf{z}}_k;\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1},\mathbf{S}\mathbf{X}_k)\ \mathcal{L}_{scat} \\
&\quad\times \mathcal{IW}(\mathbf{X}_k;\nu_{k|k-1},\mathbf{V}_{k|k-1}) \\
&\quad= \frac{1}{(\pi)^{d/2}|\mathbf{S}\mathbf{X}_k|^{1/2}}\ etr(-\frac{1}{2}(\overline{\mathbf{z}}_k-\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})(\overline{\mathbf{z}}_k-\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})^T\mathbf{S}^{-1}\mathbf{X}_k^{-1}) \\
&\quad\times \frac{|\mathbf{X}_k|^{-(m_k-1)/2}}{(m_k)^{d/2}(2\pi)^{(m_k-1)d/2}}\ etr\left(-\frac{1}{2}\mathbf{Z}_k\mathbf{X}_k^{-1}\right) \\
&\quad\times \frac{|\mathbf{V}_{k|k-1}|^{\nu_{k|k-1}/2}|X_k|^{-(\nu_{k|k-1}+d+1)/2}}{2^{\nu_{k|k-1}d/2}\Gamma_d(\nu_{k|k-1}/2)}\ etr(-\frac{1}{2}\mathbf{V}_{k|k-1}\mathbf{X}_k^{-1}) \\
&\quad= \frac{1}{(\pi)^{m_kd/2}(m_k\mathbf{S})^{d/2}}\frac{|\mathbf{V}_{k|k-1}|^{\nu_{k|k-1}/2}|\mathbf{X}_k|^{-(\nu_{k|k-1}+m_k+d+1)/2}}{2^{\nu_{k|k-1}d/2}\Gamma_d(\nu_{k|k-1}/2)} \\
&\quad\times \frac{2^{(\nu_{k|k-1}+m_k)d/2}\Gamma_d((\nu_{k|k-1}+m_k)/2)}{|\mathbf{V}_{k|k-1}+\mathbf{N}+\mathbf{Z}_k|^{(\nu_{k|k-1}+m_k)/2}}\ \frac{|\mathbf{V}_{k|k-1}+\mathbf{N}+\mathbf{Z}_k|^{(\nu_{k|k-1}+m_k)/2}}{2^{(\nu_{k|k-1}+m_k)d/2}\Gamma_d((\nu_{k|k-1}+m_k)/2)} \\
&\quad= (\pi^{m_k}m_k\mathbf{S})^{-d/2}\frac{\Gamma_d((\nu_{k|k-1}+m_k)/2)}{\Gamma_d(\nu_{k|k-1}/2)}\frac{\mathbf{V}_{k|k-1}^{\nu_{k|k-1}/2}}{|\mathbf{V}_{k|k-1}+\mathbf{N}+\mathbf{Z}_k|^{(\nu_{k|k-1}+m_k)/2}} \\
&\quad\times \mathcal{IW}(X_k;\nu_{k|k},\mathbf{V}_{k|k}) \\
&\quad= \mathcal{L}_{rm}\ \mathcal{IW}(X_k;\nu_{k|k},\mathbf{V}_{k|k})
\end{aligned}
\tag{3.17}
$$

where $\mathbf{N}=(\overline{\mathbf{z}}_k-\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})(\overline{\mathbf{z}}_k-\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})^T\mathbf{S}^{-1}$, and $\mathcal{L}_{rm}$ is the likelihood of the measurement set Z_k given all the prior probability density parameters. Here, $\Gamma_d(\cdot)$ is the d dimensional Gamma function. In this result, the set of posterior parameters for the extent are

$$
\begin{aligned}
&\mathbf{V}_{k|k} = \mathbf{V}_{k|k-1}+\mathbf{N}+\mathbf{Z}_k \\
&\nu_{k|k} = \nu_{k|k-1}+m_k \\
&\mathbf{S} = \mathbf{h}_k\mathbf{p}_{k|k-1}\mathbf{h}_k^T+\frac{1}{m_k} \\
&\mathbf{N} = (\overline{\mathbf{z}}_k-\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})(\overline{\mathbf{z}}_k-\mathbf{H}_k\hat{\mathbf{x}}_{k|k-1})^T\mathbf{S}^{-1}
\end{aligned}
\tag{3.18}
$$

with $\mathbf{h}_k\otimes\mathbf{I}_d=\mathbf{H}_k$ it the observation matrix along all the spatial dimensions.

The correction equations (3.15) and (3.18) prove that the Gaussian Inverse Wishart (GIW) prior is closed under the Bayes equation when combined with a Gaussian Wishart measurement likelihood. Indeed, the posterior is also a GIW probability density. As discussed in the section 1.2, it ensures that the filter is recursive when using the Bayes equation. To get a fully recursive filter, we need to ensure that the GIW model is also closed under the Chapman Kolmogorov equation.

3.2.2 Prediction step

For the prediction step, two simplifying assumptions are necessary[Koc08]: the evolution process is Markovian, and the evolution of the extent is independent of the state vector.

Thus, the evolution probability density can be written as

$$f(\mathbf{x}_{k+1}, \mathbf{X}_{k+1}|\mathbf{x}_k, \mathbf{X}_k) = f(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{X}_{k+1}) f(\mathbf{X}_{k+1}|\mathbf{X}_k) \tag{3.19}$$

where only the predicted state vector $\mathbf{x}_{k+1}$ depends on the predicted extent $\mathbf{X}_{k+1}$. This last observation raises an issue when using this density within the Chapman-Kolmogorov equation. Indeed in

$$\begin{aligned} p(\mathbf{x}_{k+1}, \mathbf{X}_{k+1}|Z_{\{1:k\}}) = \int & f(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{X}_{k+1}) f(\mathbf{X}_{k+1}|\mathbf{X}_k) \\ & p(\mathbf{x}_k|\mathbf{X}_k, Z_{\{1:k\}}) p(\mathbf{X}_k|Z_{\{1:k\}}) d\mathbf{x}_k d\mathbf{X}_k \end{aligned} \tag{3.20}$$

it is desirable to have two distinct integrals for simplification purposes. Hence, it can be assumed that the object extension evolution has no impact on the state vector during the prediction[Koc08]: $f(\mathbf{x}_{k+1}|\mathbf{X}_{k+1}, \mathbf{x}_k) \approx f(\mathbf{x}_{k+1}|\mathbf{X}_k, \mathbf{x}_k)$. With this approximation, the single integral of the prediction step becomes separable

$$\begin{aligned} p(\mathbf{x}_{k+1}, \mathbf{X}_{k+1}|Z_{\{1:k\}}) = & \int f(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{X}_k) p(\mathbf{x}_k|\mathbf{X}_k, Z_{\{1:k\}}) d\mathbf{x}_k \\ & \int f(\mathbf{X}_{k+1}|\mathbf{X}_k) p(\mathbf{X}_k|Z_{\{1:k\}}) d\mathbf{X}_k \end{aligned} \tag{3.21}$$

The first integral corresponds to the predicted state vector and the second to the predicted extent.

The evolution process for the state vector is assumed linear Gaussian. Thereby, with the help of equations (3.21) and the posterior probability density, the state vector prediction process is similar to a standard Kalman prediction

$$\begin{aligned} p(\mathbf{x}_{k+1}|\mathbf{X}_k|Z_{\{1:k\}}) &= \int \mathcal{N}(\mathbf{x}_{k+1}; (\mathbf{F}_{k+1} \otimes \mathbf{I}_d)\mathbf{x}_k, \mathbf{Q}_{k+1} \otimes \mathbf{X}_k) \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k} \otimes \mathbf{X}_k) d\mathbf{x}_k \\ &= \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1|k}, \mathbf{P}_{k+1|k} \otimes \mathbf{X}_k) \end{aligned} \tag{3.22}$$

with $\mathbf{F}_{k+1}$ and $\mathbf{Q}_{k+1}$ the evolution matrix and prediction error covariance matrix of chapter 1 projected on one axis, such as

$$\mathbf{F}_k = \begin{bmatrix} 1 & d_t \\ 0 & 1 \end{bmatrix} \tag{3.23}$$

and

$$\mathbf{Q}_k = \sigma_p^2 \begin{bmatrix} \frac{d_t^3}{3} & \frac{d_t^2}{2} \\ \frac{d_t^2}{2} & d_t \end{bmatrix} \tag{3.24}$$

where d_t is the period between two measurements. The equation (3.22) leads to the prediction equations

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= (\mathbf{F}_{k+1} \otimes I_d)\hat{\mathbf{x}}_{k|k} \\ \mathbf{P}_{k+1|k} &= \mathbf{F}_{k+1|k}\mathbf{P}_{k|k}\mathbf{F}_{k+1}^T + \mathbf{Q}_{k+1} \end{aligned}. \tag{3.25}$$

As for the extent density prediction, a heuristic approach can be considered to induce a prediction equation closed under Chapman-Kolmogorov. Even if a more formal proof exists, the heuristic approach is more straightforward and based on a few hypotheses [Koc08]:

- the extent posterior probability density should be an Inverse Wishart, $p(\mathbf{X}_{k+1}|Z_{\{1:k\}}) = \mathcal{IW}(\mathbf{X}_{k+1}; \nu_{k+1|k}, \mathbf{V}_{k+1|k})$, in order to get a closed prediction equation.
- the extent's shape does not change much between two acquisitions, its expectation remains the same: $\mathbb{E}[\mathbf{X}_{k+1}] = \mathbb{E}[\mathbf{X}_k]$, where $\mathbb{E}[\mathbf{X}_k] = \mathbf{V}_{k|k}/(\nu_{k|k} - 2d - 2)$.
- $\nu_{k|k}$ can be assimilated to the precision of the estimated extent. Since the precision of the estimation should decrease with the prediction step, its effect on $\nu_{k|k}$ should be to decrease it. Hence, it is assumed to be subject to exponential decay, with a temporal decay τ.

With these assumptions, the evolution model of the extent is deduced

$$\begin{aligned} \nu_{k+1|k} &= e^{-\frac{d_t}{\tau}} \nu_{k|k} \\ \mathbf{V}_{k+1|k} &= \frac{\nu_{k+1|k} - 2d - 2}{\nu_{k|k} - 2d - 2} \mathbf{V}_{k|k} \end{aligned} \tag{3.26}$$

The complete expression for the prediction step of the Gaussian Inverse Wishart can be summarized as:

$$\begin{aligned} p(\mathbf{x}_{k+1}, \mathbf{X}_{k+1}|Z_{\{1:k\}}) &= \int f(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{X}_k) p(\mathbf{x}_k|\mathbf{X}_k, Z_{\{1:k\}}) d\mathbf{x}_k \\ &\quad \int f(\mathbf{X}_{k+1}|\mathbf{X}_k) p(\mathbf{X}_k|Z_{\{1:k\}}) d\mathbf{X}_k \\ &= \mathcal{N}(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1|k}, \mathbf{p}_{k+1|k} \otimes \mathbf{X}_k) \mathcal{IW}(\mathbf{X}_{k+1}; \nu_{k+1|k}, \mathbf{V}_{k+1|k}) \end{aligned} \tag{3.27}$$

with the predicted parameters from equations (3.25) and (3.26).

Finally, the full random matrix model is closed under Bayes and Chapman-Kolmogorov equation with the Gaussian Inverse Wishart probability density as posterior and prior probability densities. However, numerous assumptions were made to achieve this goal, some of which are questionable for group object tracking. First, the measurement noise of individual measurements is not taken into account. Second, a non-linear model can not be used. Third, the number of measurements returned by a group is assumed constant. These problems should be addressed to model groups.

3.3 Modeling Groups as Ellipses

A group of objects is analogous to an extended object in some aspects: the group size is superior to the resolution cell of the sensor, so it returns more than one measurement per acquisition, and the objects composing the group behave similarly. However, a group has its specificities compared to a rigid extended object. Its shape and number of elements might evolve through time. Also, the measurement noise can be non-negligible compared to the size of the extent. This section addresses several problems encountered when modeling groups with the RM model, starting with the measurement noise.

3.3.1 Taking into account the measurement noise

3.3.1.1 The additive Gaussian noise model

In equation (3.8), the measurement noise was implicitly assumed negligible when compared to the size of the extent. It might not always be the case, whether an extended object or a group object is considered.

To tackle this problem, Feldmann *et al.* propose to use an additive Gaussian noise in the measurement model. Hence, instead of the likelihood (3.8),

$$\begin{aligned} p(Z_k|m_k,\mathbf{x}_k,\mathbf{X}_k) &= \prod_{j=1}^{m_k}\mathcal{N}(\mathbf{z}_k^j;\mathbf{H}_k\mathbf{x}_k,\mathbf{Y}) \\ &= \frac{1}{(2\pi)^{m_k d/2}\,|\mathbf{Y}|^{m_k/2}}\; etr\left(-\frac{1}{2}\left(\sum_{j=1}^{m_k}(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)(\mathbf{z}_k^j-\mathbf{H}_k\mathbf{x}_k)^T\right)(\mathbf{Y})^{-1}\right) \\ &= \frac{1}{(m_k)^{d/2}(2\pi)^{(m_k-1)d/2}\,|\mathbf{Y}|^{(m_k-1)/2}}\; etr\left(-\frac{1}{2}\mathbf{Z}_k\mathbf{Y}^{-1}\right)\mathcal{N}\left(\overline{\mathbf{z}}_k;\mathbf{H}_k\mathbf{x}_k,\frac{\mathbf{Y}}{m_k}\right) \end{aligned} \tag{3.28}$$

where $\mathbf{Y}=\lambda\mathbf{X}_k+\mathbf{R}_k$ is the measurement noise covariance matrix with the additive Gaussian noise model. Here, $\mathbf{R}_k$ is the individual measurement noise, and λ is a weighting parameter that describes the measurement scattering in the extent. As discussed by Feldmann *et al.*, a value of $\lambda=0.25$ leads to an almost uniform scattering: this is the value we have selected in this manuscript.

Because the likelihood (3.28), the correction equation cannot be solved using the prior (3.11). To solve this problem, Feldmann *et al.* propose to consider that the extent prior probability density and the state vector prior probability density are independent

$$p(\mathbf{x}_k,\mathbf{X}_k|Z_{\{1:k-1\}}) = \mathcal{N}(\mathbf{x}_k;\hat{\mathbf{x}}_{k|k-1},\mathbf{P}_{k|k-1})\mathcal{IW}(\mathbf{X}_k;\nu_{k|k-1},\mathbf{V}_{k|k-1}) \tag{3.29}$$

The prior probability density (3.29) is not entirely justifiable, since the correction of $\mathbf{x}_k$ and $\mathbf{X}_k$ using the measurement set $\mathbf{Z}_k$ results in a dependence of $\mathbf{x}_k$ and $\mathbf{X}_k$. However, this prior enables the use of non-linear models for the state vector since its covariance matrix does not depend on the extent anymore. To these assumptions, Feldmann *et al.* also assume that the true extent $\mathbf{X}_k$ is almost equal to the prior extent $\mathbf{X}_{k|k-1}$: this is necessary to solve the Bayesian equation analytically for the extent (3.17). The resulting equations are detailed in the table 3.1.

3.3.1.2 The deformation matrix model

Another approach to the non-negligible measurement noise problem was proposed by Lan *et al.* in [LL12a]. They consider that the observation of the extended object is deformed, and

Table 3.1: The prediction and update proposed by Feldmann *et al.* [FFK11]

Prediction	$\hat{\mathbf{x}}_{k\|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1\|k-1}$ $\mathbf{P}_{k\|k-1} = \mathbf{F}_k \mathbf{P}_{k-1\|k-1} \mathbf{F}_k^T + \mathbf{Q}_k$ $\nu_{k\|k-1} = e^{-d_t/\tau} \Upsilon$ $\mathbf{V}_{k\|k-1} = \frac{(\nu_{k\|k-1} - 2d - 2)}{\Upsilon} \mathbf{V}_{k-1\|k-1}$ $\Upsilon = (\nu_{k-1\|k-1} - 2d - 2)$
Correction	$\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}\varepsilon$ $\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{K}\mathbf{S}\mathbf{K}^T$ $\nu_{k\|k} = \nu_{k\|k-1} + m_k$ $\mathbf{V}_{k\|k} = \mathbf{V}_{k\|k-1} + \mathbf{N} + \mathbf{Z}$ $\varepsilon = \overline{\mathbf{z}}_k - \mathbf{H}_k \mathbf{x}_{k\|k-1}$ $\mathbf{X} = \mathbb{E}[\mathbf{X}_k] = \frac{\mathbf{V}_{k\|k-1}}{\nu_{k\|k-1} - 2d - 2}$ $\mathbf{S} = \mathbf{H}_k \mathbf{P}_{k\|k-1} \mathbf{H}_k^T + \frac{\mathbf{Y}}{m_k}$ $\mathbf{K} = \mathbf{P}_{k\|k-1} \mathbf{H}_k S^{-1}$ $\mathbf{N} = \mathbf{X}^{1/2} \mathbf{S}^{-1/2} \varepsilon\varepsilon^T \mathbf{S}^{-T/2} \mathbf{X}^{T/2}$ $\mathbf{Z} = \mathbf{X}^{1/2} \mathbf{Y}^{-1/2} \mathbf{Z}_k \mathbf{Y}^{-T/2} \mathbf{X}^{T/2}$ $\mathbf{Y} = \lambda \mathbf{X} + \mathbf{R}_k$

they rely on a deformation matrix $\mathbf{B}$ that describes the distortion of the measured extension $\mathbf{X}_k$. The resulting measurement model is

$$p(Z_k|m, \mathbf{x}_k, \mathbf{X}_k) = \prod_{j=1}^{m} \mathcal{N}(\mathbf{z}_k^j; \mathbf{H}_k \mathbf{x}_k, \mathbf{B}\mathbf{X}_k\mathbf{B}^T) \tag{3.30}$$

with the observation matrix $\mathbf{H}_k$.

This formulation is flexible enough to model an approximation of the additive Gaussian noise, indeed when

$$\mathbf{B} = \lambda \mathbb{E}[\mathbf{X}_k] + \mathbf{R}_k)^{\frac{1}{2}} \mathbb{E}[\mathbf{X}_k]^{\frac{-1}{2}} \tag{3.31}$$

then the resulting estimated error covariance is

$$\begin{aligned} \mathbf{B}\mathbf{X}_k\mathbf{B}^T &= (\lambda \mathbb{E}[\mathbf{X}_k] + \mathbf{R}_k)^{1/2} \mathbb{E}[\mathbf{X}_k]^{-1/2} \mathbf{X}_k \mathbb{E}[\mathbf{X}_k]^{-T/2} (\lambda \mathbb{E}[\mathbf{X}_k] + \mathbf{R}_k)^{T/2} \\ &\approx \lambda \mathbf{X}_k + R_k \end{aligned} \tag{3.32}$$

where λ is a scalar weight, and $\mathbf{R}_k$ is the covariance matrix of the sensor measurement error. The value of λ has already been discussed by Feldmann *et al.*, or Vivone *et al.* [Viv+15]. It should be set to $\frac{1}{4}$ to model a uniform distribution of the measurements inside the extent.

This new model resembles the additive Gaussian noise model of Feldmann *et al.* [FFK11], without the assumption of independence between the state vector and the extent densities after the correction step. However, we chose nonetheless to keep the independence hypothesis, in

Table 3.2: The prediction and update proposed by Lan *et al.* with our modification [LL12a]

Prediction	$\hat{\mathbf{x}}_{k\|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1\|k-1}$ $\mathbf{P}_{k\|k-1} = \mathbf{F}_k\mathbf{P}_{k-1\|k-1}\mathbf{F}_k^T + \mathbf{Q}_k$ $\nu_{k\|k-1} = \frac{2\upsilon(\Upsilon-1)(\Upsilon-2)}{\Upsilon(\Upsilon+\upsilon)} + 2d + 4$ $\mathbf{V}_{k\|k-1} = \frac{\upsilon(\nu_{k\|k-1}-2d-2)}{\Upsilon}\mathbf{A}\mathbf{V}_{k-1\|k-1}\mathbf{A}^T$ $\Upsilon = (\nu_{k-1\|k-1} - 2d - 2)$
Correction	$\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}\varepsilon$ $\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{K}\mathbf{S}\mathbf{K}^T$ $\nu_{k\|k} = \nu_{k\|k-1} + m_k$ $\mathbf{V}_{k\|k} = \mathbf{V}_{k\|k-1} + \mathbf{N} + \mathbf{Z}$ $\varepsilon = \overline{\mathbf{z}}_k - \mathbf{H}_k\mathbf{x}_{k\|k-1}$ $\mathbf{X} = \mathbb{E}[\mathbf{X}_k] = \frac{\mathbf{V}_{k\|k-1}}{\nu_{k\|k-1}-2d-2}$ $\mathbf{S} = \mathbf{H}_k\mathbf{P}_{k\|k-1}\mathbf{H}_k^T + \frac{\mathbf{B}\mathbf{X}\mathbf{B}^T}{m_k}$ $\mathbf{K} = \mathbf{P}_{k\|k-1}\mathbf{H}_k S^{-1}$ $\mathbf{N} = \mathbf{S}^{-1/2}\varepsilon\varepsilon^T\mathbf{S}^{-T/2}$ $\mathbf{Z} = \mathbf{B}^{-1}\mathbf{Z}_k\mathbf{B}^{-T}$ $\mathbf{B} = (\lambda\mathbf{X} + \mathbf{R}_k)^{1/2}\mathbf{X}^{-1/2}$

order to simplify the integration of non-linear models. The prediction and correction equations of this filter are given in table 3.2.

In addition to the deformation matrix, Lan *et al.* introduce another improvement for the RM[1] model: they propose to consider the extent's evolution process. This is interesting for group objects with an extent that continuously evolves.

We propose a simple evolution model for the extent: the group expansion at each time step, which can be interpreted as a loss of precision due to the prediction step. The model associated with this transition probability density is a Wishart density since it is conjugate to the Inverse Wishart density from the update. Thus, the usage of the Chapman-Kolmogorov equation gives [LL12a]:

$$\begin{aligned} p(\mathbf{X}_{k+1}|Z_{\{1:k\}}) &= \int f(\mathbf{X}_{k+1}|\mathbf{X}_k)p(\mathbf{X}_k|Z_{\{1:k\}})d\mathbf{X}_k \\ &= \int \mathcal{W}(\mathbf{X}_{k+1}; \upsilon_k, \mathbf{A}\mathbf{X}_k\mathbf{A}^T)\mathcal{IW}(\mathbf{X}_k; \nu_{k|k}, \mathbf{V}_{k|k})d\mathbf{X}_k \end{aligned} \tag{3.33}$$

where $\mathbf{A}$ is the evolution matrix, and υ_k is the degree of freedom of the prediction. υ_k can be related to the precision of the prediction. In this equation, the integration step results in a Generalized Beta Type II density[LL12a], which can be approximated via moment matching to an Inverse Wishart prior probability density $\mathcal{IW}(\mathbf{X}_{k+1}; \nu_{k+1|k}, \mathbf{V}_{k+1|k})$. The extent's prior probability parameters are presented in the table 3.2. The demonstration of these results is

[1]Random Matrix

out of the scope of this manuscript; more details are available in the aforementioned paper [LL12a], and in the original description of the RM model by Koch [Koc08].

In this work, the evolution model is an expansion model, hence we chose to express $\mathbf{A}$ such as

$$\mathbf{A} = \begin{pmatrix} \rho & 0 \\ 0 & \rho \end{pmatrix} \tag{3.34}$$

where ρ is an expansion coefficient.

Both the additive Gaussian noise approximation and the deformation matrix approximation are used in this manuscript. They solve an issue with the first RM model, taking into account the noise of each measurement. However, there is still one remaining problem. In the section 3.2, the number of measurements was assumed constant, a hypothesis that can be questioned.

3.3.2 Time evolving number of measurements

An extended object is not guaranteed to return a constant amount of measurements m_k at each time step. To tackle this problem, Koch proposes to set a measurement density parameter dependent on the size of the extent. Hence, the number of measurements returned by an extended object is proportional to its size. While it can be the case in many applications, there is no certainty that a group will keep its density constant through time. For instance, the group can shrink or expand, while the number of objects that compose it remains constant: the number of measurements returned by such a group remains constant while the extent varies.

Another solution is to consider that the number of measurements is an additional random variable to estimate. Granström *et al.* propose this approach with the measurement likelihood $p(Z_k|m_k, \mathbf{x}_k, \mathbf{X}_k)$ rewritten as $p(Z_k|\gamma_k, \mathbf{x}_k, \mathbf{X}_k)$, where γ_k is the expected number of measurements for a group [GO14].

Counting the average number of occurrences γ_k of an event happening during a time interval d_t is the typical application of a Poisson probability density. This Poisson density is supposed to be independent of the shape of the group object [Gil+05]; [GS05]. Hence the measurement likelihood (3.8) can be augmented

$$\begin{aligned} p(Z_k|\gamma_k, \mathbf{x}_k, \mathbf{X}_k) &\propto p(m_k|\gamma_k)p(\overline{\mathbf{z}}_k|\mathbf{x}_k, \mathbf{X}_k)p(\mathbf{Z}_k|\mathbf{X}_k) \\ &= \mathcal{P}(m_k; \gamma_k)\mathcal{N}(\overline{\mathbf{z}}_k; \mathbf{H}_k\mathbf{x}_k, \frac{\mathbf{X}_k}{m_k})\mathcal{W}(\mathbf{Z}_k; m_k - 1, X_k) \end{aligned} \tag{3.35}$$

Once again, the filter should be closed under the Bayes correction equation. Since the conjugate prior of a Poisson distribution is a Gamma distribution, the prior probability density of the RM model can be rewritten

$$p(\mathbf{x}_k, \mathbf{X}_k|Z_{\{1:k-1\}}) = \mathcal{G}(\gamma_k; \alpha_{k|k-1}, \beta_{k|k-1})\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\mathcal{IW}(\mathbf{X}_k; \nu_{k|k-1}, \mathbf{V}_{k|k-1}) \tag{3.36}$$

where $\alpha_{k|k-1}$ is the expected number of measurement arriving at an expected rate $\beta_{k|k-1}$, such as $\mathbb{E}(\gamma_k) = \alpha_{k|k-1}/\beta_{k|k-1}$. This prior probability density is a Gamma Gaussian Inverse Wishart (GGIW) density [LGO13]. The posterior probability density can be computed using the prior probability density and the measurement likelihood: it is also a GGIW density. The correction procedure for the Gaussian and Inverse Wishart parameters are not affected by the additional Gamma density, but two more parameters need to be estimated throughout the Bayesian recursion: α and β. α and β update equations are available. The table 3.3. Their derivation is available in Granström *et al.*'s paper [GO14].

For the prediction step, exponential forgetting is retained for $\alpha_{k+1|k}$ and $\beta_{k+1|k}$. This exponential forgetting is tuned such as γ_{k+1} evolve over a time window $w_e = \frac{\eta}{\eta-1}$ where η can be seen as a forgetting factor[LGO13]. Hence the prediction step for the gamma density maintains the expected number of measurements, $\mathbb{E}(\gamma_k) = \mathbb{E}(\gamma_{k+1})$. The prediction equations remain the same as before for the other densities constituting the posterior probability density; all the equations can be found in the table 3.3.

Table 3.3: Prediction and update equations with the additional Gamma density

	Granström *et al.* proposal [GFS19]	Our proposal [Gue+20]
Prediction	$\alpha_{k\|k-1} = \alpha_{k-1\|k-1}/\eta$ $\beta_{k\|k-1} = \beta_{k-1\|k-1}/\eta$ $\hat{\mathbf{x}}_{k\|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1\|k-1}$ $\mathbf{P}_{k\|k-1} = \mathbf{F}_k\mathbf{P}_{k-1\|k-1}\mathbf{F}_k^T + \mathbf{Q}_k$ $\nu_{k\|k-1} = e^{-d_t/\tau}\Upsilon$ $\mathbf{V}_{k\|k-1} = \frac{(\nu_{k\|k-1}-2d-2)}{\Upsilon}\mathbf{V}_{k-1\|k-1}$ $\Upsilon = (\nu_{k-1\|k-1} - 2d - 2)$	$\alpha_{k\|k-1} = \alpha_{k-1\|k-1}/\eta$ $\beta_{k\|k-1} = \beta_{k-1\|k-1}/\eta$ $\hat{\mathbf{x}}_{k\|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1\|k-1}$ $\mathbf{P}_{k\|k-1} = \mathbf{F}_k\mathbf{P}_{k-1\|k-1}\mathbf{F}_k^T + \mathbf{Q}_k$ $\nu_{k\|k-1} = \frac{2\upsilon(\Upsilon-1)(\Upsilon-2)}{\Upsilon(\Upsilon+\upsilon)} + 2d + 4$ $\mathbf{V}_{k\|k-1} = \frac{\upsilon(\nu_{k\|k-1}-2d-2)}{\Upsilon}\mathbf{A}\mathbf{V}_{k-1\|k-1}\mathbf{A}^T$ $\Upsilon = (\nu_{k-1\|k-1} - 2d - 2)$
Correction	$\alpha_{k\|k} = \alpha_{k\|k-1} + m_k$ $\beta_{k\|k} = \beta_{k\|k-1} + 1$ $\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}\varepsilon$ $\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{K}\mathbf{S}\mathbf{K}^T$ $\nu_{k\|k} = \nu_{k\|k-1} + m_k$ $\mathbf{V}_{k\|k} = \mathbf{V}_{k\|k-1} + \mathbf{N} + \mathbf{Z}$ $\varepsilon = \overline{\mathbf{z}}_k - \mathbf{H}_k\mathbf{x}_{k\|k-1}$ $\mathbf{X} = \mathbb{E}[\mathbf{X}_k] = \frac{\mathbf{V}_{k\|k-1}}{\nu_{k\|k-1}-2d-2}$ $\mathbf{S} = \mathbf{H}_k\mathbf{P}_{k\|k-1}\mathbf{H}_k^T + \frac{\mathbf{Y}}{m_k}$ $\mathbf{K} = \mathbf{P}_{k\|k-1}\mathbf{H}_k S^{-1}$ $\mathbf{N} = \mathbf{X}^{1/2}\mathbf{S}^{-1/2}\varepsilon\varepsilon^T\mathbf{S}^{-T/2}\mathbf{X}^{T/2}$ $\mathbf{Z} = \mathbf{X}^{1/2}\mathbf{Y}^{-1/2}\mathbf{Z}_k\mathbf{Y}^{-T/2}\mathbf{X}^{T/2}$ $\mathbf{Y} = \lambda\mathbf{X} + \mathbf{R}_k$	$\alpha_{k\|k} = \alpha_{k\|k-1} + m_k$ $\beta_{k\|k} = \beta_{k\|k-1} + 1$ $\hat{\mathbf{x}}_{k\|k} = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}\varepsilon$ $\mathbf{P}_{k\|k} = \mathbf{P}_{k\|k-1} - \mathbf{K}\mathbf{S}\mathbf{K}^T$ $\nu_{k\|k} = \nu_{k\|k-1} + m_k$ $\mathbf{V}_{k\|k} = \mathbf{V}_{k\|k-1} + \mathbf{N} + \mathbf{Z}$ $\varepsilon = \overline{\mathbf{z}}_k - \mathbf{H}_k\mathbf{x}_{k\|k-1}$ $\mathbf{X} = \mathbb{E}[\mathbf{X}_k] = \frac{\mathbf{V}_{k\|k-1}}{\nu_{k\|k-1}-2d-2}$ $\mathbf{S} = \mathbf{H}_k\mathbf{P}_{k\|k-1}\mathbf{H}_k^T + \frac{\mathbf{B}\mathbf{X}\mathbf{B}^T}{m_k}$ $\mathbf{K} = \mathbf{P}_{k\|k-1}\mathbf{H}_k S^{-1}$ $\mathbf{N} = \mathbf{S}^{-1/2}\varepsilon\varepsilon^T\mathbf{S}^{-T/2}$ $\mathbf{Z} = \mathbf{B}^{-1}\mathbf{Z}_k\mathbf{B}^{-T}$ $\mathbf{B} = (\lambda\mathbf{X} + \mathbf{R}_k)^{1/2}\mathbf{X}^{-1/2}$

3.4 Conclusion

This chapter discussed the notion of Extended Objects, objects generating multiple measurements at each timestep. They exist in two types: with a rigid or a non-rigid extent. Since we are targeting applications with both types of EOs, several solutions have been discussed in this chapter.

In order to detect and estimate the extent of EOs, either large datasets are available and can be used to train state-of-the-art deep learning models, or only a few small datasets are available, and the extent cannot be inferred from the data, so it must be modeled. We have left the case of deep learning approaches to the chapter 5, and we have focused on small datasets in this chapter, favoring model-based approaches.

Amongst the existing methods, we have selected the Random Matrix model. Even if it proposes a coarse description of the extent, relying on an elliptical shape, it is lightweight and simple enough to cover many applications. Moreover, the precise shape of the extent is not the most critical concern since the objective is to track group objects, and groups have fuzzy boundaries. In addition, the RM model relies on Bayesian inference, which facilitates its integration to Random Finite Sets Multi Object Tracking algorithms.

However, the first formulation of the RM model has some drawbacks that prevent its usage for group objects: it considers that the EO returns a constant number of measurements, and the measurement noise is considered negligible in front of the size of the extent. The resolution of these problems led to the presentation of 4 prediction and correction recursion propositions for the RM model, including our proposition.

The results of this chapter were published in [Gue+20], and [Gue+21]. These publications proposed to bridge the gap between EO tracking and the MOT problem. With the help of the RM models and the RFS-based MOT filters, the groups of drones tracking application is discussed in the next chapter.

Chapter 4

Multiple Extended Object Tracking

Contents

4.1 Models for swarms and groups of drones

The last decade has seen the democratization of drones. Buying and flying a drone is simple enough not to require a permit. Thanks to their compactness and stability, drones can even fly in groups. For instance, during the Olympic Games of Tokyo 2020, Intel presented a flight performance using 1824 drones evolving in close formation. However, people with malicious intent can use drones as threats because of their ease of use. For example, two drones trespassing London-Gatwick airport caused a 36 hours long interruption of the aerial traffic in 2018. Hence, the surveillance of aerial spaces to prevent such attacks is mandatory nowadays. The main objective of this chapter is to study the performances of the methods developed in the previous two chapters using detections from radar sensors.

Our application considers that a group of drones moves following a virtual leader. A virtual leader is the mirror of the average position and dynamic of the group, as discussed by Blackman in [Bla86]. Early work by Salmond and Gordon in [GSF97] and [SG99], modeled a group as a set of separable objects, with their of the of the individual state models and an additional group model; or "bulk model" in their case. This modeling is analogous to the "virtual leader-follower" model discussed by Mahler in [Mah07]. An implementation of this "virtual leader-follower" model was proposed by Clark and Godsill in [CG07], using a Gaussian Mixture Poisson Hypothesis Density filter, and further work by Swain and Clark on the derivation of a group specific PHD filter can be found in [SC11]. These approaches all consider that a group consists of separable individual objects. Thus, they aim to estimate the group's dynamic and the movements of individual objects. For our groups of drones tracking application, the drones are not separable, which prevents us from using these approaches.

In this manuscript, the groups are composed of single indiscernible objects. They resemble the non-rigid extended object model described in the previous chapter. Here, estimating a group object is restricted to estimating a non-rigid extended object. The simplicity induced by tracking a single group object instead of indiscernible close objects allows to tackle several issues:

- the permutation of tracked objects.
- the mismatch between measurement and tracked objects.
- the early destruction of tracked objects because of successive mismatches causing their likelihood to decrease.

Representing a group object as a non-rigid extended object is a good workaround to these problems.

Specific additional problems tend to make groups of drones tracking a challenging task because of the following points:

- Multiple groups of drones can merge into bigger groups or split into smaller groups. We relied on RFS filters to model this group dynamic: we have implemented a Probability

Hypothesis Density (PHD) filter and a Poisson Multi Bernoulli (PMB) filter for groups of drones using the RM extended object modeling. The first filter is easy to implement and tune because it relies on fewer parameters than the PMB filter. However, the PHD lacks the explanatory power of the PMB filter for the existence probability and the label.

- Groups can merge or split, resulting in multiple objects where some objects might be under-resolved. An under-resolved is a regular, isolated object: it returns at most one measurement per sensor sweep. The under-resolved denomination emphasizes the difference with a group or extended object returning multiple measurements. Under-resolved objects are typically single drones separating from a group. The main problem is that under-resolved drones are not extended: a regular linear Gaussian model can model them. We need to consider the cohabitation of two types of objects, under-resolved and extended objects, with their measurement and evolution models.

- The shape of a group of drones depends on the sensor used for the measurement: we rely on a passive and active radar in our applications. These sensors return two-dimensional measurements subject to Gaussian noise. They are subject to false alarms and missed detections.

- Radar sensor datasets are scarce. This prevents the use of deep learning detection algorithms to detect and estimate the shape of a group of drones. Bayesian inference is a good alternative to evaluate the extent of a group of drones. The Random Matrix model, introduced in the chapter 3, assumes that the extent has an elliptical shape and is subject to random effects. When tracking a group of drones with a radar sensor, the elliptical shape assumption is a valid approximation of the reality. The shape of a group evolves through time, and the group can shrink or expand. Even if multiple drones can form groups with diverse shapes, an ellipse remains a flexible enough approximation for our application.

This chapter is an application of the work presented in chapters 2 and 4. The groups of drones tracking problem is considered from a multiple non-rigid extended objects tracking perspective. As such, the first two sections focus on the description of the algorithms we developed, respectively the Group Object Probability Hypothesis Density filter and the Group Object Poisson Multi-Bernoulli filter. These filters are then tested on simulated datasets and real datasets. The third section focuses on the simulation and real dataset of an active radar sensor with groups of drones. The fourth section focuses on the MOT using a passive radar sensor. Both of the last sections present the sensor's characteristics and the parameters used for the tunning. The final section concludes this chapter and our work on extended MOT with a few datasets for the training.

4.2 Integrating the Random Matrix model in RFS filters

This first section aims to present the notation of the chapter and introduce the likelihood that is used when coupling the Random Matrix approach to the Random Finite Set-based filters.

In the first chapter, we have described three different RFS filters: the Gaussian Mixture Poisson Hypothesis Density, the Generalized Labeled Multi-Bernoulli, and the Poisson Multi-Bernoulli filters. They were introduced with the simple linear Gaussian model, where each object $\mathbf{x}_k^i \in \{\mathbf{x}_k^1, ..., \mathbf{x}_k^{n_k}\}$ is characterized by an expected state vector $\hat{\mathbf{x}}_k^i$ and an error covariance matrix $\mathbf{P}_k^i$: an estimation of $\mathbf{x}_k^i$ is given by $(\hat{\mathbf{x}}_k^i, \mathbf{P}_k^i)$.

We have selected the Random Matrix approach for groups of drones, as explained in the previous chapter. It relies on the Gaussian Inverse Wishart (GIW) state probability density introduced in the chapter 3. We have shown that this density can be estimated using Bayesian inference: for an extended object $\mathbf{x}_k^i \in \{\mathbf{x}_k^1, ..., \mathbf{x}_k^{n_k}\}$, four parameters need to be estimated:

- the expected state vector $\hat{\mathbf{x}}_k^i$ and the error covariance matrix $\mathbf{P}_k^i$ corresponding to the Gaussian probability density of the state vector,
- the degree of freedom ν_k^i and the scale matrix $\mathbf{V}_k^i$, that correspond to the Inverse Wishart probability density modeling the extent.

This result in the set of parameters $(\hat{\mathbf{x}}_k^i, \mathbf{P}_k^i, \nu_k^i, \mathbf{V}_k^i)$ for each object $\mathbf{x}_k^i$, with the addition of the weight of the component w_k^i in the case of a Poisson RFS density, or the existence probability r_k^i in the case of the Multi-Bernoulli RFS density.

The measurement $Z_k = \{\mathbf{z}_k^1, ..., \mathbf{z}_k^{m_k}\}$ is divided into subsets $Z_k^j = \{\mathbf{z}_k^1, ..., \mathbf{z}_k^{m^j}\}$, such as $Z_k = \bigcup_{j=1}^{M} Z_k^j$. Here there are M subsets, or clusters, and each subset consists of m^j measurements, that corresponds to an extended object. All the notations are summed up in the figure 4.1.

The partitioning of the measurement set Z_k into M subsets is a critical question regarding the quality of the extent estimation process. If the clusters are too small, the groups will be fragmented, and on the contrary, if it is too coarse, there is a risk of merging the measurements coming from different extended objects. Several methods exist to tackle this problem. For instance, Granström proposes in [GO12c] to perform the extension estimation using several clustering scales and merge the result depending on the likelihood of each scale. Since the result is computed from a fine to a coarse partitioning of the measurement set, it should converge to the correct size.

The main disadvantage of this method is its combinatorial complexity. If the result is computed over four scales, then the Probability Hypothesis Density filter needs to be run once per scale to get the final result.

In our case, we want to track groups of drones. We know the minimal size of a drone, and we can fix the maximal distance between two drones to consider them as a group. Hence, it is sufficient to use one partitioning of the measurement space. Since groups of drones have varying densities, we relied on the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm proposed by Ester in [Est+96].

Once the measurement set is partitioned, the likelihood that the subset $Z_k^j = \{\mathbf{z}_k^1, ..., \mathbf{z}_k^{m^j}\}$

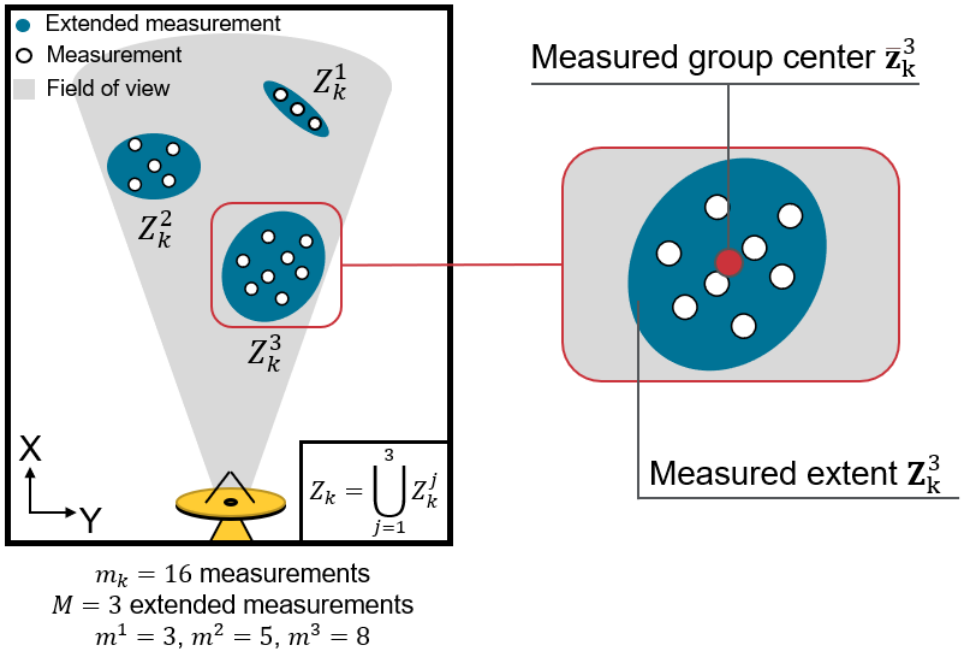


Figure 4.1: Notations for the extended measurements.

arises from the GIW object $\mathbf{x}_k^i$ is given by:

$$c^{i,j} = \frac{p_D\ (\gamma_k)^{m^j}\ e^{-\gamma_k}}{(\lambda_{FA})^{m^j}((\pi)^{m^j} m^j\ |\mathbf{S}^{i,j}|)^{d/2}} \frac{\left|\mathbf{V}_{k|k-1}^i\right|^{\nu_{k|k-1}^i/2}}{\left|\mathbf{V}_{k|k}^{i,j}\right|^{\nu_{k|k}^{i,j}/2}} \frac{\mathbf{\Gamma_d}(\nu_{k|k}^{i,j}/2)}{\mathbf{\Gamma_d}(\nu_{k|k-1}^i/2)}\ w_{k|k-1}^i \tag{4.1}$$

This cost is the likelihood computed in the chapter 3, more specifically the equation (3.17) for Feldmann and Lan RM proposal. As a brief reminder, m^j is the number of measurements of the measurement cluster j. Moreover, the parameters with the subscript $k|k$ are the posterior parameters based on the object $\mathbf{x}_k^i$ and the measurement set Z_k^j. In addition, $\Gamma_d(\cdot)$ is the d dimensional Gamma function and $\Gamma(\cdot)$ is the Gamma function they are both given in the appendix A. The expected number of measurements constituting the group is given by γ_k, and the notation $|\mathbf{V}_{k|k}^i|$ stands for the determinant of the matrix $\mathbf{V}_{k|k}^i$.

The likelihood (4.1) of associating the measurement cluster Z_k^j with the tracked object $\mathbf{x}_k^i$ is the same whether $\mathbf{x}_k^i$ belongs to a Poisson RFS[1] or a Multi-Bernoulli RFS. In any case, it must be computed for each $i \in [\![1, n_{k|k}]\!]$ and $j \in [\![1, M]\!]$. One of the only difference is the replacement of the factor $w_{k|k-1}^i$ by $r_{k|k-1}$ in the case of the Multi-Bernoulli multi-object probability density.

In the likelihood (4.1), the expected number of measurements returned by an EO[2] is known. To estimate it, Koch first proposed to set the expected number of measurements γ_k

[1]Random Finite Sets
[2]Extended Object

proportionally to the size of the extent [Koc08]. We used this approach for Feldmann's and Lan's GIW RM implementations.

However, as discussed in the chapter 3, the density of a group object might vary during the tracking since it is not rigid. For instance, the group of drones can switch between different formations, changing the density of the measurements it returns at the same type. This example is pictured in the figure 4.2. To tackle this problem, Granström *et al.* proposed to estimate γ_k using Bayesian inference in [GO12a]. They modeled it with a Gamma probability density, leading to the Gamma Gaussian Inverse Wishart density (GGIW).

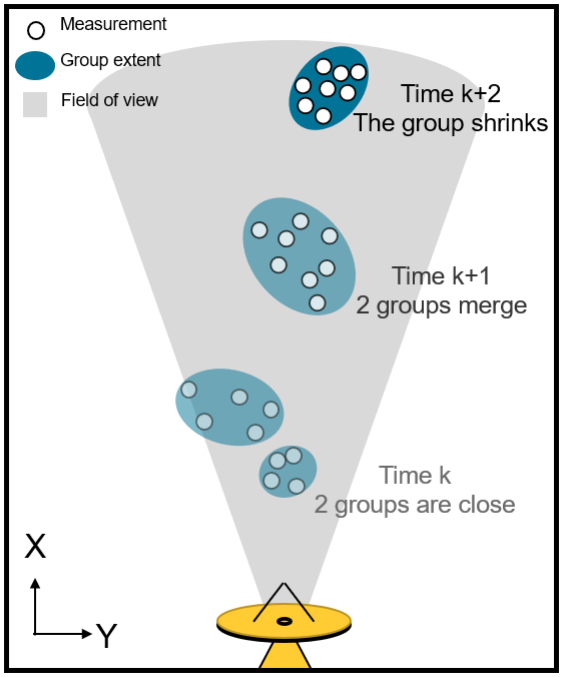


Figure 4.2: Two groups of drones merge, changing the density of the resulting swarm of drones.

The GGIW probability density is a state and extent model that can be estimated using Bayesian inference. With this model, an object $\mathbf{x}_k^i \in \{\mathbf{x}_k^1, ..., \mathbf{x}_k^{n_k}\}$ is estimated using two additional parameters when compared to the GIW density: α_k^i, and β_k^i, respectively the shape and the rate of the Gamma density. Thus $\mathbf{x}_k^i$ is determined by the set of parameters $(\hat{\mathbf{x}}_k^i, \mathbf{P}_k^i, \nu_k^i, \mathbf{V}_k^i, \alpha_k^i, \beta_k^i)$, excluding the RFS probability density additional parameters in the case of MOT. When considering a MOT problem, the likelihood of associating the GGIW

extended object $\mathbf{x}_k^i$ with the group of measurement Z_k^j is

$$\begin{aligned} c^{i,j} =& \frac{p_D\ (2\pi)^{d\ m^j/2}\ (2)^{m^j/2}}{(\lambda_{FA})^{m^j}(m^j)^{d/2}} \frac{\left|\mathbf{V}_{k|k-1}^i\right|^{\nu_{k|k-1}^i/2}}{\left|\mathbf{V}_{k|k}^{i,j}\right|^{\nu_{k|k}^{i,j}/2}} \frac{\mathbf{\Gamma_d}(\nu_{k|k}^j/2)}{\mathbf{\Gamma_d}(\nu_{k|k-1}^i/2)} \\ &\times \frac{\left|\mathbf{Y}^{i,j}\right|^{m^j/2}}{\left|\mathbf{R}_k^{i,j}\right|^{m^j-1/2}|\mathbf{S}^{i,j}|^{1/2}} \frac{\mathbf{\Gamma}(\alpha_{k|k}^{i,j})(\beta_{k|k-1}^i)^{\alpha_{k|k-1}^i}}{\mathbf{\Gamma}(\alpha_{k|k-1}^i)(\beta_{k|k}^{i,j})^{\alpha_{k|k}^{i,j}}}\ w_{k|k-1}^i \end{aligned} \tag{4.2}$$

The computations to obtain this result are straightforward using the measurement likelihood (3.35), and carrying out the same computations as in equation (3.17). The Gamma function and d-dimensional Gamma functions are still noted $\Gamma(\cdot)$ and $\Gamma_d(\cdot)$. The indexing is the same as the one found in the GIW measurement likelihood (4.1). Several parameters should be highlighted, $\mathbf{Y}^{i,j}$ describes how are distributed the measurements inside the extent, taking into account the error covariance matrix $\mathbf{R}_k^j$ and the extent covariance matrix $\mathbf{X}_{k|k-1}^j$. The computation of $\mathbf{Y}^{i,j}$ is given in the chapter 3 for Lan and Feldmann RM proposals.

We have used the GIW and GGIW models with two RFS filters, the Probability Hypothesis Density filter, and the Poisson Multi Bernoulli filter. The implementations are discussed in the following sections.

4.3 The Group Object Probability Hypothesis Density filter

4.3.1 The GIW mixture as a Poisson intensity, for a stable number of measurements

In the chapter 2, the intensity corresponding to isolated objects was assumed to be a Gaussian Mixture. Since we are dealing with group objects in this chapter, considering a Gaussian Inverse Wishart mixture intensity is appropriate. Being an unlabeled RFS filter, the PHD filter is adapted to simple scenarios. In that regard, we will assume a low clutter use case with a stable number of measurements returned by group objects for this filter. Hence, we drop the Gamma Gaussian Inverse Wishart group state model in this section.

For the Group Object Probability Hypothesis Density (GO-PHD) filter, our goal is to estimate the intensity $I(\mathbf{x}_k)$ of the Poisson multi-object probability density that models the tracked group objects:

$$I(\mathbf{x}_k) = \sum_{i=1}^{n_k} w_k^i\ \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k^i, \mathbf{p}_k^i \otimes \mathbf{V})\ \mathcal{IW}(\mathbf{X}; \nu_k^i, \mathbf{V}_k^i) \tag{4.3}$$

With this probability density, for each tracked object $\mathbf{x}_k^i \in X_k$ we need to estimate the state vector expectation $\hat{\mathbf{x}}_k^i$ and covariance matrix $\mathbf{p}_k^i$ of the Gaussian density, as well as the extent degree of freedom ν_k^i and the scaling matrix $\mathbf{V}_k^i$. When considering the addition

of the GIW mixture weight w_k^i, this model result in the estimation of n_k sets of parameters $(w_k^i, \hat{\mathbf{x}}_k^i, \mathbf{p}_k^i, \nu_k^i, \mathbf{V}_k^i)$, for each of the n_k tracked objects $\mathbf{x}_k^i$. However, with this parametrization, the number m^j of measurements returned per object is assumed stable through time.

The integration of the GIW mixture into the Probability Hypothesis Density (PHD) filter is seamless: it is just a matter of using the predictions and corrections equations described in the previous chapter coupled with the measurement likelihood (4.1) for mixture weights. Indeed, the PHD filter architecture does not depend on the object state model, meaning that Feldmann's or Lan's state models described in the chapter 3 can be used interchangeably.

4.3.2 GO-PHD filter prediction

The prediction step can be decomposed into two parts: first, forecasting the state of already tracked objects and then predicting the birth of new objects.

4.3.2.1 Already tracked group objects

At time $k-1$, the $n_{k-1|k-1}$ tracked group objects are described by a GIW mixture. Each tracked group object $\mathbf{x}_{k-1|k-1}^i \in X_{k-1|k-1}$ consists of the set of parameters $(w_{k-1|k-1}^i, \hat{\mathbf{x}}_{k-1|k-1}^i, \mathbf{p}_{k-1|k-1}^i, \nu_{k-1|k-1}^i, \mathbf{V}_{k-1|k-1}^i)$.

For each group, the prediction step consists of the application of the prediction equations of the chosen state model, whether it is Lan's or Feldmann's RM model, and the predicted weight is given by

$$w_{k|k-1} = p_S \; w_{k-1|k-1} \tag{4.4}$$

where p_S is the survival probability of the group object. The equation (4.4) is the same as the one found in the prediction of the PHD filter in the chapter 1. The resulting intensity is denoted $I_{k|k-1}^s(\mathbf{x})$.

4.3.2.2 New group objects

New group objects are created during the prediction step. The set of n^b new group objects X^b is subject to Poisson RFS with intensity $I^b(\mathbf{x})$. This intensity is a GIW mixture consisting of n^b components $\mathbf{x}^{b,i} \in X^b$, where each $\mathbf{x}^{b,i}$ is given by the set of parameters $(w^{b,i}, \hat{\mathbf{x}}^{b,i}, \mathbf{p}^{b,i}, \nu^{b,i}, \mathbf{V}^{b,i})$.

Like the PHD filter, the GO-PHD filter prediction is handled by concatenating the components of the surviving tracked object intensity, $I_{k|k-1}^s(\mathbf{x})$, with the components of the birth object intensity, $I^b(\mathbf{x})$. The resulting intensity is denoted $I_{k|k-1}(\mathbf{x}) = I_{k|k-1}^s(\mathbf{x}) + I^b(\mathbf{x})$, and it consists of $n_{k|k-1} = n_{k-1|k-1} + n^b$ components.

4.3.3 GO-PHD filter correction

The correction step of the GO-PHD filter can also be decomposed into two parts: first, the creation of a missed detection hypothesis for each tracked object, then the creation of a detection hypothesis for each object and measurement couple.

4.3.3.1 Missed group objects

It is the same as for the PHD filter: if a group object is missed, the predicted component is validated, and the weight is updated, such as

$$w^i_{k|k} = (1 - p_D) w^i_{k|k-1} \tag{4.5}$$

where p_D is the detection probability. For each of the $n_{k-1|k}$ objects $\mathbf{x}^i_{k|k}$, it results in the missed detection hypothesis components $(w^i_{k|k}, \hat{\mathbf{x}}^i_{k|k-1}, \mathbf{p}^i_{k|k-1}, \nu^i_{k|k-1}, \mathbf{V}^i_{k|k-1})$.

In addition to the missed hypothesis, the detection hypotheses must be computed.

4.3.3.2 Detected group objects

Each association between one extended measurement $Z^j_k = \mathbf{z}^1_k, ..., \mathbf{z}^{m^j}_k \subseteq Z_k = \bigcup_{j=1}^{M} Z^j_k$ and one predicted object intensity component $\mathbf{x}^i_{k|k-1} \in X^{k|k-1} = \{\mathbf{x}^1_{k|k-1}, ..., \mathbf{x}^{n_{k|k-1}}_{k|k-1}\}$ gives rise to an updated GIW component $\mathbf{x}^{i'}_{k|k}$. This new component has the index $i' = i + j \times n_{k|k-1}$. It should be noted that the index i' starts at $n_{k|k-1} + 1$ because of the missed detection hypotheses. The weight of this association is given by the equation (4.1), normalized over the measurements:

$$\begin{aligned} i' &= i + j \times n_{k|k-1} \\ w^{i'}_{k|k} &= \frac{c^{i,j}}{\sum_{j'=1}^{M} c^{i,j'}} \end{aligned} \tag{4.6}$$

The rest of the components for this association are computed according to the chosen RM state model described in chapter 3. For each $\mathbf{x}^{i'}_{k|k}$, the resulting parameters are $(w^{i'}_{k|k}, \hat{\mathbf{x}}^{i'}_{k|k}, \mathbf{P}^{i'}_{k|k}, \nu^{i'}_{k|k}, \mathbf{V}^{i'}_{k|k})$.

In total, there are $M \times n_{k|k-1}$ associations hypotheses and $n_{k|k-1}$ missed hypotheses that compose the updated components of the intensity $I_{k|k}(\mathbf{x})$. This leads to $n_{k|k} = M \times n_{k|k-1} + n_{k|k-1}$ updated mixture components for the intensity $I_{k|k}(\mathbf{x})$.

4.3.4 Implementation of the Group Objects Probability Hypothesis Density filter

The risk of combinatorial explosion, due to the correction step of the GO-PHD filter, is managed with a merging algorithm and a pruning algorithm. They can be found in the

chapter 1.

For the clustering algorithm, the DBSCAN algorithm with one cluster size is enough for our application, as discussed in the section 4.2. Even if the clustering size is too small, leading to an over-clustered measurement space, the merging algorithm will merge the resulting group objects.

Regarding the initialization of new objects, instead of a birth object GIW intensity placed at the edged of the field of view, we proposed a uniform intensity covering the whole field of view. This idea is pictured in the figure 4.3.

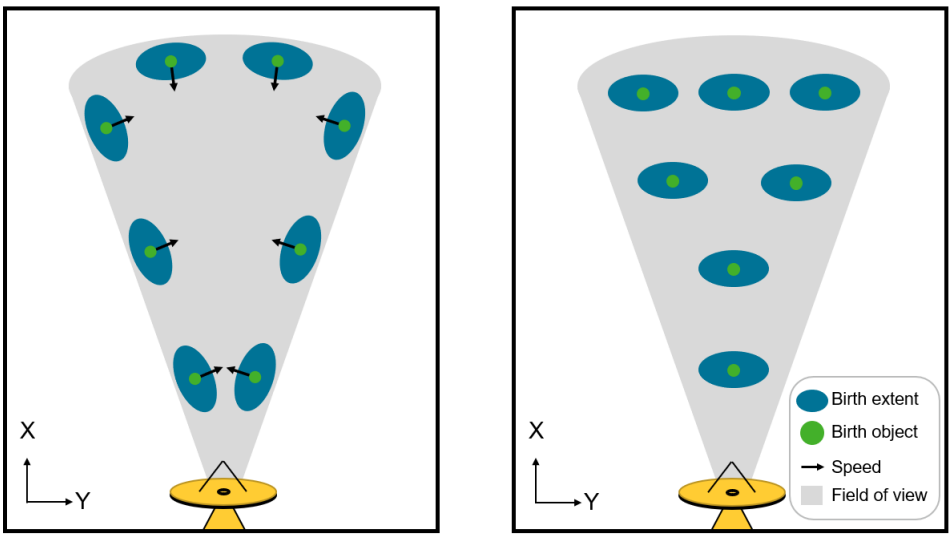


Figure 4.3: Birth objects layout.

The implementation of the filter is detailed in the block diagram of the figure 4.4. The GIW update or prediction equations can be found in the previous chapter. The complete filter is not very different from the original PHD filter. It only adds the estimation of ν_k^i and $\mathbf{V}_k^i$ for the extent, as well as a clustering algorithm.

Granström *et al.* proposed a similar filter in [GFS19]. This work was further extended in [Gra+15], including Feldmann's RM state model. When compared to their work, our contributions to the extended objects tracking fields are the following:

- we have considered the measurements noise in addition to the extent by integrating Lan's RM model in our proposal.
- we have modified Lan's RM model, assuming a state vector independent from the extent.
- in our application, we considered the under-resolved aspect of group objects.
- we have considered a hybrid approach to solve the cohabitation between isolated objects, under-resolved group objects, and group objects.

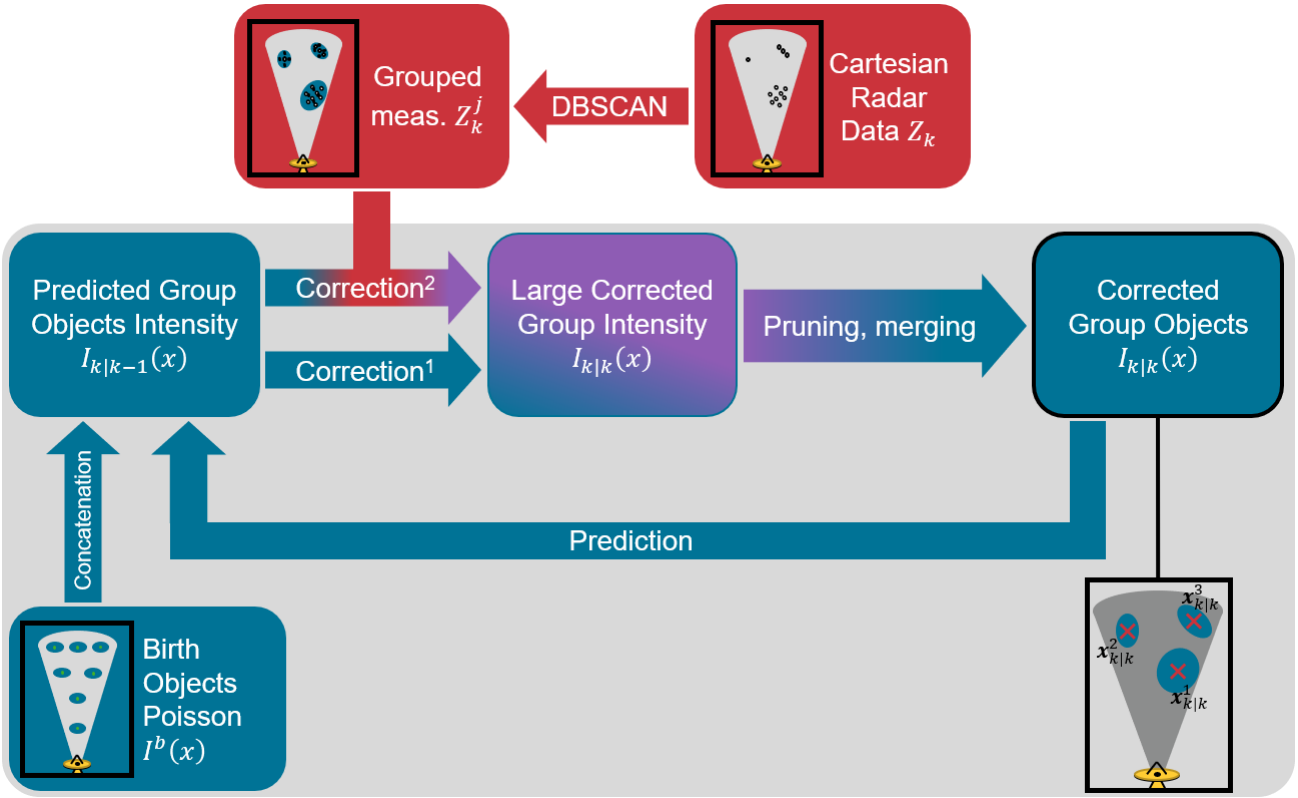


Figure 4.4: Block diagram of the GO-PHD filter.

- we have used this work on a real dataset on a passive radar sensor, in noisy conditions.
- we showed that in cases with high measurement noise, our approach can estimate the size of the extent where Feldmann's RM model fails.

These contributions can be found in the conference paper [Gue+21].

The main drawback of this filter is that it is an unlabeled approach, leading to the swift destruction of tracked objects when the detection probability is low. In addition, the GIW Random Matrix state model fails to estimate the number of detection an object should return. Because of these inconveniences, we explored the Group Object Poisson Multi-Bernoulli (GO-PMB) filter for more challenging datasets: a labeled RFS filter coupled with the Gamma Gaussian Inverse Wishart (GGIW) state probability density.

4.4 The Group Object Poisson Multi-Bernoulli filter

The Poisson Multi-Bernoulli filter was presented in the chapter 2. The proposed implementation was linear Gaussian. In this chapter, we aim at tracking multiple extended objects. Hence, this section introduces the Group Object Poisson Multi-Bernoulli (GO-PMB) filter. It

is a Poisson Multi-Bernoulli filter associated with a Random Matrix group object model. It relies on the efficiency of the Poisson density for unknown objects and the descriptive power of the Multi-Bernoulli density for tracked objects. Unknown objects are objects that are assumed to exist but that have not been detected yet.

The labeled implementation of the PMB filter was introduced in the chapter 2. Since two RFS probability densities need to be estimated, a Poisson density and a Multi-Bernoulli density, interfacing the Random Matrix model with a PMB filter involves two group objects parameterization.

The group object state model we chose for this filter is the Gamma Gaussian Inverse Wishart (GGIW) RM model first proposed by Granström *et al.* in [GO12a]. They suggested adding a Gamma density to the GIW model to estimate the average number of measurements returned by a group object.

When using the GGIW state model, two additional parameters need to be estimated compared to the GIW model: α_k^i, and β_k^i for each object $\mathbf{x}_k^i \in X_k$. They represent the Gamma density parameters that model the expected number of measurements returned by a group object.

4.4.1 The GGIW mixture as a Poisson intensity for unknown and new objects

The GO-PMB requires two Poisson intensities for unknown and birth group objects. They are described by their intensity: a GGIW mixture. It resembles the GIW mixture, equation (4.3), with the addition of the Gamma density:

$$I(\mathbf{x}) = \sum_{i=1}^{n_k} w_k^i \; \mathcal{G}(\gamma; \alpha_k^i, \beta_k^i) \; \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_k^i, \mathbf{p}_k^i) \; \mathcal{IW}(\mathbf{X}; \nu_k^i, \mathbf{V}_k^i) \tag{4.7}$$

where, for each tracked object $\mathbf{x}_k^i$, we need to estimate the same parameters as in (4.3), with the addition of the shape α_k^i and the rate β_k^i of the Gamma density: $(w_k^i, \hat{\mathbf{x}}_k^i, \mathbf{P}_k^i, \nu_k, \mathbf{V}_k^i, \alpha_k^i, \beta_k^i)$.

4.4.2 Coupling the Multi-Bernoulli density with a GGIW Random Matrix model

Integrating the Gamma Gaussian Inverse Wishart modeling for Extended Objects in a Multi-Bernoulli density is straightforward. From a parametrization perspective, for each tracked object $\mathbf{x}_k^i$, it just consists in adding an existence probability density r_k^i to the GGIW model.

More explicitly, a Multi-Bernoulli density is just the union of n_k single Bernoulli densities.

One GGIW Bernoulli density can be expressed as

$$p_b^{h,i}(X) = \begin{cases} 1 - r_k^i & \text{if } X = \{\emptyset\} \\ r_k^i \ \mathcal{G}(\gamma; \alpha_k^i, \beta_k^i)\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}_k^i, \mathbf{P}_k^i\mathbf{V}) \ \mathcal{IW}(\mathbf{X}; \nu_k^i, \mathbf{V}_k^i) & \text{if } X = \{x\} \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Hence, for a single object $\mathbf{x}_k^i$, the set of parameters to estimate is $(r_k^i, l^i, \hat{\mathbf{x}}_k^i, \mathbf{P}_k^i, \nu_k, \mathbf{V}_k^i, \alpha_k^i, \beta_k^i)$. In our case, we add a label l^i to identify the object $\mathbf{x}_k^i$.

4.4.3 GO-PMB filter prediction

The GO-PMB filter prediction equations are composed of the prediction equations for the MB density and the prediction equations for the Poisson unknown and birth groups intensities.

4.4.3.1 Already tracked group objects

At time $k-1$, the $n_{k-1|k-1}$ tracked group objects are modeled as a Multi Bernoulli (MB). Each tracked group object $\mathbf{x}_{k-1|k-1}^i \in X_{k-1|k-1}$ consists of the set of parameters $(w_{k-1|k-1}^i, \hat{\mathbf{x}}_{k-1|k-1}^i, \mathbf{P}_{k-1|k-1}^i, \nu_{k-1|k-1}^i, \mathbf{V}_{k-1|k-1}^i, \alpha_{k-1|k-1}^i, \beta_{k-1|k-1}^i)$.

For each group $\mathbf{x}_{k-1|k-1}^i \in X_{k-1|k-1}$, the goal of the prediction step is to compute the prior probability density parameters $(w_{k|k-1}^i, \hat{\mathbf{x}}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i, \nu_{k|k-1}^i, \mathbf{V}_{k|k-1}^i, \alpha_{k|k-1}^i, \beta_{k|k-1}^i)$. It corresponds to the application of the GGIW model prediction equations for the state model, and the predicted existence probability is given by

$$r_{k|k-1}^i = p_S \ r_{k-1|k-1}^i \tag{4.9}$$

where p_S is the survival probability of the group object. The equation (4.9) is the same as the one found in the prediction of the PMB filter in the chapter 1. The cardinality of the resulting MB density remains the same, $n_{k|k-1} = n_{k-1|k-1}$.

4.4.3.2 Unknown and new group objects

The prediction step of the unknown objects' Poisson density is the same as any Poisson density. For each component of the updated unknown object intensity $I_{k-1|k-1}^u$, the prediction step consists of the application of the GGIW state model prediction equations. The resulting surviving unknown objects intensity is denoted $I_{k|k-1}^{su}$, it consists of $n_{k-1|k-1}^u$ components.

New group objects are created during the prediction step. The set of n^b new group objects X^b is subject to Poisson RFS with intensity $I^b(\mathbf{x})$. This intensity is a GIW mixture consisting of n^b components $\mathbf{x}^{b,i} \in X^b$, with the parameters $(w^{b,i}, \hat{\mathbf{x}}^{b,i}, \mathbf{P}^{b,i}, \nu^{b,i}, \mathbf{V}^{b,i})$.

Like the GO-PHD filter prediction, the components of the surviving unknown objects intensity, $I_{k|k-1}^{su}(\mathbf{x})$, are concatenated with the components of the birth object intensity, $I^b(\mathbf{x})$.

The resulting intensity is denoted $I^u_{k|k-1}(\mathbf{x}) = I^{su}_{k|k-1}(\mathbf{x}) + I^b(\mathbf{x})$, and it consists of $n^u_{k|k-1} = n^u_{k-1|k-1} + n^b$ components.

4.4.4 GO-PMB filter correction

The GO-PMB filter correction step can be decomposed into three parts: the missed tracked objects MB update, the detected object MB update, and the Poisson intensity update.

4.4.4.1 Missed group objects

If a tracked group object is missed, it either does not exist or was just not detected by the sensor. These two cases create a bimodal Multi Bernoulli component that is reduced to a mono-modal one using the MB mixture reduction. The resulting Bernoulli component is the validation of the predicted GGIW density with the updated existence probability

$$r^i_{k|k} = \frac{r^i_{k-1|k} q_D}{1 - r^i_{k-1|k} + r^i_{k-1|k} q_D} \tag{4.10}$$

where q_D is given by

$$q_D = 1 - p_D + p_D(\frac{\beta^i_{k-1|k}}{\beta^i_{k-1|k} + 1})^{\alpha^i_{k-1|k-1}} \tag{4.11}$$

with the detection probability p_D. In addition, to select only one Multi Bernoulli after the update, the likelihood $w^{i,0}$ of the missed detection is computed, such as

$$w^{i,0}_{k|k} = \frac{r^i_{k-1|k} q_D}{1 - r^i_{k-1|k} + r^i_{k-1|k} q_D} \tag{4.12}$$

It results in the $n_{k-1|k}$ missed detection hypothesis components $\mathbf{x}^i_{k|k}$ consisting of the parameters $(w^i_{k|k}, \hat{\mathbf{x}}^i_{k|k-1}, \mathbf{P}^i_{k|k-1}, \nu^i_{k|k-1}, \mathbf{V}^i_{k|k-1})$.

In addition to the missed hypotheses, the detection hypotheses must be computed.

4.4.4.2 Detected group objects

The detection hypotheses are computed for each tracked group object and extended measurement couple. With the extended measurement Z^j_k and the Bernoulli density $\mathbf{x}^i_{k|k} \in X_{k|k-1}$, it just consists in the application of the GGIW update equations discussed in the chapter 3. In addition, the existence probability is also updated

$$r^i_{k|k} = 1 \tag{4.13}$$

where p_D is the detection probability. The likelihood of this association is also computed to select only one Multi Bernoulli component at the end of the update. It is given by

$$w^{i,j} = r^i_{k|k-1}\ c^{i,j} \tag{4.14}$$

with $c^{i,j}$ the likelihood given in equation (4.2).

This results in the computation of $M \times n_{k|k-1}$ detection hypotheses that are added to the $n_{k|k-1}$ missed detection hypotheses. The correction step also initializes new objects. They are created with the help of the unknown object Poisson intensity.

4.4.4.3 From unknown Poisson intensity to newly detected group objects

The unknown objects intensity $I^u_{k|k-1}(\mathbf{x})$ unique purpose is to create new detected object Bernoulli densities. The process is simple: each extended measurement Z^j_k gives rise to a new Bernoulli density that is computed using the Poisson intensity $I^u_{k|k-1}(\mathbf{x})$. This is a two steps procedure, first updating each of the $n^u_{k|k-1}$ components of the unknown object intensity with the measurement Z^j_k:

$$\begin{aligned}
\mu^l &= w^{u,l}_{k|k-1}\ c^{l,j} \\
\mathbf{y}^l &= \hat{\mathbf{x}}^{u,l}_{k|k-1} - \mathbf{K}\epsilon \\
\mathbf{P}^l &= \mathbf{P}^{u,l}_{k|k-1} - \mathbf{K}\mathbf{H}\mathbf{P}^{u,l}_{k|k-1} \\
\alpha^l &= \alpha^l_{k|k-1} + m^j \\
\beta^l &= \beta^l_{k|k-1} + 1 \\
\nu^l &= \nu^l_{k|k-1} + m^j \\
\mathbf{V}^l &= \mathbf{V}^l_{k|k-1} + \mathbf{Z}^j_k + \mathbf{N}
\end{aligned}
\qquad
\begin{aligned}
&\text{with} \quad l \in [\![1, n^u_{k|k-1}]\!], \text{ and} \\
\overline{\mathbf{z}}^j_k &= \tfrac{1}{m^j} \textstyle\sum_{j'=1}^{m^j} \mathbf{z}^{j'}_k \\
\mathbf{Z}^j_k &= \tfrac{1}{m^j-1} \textstyle\sum_{j'=1}^{m^j} (\mathbf{z}^{j'}_k - \overline{\mathbf{z}}^j_k)(\mathbf{z}^{j'}_k - \overline{\mathbf{z}}^j_k)^T \\
\mathbf{X} &= \tfrac{\mathbf{V}^l_{k|k-1}}{\nu^l_{k|k-1} - 2\ d - 2} \\
\epsilon &= (\overline{\mathbf{z}}^j_k - \mathbf{H}\hat{\mathbf{x}}^{u,l}_{k|k-1}) \\
\mathbf{S} &= \mathbf{H}\mathbf{P}^{u,l}_{k|k-1}\mathbf{H}^T + \tfrac{\mathbf{X}}{m^j} \\
\mathbf{K} &= \mathbf{P}^{u,l}_{k|k-1}\mathbf{H}^T\mathbf{S}^{-1} \\
\mathbf{N} &= \mathbf{X}^{1/2}\mathbf{S}^{-1/2}\epsilon\epsilon^T\mathbf{S}^{-T/2}\mathbf{X}^{T/2}
\end{aligned}
\tag{4.15}$$

where the index l denotes that these updated parameters are temporary, and $c^{l,j}$ is the GGIW association likelihood given by the equation (4.2). The equations (4.15) are given using Gränstrom *et al.* proposal [GFS16], and nothing prevents us from using our proposal. Indeed, the new Bernoulli component is computed as the average of these parameters, such as

$$\begin{aligned}
C &= \textstyle\sum_{l=1}^{n^u_{k|k-1}} \mu^l \\
w^{i',j} &= C + (\lambda^{FA})^{m^j} \\
r^i_{k|k} &= \tfrac{C}{C + (\lambda_{FA})^{m^j}} \\
\hat{\mathbf{x}}^i_{k|k} &= \tfrac{\sum_{l=1}^{n^u_{k|k-1}} \mu^l\ \mathbf{y}^l}{C} \\
\mathbf{P}^i_{k|k} &= \tfrac{\sum_{l=1}^{n^u_{k|k-1}} \mu^l \left(\mathbf{P}^l + (\hat{\mathbf{x}}^i_{k|k} - \mathbf{y}^l)(\hat{\mathbf{x}}^i_{k|k} - \mathbf{y}^l)^T\right)}{C}
\end{aligned}
\qquad
\begin{aligned}
\alpha^i_{k|k} &= \tfrac{\sum_{l=1}^{n^u_{k|k-1}} \mu^l\ \alpha^l}{C} \\
\beta^i_{k|k} &= \tfrac{\sum_{l=1}^{n^u_{k|k-1}} \mu^l\ \beta^l}{C} \\
\nu^i_{k|k} &= \tfrac{\sum_{l=1}^{n^u_{k|k-1}} \mu^l\ \nu^l}{C} \\
\mathbf{V}^i_{k|k} &= \tfrac{\sum_{l=1}^{n^u_{k|k-1}} \mu^l\ \mathbf{V}^l}{C}
\end{aligned}
\tag{4.16}$$

where i starts at $M \times n_{k|k-1} + n_{k|k-1} + 1$ to account for the missed and detected hypotheses discussed above. Since $w^{i',j}$ is the weight of this association, i' starts at $n_{k|k-1} + 1$, following the association matrix layout given in figure 2.4.

At this point, the missed, detected, and birth Bernoulli hypotheses are computed. Similar to what was done with the PMBM filter in the chapter 2, the best Multi Bernoulli density is selected using Murty's assignment algorithm. To complete the correction step of the GO-PMB filter, the unknown objects' Poisson intensity must be updated.

The Poisson intensity corresponding to unknown group objects $I^u_{k|k-1}(\mathbf{x})$ is updated as a missed hypothesis. Hence, like for the GO-PHD or PHD filters missed detection, $I^u_{k|k}(\mathbf{x})$ is just the validation of $I^u_{k|k-1}(\mathbf{x})$ with every components weight tempered by the non detection probability $1 - p_D$:

$$w^{u,i}_{k|k} = (1 - p_D) w^{u,i}_{k|k-1} \tag{4.17}$$

This leads to the updated uknown objects intensity $I^u_{k|k}(\mathbf{x})$, that is a mixture of $n^u_{k|k} = n^u_{k|k-1}$ GGIW state probability densities.

4.4.5 Implementation of the Group Object Poisson Multi Bernoulli filter

The implementation of this filter is straightforward: as for the GO-PHD filter, it is just the integration of the GGIW Random Matrix model in the PMB filter. The final implementation also benefits from the performance improvements brought by the gating, pruning, and merging algorithms detailed in the chapter 1. As for the clustering, it is also left to the DBSCAN algorithm. The complete implementation block diagram can be found in the figure 4.5.

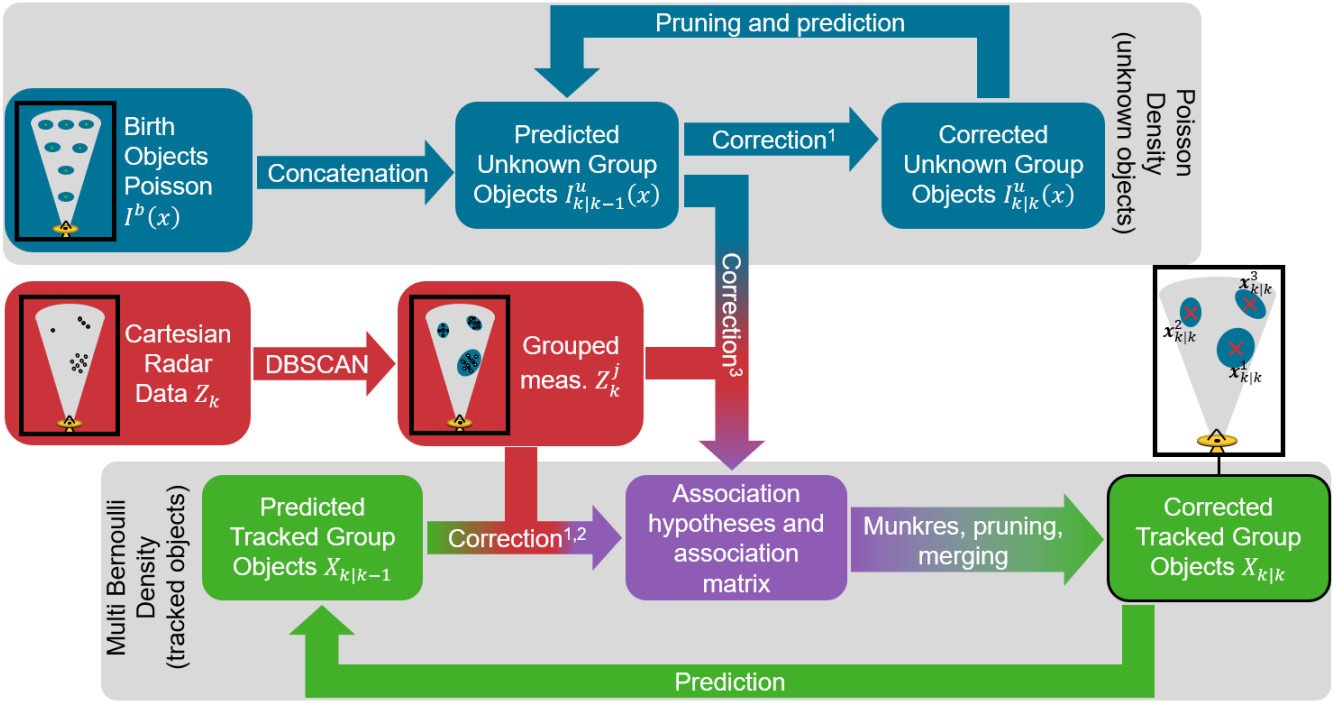


Figure 4.5: Block diagram of the GO-PMB filter.

Granström *et al.* proposed in [GFS16] a GGIW implementation of the PMB filter based on Feldmann's RM state model. They only proposed a simulation study to validate their results.

Building on this work, we have proposed in [Gue+20] the following contributions:

- using our GGIW RM state model to estimate the extent of groups of drones.
- taking into consideration under-resolved objects.
- testing the algorithms on both a simulated and a real dataset.
- incorporating an evolution model to consider the growth of groups of drones.

4.5 Tracking groups of drones with RM state models and RFS filters

Tracking groups of drones with the RFS filters and RM state models necessitates several additional contributions and adaptations, starting with the under-resolved objects problem. Also, this section highlights the object management procedures we have implemented for both filters.

4.5.1 Switching between the under-resolved and extended object state

An under-resolved extended object is an object that returns less than the minimum number of measurements required by the RM model. In fact, ν_k^i, the degree of freedom of the Inverse Wishart density must be greater than $2\ d-2$, since the extent is the expectation of the Inverse Wishart density, given by

$$\mathbf{X}_k^i = \frac{\mathbf{V}_k^i}{\nu_k^i - 2\ d - 2} \tag{4.18}$$

The degree of freedom ν_k^i decreases with each prediction step and increases with each correction step. In our case, it should always be greater or equal to 7 since the number of spatial dimensions of the extent is $d = 2$.

To enforce this result, several solutions are conceivable:

- groups that return less than 4 measurements are ignored.
- switch between a model for under-resolved objects and group objects. For instance, switching between a Gaussian Mixture model for single objects and a Random Matrix model for groups.
- if a group is under-resolved and returns less than 3 measurements, virtual measurements can be created to turn any under-resolved objects into regular objects.

As illustrated in the figure 4.6, we have chosen the third option, turning under-resolved objects into regular extended objects by adding virtual measurements to the groups of measurements.

While the second proposal is more accurate than creating virtual measurements, the complexity overhead can be simply avoided by using the third option.

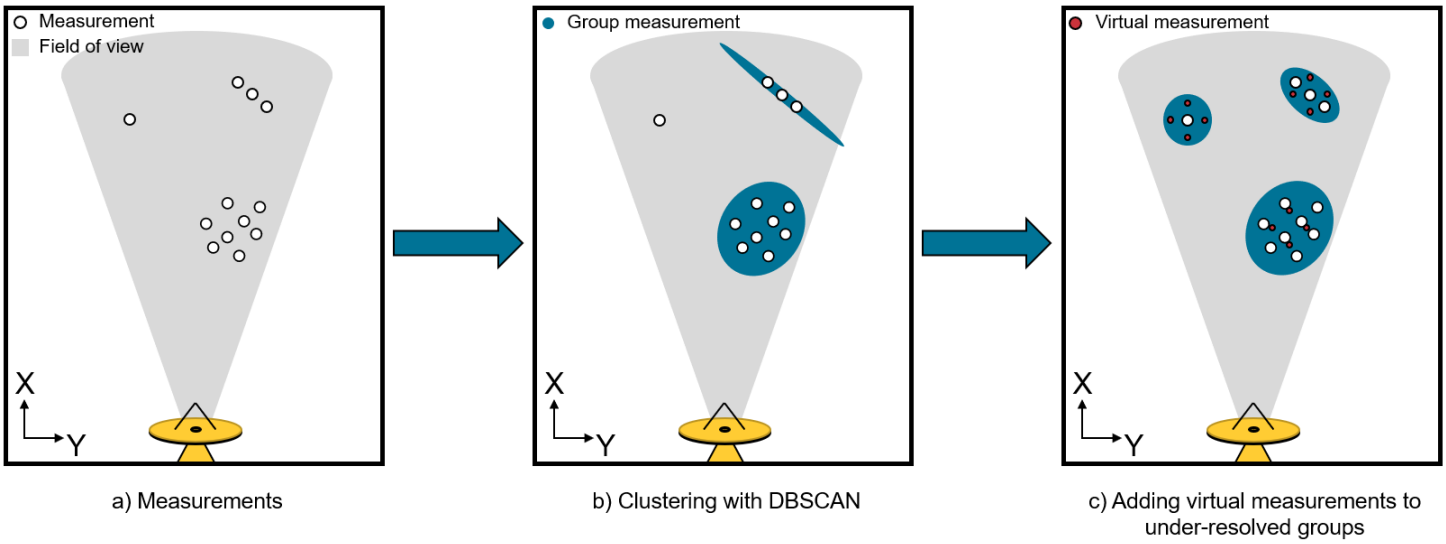


Figure 4.6: The creation of virtual extended measurements.

In addition, creating virtual measurements is not inaccurate. Any object is extended, and under-resolved objects are just smaller than the cell resolution of the sensor. Hence using virtual measurements that are spread at a distance smaller than the cell resolution of the sensor is a valid strategy.

In addition, the use of virtual measurements tackles another problem we encountered during our experiments: it enforces that the measured extent matrix is a symmetric definite positive matrix. For instance, if a group returns several measurements that are aligned, its measured extent will be ill-conditioned, with a rank inferior to the dimension of the matrix. Another example is when the measurements of a group are almost aligned, the measured extent might appear as a flattened ellipse with an overestimated size along its main axis. This issue is illustrated in the figure 4.6, with the group consisting of three measurements.

4.5.2 Objects management

The initialization procedure is different from what is described in most of the RFS literature. Since both filters, the GO-PHD[3] and the GO-PMB[4], rely on a Poisson RFS to model the birth of new objects, the usual layout for the initialization is to put objects at the edge of the field of view of the sensor.

However, we chose to use a uniform distribution of the birth densities in the sensor's field of view. It seems appropriate since the drones can take off and land anywhere in the field of

[3]Group Object Probability Hypothesis density
[4]Group Object Poisson Multi-Bernoulli

view. This method ensures that measurements are always close to a predicted birth object. However, it comes with a drawback, the initialization of unwanted false alarms.

To tackle this problem, an initialization delay exists in our implementations. New objects need to go through 3 recursions of the algorithm to be displayed. In addition, Each object's weight or existence probability needs to be higher than a threshold to be displayed. Both of these assumptions ensure a good filtration of the false alarms. The birth object density is illustrated in the figure 4.3.

Both algorithms use a delay and a pruning threshold to terminate objects that are no longer detected by our sensor. For the GO-PHD, the pruning threshold is tuned. On the contrary, the GO-PMB threshold is set to

$$T_{tracked\ objects} = w^b\ p_S^2\ (1 - p_D)^3 \tag{4.19}$$

where w_b is the initialization weight of an object, p_S is the survival probability of a tracked object, and p_D is its detection probability. This threshold ensures that tracked objects are terminated after three iterations.

We could not take the approach of the equation (4.19) with the GO-PHD filter since it is not a labeled filter. Pruning the weight of a mixture with such a threshold would lead to a combinatorial explosion of the number of tracked objects. Indeed, at each update, one component gives birth to $m + 1$ component, m being the number of measurements.

Finally, the pruning and merging procedures of the objects are also implemented in the GO-PHD and GO-PMB filters. As in the chapter 1, they rely on the Mahalanobis distance between objects and a threshold T_{merge} for the merging step, and a threshold T_{thresh} for the pruning step.

4.5.3 Shape changing group objects

One of the main contributions of our work is taking into account the varying shape aspect of the group objects. In the literature, this subject is discussed by Lan *et al.* in [LL12a]. They propose to use an evolution matrix $\mathbf{A}$ during the prediction step. Apart from the simulated dataset they use to validate their model, there is a lack of application to real data regarding this approach. In addition, the data they consider corresponds to a rigid extended object, a simulated aircraft carrier in their case. Lan *et al.* also proposed results on a group object in another paper about this subject, [LL16]. However, it is a simulated dataset of a well-resolved group object.

Thus, we propose using this model and evaluating it on real data with under-resolved objects. In this work, the evolution matrix $\mathbf{A}$ is set to

$$\mathbf{A} = \begin{pmatrix} \rho & 0 \\ 0 & \rho \end{pmatrix} \tag{4.20}$$

with ρ the inverse of the square root of δ_k. δ_k is the number of degrees of freedom of the evolution model for the extent. The higher δ_k, the higher the uncertainty regarding the extent. Here, δ_k is set to 2 and $\rho = \frac{1}{\sqrt{2}}$.

4.6 Presentation of the metrics

Five metrics are retained here: the Optimal Sub-Pattern Assignment (OSPA) metric [SVV08], the Generalized Optimal Sub-Pattern Assignment (GOSPA) metric [RGFS17], the cardinality of the estimated set of objects, the average Root Mean Squared Error (RMSE), and the average group object size.

The OSPA metric is straightforward to understand: either an object is tracked and its position RMSE compared to the ground truth can be computed, or it is missed or non-existent. In the figure 4.7, denote by X_k^{estim} the set of n estimated objects $\mathbf{x}_k^i$ at time k, and X_k^{gt} the ground truth set that includes m true objects $\mathbf{x}_k^{gt,j}$. Before computing the OSPA, an association map Π between X_k^{estim} and X_k^{gt} is computed using the Munkres assignment algorithm [Mun57]. This association map Π is a set of permutations over the set with the highest cardinality. As an example, let $m \leq n$, then $\Pi = \{\pi^1, ..., \pi^m\}$ is a set of m permutations, taking their values in $[\![1, n]\!]$. Using this set, the euclidean distance between each estimated object and its ground truth can be computed such as

$$d^c(\mathbf{x}_k^{gt,j}, \mathbf{x}_k^{\pi^j}) = min\left(||\mathbf{x}_k^{gt,j} - \mathbf{x}_k^{\pi^j}||, c\right) \tag{4.21}$$

where $||\cdot||$ stands for the euclidean distance between two vectors. The idea is that when these vectors are further apart than a maximum cutoff distance c, then the cost c is applied, and both vectors are considered independent: c can be seen as an error cost for a missed object.

The full OSPA metric takes into account the estimated yet false objects using the same cost c. Using the example of the figure 4.7, the $n - m$ remaining estimated objects are false objects and participate in the OSPA with the value of $(n - m)$ c.

$$d_{OSPA}(X_k^{estim}, X_k^{gt}) = \left(\frac{1}{n}\min_{\pi^j \in \Pi}\left(\sum_{j=1}^{m} d^c(\mathbf{x}_k^{gt,j}, \mathbf{x}_k^{\pi^j})^p + c^p(n-m)\right)\right)^{1/p} \tag{4.22}$$

In this equation, c is the cost of a mismatch between the ground truth set and the estimated set. The role of p is to weigh the impact of a mismatch between these sets. Indeed, if c remains constant and p increases, the metric will drastically increase with outliers.

Nonetheless, this metric has some drawbacks. Rahmathullah *et al.* showed in [RGFS17] that the normalization introduced in equation (4.22), the factor $\left(\frac{1}{n}\right)^{1/p}$, has some undesirable effect on the metric's behavior. For instance, let $X_k^{gt} = \{\emptyset\}$ and $X_k^{estim} = \{\mathbf{x}^1, ..., \mathbf{x}^n\}$, that is the case where only false alarms are tracked, then the OSPA metric is equal to $\left(\frac{n}{n}\right)^{1/p} c = c$. Ideally, this metric should increase with the number of false alarms, which is not the case here. That's why Rahmathullah *et al.* propose to remove the normalization in the Generalized OSPA

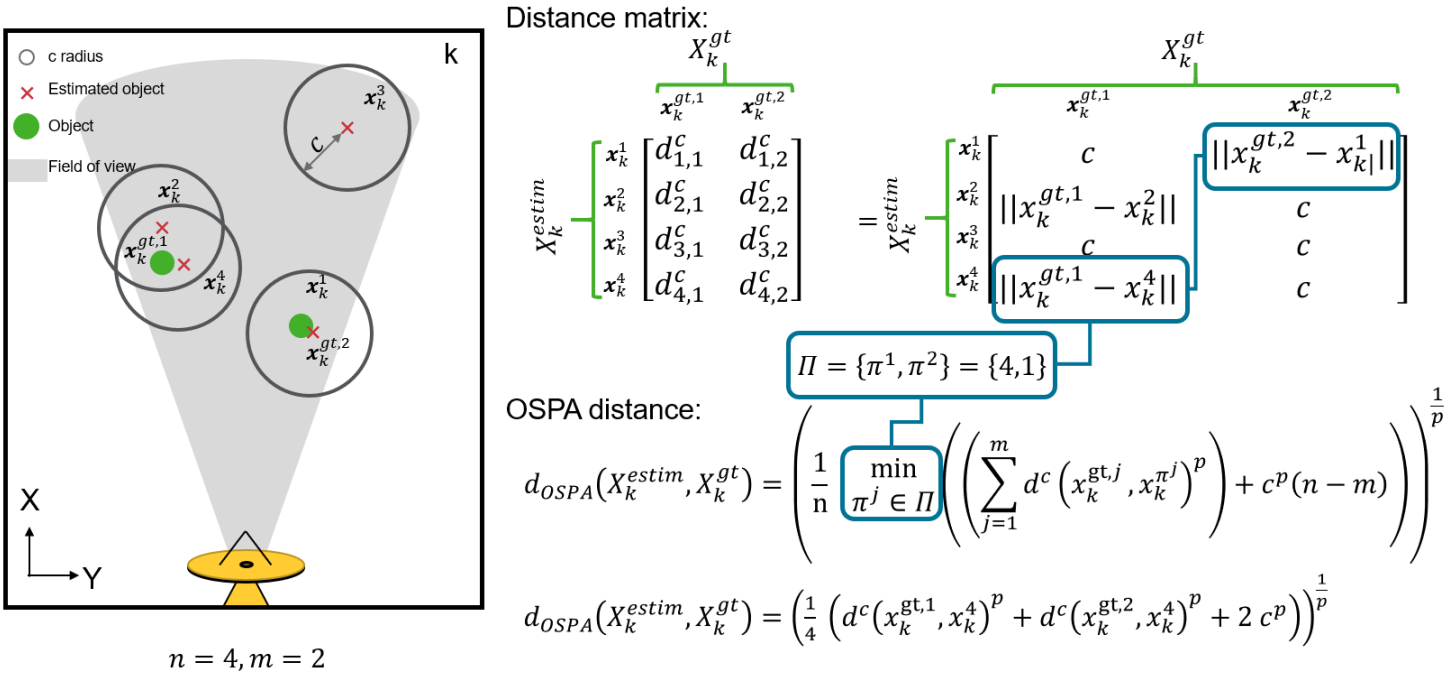


Figure 4.7: Example of OSPA metric computation.

metric (GOSPA). With $X_k^{gt} = \{\emptyset\}$ and $X_k^{estim} = \{\mathbf{x}^1, ..., \mathbf{x}^n\}$, the GOSPA metric is equal to $(n)^{1/p}\, c$, that augments with the number of false alarms.

The GOSPA metric can be defined as

$$d_{GOSPA}(X_k^{estim}, X_k^{gt}) = \left(\min_{j \in \Pi}\left(\sum_{i=1}^{m} d^c(x_k^{gt,j}, x_k^{\pi^j})^p + \frac{c^p}{\alpha}(n-m)\right)\right)^{1/p} \tag{4.23}$$

where $X_k^{gt} = \{\mathbf{x}_k^1, ..., \mathbf{x}_k^m\}$ and $X_k^{estim} = \{\mathbf{x}_k^1, ..., \mathbf{x}_k^n\}$, and $m < n$. α is a normalization term: if $\alpha = 1$ there is no normalization, and if $\alpha = 2$, the cost for a wrong assignment between the ground truth and the estimation is the same whether it is due to a wrong assignment or a non detected object. In our case we will always take $\alpha = 2$, however, the value of c will vary depending on the problem at hand.

In addition to the GOSPA metric, the average RMSE, the cardinality of the estimated set, and the estimated extent are used as metrics. For the estimated extent, two parameters are evaluated: the small half-axis size "a" and the big half-axis size "b", as shown in the figure 4.8.

These metrics will help us to compare the results of all the filters discussed in this chapter. We will first go through the performance of the unlabeled GO-PHD filter using two RM state models: Lan's approach and Feldmann's approach. Then, we will test our RM state model implementation using a labeled GO-PMB filter on a challenging active radar dataset.

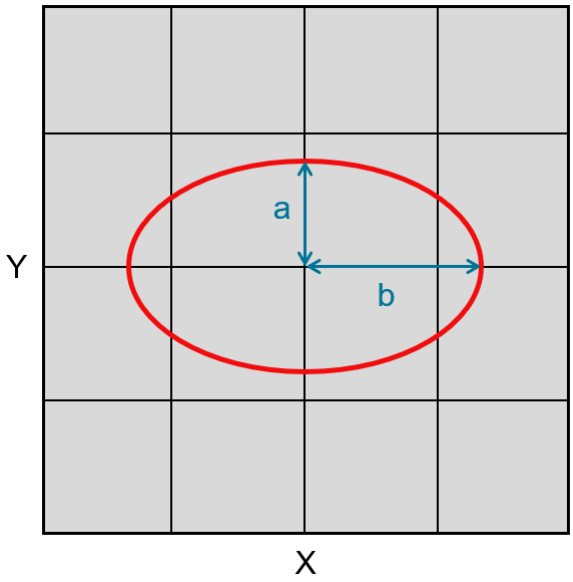


Figure 4.8: Extent size.

4.7 Results with a passive radar sensor

4.7.1 The passive radar sensor

Over the last few years, the interest in passive radar sensors to counter drones and swarms of drones intrusions in protected aerial spaces has dramatically increased [Mal+17]. The passive radar sensor technology relies on civilian radio frequency emitters, or opportunistic donors, to detect moving objects in the electromagnetic field they emit. Some interesting opportunistic donors are FM Frequency Modulation (FM) or Amplitude Modulation (AM) radio waves, Digital Audio Broadcasting (DAB), Terrestrial Digital Video Broadcasting (DVB-T).

When tracking drones and groups of drones, several parameters need to be considered to choose the most efficient emitter. For drones, the waveform parameters and properties are important. The carrier frequency, useful bandwidth, coverage at low altitude, efficient clutter cancellation of DVB-T emitters offer the best compromise for drone detection and tracking. These parameters are summed up in the table 4.1.

The DVB-T emitters are opportunistic donors for passive radar sensors. They constitute an array of civilian broadcasters working 24/7 with an “omnidirectional” illumination of the emitters’ neighborhood. Consequently, the detection of moving objects in this widely illuminated area can be conducted simultaneously for all the objects. In contrast, for classical scanning radar sensors, a wide area survey is achieved sequentially.

Another direct advantage of such a broad and constant illumination relies on the long Coherent Integration Time (CIT). The CIT duration of observations is directly based on the kinematic parameters of tracked objects to avoid range and Doppler migration. The typical

Parameters	Carrier Freq.v	Bandwidth	Power
Values	460-700 MHz	7.5 MHz	10-50 kW
Parameters	Min. range	Max. Range	Range standard deviation
Values	10 m	3000 m	30 m
Parameters	Azimuth beamwidth	Elevation beamwidth	Azim. standard deviation
Values	100 °	2 °	0.9 °
Parameters	Clutter rate	Sampling period	p_D
Values	15 FA/s	2 s	0.99

Table 4.1: DVB-T parameters.

DVB-T CIT for drones is close to 0.5 seconds (up to 1 second), leading to a 2 Hz resolution for Doppler and frequency effects.

The propellers of a fixed-wing or multi-rotor drone are responsible for a phenomenon called "micro-Doppler". Indeed, the blades of a propeller can be modeled as high-velocity rotating parts of the drone. Their contribution to the radar signature is often called the "micro-Doppler" phenomenon.

However, we prefer the following interpretation: since the contributions of the blades are not dependent on the carrier frequency or the drone's motion, their distinctive contribution cannot be considered a "micro-Doppler" phenomenon. Indeed, in its formal definition, the Doppler effect depends on the carrier frequency and the object motion. However, the contribution of the rotating blades relies on the induced periodicity of the global drone signature. For example, when considering typical rotation speeds between 6000 revolutions per minute (rpm) and 9000 rpm, the drone signature will present a periodicity between 100 and 150 Hz. Since these numbers are obtained using a Doppler analysis, the "micro-Doppler" naming shortcut appeared, even though it is not a Doppler effect. We prefer the term blade modulations for this phenomenon.

The figure 4.9 illustrates an example of the detection of blade modulations. In this figure, the drone is a quad-rotor, and an array of passive DVB-T receivers allow to estimate the bi-static range, bi-static Doppler/velocity, and azimuth of the drone. In this example, the drone performs a stationary flight for several minutes, meaning that the drone's Doppler/velocity is null. Nevertheless, because of the blades' modulations and the period they induce, the drone can be detected, and we can regress to the drone localization. The blades modulations show a period close to 120 Hz and a harmonic around 240 Hz. Of course, the modulations are positive and negative: when the propellers spin, their blades are getting closer and further from the array of DVB-T antennas.

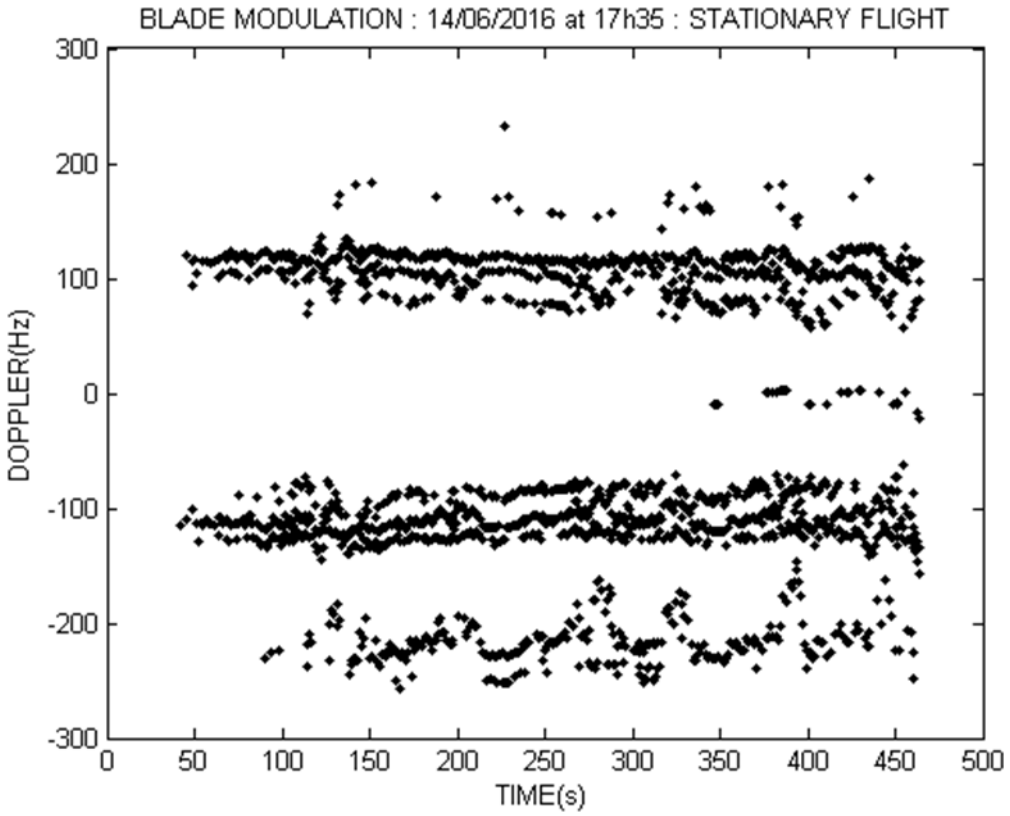


Figure 4.9: Blade modulations detection.

4.7.2 The group of drones dataset

In this scenario, three UAS are moving in close formation. Only one of them, the "Matrix 600-Pro", is equipped with a GPS sensor. The GPS's black trajectory gives the ground truth in the figure 4.10. The swarm starts the take-off at the *alpha* point and moves to the *beta* point (at 1.2 km from the passive radar sensor). After a maneuver, the swarm continues its way to the *gamma* point and returns to the *beta* point. It takes 12 minutes in total to complete this trajectory.

It should be noted that the radar output is aliased due to the radar post-processing: the measurements of an extended object are therefore mainly spread along one axis. Another remark concerns the ground truth between the *alpha* and *gamma* points: due to an error of the GPS, the ground truth is not accurate in this area.

As highlighted in figure 4.11.f, the ground truth is also inaccurate in two other places. First, the ground truth is not in the sensor's field of view during a brief moment at the point *gamma*. Second, one of the drones separated from the swarm at the end of the scenario because of an empty battery. Since it was not equipped with a GPS, there is no ground truth.

The main parameters of the filters for this scenario are summarized in the appendix B.1.

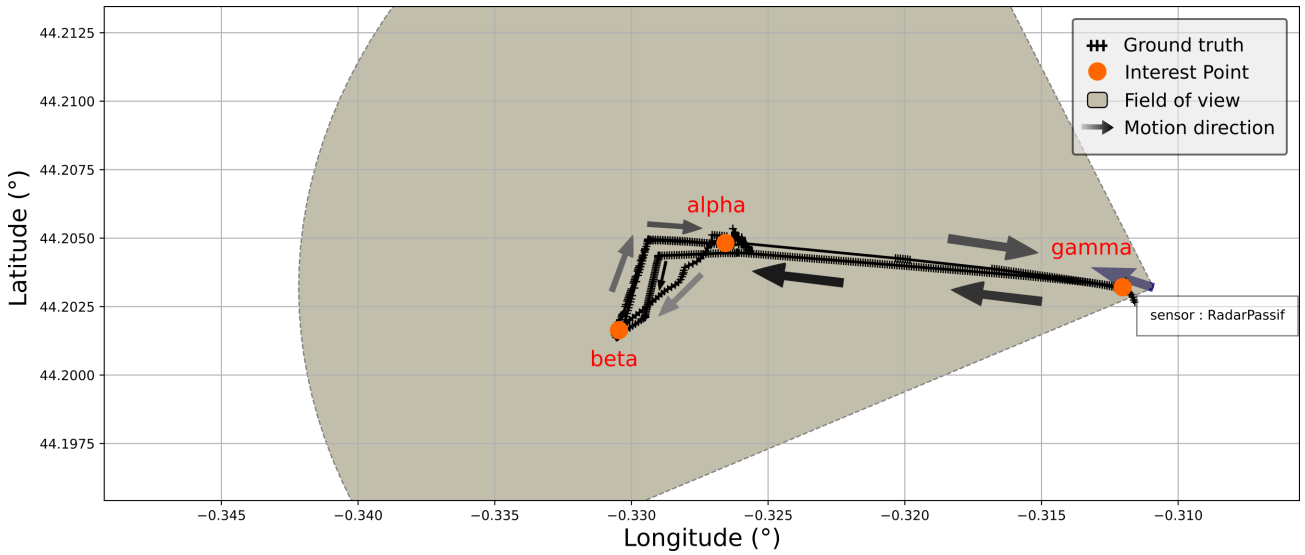


Figure 4.10: The passive radar sensor dataset

Whether it is for the GO-PHD with Feldmann's RM model, noted GO-PHD/F, or the GO-PHD with Lan's model, noted GO-PHD/L, the parameters are similar. The Generalized Optimal Sub Pattern Assignment metric (GOSPA) is used to evaluate their performances with the following parameters: $c = 30$ meters, $\alpha = 2$ and $p = 1$.

As shown in figure 4.11.c, the positioning error for both approaches is relatively high, at 28.5 meters on average for the first one and 28.6 meters on average for the second one. These numbers might seem significant, but the center of the group is not the position of the GPS. Moreover, the ground truth is not perfectly aligned with the measurements, and the GPS has its own positioning errors, inducing supplementary biases.

The sensor's output is subject to high noise. The cell resolution of the passive radar is low, and the drones seem to jump from one cell to another. This directly impacts the track structure and should be considered in the cinematic model in future work. Even if the position error is high for both approaches, they behave similarly, as attested by their GOSPA metric in figure 4.11.a: the plots are superposed during most of the scenario.

Both algorithms seem to perform similarly tracking-wise, but we experienced that the GO-PHD/F filter was more sensitive to false alarms and wrong associations, hence the higher pruning threshold given in B.1. There is a high delay of 7 recursions before validating a new object for both filters because they tend to validate false alarms too quickly.

The estimated extent of the group of drones is given in graphs 4.11.d, and 4.11.e. When comparing the extent estimated by both approaches, Lan's proposal provides an estimate within the good range but with high variance, and Feldmann's proposal converges rapidly to a negligible estimate. This difference in terms of estimation is due to the high variance of the measurements compared to the size of the group of drones. Feldmann's proposal assumes an exponential decay model for the extent estimate during the prediction, which appears

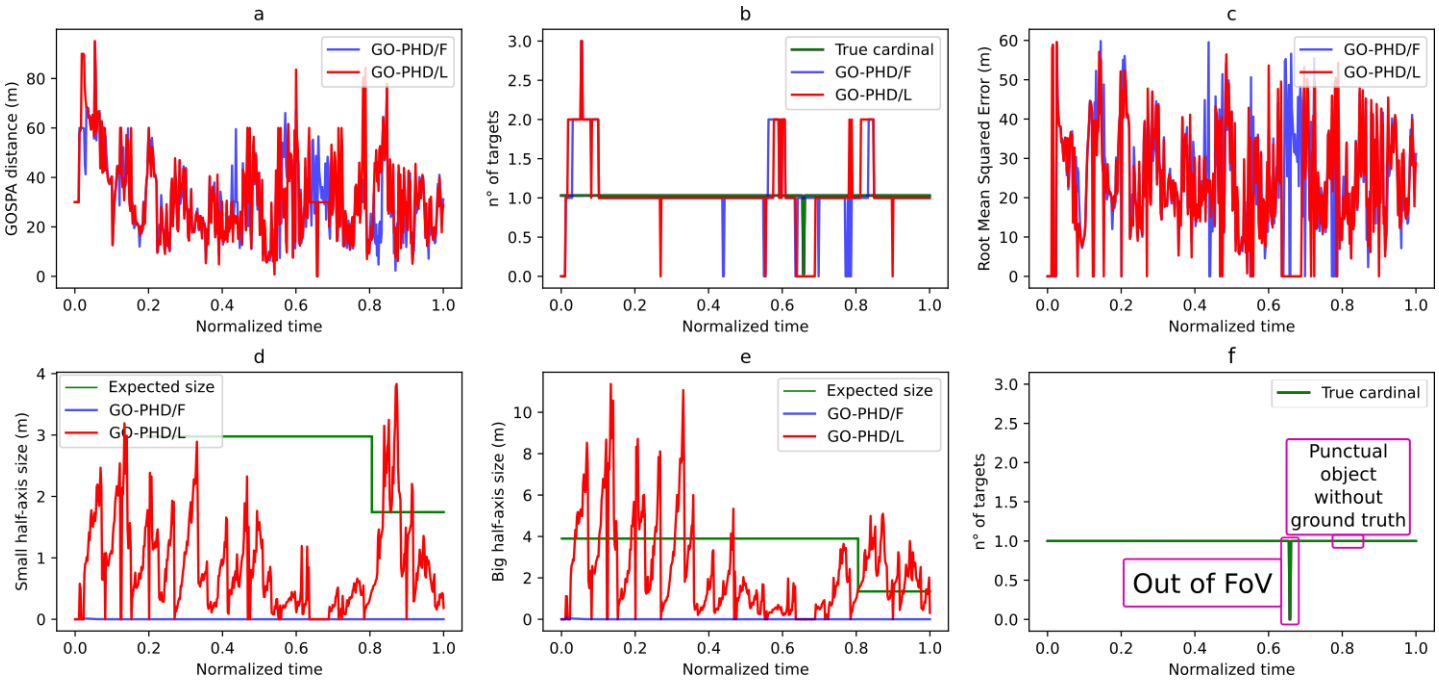


Figure 4.11: Metrics for the passive radar dataset

to decrease exponentially because the contribution of the measured extent is not significant during the update step. We could not get good results for this approach, except by minimizing the exponential decay.

Regarding the drone that separates itself from the swarm of drones, at normalized time 0.8 in figure 4.11.f, the GO-PMB filter with Lan's RM model, denoted GO-PMB/L, tracks it during almost all its return path to the *gamma* point. However, it loses it midway because the drone does not return any measurement during this timelapse. On the other hand, the GO-PMB filter with Feldmann's RM, denoted GO-PMB/F, struggles with this single isolated object, tracking only the beginning and the end of its return path.

Finally, when they occur, the mismatches of the tracks are due to the unlabeled nature of the PHD filter. As discussed in the chapter 2, one predicted object creates $M+1$ new objects after the correction, M being the number of extended measurements. Since we rely on the estimate with the highest weight to assign the labels, as proposed by Clark *et al.* in [CPV06], we have experienced some issues with tracks that seem to jump from one place to another if the label is passed to the wrong association. In addition, a deterministic labeling approach would allow the easy implementation of a label recycling algorithm.

In conclusion, the GO-PHD/L filter performed well on this dataset, while the GO-PHD/F filter suffered from the high measurement noise. Our implementation of the GO-PHD/L filter showed a more appropriate group extent size estimate. These results are in part due to the quality of the passive radar sensor, delivering low clutter and a high detection probability for large enough objects, despite its high measurement error noise. We propose two improvements: first, implementing a labeled RFS filter and estimating the average number of measurements

generated by the group object. These improvements are tested on a more complex dataset in the next section.

4.8 Results with an active radar sensor

4.8.1 The pySim simulation tool

pySim is a python based software impulsed by Benjamin Pannetier to simulate MOT[5] scenarios with multiple heterogeneous sensors. It can also handle real datasets and compute multiple metrics. An illustration of pySim is proposed in the figure 4.12. Our contributions to the groups of drones tracking problem were developed in pySim.

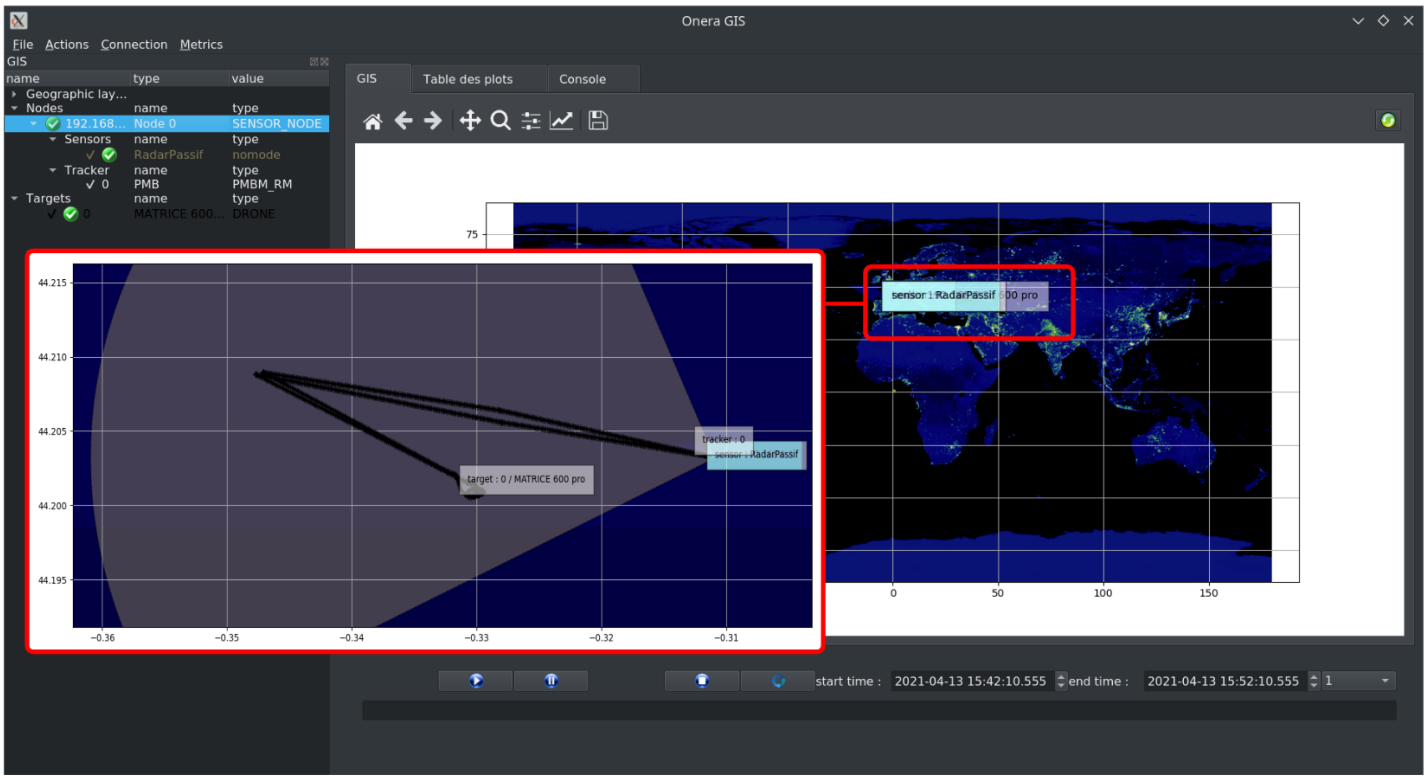


Figure 4.12: pySim software user interface

For the simulation of radar scenarios, pySim works in three steps:

1. the creation of a scenario,
2. the launch of multiple Monte Carlo runs,
3. the computation of the metrics.

[5]Multi Object Tracking

When creating a scenario, the user picks a reference point on a world map and selects a sensor for the experiment. We use a radar sensor for the experiments; hence it is our choice for the simulation. Among the radar sensor parameters, there are the minimum and maximum detection ranges in meters, the elevation and azimuth aperture in degrees, and the performance parameters of the sensor. The radar performances are characterized by four parameters: the detection probability p_D, the clutter rate in false alarms per squared meters, and the measurement standard deviation along with the range and the azimuth.

The sensor is useless without objects traveling in its field of view, pySim allows the creation of multiple isolated objects, with spline trajectories and random speeds if necessary. The groups are created using multiple isolated objects evolving in close formation and at the same speed.

The contribution of this thesis to pySim is the implementation of the metrics, as well as the Monte Carlo procedure for the validation of the filters. In addition, we have implemented the PMB[6], GO-PHD, and GO-PMB filters. The group-oriented RFS filters exist in two forms: Granström's proposal and Our proposal.

4.8.2 Simulation with pySim

The objective of this first simulated dataset is to see how the algorithms react to a nonlinear maneuver with merging, splitting, and an uneven number of objects per group. An added difficulty comes from the fact that the top object is alone at the beginning and the end of the scenario, as shown in the figure 4.13. Indeed, the Random Matrix model assumes that several measurements are available for each extended or group object at each time step, which is not always the case in our application. The group object of this scenario is slightly under-resolved since it is the union of only five objects.

The simulated sensor is an active radar sensor. The measurements are 2D polar points converted to Cartesian coordinates, the detection probability is set to $p_D = 0.9$, the range standard deviation varies between $\sigma_r = \{1, 4\}$ meters, and the angular standard deviation ranges between $\sigma_\theta = \{0.1, 0.6\}$ degrees. The false alarm rate is set to $N^{FA} = 40$ false alarms per radar swipe. The sampling period of the sensor is $\Delta_t = 0.5$ seconds. These parameters are reminded in the table 4.2. Regarding the filter used in this section, their parameters can be found in the appendix B.2.

The graphs from the figure 4.14 are the averaged results over 100 Monte Carlo runs. The results of the PMB filter are only given in the case where $\sigma_r = 1$ meter and $\sigma_\theta = 0.1$ degree, for better readability regarding other results.

The simple Gaussian state probability density of the PMB filter leads to a highly underestimated number of objects, as shown in graph 4.14.b. It should be noted that the PMB filter is tracking the actual cardinality of the problem, estimating six individual objects. However, this filter shows poor track continuity results: it struggles to properly initialize the objects

[6]Poisson Multi-Bernoulli

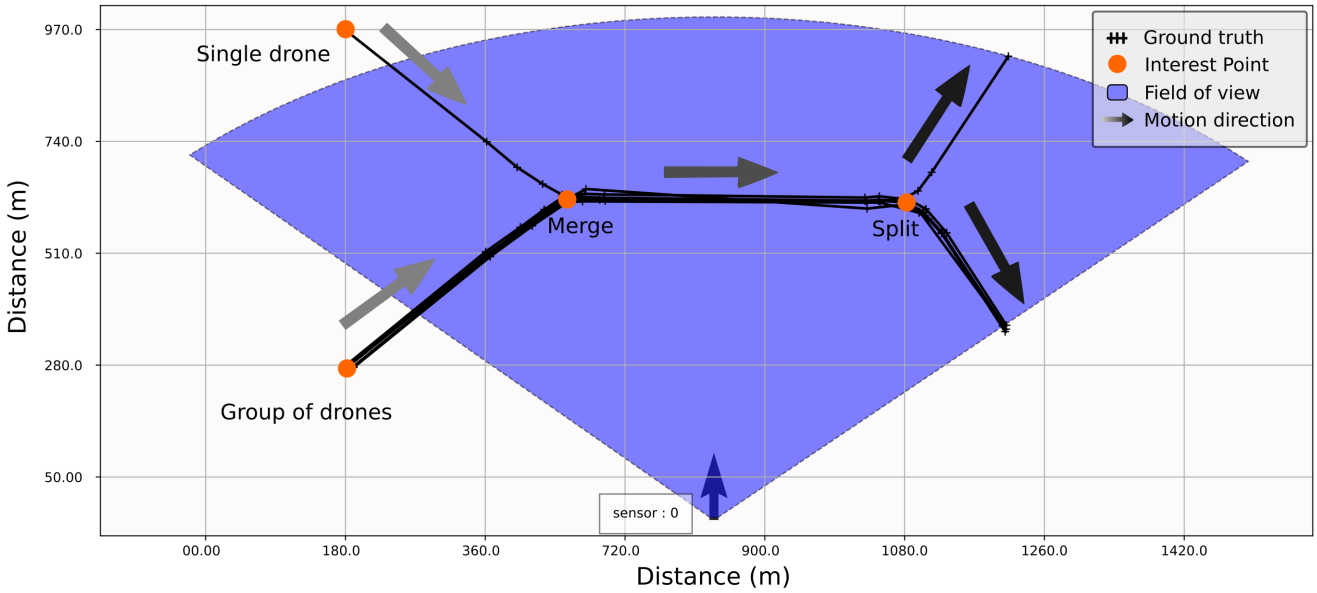


Figure 4.13: Simulated dataset

Parameters	Min. range	Max. Range	Range standard deviation
Values	10 m	500 m	$\{1, 4\}$ m
Parameters	Azimuth beamwidth	Elevation beamwidth	Azim. standard deviation
Values	90 °	30 °	$\{0.1, 0.6\}$ °
Parameters	Clutter rate	Sampling period	p_D
Values	80 FA/s	0.5 s	0.9

Table 4.2: Simulated radar parameters.

composing the group. When it does, it fails to maintain them because of wrong association and phantom movements. It should be noted that the RMSE is lower for the PMB filter due to a stark pruning policy: it prevents the wrong associations with false alarms, but at the cost of track continuity once again.

Regarding the RM models coupled with the PMB filter, Granström RM implementation is identified as the GO-PMB/G, and our RM implementation is denoted GO-PMB/O. The GOSPA metric is shown in the graph 4.14.a clearly shows the impact of the Random Matrix state models compared to the Gaussian state model. With $\sigma_r = 1$ and $\sigma_\theta = 0.1$, the GO-PMB/O and the GO-PMB/G filters seem to perform similarly. However, the higher the noise, the more the GO-PMB/G filter struggles to estimate the number of objects properly, as shown in graphs 4.14.e.

The additive Gaussian noise model of the GO-PMB/O shows better tracking performances with the group object, as seen in the middle of the scenario when only one group is tracked on the graph 4.14.e. This improved tracking performance comes with a higher RMSE with

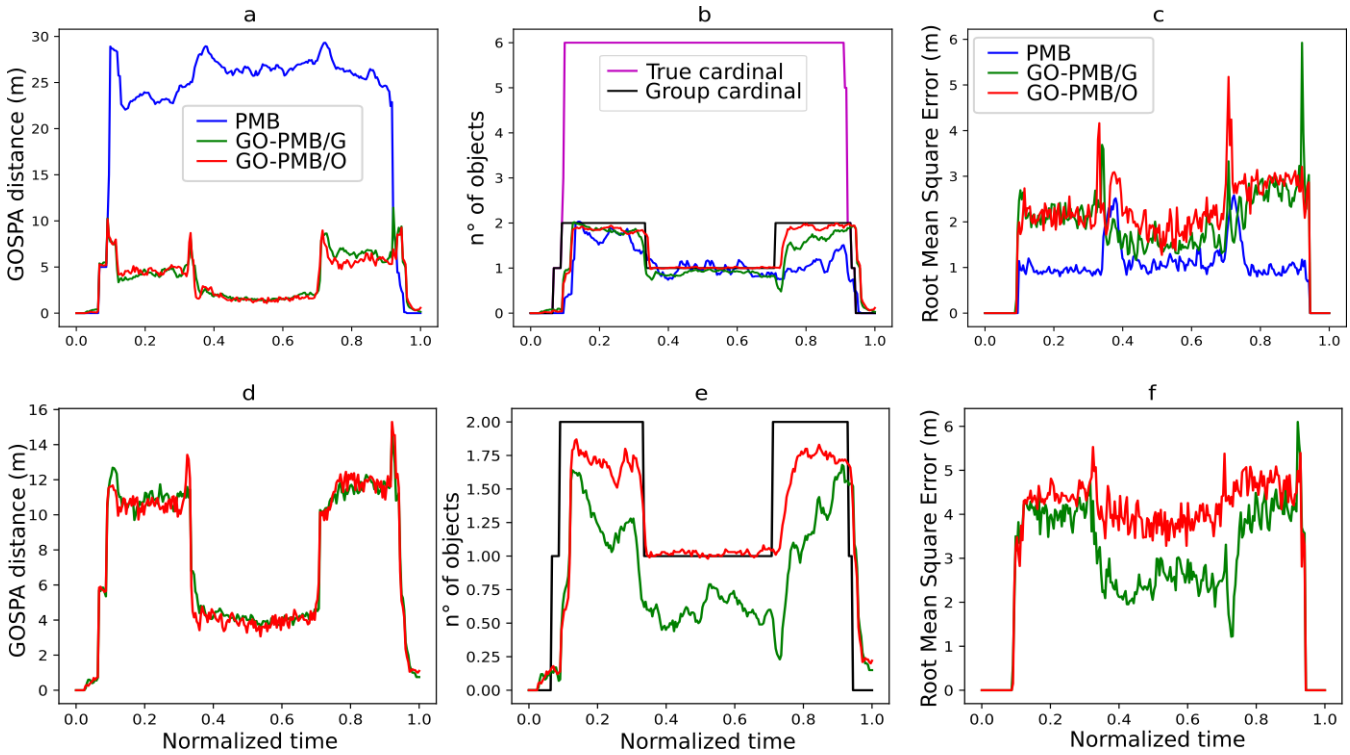


Figure 4.14: Metrics of the simulated dataset. One line corresponds to one scenario: a-c for $\sigma_r = 1$ meter and $\sigma_\theta = 0.1$ degrees, and d-f for $\sigma_r = 4$ meters and $\sigma_\theta = 0.6$ degrees

respect to the GO-PMB/G filter: in fact, the GO-PMB/G filter converges quickly, but because of the high noise when $\sigma_r = 4$ and $\sigma_\theta = 0.6$, it fails to maintain the track continuity. Hence, what happens is a lower RMSE for the GO-PMB/G filter, at the cost of a worst tracking continuity, meaning a higher cardinality error, as seen in graphs 4.14.e and 4.14.f.

In conclusion, both of our implementations of the Group Object Poisson Multi-Bernoulli filter work well with under-resolved isolated and group objects, which is very important when tracking swarms of drones. When a drone separates from a swarm, its extent becomes negligible with respect to the sensor noise. Moreover, compared to the GO-PMB/G filter, the GO-PMB/O filter proved to be more robust to higher sensor noise. However, even with the contributions we proposed, these results are based on a simulated dataset. To complete our results, we propose to run the GO-PMB filter on a real pedestrian dataset.

4.8.3 The pedestrian dataset

We have used an active radar sensor for this dataset, relying on Frequency Modulated Continuous Wave (FMCW). It was developed for aerial space surveillance applications, including the detection and tracking of drones and swarms of drones. It can detect human-sized objects up to 500 meters and vehicles up to 900 meters. Its parameters can be found in the table 4.3.

Parameters	Radar type	Carrier frequency	Power
Values	FMCW	4 - 8 GHz	25 mW
Parameters	Min. range	Max. Range	Range standard deviation
Values	10 m	500 m	1 m
Parameters	Azimuth beamwidth	Elevation beamwidth	Azim. standard deviation
Values	90 °	30 °	0.1 °
Parameters	Clutter rate	Sampling period	p_D
Values	30 FA/s	1 s	0.9

Table 4.3: PSR-500 radar parameters.

The measurements are 2D polar points converted to Cartesian coordinates, and the detection probability is equal to $p_d = 0.9$. Moreover, the range standard deviation is $\sigma_r = 1$ meter and the angular standard deviation is equal to $\sigma_\theta = 0.1$ degrees. The false alarm rate is moderate, with an average of $N^{FA} = 30$ false alarms per radar swipe, and the sampling period of the sensor is $\Delta_t = 1$ second. Finally, it should be noted that the ground truth is empirical.

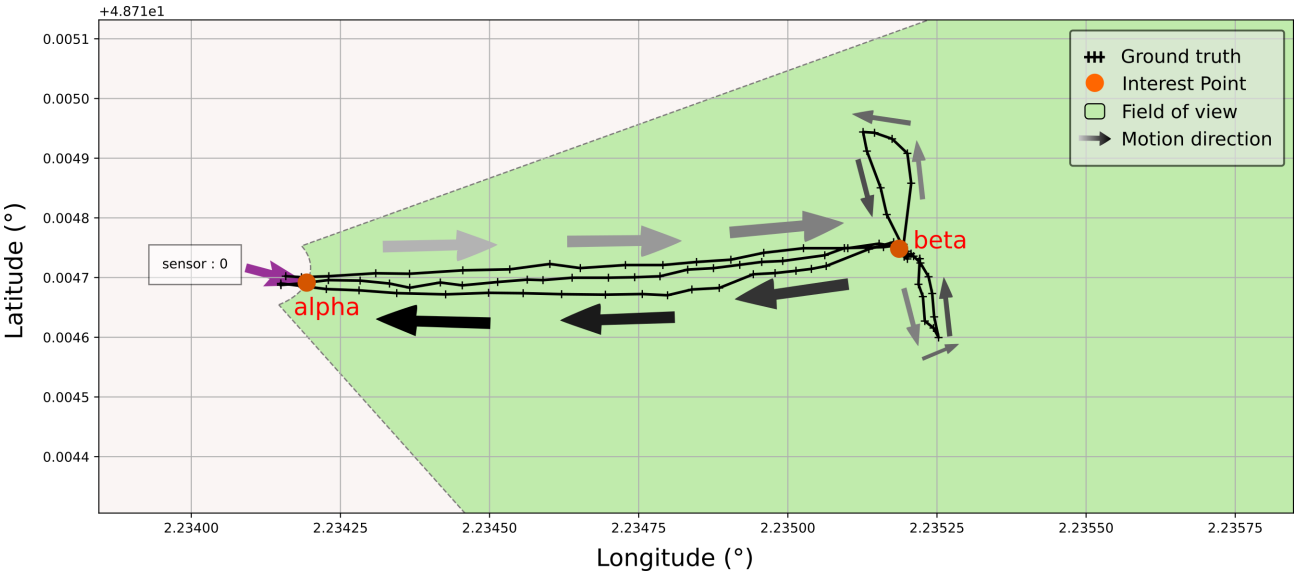


Figure 4.15: Pedestrians dataset

The scenario of this dataset is the following: an initial group of 9 people walks away from the radar sensor, denoted *alpha* in the figure 4.15. Then they separate at a ninety-degree angle, at point *beta*, and go on the side of the road. Shortly after, they walk back to the point *beta*, regroup and pause. In the last part, the pedestrians walk back to the radar sensor, the point *alpha*. During this last part, the group separates once again into two groups. The resulting tracks form a T-shape.

The first difficulty with this scenario is the low number of measurements returned by the

group of nine walking people. Even though they are in front of the radar, they generate around 4 to 5 measurements per swipe. In addition, the detection probability of the tracked groups is low twice during this scenario: when they separate at point *beta* and when they gather and walk back to the point *alpha*. In the first case, they are almost invisible because of the ground topology, both sides of the road forming depressions that occlude the groups. In the second case, one group is occluding the other one.

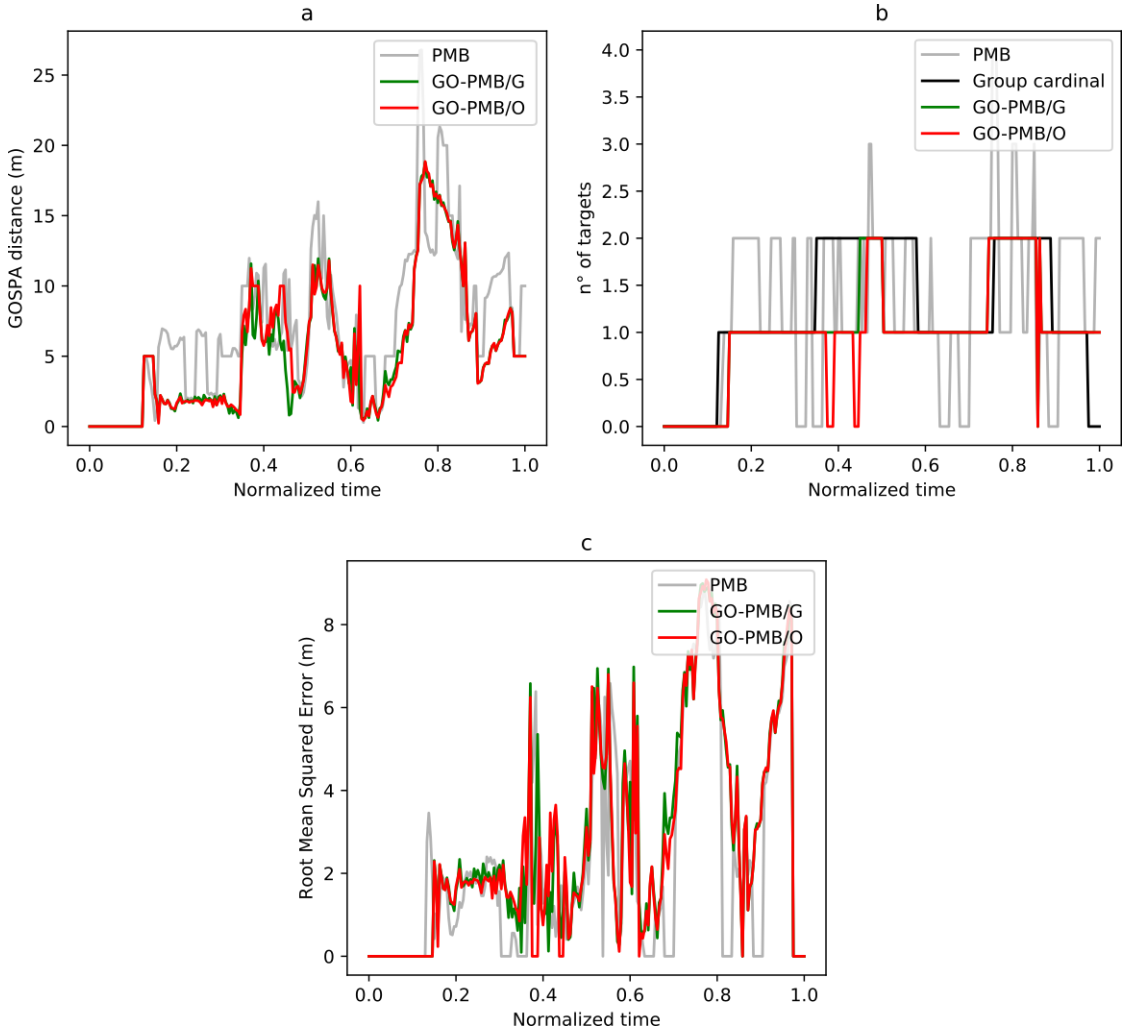


Figure 4.16: Metrics for the real dataset

Both the PMB and the GO-PMB filters were tested on this dataset. For the RM state models, Granström's RM implementation is denoted GO-PMB/G, and our RM implementation is denoted GO-PMB/O. The parameters of all the filters can be found in the appendix B.3.

Once again, the PMB filter struggles with track continuity. Regarding the GO-PMB/G and the GO-PMB/O filters, they are on par with each other. Indeed, this scenario is composed of slowly evolving groups, and the false alarm rate is low. Moreover, the sensor noise error

covariance is also low since the objects are moving close to the sensor. It can even be approximated as negligible in regard to the extent in this specific scenario. Thus the derivation and update procedures of the original Random Matrix model[Koc08] are sufficient here. Hence, the cardinality and errors are very similar on the graphs presented in the figure 4.16.

4.9 Conclusion

We have proposed a solution to the MOT problem for group objects and under-resolved objects when only scarce datasets are available. For that matter, we proposed to rely on the Random Matrix (RM) state model associated with two different RFS-based filters.

The solutions were built incrementally, using the Probability Hypothesis Density filter coupled with Feldmann's and Lan's RM models on a passive radar dataset first. In this dataset, the measurement uncertainty is greater than the size of the extent. It is especially problematic for Feldmann's RM model, whose extension estimate is poor because of the combined effect of the exponential decay prediction step. The extent was better estimated by Lan's RM model. Still, the overall performance of the filter was not the best due to the unlabeled aspect of this filter and the fixed expected number of measurements for each correction step.

We proposed two solutions using an active radar sensor dataset for validation to correct both of these problems. First, we switched to the PMB filter to label tracked objects. Second, we augmented Lan's RM model with a Gamma density to estimate the number of measurements returned by an extended object. The resulting filter, the GO-PMB/O filter, is compared to two other RFS-based filters: the linear Gaussian PMB filter developed in chapter 2 and the PMB filter coupled with Granström's RM state model, the GO-PMB/G filter.

The PMB, GO-PMB/G, and GO-PMB/O filters are compared using two active radar datasets, one from simulated data and one from real data. The GO-PMB/O outperforms the other filters in terms of shape estimation and tracking continuity. In addition, it shows better performance when dealing with under-resolved objects.

To summarize the contributions we proposed in this work. First, we have modified and implemented our RM model based on previous work. In addition, we proposed to add the estimation of the number of measurements to the state model. This model was first used with a PHD, and then a PMB RFS filter, and they were both compared to similar filters in high noise scenarios. Amongst the scenarios, we proposed to use our real datasets and a simulation using a MOT software we developed, pySim. These scenarios confronted us with the under-resolved objects problem that we solved using a virtual measurement approach.

To further push this work, numerous perspectives exist. We could implement a Generalized Labeled Multi-Bernoulli (GLMB) RFS filter and couple it with our RM state model to compare it to the GO-PMB/O labeled filter. In addition, working on the extension estimation stability could be interesting, especially in our high measurement noise scenarios when compared to the size of the extent. Finally, a study of the impact of the deformation matrix and the

evolution matrix on the extent estimation could be conducted. It would require a group of drones dataset with ground truth data of the extent.

Chapter 5

Multi Object Tracking for automotive applications

Contents

5.1 The problems of automotive Multi Object Tracking

5.1.1 Context of the study

An autonomous vehicle first needs to perceive its environment before planning its trajectory and executing its decision. The perception task of an autonomous vehicle concerns the detection, tracking, and forecasting of the surrounding environment state. The perception task has several applications, for instance, lane tracking to position the vehicle on the road or the detection and estimation of other road users. In this work, we focus on the latter. To put it in perspective of the Simultaneous Localization And Mapping (SLAM) computational problem,

this work focuses on the front end, that is, the sensor data processing and the tracking of other road users.

Since all the software blocks of autonomous driving are built upon the perception task, enforcing the redundancy and integrity of information through this procedure is mandatory. The redundancy is fundamental; it asserts that in the eventual case of a system failure, the system integrity can be certified until the vehicle's passengers are safe. Failure causes can be software or hardware, and assessing the perfection of a system is impossible: the redundancy features must be built at the root of the perception of an autonomous driving system.

Heterogeneous and complementary sensors equip an autonomous vehicle. Their multiplicity and diversity enforce the redundancy from the start of the perception task, but it comes at the cost of higher complexity than single-sensor approaches. Indeed, the sensors return asynchronous measurements that may have different representations or coordinate systems. In addition, they can contradict each other, and they are subject to varying types of noises. The main goal of data fusion is to operate the synthesis of all the information returned by the sensors. At the same time, it aims at leveraging the advantages of each type of sensor. Data fusion is a field on its own, and it will not be discussed in this chapter. We will only rely on a lidar sensor for the detection and tracking of other road users.

Several large datasets offering lidar point clouds exist: for instance, the Kitti dataset [GLU12] or the nuScenes dataset [Cae+20]. These large datasets allow testing the performances of the perception system in a wide variety of situations. In addition, they propose a high-quality ground truth, together with metrics and performance indicators that allow the scientific community to compare different perception solutions. The object perception task is often divided into two challenges: the *object detection* and the *Multiple Object Tracking* (MOT) challenges.

Since we focus on the MOT challenge, the detection algorithm is selected amongst the state-of-the-art algorithms for automotive applications. The current best object detection algorithms are based on deep learning, a supervised machine learning approach. Using massive datasets allows automating the learning object detection. Recent research tends to prove that most of the complexity lies in the detection process, and even with a basic MOT algorithm, near state-of-the-art performance can be achieved with a good off-the-shelf detector. For instance, Kim *et al.* achieved state-of-the-art performances with a Deep Learning based object detector and a Global Nearest Neighbors Standard Filter (GNNSF) [KOLT21]. The same conclusion can be drawn from the work of Yin *et al.*, where the MOT algorithm limits itself to a nearest neighbor association between two consecutive sensor swipes [YZK21].

While these prove to be well-performing solutions, integrity is not always treated, whereas it is a crucial one for the automotive industry. Moreover, the comparison does not incorporate the latest works done in MOT, represented by the Random Finite Sets (RFS) theory.

Building on the study [YZK21], the objective of this chapter is to compare the current state of the-art of automotive MOT with an RFS based filter: the Generalized Labeled Multi Bernoulli (GLMB) filter [VVH16]. After a short reminder on the state-of-the-art of automotive

MOT, the object detection solution we have implemented is presented in the second section. The third section discusses the implementation of the GLMB filter. All the algorithms discussed in this chapter are then compared using the automotive dataset nuScenes, and they are displayed in the results section.

5.1.2 Choosing deep learning over modeling for the object detection and extent estimation

Contrary to the previous chapter, where the detection and shape estimation process was handled by the Random Matrix state model, in this chapter, we rely on deep learning for these tasks. Only scarce datasets were available for the drone tracking application, which is not the case in this application. Before these datasets were available, the perception of autonomous vehicles heavily relied on modeling.

The first leap in autonomous driving research happened during the Darpa Grand Challenge from 2005 to 2007 in the United States of America. During this event, the goal was for a vehicle to navigate in different environments autonomously. The teams did not have access to large datasets, and they relied on modeling for the object detection task. For instance, Petrovskaya *et al.* proposed in [PT09] to detect cars using a lidar together with a simple geometric ground rejection algorithm and a segmentation algorithm based on the difference between two successive frames. While effective, this approach is based on a specific heuristic since the navigation environment is fully controlled: it can not be easily generalized to other scenarios because of security requirements.

As soon as the first large datasets appeared, together with the processing power, the shift to deep learning algorithm has operated. Deep learning takes advantage of the "Unreasonable Effectiveness of Data", as stated by Halevy *et al.* in the eponymous paper [HNP09]. The idea is that data matters more than algorithms for complex problems. It is even more valid for deep learning, where the larger the dataset, the more a deep neural network can learn generic and very specific features regarding the problem at hand. In that regard, deep learning approaches seem less prone to "the curse of dimensionality" encountered with more classical techniques.

Despite all these advantages, there are some drawbacks to using deep learning. The necessity to gather large annotated datasets is not the most problematic. The main inconvenience comes from the explainability of such algorithms. Since it relies on an automatized learning step, no one can explain why a network converges to one solution or another. In that regard, a deep learning algorithm behaves like a black box algorithm: the inputs and the outputs are known, but anything in between is out of reach. Most of the works in fundamental research aim at solving this issue, such as the work of Ribeiro *et al.* [RSG16].

In the case of our application, this black box approach resembles what a car manufacturer faces when outsourcing its research and development effort to suppliers. Indeed, when buying off-the-shelf solutions, the manufacturer does not necessarily know the algorithm involved. That's why this research is interesting: suppliers' algorithms are subject to errors, just like sensors, so they require filtering to be usable. A deep learning network can be seen as a

supplier solution that needs to be filtered, using, in our case, an RFS-based MOT tracker.

To summarize, the first object detection approaches used modeling heuristics because only scarce datasets were available, like for the drone application. Then large datasets have been collected and annotated, enabling the use of deep learning. Large datasets allow the algorithm to learn the extended object representation from the data directly. It is more effective than using a model such as the Random Matrix extended object model. However, it comes at the cost of computing and explainability, trading a clear modeling for a black-box approach. Nonetheless, black-box approaches solutions are common in the automobile industry. From the MOT algorithm perspective, it can be seen as a "meta-sensor", subject to measurement noise, false alarms, and missed detections.

This "meta-sensor" description is compatible with RFS-based filters. Mahler described RFS and FISST[1] as a "fire and forget" method, where the fusion or detection step can be handled in a meta-sensor that gathers all the sensors' information [Mah04]. This chapter uses a lidar-only approach coupled with a deep learning detection algorithm to simulate this "meta-sensor".

A lidar sensor returns point clouds, with each data point consisting of a 3-dimensional position and a reflectance. An example of a lidar point cloud can be seen in the figure 5.1. From this lidar point cloud, the goal is to estimate the bounding boxes that correspond to the objects surrounding the autonomous vehicle. These bounding boxes are the ground truth of the dataset: they are characterized by a position, a speed, a length, a width, a height, an orientation, a class, and an identifier. Examples of bounding boxes can be found in the figure 5.1. They can be considered as extended objects since they return more than one measurement.

Estimating a bounding box is equivalent to estimating its extent, position, and kinematic parameters. As discussed in the previous chapter, this task is achieved using Yin *et al.* CenterPoint, a deep learning algorithm [YZK21]. Other deep learning object detection algorithms exist, but this chapter is not state-of-the-art about deep learning. We chose to use it as a black box. CenterPoint is amongst the best performers on Waymo and nuScenes automotive datasets: it was the overall best choice when we began our research, even though it has since been outperformed in terms of precision, shape, and class estimation.

The workflow of CenterPoint is summarized in the figure 5.1, it starts with a birds-eye view projection of the lidar point cloud, turning it into an image. Hence, the point cloud is discretized, and each pixel has three components: the maximum height reached inside it, the median reflectance, and the number of data points that fall in this pixel. This birds-eye view image is then passed to a first Convolutional Neural Network that detects the center of the objects and regress to their extent, class, and orientation. We are treating it as a "black box" algorithm.

The result is a 3-dimensional bounding box, characterized by its extent, position, orientation, and class, as illustrated in the result example of the figure 5.1. As pictured in this

[1] FInite Set STatistics

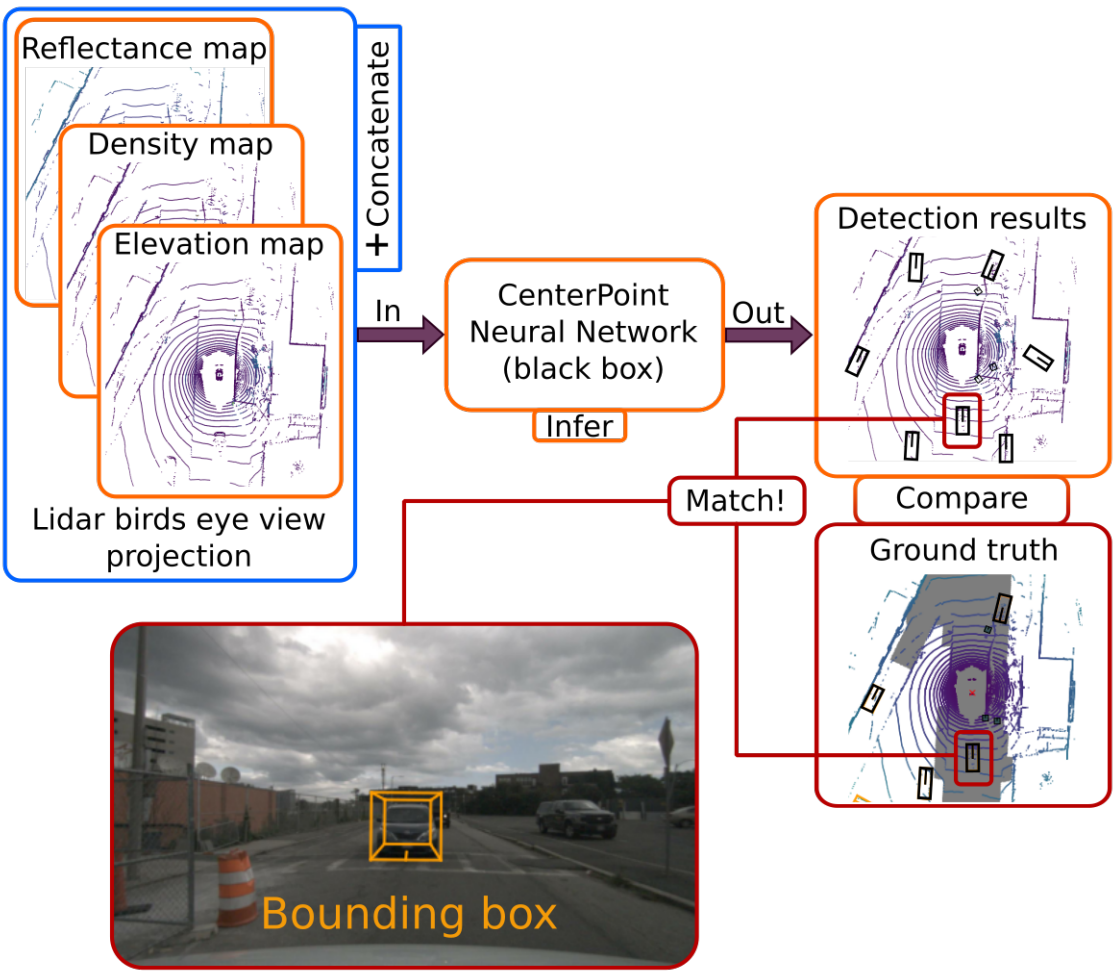


Figure 5.1: Lidar only detection algorithm workflow.

example, there are a lot of false alarms, which is generally true for most object detection algorithms for automotive applications. The false alarms are not the only problem, and three random phenomena mainly impact a typical detector:

- false alarms, or false positives, corresponding to detections that are not associated with an object,
- missed detections, or false negatives, whether it is due to the occlusion of the object or the detection process,
- noisy detections inducing uncertainty directly from the measurements.

These problems are critical for an autonomous vehicle. Thus, a perception layer cannot limit itself to the detection paradigm. In addition, the output of the object detection algorithm lacks an identifier for each object. The role of a tracking algorithm is to tackle all these

problems. The following section discusses MOT algorithms to filter out the false alarms and track each extended object through time.

5.1.3 State of the art Multi-Object Tracking algorithms for autonomous driving

In this chapter, tracked objects are referred as a set $X = \{(\mathbf{x}^1, l^1), ..., (\mathbf{x}^n, l^n)\}$, where each $\mathbf{x}^i = [p_x, v_x, p_y, v_y, p_z, v_z, \theta, w_b, l_b, h_b, C, s]^T$ is a state vector describing respectively the position (p_x, p_y, p_z) and the speed (v_x, v_y, v_z) of the center of the object, its orientation (θ), its bounding box dimensions (w_b, l_b, h_b), its class (C), and its score (s). Each state vector $\mathbf{x}^i$ is associated to a tracking identity, or label l_i. The set $X = \{(\mathbf{x}^1, l^1), ..., (\mathbf{x}^n, l^n)\}$ is estimated recursively with detections $Z = \{\mathbf{z}_1, ..., \mathbf{z}_m\}$, where $\mathbf{z}_i = [p_x, v_x, p_y, v_y, p_z, v_z, \theta, w_b, l_b, h_b, C, s]^T$ in the case of CenterPoint, using the same notations as above: hence the tracking algorithms could be reduced to the addition of a tracking identity to the measurement. It should be noted that the measurement set and the tracked objects set cardinalities are generally different, $m \neq n$. To link these two sets, the state of the art consists of two classes of algorithms:

- *detect to track*: the tracking occurs after a first detection step.
- *simultaneous detection and tracking*: tracking and detections are joints. Tracking results are used to detect new objects, and detections can create new tracks.

We are interested in the *detect to track* architecture presented in [FPZ17]. It consists of creating an association matrix between measurements and tracks that is solved using an assignment algorithm. Afterward, the state of each track for the next time step is predicted using an evolution model. Then some track management is done for unassigned measurements and undetected objects. This procedure is repeated recursively throughout the tracking. Depending on the algorithmic choices for these building blocks, we have selected three algorithms from the automotive MOT state of the art: AB3DMOT, CenterPoint, and EagerMOT [Wen+20]; [YZK21]; [KOLT21]. These three propositions propose to use a deep learning detection algorithm coupled to a simple MOT algorithm for the tracking.

AB3DMOT is the baseline tracker of nuScenes. Using only lidar point clouds, it relies on a Munkres algorithm for the assignment, coupled with a Kalman filter for the state estimation: it is a derivative of the Global Nearest Neighbours Standard Filter (GNNSF) algorithm presented in the introduction.

EagerMOT also uses a GNNSF MOT algorithm, but its deep learning detector relies on two sensors modalities: the cameras and the lidar. Thanks to a specific 2D to 3D data information fusion strategy, it outperforms the other methods tested in this chapter. Even though it relies on more sensors, we have decided to include it nonetheless to understand what is undermining the performances of lidar-only solutions.

Compared to these algorithms, the tracker from CenterPoint is very straightforward: after receiving the detection from Centerpoint, the assignment between objects and detections is

done with a greedy algorithm. After this greedy association, the prediction is carried out using a constant velocity model, without considering the measurement uncertainty: a Kalman filter is not even involved for the state estimation.

The main differences between these MOT algorithms are the heuristics for tracked objects' death and birth and the way to build the association matrix: the objects' track management is at the center of MOT challenges. For instance, CenterPoint initializes any unmatched measurement as a new tracked object after the assignment, while other algorithms might require a delay.

From a more formal perspective, solutions such as EagerMOT, AB3DMOT, and CenterPoint are badly conditioned, even if they are effective. Especially, the formulation of the association map, resulting from the resolution of the association matrix between measurements and tracked objects, is both considered as a state variable to estimate and an observable parameter. This assumption is not correct, as discussed in the chapters 1 and 2.

RFS-based filters offer an accurate description of the multi-object system. We want to leverage this descriptive power to model the MOT problem for the automotive task.

5.2 Automotive tracking with a GLMB filter

We have selected the Generalized Labeled Multi-Bernoulli filter introduced in the chapter 2 for the automotive MOT task. It is the only RFS filter relying only on Multi Bernoulli densities, and Vo *et al.* have proven that it can run efficiently using Gibbs Sampling and a gating algorithm in [VVH16] and [VVB19]. In addition, the GLMB filter prediction and measurement models take into account the issues of the detection algorithm presented in the previous section. The pseudo-code of this algorithm can be found in the section 2.6.

To compare the GLMB filter to the MOT algorithm of CenterPoint [YZK21], a similar evolution motion model is selected: the constant acceleration model. The state vector to estimate, $\mathbf{y}_k = [p_x, v_x, a_x, p_y, v_y, a_y]^T$, consists of the position, the speed, and the acceleration along 2 dimensions, and is a partial representation of the state vector $\mathbf{x}_k$ introduced in 5.1.3. We do not have access to $\mathbf{y}_k$ directly, but we do have access to its prior probability density that is assumed to be linear Gaussian, with the expectation $\hat{\mathbf{y}}_{k|k-1}$ and the error covariance matrix $\mathbf{P}_{k|k-1}$. At each time step, the set $(\hat{\mathbf{y}}_{k|k}, \mathbf{P}_{k|k})$ need to be estimated.

In order to bridge the gap between the partial state vector $\mathbf{y}_k$ and the desired state vector $\mathbf{x}_k$, $\mathbf{y}_k$ is enriched with the class (C), the score (s), the orientation (θ), the bounding box dimensions (w_b, l_b, h_b), and the position and speed along the vertical axis (p_z, v_z), directly taken from the measurement $\mathbf{z}_k$ defined in 5.1.3. In other words, $\mathbf{x}_k = [(\mathbf{H}_k\mathbf{y}_k)^T, p_z, v_z, \theta, w_b, l_b, h_b, C, s]^T$, with $\mathbf{H}_k$ the observation matrix defined bellow.

The idea behind this truncation is that apart from the position $[p_x, v_x, p_y, v_y]$ of the tracked object $\mathbf{x}_k$, all the other state variables keep approximately the same value during the inference. Hence, only $\mathbf{y}_k = [p_x, v_x, a_x, p_y, v_y, a_y]^T$ is estimated with a Bayesian filter.

5.2.1 State Evolution Model Assumptions

The constant acceleration evolution model assumes that:

- the prediction of the state is linear; thus it is similar to a classical Kalman filter prediction,
- the evolution of each dimension is independent,
- the object maintains its acceleration between two sensor acquisitions, its evolution is given by a matrix $\mathbf{F}_k$,
- the evolution process is noisy, each prediction gives rise to the augmentation of the error covariance matrix by a term $\mathbf{Q}_k$.

This leads to an evolution matrix $\mathbf{F}_k$ given by

$$\mathbf{F}_k = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & d_t & \frac{1}{2}d_t^2 \\ 0 & 1 & d_t \\ 0 & 0 & 1 \end{pmatrix} \tag{5.1}$$

where d_t denotes the measurement interval in seconds, $\mathbf{I}_2$ is a two dimensions identity matrix and $\otimes$ stands for the Kronecker product. As for the error covariance matrix $\mathbf{Q}_k$

$$\mathbf{Q}_k = \sigma_{pred}^2 \, \mathbf{I}_2 \otimes \begin{pmatrix} \frac{d_t^4}{4} & \frac{d_t^3}{2} & \frac{d_t^2}{2} \\ \frac{d_t^3}{2} & d_t^2 & d_t \\ \frac{d_t^2}{2} & d_t & 1 \end{pmatrix} \tag{5.2}$$

with σ_{pred} standing for the evolution model standard deviation. The resulting state prediction equations are given by

$$\begin{aligned} \hat{\mathbf{y}}_{k+1|k} &= \mathbf{F}_k \hat{\mathbf{y}}_{k|k} \\ \mathbf{P}_{k+1|k} &= \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^T + \mathbf{Q}_k \end{aligned} \tag{5.3}$$

5.2.2 State Measurement Model Assumptions

The measurement model assumptions for detected objects are the following:

- the measurement model is linear Gaussian, the correction step is similar to the regular Kalman filter,
- a measurement $\mathbf{z}_k^t = [p_x, v_x, p_y, v_y]$ comprises the position and speed along each dimension. The associated observation matrix is denoted $\mathbf{H}_k$,
- the measurement noise is independent along each dimension, and for each measured parameter. It is represented by a diagonal covariance matrix $\mathbf{R}_k$.

The measurement $\mathbf{z}_k^{trun}$ is a truncated version of the full measurement z introduced in 5.1.3: $\mathbf{z}_k = [(\mathbf{z}_k^{trun})^T, p_z, v_z, \theta, w_b, l_b, h_b, C, s]^T$. y relies only on $\mathbf{z}^{trun}$ for the estimation process. As stated earlier, all the additional information constitutive of $\mathbf{x}_k$ are directly stripped from $\mathbf{z}_k$ such as $\mathbf{x}_k = [(\mathbf{H}_k\mathbf{y}_k)^T, p_z, v_z, \theta, w_b, l_b, h_b, C, s]^T$.

The observation matrix is given by

$$\mathbf{H}_k = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{5.4}$$

and the measurement error covariance matrix

$$\mathbf{R}_k = \mathbf{I}_2 \otimes \begin{pmatrix} \sigma_{pos}^2 & 0 \\ 0 & \sigma_{speed}^2 \end{pmatrix} \tag{5.5}$$

where σ_{pos}, and σ_{speed} respectively stand for the position, and velocity standard deviations. The final state correction equations are similar to the standard Kalman filter:

$$\begin{aligned} \hat{\mathbf{y}}_{k|k-} &= \hat{\mathbf{y}}_{k|k-1} + \mathbf{K}(\mathbf{z}^{trun} - \mathbf{H}_k\hat{\mathbf{y}}_{k|k-1}) \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}\mathbf{H}_k\mathbf{P}_{k|k-1} \\ K &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \end{aligned} \tag{5.6}$$

5.2.3 Association Likelihood

In equations (2.25) and (2.26), the association likelihood was introduced. Here it is augmented with a class specific identification function $\delta_{C^j}(C^i)$, using the class of $\mathbf{z}^j$ and $\mathbf{x}_k^i$. The notation for the association likelihood is $g_{(\mathbf{z}_k^{trun,j}, C^j)}(\hat{\mathbf{y}}_{k|k-1}^i, C^i)$, and it is set to

$$\begin{aligned} g_{(\mathbf{z}_k^{trun,j}, C^j)}(\hat{\mathbf{y}}_{k|k-1}^i, C^i) &= \frac{\delta_{C^j}(C^i)\mathcal{N}(\mathbf{z}_k^{trun,j} - \mathbf{H}_k\hat{\mathbf{y}}_{k|k-1}^i; 0, \mathbf{S})}{\rho_{FA}} \\ S &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \\ \delta_{C^j}(C^i) &= \begin{cases} 1 & \text{if } C^i = C^j \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5.7}$$

where ρ_{FA} is the false alarms intensity of the measurement model, and $\mathcal{N}(\mathbf{z}_k^{trun,j} - \mathbf{H}_k\hat{\mathbf{y}}_{k|k-1}^i; 0, \mathbf{S})$ is the marginal probability density of the measurement $\mathbf{z}_k^{trun,j}$. $\delta_{C^j}(C^i)$ is equal to 1 when the class C^j of the detection z_v is the same as the class C^i of the tracked object $\mathbf{x}_k^i$.

5.2.4 Objects' Initialization And Destruction

The object initialization procedure is straightforward: each measurement $\mathbf{z}_k^j$ with a detection score higher than a class-specific threshold T_{new} is considered as a possible new object whose

acceleration is set to 0 m/s^2. Then, during the resolution of the association matrix, the choice is made between a measurement generated by a false alarm, a new object, and an already tracked object.

Once a new object is initialized, for instance the object $(\mathbf{x}^{b,i}, l^{b,i})$, its partial state vector $\mathbf{y}_k^i$ is created, and its survival probability $P_S(\mathbf{y}_k^i, l^i)$ increases according to the number of times the object is detected. It decreases when it is missed, and when an object is missed three times, it is destroyed.

This concludes the presentation of our implementation choices, the results of the joint prediction, and update GLMB filter are provided in the next section.

5.3 Results

5.3.1 Overview of nuScenes dataset

We rely on the nuScenes dataset that offers a *Multi-Object Tracking* (MOT) challenge for the result section [Cae+20].

The ego vehicle from nuScenes relies on the typical sensors found in such datasets: 5 radars, one lidar, six cameras, a GPS, and an inertial measurement unit. Each sensor modality covers roughly all the surroundings of the ego vehicle. This dataset consists of 850 labeled scenes and 150 test scenes of 20 seconds. They were collected in Singapore and Boston. Several tasks are proposed with this dataset, including a MOT challenge.

The goal of nuScenes' MOT challenge is to track and label objects surrounding the ego vehicle, whether the sensors detect them or not, and despite the presence of false alarms and noise in the measurements. Because of the heterogeneity of the sensors and due to the complexity of the observed scene, we rely on neural networks for the detection step.

The detection algorithm we have selected is one of the state of the art solution found in the *object detection* challenge of nuScenes. We have retained the CenterPoint approach [YZK21]. In spite of its state-of-the-art performances, it suffers from certain issues regarding false alarms rate and bike detections, which will be discussed in the result section.

One of the advantages of nuScenes is its clean ground truth. The dataset is annotated at a rate of 500 milliseconds, and each frame is composed of multiple 3D bounding boxes. Comparing the ground truth to the results of MOT algorithms permits the computation of many metrics. For the MOT challenge [Cae+20], the two main metrics are the Average Multi-Object Tracking Accuracy (AMOTA) and the Average Multi-Object Tracking Precision (AMOTP): they are presented below.

5.3.2 Metrics and Tuning Parameters

The AMOTA is an integral metric of the Multi Object Tracking Accuracy with Recall normalization (MOTAR) over a number L of recall values:

$$\text{AMOTA} = \frac{1}{L-1} \sum_{r \in \{\frac{1}{L-1}, \frac{2}{L-1}, 1\}} \text{MOTAR} \tag{5.8}$$

For a given recall r, the MOTAR metric takes into account the number of Identity Switches (IDS_r), the number of False Positives (FP_r), and the number of False Negatives (FN_r), and compare them to the number of Ground Truth True Positives (GT_{tp}). It relies on a recall normalization term $(1-r)\ GT_{tp}$ to keep it in the $[0,1]$ interval, and to avoid the fading contributions of IDS_r and FP_r at low recall rates; there are at least 90% FN_r at 10% recall.

$$\text{MOTAR} = \max\left(0\ ,\ 1 - \frac{IDS_r + FP_r + FN_r - (1-r)\ GT_{tp}}{r\ GT_{tp}}\right) \tag{5.9}$$

The closer to 1 the MOTAR and AMOTA metrics are, the better.

The AMOTP metric is complementary to the AMOTA metric. It is also an integral metric, but this time it is interested in the Multi Object Tracking Precision (MOTP) for a number L of recall values:

$$\text{AMOTP} = \frac{1}{L-1} \sum_{r \in \{\frac{1}{L-1}, \frac{2}{L-1}, 1\}} \text{MOTP} \tag{5.10}$$

For a given recall r, the MOTP metric computes the sum of the error in localization for the position of each tracked object i during all the time of the tracking ($\sum_{i,t} d^i_{r,t}$), and compares it to the total number of True Positives selected by the tracking algorithm ($\sum_t TP_{r,t}$):

$$\text{MOTP} = \frac{\sum_{i,t} d^i_{r,t}}{\sum_t TP_{r,t}} \tag{5.11}$$

The result is a metric independent of the tracking accuracy and only retains information about precision: this metric is given in meters (m). The lower the MOTP and AMOTP metrics are, the better.

In the nuScenes dataset, the AMOTA and AMOTP metrics are computed across a number $L = 40$ recall values. Secondary metrics are also calculated,

- the number of Mostly Tracked (MT) objects that are objects tracked during at least 80% of their lifespan,
- the number of Mostly Lost (ML) objects that are objects tracked during at most 20% of their lifespan,
- the number of times the trajectory of a tracked object is Fragmented over time (Frag),

Table 5.1: Main hyperparameters for the GLMB filter

	Car	Truck	Bus	Trailer	Pedestrian	Motorcycle	Bicycle
Existence prob.	0.74	0.74	0.74	0.49	0.74	0.74	0.62
σ_{pos} (m)	1.3	1.4	1.1	1.2	0.6	1.7	1
σ_{speed} (m/s)	1.66	1.33	2.29	1.25	1.18	2.20	1.25
σ_{accel} (m/s^2)	1.125	1.75	1.75	1.75	2.375	1.75	1.75
σ_{pred}	0.5	0.5	0.5	3	0.5	1	0.5

- the Longest Gap Duration (LGD) that gives the average time a tracked object is missed in seconds,
- the average time before the initialization of a new object in seconds (TID): if an object is never initialized, it is accounted for in the TID metric,
- the average number of False Alarms per Frame (FAF).

These metrics are computed on the nuScenes test dataset. The object tracking is done over seven classes: cars, bicycles, pedestrians, buses, trailers, motorcycles, and trucks. The tracking range is limited to 40 meters for bicycles, pedestrians, and motorcycles. It extends to 50 meters for the rest of the classes: these are the limitations imposed by the *tracking challenge*.

The tuning of the GLMB is specific for each class for most of the hyperparameters. The most important ones are reported in the table 5.1. They were tuned using randomized search among a list of possible values with the help of the nuScenes training dataset.

5.3.3 Performance Results

The results are presented in the table 5.2. Looking at the main metrics, the tracking accuracy (AMOTA), and the tracking precision (AMOTP), the GLMB filter outperforms the baseline represented by AB3DMOT and is on par with CenterPoint. Of course, overall, EagerMOT shows better metrics since it relies on two sensors modalities.

Taking a look at the AMOTA metric in the table 5.2, EagerMOT has a slight edge compared to GLMB and CenterPoint MOT algorithms. The GLMB filter and CenterPoint show very close results for the tracking accuracy, and they perform far better than the baseline, AB3MOT.

Compared to CenterPoint and EagerMOT, the GLMB filter is lagging regarding the AMOTP metric. Both CenterPoint and EagerMOT do not rely on a state prediction model, while the GLMB filter average between the object's estimated position and the measurement. This led to a lower tracking precision for the GLMB filter. It is probably due to the overestimation of the measurement errors for the position. A solution to this issue could be to

Table 5.2: Performance results

	AMOTA	AMOTP (m)	MOTAR	MOTP (m)	MT	ML	Frag	LGD (s)	TID (s)	FAF
AB3DMOT	0.151	1.501	0.552	0.402	1006	4428	2557	3.742	1.972	55.83
CenterPoint-Single	0.638	0.555	**0.794**	**0.284**	**5584**	1681	**529**	**0.698**	**0.372**	57.42
EagerMOT	**0.677**	**0.550**	0.793	0.335	5303	1842	601	0.801	0.448	56.85
GLMB (ours)	0.635	0.591	0.789	0.325	5451	1622	690	0.910	0.560	**53.80**

fine-tune the GLMB filter. The precision could also be improved with a better evolution model.

As for the secondary metrics, these observations remain valid, except for the average number of False Alarms per Frame (FAF): the GLMB filter improves the performances and is followed by AB3DMOT. This implies that false positives have a smaller impact on the AMOTA metric when compared to the false negatives and the identity switches, despite the regularization term discussed in the last section. While missing an object can have dramatic effects, the validation of too many non-existing objects can also be very dangerous for an autonomous vehicle, leading to undesired emergency breaking. It is remarkable that even the AB3DMOT tracker, which is lagging behind in every aspect when compared to CenterPoint and EagerMOT solutions, remains one of the best performers for this metric on the *tracking challenge* leader board.

A class-by-class comparison of the AMOTA metric is proposed in the next section, and a plot of the resulting output to understand the situation better.

5.3.4 Further Discussions

Table 5.3: AMOTA per class

	AB3DMOT (lidar)	EagerMOT (cam+lidar)	CenterPoint (lidar)	GLMB (lidar, ours)
Bicycle	0.00	**0.58**	0.32	0.32
Bus	0.41	**0.74**	0.71	0.72
Car	0.28	0.81	**0.83**	**0.83**
Motorcycle	0.08	**0.63**	0.59	0.57
Pedestrian	0.14	0.74	**0.77**	0.75
Trailer	0.14	0.64	**0.65**	**0.65**
Truck	0.01	0.60	0.60	**0.61**

The average results show that our proposal is relevant for autonomous driving applications, but the class-specific results are informative.

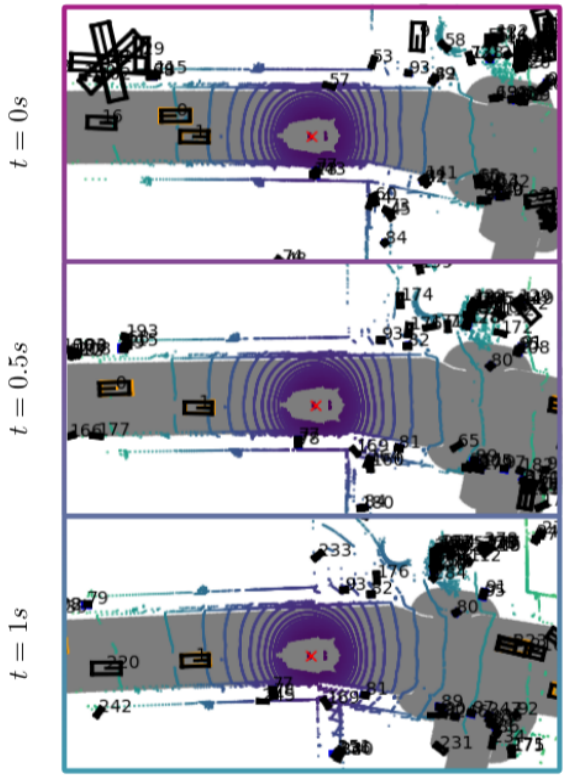


Figure 5.2: The output of the GLMB filter for the first scene of the *mini-test* dataset of nuScenes. The black bounding boxes represent the tracked objects and their labels. The colored bounding boxes represent the ground truth. The lidar point cloud is visible, as well as the road, in grey.

First, the GLMB filter and CenterPoint lag behind EagerMOT for one class, the bicycle class. Indeed, the bicycle class is amongst the most difficult for lidar-only detectors. A bicycle returns only a few lidar points, and it is not the most represented class; hence it is hard to train a detector for it. In comparison, EagerMOT's detector also relies on camera images, where a bike is represented by thousands of pixels. Hence, adding a sensor modality brings some improvements, but only with classes that are difficult to track with lidar-only methods. This shows the main strength of the fusion approach compared to single sensors ones: exploiting the benefits of each sensor while overcoming their drawbacks.

The results are very close to one another regarding the other classes: the objects are more prominent and easier to detect with a lidar. From this observation, a second conclusion can be drawn: the detector's output strongly influences the MOT algorithm. In fact, for this chapter, we tuned the GLMB filter to perform close to the CenterPoint solution, proving that RFS filters can achieve state-of-the-art results. Hence the output of the GLMB filter is very close to the detector's output.

This statement is confirmed in the figure 5.2. This plot is representative of most of the

GLMB outputs: the tracking of existing objects is effective, but false alarms are everywhere. This leads to unusable results for real applications. It should be noted that we succeeded in lowering the number of false alarms, at the detriment of the AMOTA metric.

The problem of false alarms is not limited to the GLMB filter: all of the MOT solutions of the nuScenes *tracking challenge* return an average of over 40 False Alarms per Frame (FAF). As discussed above, false alarms can have a dramatic effect on driving safety. Thus, this problem should be tackled in future work, even if the FAF metric is considered as a secondary metric.

5.4 Conclusion

We have proposed an implementation of the GLMB filter for automotive MOT using a deep learning detection algorithm. The GLMB filter achieves state-of-the-art results and catches up with the original CenterPoint MOT filter in terms of metrics. It proves to perform a little better on the False Alarms metric than the other solutions we are referring to in this chapter.

However, just looking at the metrics is not very informative about the tracking quality, and even if the performances of the GLMB filter we designed are similar to the baseline, plotting the MOT results shows a high number of undesired false alarms. They have a minor effect on the tracking accuracy metric. Still, their impact on driving safety is real: future work should be done to improve the consideration of false alarms in MOT challenges.

The performances of the GLMB are deceptive compared to the other algorithms: it is just on par with them, despite the complexity overhead. Even if our solution is more robust by design, which is a desirable characteristic for autonomous vehicles, it is not apparent in the results. Here, more robust by design means that it can retain multiple association hypotheses and evaluate a comprehensive likelihood for each object. Nonetheless, the results highlight a high dependence on the detection algorithm.

To improve the results, increase the AMOTP metric, and lower the FAF metric, we propose the following research perspectives:

- changing the motion model for a more accurate one. Because the constant acceleration model does not fit all situations, switching to a Multiple Model Approach might improve the results.
- for some vehicle class, most samples are in the *parked* state: a specific model could improve the tracking results, especially for the bicycle class. Indeed, parked vehicles on the sidewalks are often occluded by parked vehicles on the roadsides: wrong associations or undesired object destruction might happen.
- a detector whose output contains information about the survival likelihood, the detection likelihood, and the measurement error.

- an object rejection algorithm based on the High Definition map (HD map).
- to gain both in robustness and accuracy, filtering over multiple frames could be interesting.
- most of the algorithms only rely on keyframes for the MOT challenge locking the filter rate to half a second: exploiting all lidar frames lead to a rate of 50 milliseconds, which should give better results.
- of course, sensor data fusion should lead to a leap in performance.

Another exciting perspective could be to introduce deep learning to RFS-based filters. In particular, to simplify the birth process of new objects: it could be learned from the dataset, as well as the detection and survival probabilities of a given object, the false alarm rate, and all other parameters involved in the MOT filter.

Conclusion and perspectives

Summary of the work

This work is a collaboration between the ONERA, the Gipsa-lab and the Renault Group. It addresses two Multi-Object Tracking applications. For the Renault Group, it is the autonomous vehicle perception task, while for the ONERA, it is the groups of drones tracking problem. Both problems are different versions of the multiple extended object tracking problem. We proposed to solve them using Bayesian modeling tools to infer the information returned by the sensors.

To tackle the MOT problem, we have used Random Finite Sets-based filters. We have seen that adopting the RFS representation allows us to model the issues encountered in MOT using probability densities instead of heuristics. The formalism induced by this description led us to the presentation of three MOT filters we have implemented in this work.

In addition to MOT, both applications involved extended objects. Among all the approaches available for extended object shape estimation, we have retained two methods. For the group of drones application, we have focused on Bayesian modeling using the Random Matrix model. In contrast, we have relied on deep learning for the extended objects detection task of the automotive application. Indeed, when large datasets are available, as is the case for the automotive application, the task of detecting and estimating extended objects can be learned directly from the data.

In the case of the groups of drones tracking application, we have studied two RFS filters combined with four Random Matrix (RM) models. Among the RM model recursions, we have proposed our model, integrating the measurement noise and the random number of measurements to the extent model. In addition, we have proposed a solution to consider under-resolved objects. We have developed pySim, a python software that can playback radar datasets to test the algorithms we have implemented. Using simulated and real radar datasets, we were able to validate the effectiveness of our contributions.

As for the automotive application, we have implemented an RFS tracker to filter the output of a deep learning detection algorithm. We have achieved state-of-the-art results and have caught up with some well-known MOT algorithms metrics-wise. Our approach to this problem proved to perform better on the False Alarm metric while not solving this issue: future works should address this problem.

Perspectives

The first perspective is the integration of deep learning algorithms to improve the prior knowledge available about the studied system. For MOT, the growing amount of data available for most applications offers many perspectives. Deep learning algorithms are already bringing major performance leaps in the detection and tracking fields. It would be interesting to learn an adapted "meta-sensor" for RFS-based filters. By "meta-sensor", we refer to the proposition of Mahler in [Mah04], where the information returned by multiple sensors are merged into "meta-observations". Hence, these "meta-observations" seem to originate from a "meta-sensor" adapted to an RFS-based filter. Such a sensor could bring the advantages of sensor data fusion while offering "meta-observations" compatible with RFS-based filters.

The usage of deep learning could also benefit the evolution and measurement models of MOT filters. For instance, one could learn the birth probability density of new objects or the false alarm rate and detection probability. For this purpose, using a map or a High Definition map could give cues about where to initialize new objects while preventing the validation of false alarms occurring in odd areas.

In addition, the state evolution model could profit from the latest advancement in motion forecasting. For instance, the work of Mercat *et al.* [Mer+20] takes into account the objects' interactions and the surrounding environment to forecast the motion of tracked objects. This perspective seems very attractive, especially to detect future occlusions between the tracked objects or discard false alarms.

Because of the leap in performances offered by deep learning and the current research effort channeled by this topic, it had to be mentioned for the perspectives of this thesis. However, some applications lack data or necessitate explainable results, which prevent the usage of deep learning. For them, the Bayesian modeling effort put into the work of this manuscript is a good starting point.

First, most of the perspectives we have proposed using deep learning could be done using modeling assumptions and Bayesian inference. For instance, only the classical linear evolution model was considered for the automotive application, while a constant turn rate constant acceleration model is more appropriate in this case. Before using deep learning methods, numerous evolution models exist.

In addition, much information can be leveraged about the objects' motion models using prior knowledge. For example, in the case of autonomous vehicles, a lane detection and tracking algorithm such as the one proposed by Camarda *et al.* [Cam+20], could help forecast the future position of surrounding vehicles without relying on deep learning.

At last, a new type of sensor is on the verge of industrialization: the event camera. This technology is promising for tracking applications: its high refresh rate and high dynamic range offer interesting research perspectives. For instance, it can be turned into an inexpensive high-speed camera with unmatched low light capabilities when associated with a camera. This can be useful for demanding night environments with fast-moving objects. In addition, this

sensor's data stream is very lightweight compared to a camera sensor, which is an advantage in an embedded system offering limited computational capabilities.

Appendix A

Useful functions and probability density functions

A.1 Multivariate Gaussian probability density function

A multivariate Gaussian probability density of a vector variable $\mathbf{x}$ is defined as $\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})$, with

$$\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{P}|}} e^{-\frac{1}{2}(\hat{\mathbf{x}}-x)^T \mathbf{P}^{-1}(\hat{\mathbf{x}}-x)} \tag{A.1}$$

Here $\hat{\mathbf{x}}$ stands for the expectation of $\mathbf{x}$, and $\mathbf{P}$ is the error covariance matrix. d is the number of state dimensions of $\mathbf{x}$.

A.2 d-dimensional Gamma function

The d-dimensional Gamma function is a multivariate generalization of the Gamma function. For any positive integer $m > 1$, it is defined as

$$\Gamma_d(m) = \pi^{p(p-1)/4} \prod_{j=1}^{d} \Gamma\left(m + \frac{(1-j)}{2}\right) \tag{A.2}$$

with $\Gamma(m)$ the regular, univariate Gamma function:

$$\begin{aligned} \Gamma(m) &= (m-1)! \\ &= \Gamma_1(m) \end{aligned} \tag{A.3}$$

A.3 Wishart probability density function

A Wishart density is a probability density on d dimensional symmetric positive definite matrices. It models the uncertainty of the empirical covariance computed from Gaussian measurements:

$$\mathcal{W}(\mathbf{Z}; m, \mathbf{X}) = \frac{1}{(2)^{md/2} |\mathbf{X}|^{m/2} \Gamma_d(\frac{m}{2})} |\mathbf{Z}|^{(m-d-1)/2} e^{-\frac{1}{2} tr(\mathbf{X}^{-1}\mathbf{Z})} \tag{A.4}$$

where Γ_d is the d dimensional Gamma function, m is the degrees of freedom, and $\mathbf{X}$ is the scaling matrix. It should be noted that $(m > d-1)$ at all times.

A.4 Inverse Wishart probability density function

An Inverse Wishart density is a probability density on d dimensional symmetric positive definite matrices. The Inverse Wishart density is expressed as the inverse of the Wishart density, such as :

$$\mathcal{IW}(\mathbf{X};\nu,\mathbf{V}) = \frac{|\mathbf{V}|^{\nu/2}}{(2)^{\nu d/2}\Gamma_d(\frac{\nu}{2})}\,|\mathbf{X}|^{-(\nu+d+1)/2}\,e^{-\frac{1}{2}tr(\mathbf{V}\mathbf{X}^{-1})} \tag{A.5}$$

where Γ_d is the d dimensional Gamma function, ν is the degrees of freedom, and $\mathbf{V}$ is the scaling matrix. It should be noted that $(\nu > d-1)$ at all times.

The expectation of this density is

$$\mathbb{E}(\mathbf{X}) = \frac{\mathbf{V}}{\nu - d - 1} \tag{A.6}$$

In the RM model, this expected value is the extent.

A.5 Poisson probability density function

An extended object may return a random number of measurements. A discrete Poisson probability density can model these measurements. The discrete Poisson probability density is given by

$$\mathbf{P}(m;\gamma) = \frac{\gamma^m}{m!}e^{-\gamma} \tag{A.7}$$

with γ the expected number of measurements.

A.6 Gamma probability density function

The conjugate prior of the Poisson probability density is the Gamma probability density

$$\mathbf{G}(\gamma;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\gamma^{\alpha-1}e^{-\gamma\beta} \tag{A.8}$$

with α the shape and β the rate of the studied phenomenon. Here $\Gamma(\cdot)$ denotes the Gamma function, not to be confused with the Gamma probability density function.

The expected value of a Gamma probability density is given by

$$\mathbb{E}(\gamma) = \frac{\alpha}{\beta} \tag{A.9}$$

In the case of the Gamma Gaussian Inverse Wishart model, it is the expected number of measurements returned by an object.

Appendix B

Filters parameters

B.1 Parameters for the passive radar sensor datasets

This appendix presents the GO-PHD/F and GO-PHD/L filters' parameters for the results presented in the section 4.7.

Parameters	Birth Probability w_b	Num. Births	p_S	p_D
Values	0.001	348	0.99	0.98
Parameters	Clutter Rate	Cluster Size	T_{prune}	T_{merge}
Values	5×10^{-6} FA/m^2	5 m	(1)	30 m

Table B.1: GO-PHD/F and L parameters.

(1) Both filters are using a different threshold, $T_{prune} = 0.001$ for the GO-PHD/L filter, and $T_{prune} = 0.008$ for the GO-PHD/F filter.

B.2 Parameters for the pySim simulated radar dataset

This appendix presents the filters' parameters for the results presented in the section 4.8.3.

B.2.1 PMB

Parameters	Birth Probability w_b	Num. Births	p_S	p_D
Values	0.001	10	0.95	0.9
Parameters	Clutter Rate λ_{FA}	Cluster Size	T_{prune}	T_{merge}
Values	5×10^{-7} FA/m^2	Not needed	0.0001	5 m

Table B.2: PMB parameters.

B.2.2 GO-PMB/F and GO-PMB/L

Parameters	Birth Probability w_b	Num. Births	p_S	p_D
Values	0.001	10	0.95	0.9
Parameters	Clutter Rate λ_{FA}	Cluster Size	T_{prune}	T_{merge}
Values	5×10^{-7} FA/m^2	15 m	0.001	30 m

Table B.3: GO-PMB parameters.

B.3 Parameters for the active radar sensor dataset

This appendix presents the filters' parameters for the results presented in the section 4.8.3.

B.3.1 PMB

Parameters	Birth Probability w_b	Num. Births	p_S	p_D
Values	0.001	10	0.95	0.9
Parameters	Clutter Rate λ_{FA}	Cluster Size	T_{prune}	T_{merge}
Values	5×10^{-6} FA/m^2	Not needed	0.001	2 m

Table B.4: PMB parameters.

B.3.2 GO-PMB/F and GO-PMB/L

Parameters	Birth Probability w_b	Num. Births	p_S	p_D
Values	0.001	10	0.95	0.9
Parameters	Clutter Rate λ_{FA}	Cluster Size	T_{prune}	T_{merge}
Values	5×10^{-6} FA/m^2	3	0.001	2 m

Table B.5: GO-PMB parameters.

Appendix C

Choosing the structure of the state noise covariance for the Brownian models

This appendix was proposed by Professor Olivier Michel of the Gipsa-Lab.

With the goal is to determine the structure of the covariance matrix of the state noise, for a simple state model of the form

$$\begin{aligned} p_{t_2} &= p_{t_1} + v_{t_1}\ d_t + \varepsilon^p_{t_2} \\ v_{t_2} &= v_{t_1} + \varepsilon^v_{t_2} \end{aligned} \tag{C.1}$$

with p_{t_1} the position of an object x according one dimension at time step t_1, and v_{t_1} the speed along the same axis. Here t_1 and t_2 are two consecutive time indices. It is assumed that the only force driving the process is a white noise process. The question is to identify the mean and covariance matrix of the noise vector above.

This set of equations appears to be the discretized form of the following continuous time stochastic differential equation

$$dv_t = \alpha\ dw_t \tag{C.2}$$

where w_t is a Wiener-Levy Process (thus dw_t "behaves" like a random white noise process). Note that for physical interpretation, this amounts to assume that the Brownian driving force is dw_t. The constant α entails both the inertial mass of the object whose position and velocity are respectively p_t and v_t at time t. α stands for the diffusion coefficient of the Brownian process. As a consequence, the velocity v of the object x is the Wiener Levy (WL) process, and the position of x is the Riemann integral of this process.

$$p_{t_2} = p_{t_1} + \int_{t_1}^{t_2} w_s\ ds \tag{C.3}$$

Assume that $d_t = t_2 - t_1 > 0$, *i.e.*, it represents the elapsed time between the dates indexed respectively by t_1 and t_2.

Among the useful properties of WL processes, we have

$$\begin{aligned} \mathbb{E}[w_t] &= 0 \\ \mathbb{E}[w_t w_{t'}] &= \alpha\ min(t, t') \end{aligned} \tag{C.4}$$

allowing to compute the variance of x as a function of α:

$$\begin{aligned} var\left[\int_{t_1}^{t_2} w_s\ ds\right] &= \mathbb{E}\left[\left(\int_{t_1}^{t_2} w_s ds\right)^2\right] \\ &= \int_0^{d_t}\int_0^{d_t} \mathbb{E}[w_u\ w_s]\ du\ ds \\ &= \int_0^{d_t}\int_0^{d_t} \alpha\ min(u,s)\ du\ ds \\ &= \alpha\int_0^{d_t}\left[\int_0^{u} s\ ds\right]\ du + \alpha\int_0^{d_t}\left[\int_u^{d_t} u\ ds\right] du \\ &= \alpha\frac{d_t^3}{3} \end{aligned} \tag{C.5}$$

This means that under the dynamical model above, both the velocity and the position of the object x are non-stationary processes of respective variances that increase like d_t and $\frac{d_t^3}{3}$ when d_t increases. Furthermore, position and velocity being jointly consequences of the same driving force, they appear to be correlated:

$$\begin{aligned} \mathbb{E}[w_t\int_{t_1}^{t_1+d_t} w_s\ ds] &= \int_0^{d_t}\mathbb{E}[w_{d_t}\ w_s]ds \\ &= \alpha\int_0^{d_t} s\ ds \\ &= \alpha\frac{d_t^2}{2} \end{aligned} \tag{C.6}$$

Finally, for our model, the state noise covariance matrix is given by

$$\begin{aligned} \mathbf{Q} &= \mathbb{E}\left[\begin{bmatrix}\varepsilon_t^p\\ \varepsilon_t^v\end{bmatrix}\ [\varepsilon_t^p\ \varepsilon_t^v]\right] \\ &= \alpha\begin{bmatrix}\frac{d_t^3}{3} & \frac{d_t^2}{2}\\ \frac{d_t^2}{2} & d_t\end{bmatrix} \end{aligned} \tag{C.7}$$

Interpretation: Between two time samples, as our state equation is continuous and represents the evolution of a system excited by a white process accounting for the random forces, both the velocity and the position will exhibit non stationary behavior. Actually, the velocity is a Wiener process, and the position is the integral of this latter. The larger the elapsed time between consecutive samples, the larger will be their respective variances and covariance.

From a more pragmatic point of view, this quantifies the indeterminacy of both position and velocity of x and its evolution between two measurements.

Appendix D

Résumé français

D.1 Introduction

Le travail présenté dans ce doctorat s'intéresse au suivi de plusieurs objets et à l'estimation de leur trajectoire dans un environnement ouvert. Ce doctorat est une collaboration entre les équipes de recherche du groupe Renault, de l'ONERA et du Gipsa-Lab, cette synergie s'explique par la proximité des deux applications visées. En effet, le groupe Renault s'intéresse à l'estimation des paramètres cinématiques des usagers de la route entourant un véhicule autonome, tandis que l'ONERA vise la défense d'espaces aériens contre les intrusions de drones et de groupes de drones. Pour ces deux applications, l'objectif est d'estimer les paramètres cinématiques, position et vitesse par exemple, d'objets dynamiques se déplaçant dans une zone couverte par des capteurs hétérogènes.

Un objet est une entité dynamique observable par un ou plusieurs capteurs. Plusieurs paramètres caractérisent un objet, comme sa vitesse, sa position, sa taille ou son accélération. Ces paramètres sont partiellement observables en utilisant des mesures générées par des capteurs bruités. Une étape de filtrage est nécessaire lors de l'estimation pour réduire l'erreur d'estimation et estimer les paramètres non observables d'un objet. Par exemple, la vitesse et l'accélération d'un objet peuvent être déduites de ses positions successivement mesurées. L'inférence Bayésienne permet de construire des filtres qui minimisent l'erreur quadratique moyenne de l'estimation : ils sont optimaux sous certaines hypothèses.

Les applications visées ici se concentrent sur le suivi simultané de plusieurs objets. La présence de plusieurs objets ajoute un problème s'association entre mesures et objets, en plus du problème du filtrage. La question est de savoir quel objet a généré quelle mesure, tout en prenant en compte les défauts du capteur: par exemple des objets peuvent être manqués, des mesures peuvent correspondre à des objets inexistants ou inintéressants. Ces problèmes sont adressés par le domaine du suivi multi-objet, ou *Mutli Object Tracking* (MOT), à l'aide d'hypothèses de modélisation et des statistiques Bayésiennes.

Parmi les multiples objets à suivre, certains peuvent être des objets étendus, leur taille excède la résolution du capteur et ils retournent plusieurs mesures par acquisition. Le nombre de mesures et leur organisation spatiale sont liés à la forme et l'orientation de l'objet étendu: en plus des paramètres cinématiques, l'extension spatiale de l'objet peut être estimée. Deux cas de figure sont possibles pour l'estimation de la taille d'un objet étendu. Si une faible quantité de données est disponible, la construction d'un modèle de l'extension spatiale est une

première option. En revanche, si de grands jeux de données sont à disposition, l'extension spatiale peut être apprise pour chaque objet à l'aide de méthode d'apprentissage statistique. Cette distinction entre grands et petits jeux de données est essentielle, puisqu'elle conditionne l'état de l'art disponible pour l'estimation de l'extension spatiale d'un objet étendu.

Ce travail propose une approche original, qui repose sur l'utilisation de l'inférence Bayésienne pour la résolution du problème de suivi multi-objet et l'estimation de l'extension. Les deux applications proposées reposent sur la modélisation du problème multi-objet à l'aide des Ensembles Aléatoires Finis. En revanche, due à la taille des jeux de données disponibles pour chaque application, les solutions choisies pour estimer la forme et la taille des objets étendus diffèrent. Pour le suivi de groupes de drones, où peu de données sont disponibles, les objets étendus sont modélisés à l'aide de matrices aléatoires. Quant à l'application automobile, l'estimation des objets étendus est réalisé à l'aide d'un réseau de neurones profond, entrainé sur la grande quantité de données disponibles.

Le premier chapitre se concentre sur le problème du suivi multi-objet, et les filtres couramment rencontrés dans l'état de l'art pour le résoudre. Il s'attarde aussi sur les limites de ces filtres, le fait de ne considérer l'inférence Bayésienne qu'au niveau de l'objet et non du système multi-objet, ce qui est sous optimal. Le second chapitre présente donc les ensembles aléatoires finis, qui permettent de s'affranchir des limites évoquées dans le premier chapitre tout en résolvant le problème du suivi mutli-objet. Le cas des objets étendus est traité dans le troisième chapitre, surtout la modélisation basée sur les matrices aléatoires. Les deux derniers chapitres se concentrent sur les applications, d'abord le suivi de drones et de groupes de drones, et ensuite la perception pour le véhicule autonome.

D.2 Résumé des chapitres

D.2.1 Résolution du problème du suivi multi-objet avec l'inférence Bayésienne

Ce chapitre aborde l'estimation des paramètres cinématiques d'un objet: sa position, sa vitesse et son accélération. Des rappels concernant le filtrage Bayésien pour un objet seul permettent d'aborder le filtrage multi-objet et le filtre standard des plus proches voisins globaux, *Global Nearest Neighbors Standard Filter* (GNNSF).

Le filtre GNNSF est une extension du filtre de Kalman: il s'agit de scinder le problème du suivi multi-objet en deux. D'abord l'appairage entre mesures et objets est réalisé avec un algorithme d'affectation, puis un filtre de Kalman par couple objet-mesure retourne une estimation des paramètres cinématiques de l'objet. Lorsque plusieurs résultats d'appairage sont retenus, plusieurs hypothèses d'associations globales sont propagées: c'est le filtre de suivi à hypothèses multiples, *Multiple Hypothesis Tracking* (MHT).

Malheureusement, les filtres MHT et GNNSF ne décrivent pas explicitement la vraisemblance des mesures et le modèle d'évolution du système multi-objet: la création de nouveaux

objets ou les fausses alarmes sont gérées avec des hypothèses empiriques. Le formalisme offert par les ensembles aléatoires finis permet de définir un filtre Bayésien complet.

D.2.2 Modélisation du problème du suivi-multi objet avec les ensembles aléatoires finis

Les filtres à ensembles aléatoires finis sont des ensembles de vecteurs dont le cardinal est aléatoire. L'objectif est d'utiliser ces ensembles en lieu et place du vecteur d'état dans l'inférence Bayésienne. Cela permet de définir des modèles de mesure et d'évolution qui intègrent les problématiques du suivi multi-objet. Un avantage supplémentaire provient de la résolution du problème d'association, qui se fait conjointement avec l'estimation de l'état des objets.

Trois filtres basés sur les ensembles aléatoires finis sont présentés dans ce chapitre. Le premier est le filtre *Probability Hypothesis Density* (PHD), qui permet de suivre des objets sans recourir à un algorithme d'affectation. C'est un filtre léger mais sujet à l'amalgame d'objets proches, ce qui peut être un problème pour l'application automobile par exemple. Pour pallier ce problème, le filtre *Generalized Labeled Multi-Bernoulli* (GLMB) est aussi présenté: il suit la même philosophie que le filtre MHT.

Tout comme un filtre MHT, le filtre GLMB a une empreinte mémoire et un coût en calculs bien plus élevés que le filtre PHD car il propage plusieurs hypothèses d'association simultanément. Une alternative qui vise à résoudre ces problèmes est le filtre *Poisson Multi-Bernoulli Mixture filter*, c'est une approche hybride combinant les résultats du filtre PHD et ceux du filtre GLMB.

Ces filtres basés sur les ensembles aléatoires finis adressent les problèmes évoqués dans le premier chapitre. En revanche, ils reposent sur l'hypothèse que les objets retournent au plus une mesure, ce qui n'est pas vérifié dans les applications visées par cette thèse. Un nouveau modèle d'état est donc nécessaire pour ces objets qui sont qualifiés d'étendus.

D.2.3 Les modèles d'états pour objets étendus

Les objets étendus retournent plusieurs mesures par acquisition du capteur. Parmi les modèles d'états prenant cette caractéristique en compte, l'approche des matrices aléatoires pour les objets étendus considère que les mesures retournées par un objet étendu sont distribuées uniformément dans une forme elliptique. L'avantage de ce modèle est sa faible complexité: au coût de l'hypothèse d'une forme en ellipse, l'estimation de la taille de l'objet revient à estimer une matrice symétrique définie positive du même nombre de dimensions que le problème étudié.

Pour estimer une matrice symétrique définie positive, le bruit de mesure est supposé Gaussien, ce qui amène à une loi de probabilité sur les matrices aléatoires de Wishart pour le groupe de mesure et une loi de probabilité Wishart inverse pour la taille et l'orientation de l'objet étendu. Pour l'état cinématique de l'objet, la densité de probabilité Gaussienne est

conservée, ce qui donne le modèle d'état Gaussien Wishart inverse pour le modèle d'état.

Le modèle d'état Gaussien Wishart inverse pour les cibles étendues souffre de plusieurs limitations pour les applications visées dans le cadre de ce doctorat. Il est considéré que le nombre de mesures retourné est constant et que le bruit de mesure est négligeable devant la taille de l'objet étendu. Ce chapitre propose donc quatre récursions Bayésiennes avec des hypothèses supplémentaires pour résoudre ces problèmes. Ces modèles d'états basés sur les matrices aléatoires sont intégrés dans les algorithmes de suivi multi-objet dans le chapitre suivant.

D.2.4 Le suivi multi-objet étendus pour le suivi de drones

L'intégration des quatre modèles d'état à matrices aléatoires dans des filtres multi-objets basés sur les ensembles aléatoires finis concerne ici l'application au suivi de groupes de drones. Trois jeux de donnés sont utilisés pour les résultats: les données issues d'un radar actif simulé, les données issues d'un radar actif réel et les données issues d'un radar passif réel. Les tests de performance sont réalisés avec le logiciel pySim, développé en partie dans le cadre de ce doctorat.

Les contributions proposées dans ce chapitre sont les suivantes. Tout d'abord, la proposition d'un modèle à matrices aléatoires spécifiquement développé pour le suivi d'objets groupés et d'objets étendus faiblement résolus. Ensuite, son intégration dans des filtres basés sur les ensembles aléatoires finis, résultant dans des filtres limitant le nombre d'hypothèses nuisantes au formalisme Bayésien. Enfin, l'acquisition de jeux de données pour le test des algorithmes développés.

Ce travail ouvre plusieurs perspectives, notamment l'intégration de modèles cinématiques non linéaires pour le suivi de groupes de drones et un travail sur la stabilité de la forme elliptique dans le cas des groupes qui se déforment rapidement. De plus, des jeux de données avec la vérité-terrain de l'extension des groupes seraient intéressants pour calculer des métriques non empiriques concernant la taille des groupes de drones suivis. Enfin, l'étude plus complète de l'impact de la matrice d'évolution de la forme des groupes permettrait de quantifier son avantage par rapport à un modèle d'évolution constant.

D.2.5 Le suivi multi-objet pour le véhicule autonome

Contrairement au chapitre précédent, l'estimation de la taille des objets étendus est ici réalisée avec des algorithmes d'apprentissage profond. En effet, en présence de grands jeux de données, les algorithmes de détection basés sur des modèles offrent de moins bonnes performance que leurs homologues basés sur des réseaux de neurones profonds. L'objectif de ce chapitre est de comparer les performances en suivi multi-objet des filtres basés sur les ensembles aléatoires finis à d'autres algorithmes proposés sur le jeu de données nuScenes.

L'implémentation d'un filtre *Generalized Labeled Multi-Bernoulli* (GLMB) permet

d'atteindre les résultats de l'état de l'art sur le jeu de données nuScenes. L'avantage de ce filtre par rapport à l'état de l'art est sa robustesse, ce qui est désirable pour un véhicule autonome. Tous les résultats de l'état de l'art témoignent néanmoins de la dépendance à l'algorithme d'apprentissage profond pour la détection d'objets étendus.

Plusieurs perspectives peuvent être considérées pour améliorer les résultats obtenus. L'utilisation d'un modèle d'état à interactions multiples par exemple, pour prévenir le changement de mouvement des objets entourant le véhicule autonome et la fusion multi-capteurs sont les premières pistes à considérer.

D.3 Conclusion

D.3.1 Synthèse

Ce travail est une collaboration entre l'ONERA, le Gipsa-lab et le groupe Renault. Il se concentre sur deux applications de suivi multi-objet. Il s'agit de la perception du véhicule autonome pour le groupe Renault, et du suivi de groupes de drones pour l'ONERA. Ce sont deux formulations du problème du suivi multi-objet étendu. Nous avons proposé de les résoudre en utilisant majoritairement la modélisation Bayésienne.

Les ensembles aléatoires finis permettent de résoudre le problème du suivi multi-objet en utilisant uniquement l'inférence Bayésienne. Le formalisme offert par cette description permet de décrire les phénomènes observables dans les mesures du capteur et dans le modèle d'évolution du système multi-objet, plutôt que de construire un filtre basé sur un raisonnement heuristique. Ces observations ont permis la description de trois filtres basés sur les ensembles aléatoires finis.

En plus du suivi multi-objet, chacune des applications impliquent des objets étendus. Deux méthodes sont retenues pour résoudre ce problème: le modèle des matrices aléatoires dans le cas du suivi de groupes de drones et les réseaux de neurones profonds dans le cas de l'application automobile. Ces choix se sont basés sur la taille des jeux de données disponibles pour chaque application.

Le suivi de groupes de drones a donné lieu à l'utilisation de deux filtres basés sur les ensembles aléatoires finis, et quatre modèles de matrices aléatoires pour l'extension des groupes. Nous avons proposé un modèle d'extension spécifique pour les groupes de drones, intégrant le bruit de mesure et le nombre aléatoire de mesures. De plus, nous avons proposé de prendre en considération la possibilité de suivre simultanément des objets étendus et des objets sous-résolus. Ces travaux sont intégrés dans pySim, un logiciel python qui permet de jouer des scénarios de suivi de groupes de drones réels et simulés.

Quant à l'application automobile, un filtre basé sur les ensembles aléatoires finis fut aussi développé, en utilisant les détections d'un réseau de neurones profond en guise de mesures. Nous avons atteint des résultats au niveau de l'état de l'art sur la plupart des métriques

étudiées. Le filtre développé à notamment validé en moyenne moins de fausses alarmes sans résoudre totalement ce problème: cet aspect doit être abordé dans de futurs travaux.

D.3.2 Perspectives

Tout d'abord l'intégration d'algorithmes d'apprentissage profond pour tirer le plus d'information *a priori* possible des jeux de données disponibles. Pour le suivi multi-objet, cela signifierait l'apprentissage d'un "meta-capteur", comme proposé par Mahler dans [Mah04]. Il permettrait de synthétiser les mesures générées par plusieurs capteurs dans des "meta-observations", qui seraient alors utilisées en entrée d'un filtre basé sur les ensembles aléatoires finis. Ce "meta-capteur" permettrait d'explorer les avantages offerts par la fusion de données multi-capteurs.

L'apprentissage profond pourrait aussi bénéficier aux modèles de mesure et d'évolution des filtres multi-objets. Par exemple, certains paramètres du filtre pourraient être appris, tel que la probabilité de détection, ou le taux de fausses alarmes. Un autre exemple, l'utilisation de la carte haute définition permettrait d'avoir plus d'informations a priori concernant la naissance de nouveaux objets, tout en prévenant la validation de fausses alarmes dans des zones improbables.

Les modèles prédictifs proposés par Mercat *et al.* [Mer+20] serait aussi un ajout intéressant pour prendre en compte l'interaction des objets avec leur environnement. Ces modèles permettraient entre autres de prédire des variations de comportements dans la trajectoire ou des occlusions futures.

La plupart des perspectives proposées reposent sur l'apprentissage profond, notamment car ces algorithmes offrent des performances bien supérieures à ceux reposant sur la modélisation. Pourtant, la modélisation est inévitable lorsque peu de donnés sont disponibles. Dans ce cas la plupart des perspectives précédentes peuvent être abordées avec des modèles plus complexes. Par exemple, des modèles d'évolution plus complexes, comme ceux proposés par Camarda *et al.* [Cam+20], peuvent permettre de prédire les positions futures des véhicules par rapport aux voies routières sans apprentissage profond.

Enfin, de nouveaux capteurs font leur apparition: les caméras événementielles. Ces caméras sont asynchrones, elles retournent une information de changement de contraste lorsqu'il se produit dans un pixel. Cette mise à jour asynchrone permet un taux de rafraichissement et une gamme dynamique élevés. Ce capteur est intéressant dans des conditions de faible luminosité avec des objets qui se meuvent à haute vitesse, comme ça peut être le cas sur une autoroute de nuit. Pour ces raisons, leur intégration aux systèmes de suivi multi-objet existant ouvre de nouvelles perspectives de recherche.

Bibliography

[AOG15] Tohid Ardeshiri, Umut Orguner, and Fredrik Gustafsson. "Bayesian inference via approximation of log-likelihood for priors in exponential family." In: *arXiv preprint arXiv:1510.01225* (2015) (cit. on p. 7).

[BBS07] Adrian Baddeley, Imre Bárány, and Rolf Schneider. "Spatial point processes and their applications." In: *Stochastic Geometry: Lectures given at the CIME Summer School held in Martina Franca, Italy, September 13–18, 2004* (2007), pp. 1–75.

[BBS88] Henk AP Blom and Yaakov Bar-Shalom. "The interacting multiple model algorithm for systems with Markovian switching coefficients." In: *IEEE transactions on Automatic Control* 33.8 (1988), pp. 780–783.

[BH09] Marcus Baum and Uwe D Hanebeck. "Random hypersurface models for extended object tracking." In: *Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on*. IEEE. 2009, pp. 178–183 (cit. on pp. 49, 50).

[BH14] Marcus Baum and Uwe D Hanebeck. "Extended object tracking with random hypersurface models." In: *IEEE Transactions on Aerospace and Electronic systems* 50.1 (2014), pp. 149–159.

[Bla04] Samuel S Blackman. "Multiple hypothesis tracking for multiple target tracking." In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 (2004), pp. 5–18.

[Bla86] Samuel S Blackman. "Multiple-target tracking with radar applications." In: *Dedham* (1986) (cit. on p. 66).

[Bot17] F Botha. "Data Fusion of Radar and Stereo Vision for Detection and Tracking of Moving Objects." PhD thesis. Matieland 7602, South Africa: University of Stellenbosch, Mar. 2017.

[Bry+18] Daniel S Bryant et al. "A Generalized Labeled Multi-Bernoulli Filter with Object Spawning." In: *IEEE Transactions on Signal Processing* 66.23 (2018), pp. 6177–6189.

[BSF88] Yaakov Bar-Shalom and Thomas E Fortmann. *Tracking and Data Association*. Academic Press Professional, Inc., 1988 (cit. on p. 9).

[BSL95] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking: principles and techniques*. Storrs: YBS Publishing, 1995 (cit. on pp. 9, 14).

[Cae+20] Holger Caesar et al. "nuscenes: A multimodal dataset for autonomous driving." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Proc. CVF. 2020, pp. 11621–11631 (cit. on pp. 38, 100, 108).

[Cam+20] Federico Camarda et al. "Multisensor Tracking of Lane Boundaries based on Smart Sensor Fusion." In: *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2020, pp. 1649–1654 (cit. on pp. 116, 130).

[CG07] Daniel Clark and Simon Godsill. "Group target tracking with the gaussian mixture probability hypothesis density filter." In: *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. IEEE. 2007, pp. 149–154 (cit. on p. 66).

[Cou+06] Christophe Coué et al. "Bayesian occupancy filtering for multitarget tracking: an automotive application." In: *The International Journal of Robotics Research* 25.1 (2006), pp. 19–30.

[CPV06] Daniel E Clark, Kusha Panta, and Ba-Ngu Vo. "The GM-PHD filter multiple target tracker." In: *information Fusion, 2006 9th International Conference on*. IEEE. 2006, pp. 1–8 (cit. on pp. 19, 31, 90).

[CVV07] Daniel Clark, Ba-Tuong Vo, and Ba-Ngu Vo. "Gaussian particle implementations of probability hypothesis density filters." In: *2007 IEEE Aerospace Conference*. IEEE. 2007, pp. 1–11 (cit. on p. 19).

[Del+17] Emmanuel Delande et al. "Novel multi-object filtering approach for space situational awareness." In: *Journal of Guidance, Control, and Dynamics* 41.1 (2017), pp. 59–73.

[Dem67] Arthur P Dempster. "Upper and lower probabilities induced by a multivalued mapping." In: *The Annals of Mathematical Statistics*. Vol. 38. 2. Institute of Mathematical Statistics, 1967, pp. 325–339.

[Dez02] Jean Dezert. "Foundations for a new theory of plausible and paradoxical reasoning." In: *Information and Security* 9 (2002), pp. 13–57.

[Die01] KC Dietmayer. "Model-Based Object Classification and Object Tracking in Traffic Scenes from Range-Images." In: *IV2001* (2001), pp. 25–30.

[DPR18] Maxime Derome, Benjamin Pannetier, and Michèle Rombaut. *Projet de recherche: Fusion de données multi-capteurs pour la perception d'environnement appliquée aux Aides à la Conduite (ADAS) et à la Voiture Autonome*. 2018.

[DS12] Jean Dezert and Florentin Smarandache. "An introduction to DSmT for information fusion." In: *New mathematics and Natural Computation* 8.03 (2012), pp. 343–359.

[DT97] Frank Dellaert and Chuck Thorpe. "Robust car tracking using Kalman filtering and Bayesian templates." In: *Conference on intelligent transportation systems*. Vol. 1. 1997 (cit. on p. 9).

[Est+96] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *KDD*. Vol. 96. Proc. KDD. 1996, pp. 226–231 (cit. on p. 68).

[Eur16] Eurobios. *Modèles probabilistes de fusion objet ADAS*. Confidential, 2016.

[Fai+15] Florian Faion et al. "Symmetries in Bayesian Extended Object Tracking." In: *Journal of Advances in Information Fusion* 10.1 (2015), pp. 13–30.

[FFK11] Michael Feldmann, Dietrich Franken, and Wolfgang Koch. "Tracking of extended objects and group targets using random matrices." In: *IEEE Transactions on Signal Processing* 59 (2011), pp. 1409–1420 (cit. on pp. 51, 60).

[FPZ17] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. "Detect to track and track to detect." In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 3038–3046 (cit. on p. 104).

[FSU09] D Franken, M Schmidt, and M Ulmke. "" Spooky action at a distance" in the cardinalized probability hypothesis density filter." In: *IEEE Transactions on Aerospace and Electronic Systems* 45.4 (2009).

[GBR16] Karl Granström, Marcus Baum, and Stephan Reuter. "Extended object tracking: Introduction, overview and applications." In: *arXiv preprint arXiv:1604.00970* (2016) (cit. on pp. 49, 51).

[Gér19] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly, 2019.

[GF+18] Ángel F García-Fernández et al. "Poisson multi-Bernoulli mixture filter: direct derivation and implementation." In: *IEEE Transactions on Aerospace and Electronic Systems* 54 (2018), pp. 1883–1901 (cit. on pp. 35, 39).

[GF+20] Ángel F García-Fernández et al. "Trajectory poisson multi-Bernoulli filters." In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 4933–4945 (cit. on p. 44).

[GFS16] Karl Granström, Maryam Fatemi, and Lennart Svensson. "Gamma Gaussian inverse-Wishart Poisson multi-Bernoulli filter for extended target tracking." In: *2016 19th International Conference on Information Fusion (FUSION)*. IEEE. 2016, pp. 893–900 (cit. on pp. 79, 80).

[GFS19] Karl Granström, Maryam Fatemi, and Lennart Svensson. "Poisson multi-Bernoulli mixture conjugate prior for multiple extended target filtering." In: *IEEE Transactions on Aerospace and Electronic Systems* 56 (2019), pp. 208–225 (cit. on pp. 7, 63, 74).

[Gil+05] Kevin Gilholm et al. "Poisson models for extended target and group tracking." In: *Signal and Data Processing of Small Targets 2005*. Vol. 5913. Proc. SPIE. 2005, pp. 230 –241 (cit. on p. 62).

[GLO11] Karl Granström, Christian Lundquist, and Umut Orguner. "Tracking rectangular and elliptical extended targets using laser measurements." In: *14th International Conference on Information Fusion*. IEEE. 2011, pp. 1–8.

[GLO12] Karl Granström, Christian Lundquist, and Umut Orguner. "Extended target tracking using a Gaussian-mixture PHD filter." In: *IEEE Transactions on Aerospace and Electronic Systems* 48 (2012), pp. 3268–3286.

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012 (cit. on p. 100).

[GN99] Arjun K Gupta and Daya K Nagar. *Matrix variate distributions*. Chapman and Hall/CRC, 1999.

[GO12a] Karl Granström and Umut Orguner. "Estimation and maintenance of measurement rates for multiple extended target tracking." In: *2012 15th International Conference on Information Fusion*. IEEE. 2012, pp. 2170–2176 (cit. on pp. 70, 76).

[GO12b] Karl Granström and Umut Orguner. *Implementation of the GIW-PHD filter*. Linköping University Electronic Press, 2012.

[GO12c] Karl Granström and Umut Orguner. "A PHD filter for tracking multiple extended targets using random matrices." In: *IEEE Transactions on Signal Processing* 60.11 (2012), pp. 5657–5671 (cit. on p. 68).

[GO14] Karl Granström and Umut Orguner. "New prediction for extended targets with random matrices." In: *IEEE Transactions on Aerospace and Electronic Systems* 50 (2014), pp. 1577–1589 (cit. on pp. 62, 63).

[Gra+14] Karl Granström et al. "A multiple model PHD approach to tracking of cars under an assumed rectangular shape." In: *17th International Conference on Information Fusion (FUSION)*. IEEE. 2014, pp. 1–8.

[Gra+15] Karl Granström et al. "Gamma Gaussian Inverse Wishart Probability Hypothesis Density for Extended Target Tracking Using X-Band Marine Radar Data." In: *IEEE Transactions on Geoscience and Remote Sensing* 53 (2015), pp. 6617–6631 (cit. on p. 74).

[Gra+17] Karl Granström et al. "Corrections on:"Extended Target Tracking Using a Gaussian-Mixture PHD Filter"." In: *IEEE Transactions on Aerospace and Electronic Systems* 53.2 (2017), pp. 1055–1058.

[GS05] Kevin Gilholm and David Salmond. "Spatial distribution model for tracking extended objects." In: *IEEE Proceedings-Radar, Sonar and Navigation* 152 (2005), pp. 364–371 (cit. on p. 62).

[GSF97] Neil T Gordon, David J Salmond, and David J Fisher. "Bayesian target tracking after group pattern distortion." In: *Signal and Data Processing of Small Targets 1997*. Vol. 3163. SPIE. 1997, pp. 238–248 (cit. on p. 66).

[Gue+20] Louis Guerlin et al. "Study on group target tracking to counter swarms of drones." In: *Signal Processing, Sensor/Information Fusion, and Target Recognition XXIX*. Vol. 11423. International Society for Optics and Photonics. 2020, p. 1142304 (cit. on pp. 44, 46, 51, 63, 64, 81).

[Gue+21] Louis Guerlin et al. "UAS Swarm detection and tracking with passive-radar." In: *2021 International Conference on Radar*. CIE. 2021 (cit. on pp. 32, 46, 51, 64, 75).

[Gue+22] Louis Guerlin et al. "Détection et suivi multi-objets étendus par des approches Bayésiennes." In: *GRETSI'22, Accepted*. GRETSI. 2022 (cit. on pp. 38, 46, 51).

[HNP09] Alon Halevy, Peter Norvig, and Fernando Pereira. "The unreasonable effectiveness of data." In: *IEEE intelligent systems* 24.2 (2009), pp. 8–12 (cit. on p. 101).

[Jaz70] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Academic Press Inc, 1970.

[JU04] Simon J Julier and Jeffrey K Uhlmann. "Unscented filtering and nonlinear estimation." In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422 (cit. on p. 9).

[JV87] Roy Jonker and Anton Volgenant. "A shortest augmenting path algorithm for dense and sparse linear assignment problems." In: *Computing* 38.4 (1987), pp. 325–340 (cit. on p. 11).

[Kal60] Rudolph Emil Kalman. "A new approach to linear filtering and prediction problems." In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45 (cit. on pp. 4, 6).

[Kir+00] Thia Kirubarajan et al. "Ground target tracking with variable structure IMM estimator." In: *IEEE Transactions on Aerospace and Electronic Systems* 36 (2000), pp. 26–46.

[Koc08] J. W. Koch. "Bayesian approach to extended object and cluster tracking using random matrices." In: *IEEE Transactions on Aerospace and Electronic Systems* 44 (2008), pp. 1042–1059 (cit. on pp. 51–53, 55–57, 62, 70, 97).

[Koc13] Wolfgang Koch. *Tracking and sensor data fusion: methodological framework and selected applications*. Springer Science & Business Media, 2013.

[KOLT21] Aleksandr Kim, Aljoša Ošep, and Laura Leal-Taixé. "Eagermot: 3d multi-object tracking via sensor fusion." In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 11315–11321 (cit. on pp. 100, 104).

[KSD16] Christina Knill, Alexander Scheel, and Klaus Dietmayer. "A direct scattering model for tracking vehicles with high-resolution radars." In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 298–303.

[KVK97] Wolfgang Koch and Günter Van Keuk. "Multiple hypothesis track maintenance with possibly unresolved measurements." In: *IEEE Transactions on Aerospace and Electronic Systems* 33 (1997), pp. 883–892.

[LGO13] Christian Lundquist, Karl Granström, and Umut Orguner. "An extended target CPHD filter and a gamma Gaussian inverse Wishart implementation." In: *IEEE Journal of Selected Topics in Signal Processing* 7 (2013), pp. 472–483 (cit. on p. 63).

[Li+18] Peng Li et al. "Modified Gaussian inverse Wishart PHD filter for tracking multiple non-ellipsoidal extended targets." In: *Signal Processing* 150 (2018), pp. 191–203.

[LL12a] Jian Lan and X Rong Li. "Tracking of extended object or target group using random matrix — Part I: New model and approach." In: *2012 15th International Conference on Information Fusion*. Proc. ICIF. 2012, pp. 2177–2184 (cit. on pp. 51, 59, 61, 62, 83).

[LL12b] Jian Lan and X Rong Li. "Tracking of extended object or target group using random matrix—Part II: Irregular object." In: *2012 15th International Conference on Information Fusion*. IEEE. 2012, pp. 2185–2192 (cit. on p. 51).

[LL14] Jian Lan and X Rong Li. "Tracking of maneuvering non-ellipsoidal extended object or target group using random matrix." In: *IEEE Transactions on Signal Processing* 62.9 (2014), pp. 2450–2463.

[LL16] Jian Lan and X Rong Li. "Tracking of extended object or target group using random matrix: New model and approach." In: *IEEE Transactions on Aerospace and Electronic Systems* 52.6 (2016), pp. 2973–2989 (cit. on p. 83).

[LSH13] Malin Lundgren, Lennart Svensson, and Lars Hammarstrand. "A CPHD filter for tracking with spawning models." In: *IEEE Journal of Selected Topics in Signal Processing* 7.3 (2013), pp. 496–507 (cit. on p. 26).

[Mah00] Ronald Mahler. "Optimal/robust distributed data fusion: a unified approach." In: *Signal Processing, Sensor Fusion, and Target Recognition IX*. Vol. 4052. Proc. SPIE. 2000, pp. 128–139.

[Mah03] Ronald Mahler. "Multitarget Bayes filtering via first-order multitarget moments." In: *IEEE Transactions on Aerospace and Electronic systems* 39 (2003), pp. 1152–1178 (cit. on pp. 19, 25).

[Mah04] Ronald Mahler. "Statistics 101 for multisensor, multitarget data fusion." In: *IEEE Aerospace and Electronic Systems Magazine* 19 (2004), pp. 53–64 (cit. on pp. 102, 116, 130).

[Mah07] Ronald Mahler. *Statistical multisource-multitarget information fusion*. Boston: Artech House, Inc., 2007 (cit. on pp. 16, 18, 19, 22, 25, 26, 66).

[Mah12] Ronald Mahler. "On multitarget jump-Markov filters." In: *2012 15th International Conference on Information Fusion*. IEEE. 2012, pp. 149–156.

[Mah13] Ronald Mahler. "Statistics 102 for multisource-multitarget detection and tracking." In: *IEEE Journal of Selected Topics in Signal Processing* 7 (2013), pp. 376–389 (cit. on p. 25).

[Mah19] Ronald Mahler. "A GLMB filter for unified multitarget multisensor management." In: *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII*. Vol. 11018. International Society for Optics and Photonics. 2019, p. 110180D (cit. on p. 38).

[Mal+17] Mateusz Piotr Malanowski et al. *Drone detection experiment using DVB-T-based passive radar*. Tech. rep. The Institute of Electronic Systems, 2017 (cit. on p. 86).

[MCB11a] Julien Moras, Véronique Cherfaoui, and Philippe Bonnifait. "Credibilist occupancy grids for vehicle perception in dynamic environments." In: *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 84–89.

[MCB11b] Julien Moras, Véronique Cherfaoui, and Philippe Bonnifait. "Moving objects detection by conflict analysis in evidential grids." In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2011, pp. 1122–1127.

[McE66] Bruce A McElhoe. "An assessment of the navigation and course corrections for a manned flyby of mars or venus." In: *IEEE Transactions on Aerospace and Electronic Systems* 4 (1966), pp. 613–623 (cit. on p. 9).

[Mer+20] Jean Mercat et al. "Multi-head attention for multi-modal joint vehicle motion forecasting." In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 9638–9644 (cit. on pp. 116, 130).

[Mic+17] Martin Michaelis et al. "Heterogeneous multi-sensor fusion for extended objects in automotive scenarios using Gaussian processes and a GMPHD-filter." In: *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE. 2017, pp. 1–6 (cit. on p. 32).

[Mun57] James Munkres. "Algorithms for the assignment and transportation problems." In: *Journal of the society for industrial and applied mathematics* 5 (1957), pp. 32–38 (cit. on pp. 11, 84).

[Mur68] Katta G Murty. "Letter to the editor—An algorithm for ranking all the assignments in order of increasing cost." In: *Operations research* 16.3 (1968), pp. 682–687 (cit. on pp. 14, 43).

[NRL14] Amaury Nègre, Lukas Rummelhard, and Christian Laugier. "Hybrid sampling bayesian occupancy filter." In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE. 2014, pp. 1307–1312.

[Org12] Umut Orguner. "A variational measurement update for extended target tracking with random matrices." In: *IEEE Transactions on Signal Processing* 60.7 (2012), pp. 3827–3834.

[Pas+09] Syed Ahmed Pasha et al. "A Gaussian mixture PHD filter for jump Markov system models." In: *IEEE Transactions on Aerospace and Electronic systems* 45.3 (2009), pp. 919–936.

[PP02] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[PT09] Anna Petrovskaya and Sebastian Thrun. "Model based vehicle detection and tracking for autonomous urban driving." In: *Autonomous Robots* 26.2-3 (2009), pp. 123–139 (cit. on p. 101).

[Rei79] Donald Reid. "An algorithm for tracking multiple targets." In: *IEEE transactions on Automatic Control* 24.6 (1979), pp. 843–854.

[Reu+14] Stephan Reuter et al. "The labeled multi-Bernoulli filter." In: *IEEE Transactions on Signal Processing* 62 (2014), pp. 3246–3260.

[RGFS17] Abu Sajana Rahmathullah, Ángel F García-Fernández, and Lennart Svensson. "Generalized optimal sub-pattern assignment metric." In: *2017 20th International Conference on Information Fusion (Fusion)*. Proc. ICIF. 2017, pp. 1–8 (cit. on p. 84).

[Ris+12] Branko Ristic et al. "Adaptive target birth intensity for PHD and CPHD filters." In: *IEEE Transactions on Aerospace and Electronic Systems* 48.2 (2012), pp. 1656–1668.

[RS61] Howard Raiffa and Robert Schlaifer. *Applied statistical decision theory*. Harvard University, 1961.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144 (cit. on p. 101).

[SC11] Anthony Swain and Daniel Clark. "The single-group PHD filter: An analytic solution." In: *14th International Conference on Information Fusion*. IEEE. 2011, pp. 1–8 (cit. on p. 66).

[Sch+16] Alexander Scheel et al. "Multi-sensor multi-object tracking of vehicles using high-resolution radars." In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 558–565.

[SG00] David J Salmond and Neil J Gordon. "Group tracking with limited sensor resolution and finite field of view." In: *Signal and Data Processing of Small Targets 2000*. Vol. 4048. SPIE. 2000, pp. 532–540.

[SG99] David J Salmond and Neil J Gordon. "Group and extended object tracking." In: (1999) (cit. on p. 66).

[Sha76] Glenn Shafer. *A mathematical theory of evidence*. Vol. 42. Princeton university press, 1976.

[SSM62] Gerald L Smith, Stanley F Schmidt, and Leonard A McGee. *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. National Aeronautics and Space Administration, 1962 (cit. on p. 9).

[SVV08] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. "A consistent metric for performance evaluation of multi-object filters." In: *IEEE transactions on signal processing* 56.8 (2008), pp. 3447–3457 (cit. on p. 84).

[TBH18] Kolja Thormann, Marcus Baum, and Jens Honer. "Extended Target Tracking Using Gaussian Processes with High-Resolution Automotive Radar." In: *2018 21st International Conference on Information Fusion (FUSION)*. IEEE. 2018, pp. 1764–1770.

[Viv+15] G. Vivone et al. "Converted measurements random matrix approach to extended target tracking using X-band marine radar data." In: *2015 18th International Conference on Information Fusion (Fusion)*. Proc. ICIF. 2015, pp. 976–983 (cit. on p. 60).

[VM06] Ba-Ngu Vo and Wing-Kin Ma. "The Gaussian Mixture Probability Hypothesis Density Filter." In: *IEEE Transactions Signal Processing* 54 (2006), pp. 4091–4104 (cit. on pp. 19, 24, 26, 31).

[Von12] Pascal O Vontobel. "The Bethe permanent of a nonnegative matrix." In: *IEEE Transactions on Information Theory* 59.3 (2012), pp. 1866–1901.

[VV13] Ba-Tuong Vo and Ba-Ngu Vo. "Labeled random finite sets and multi-object conjugate priors." In: *IEEE Transactions on Signal Processing* 61.13 (2013), pp. 3460–3475 (cit. on pp. 7, 32, 33, 36).

[VVB19] Ba-Ngu Vo, Ba-Tuong Vo, and Michael Beard. "Multi-sensor multi-object tracking with the generalized labeled multi-Bernoulli filter." In: *IEEE Transactions on Signal Processing* 67.23 (2019), pp. 5952–5967 (cit. on p. 105).

[VVH16] Ba-Ngu Vo, Ba-Tuong Vo, and Hung Gia Hoang. "An efficient implementation of the generalized labeled multi-Bernoulli filter." In: *IEEE Transactions on Signal Processing* 65 (2016), pp. 1975–1987 (cit. on pp. 19, 32, 36, 38, 100, 105).

[Way] *Waymo Open Dataset: An autonomous driving dataset*. 2019.

[Wen+20] Xinshuo Weng et al. "Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics." In: *arXiv preprint arXiv:2008.08063* (2020) (cit. on p. 104).

[Wil12] Jason L Williams. "Hybrid Poisson and multi-Bernoulli filters." In: *2012 15th International Conference on Information Fusion*. IEEE. 2012, pp. 1103–1110.

[Wil15a] Jason L Williams. "An efficient, variational approximation of the best fitting multi-Bernoulli filter." In: *IEEE Transactions on Signal Processing* 63.1 (2015), pp. 258–273.

[Wil15b] Jason L Williams. "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA, and association-based MeMBer." In: *IEEE Transactions on Aerospace and Electronic Systems* 51 (2015), pp. 1664–1687 (cit. on pp. 19, 24, 38–40, 44).

[WL14] Jason Williams and Roslyn Lau. "Approximate evaluation of marginal association probabilities with belief propagation." In: *IEEE Transactions on Aerospace and Electronic Systems* 50 (2014), pp. 2942–2959.

[WÖ15] Niklas Wahlström and Emre Özkan. "Extended target tracking using Gaussian processes." In: *IEEE Transactions on Signal Processing* 63.16 (2015), pp. 4165–4178 (cit. on p. 50).

[Xia+18] Yuxuan Xia et al. "Extended Target Poisson Multi-Bernoulli Filter." In: *arXiv preprint arXiv:1801.01353* (2018).

[Xia+19] Yuxuan Xia et al. "Multi-scan implementation of the trajectory Poisson multi-Bernoulli mixture filter." In: *Journal of Advances in Information Fusion* 14.2 (2019), pp. 213–235 (cit. on p. 44).

[YLG14] Jin-long Yang, Peng Li, and Hong-wei Ge. "Extended target shape estimation by fitting B-spline curve." In: *Journal of Applied Mathematics* 2014 (2014).

[YZK21] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. "Center-based 3d object detection and tracking." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 11784–11793 (cit. on pp. 100, 102, 104, 105, 108).