GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Tracking and Fusion Methods for Extended Targets Parameterized by Center, Orientation, and Semi-axes

Dissertation
for the award of the degree
"Doctor rerum naturalium" (Dr.rer.nat.)
at the Georg-August University Göttingen

within the doctoral program Ph.D. Programme in Computer Science (PCS)
of the Georg-August University School of Science (GAUSS)

submitted by
Kolja Thormann from Göttingen, Germany

Göttingen, 2021

I hereby declare that I have written this thesis independently without any help from others and without the use of documents or aids other than those stated. I have mentioned all used sources and cited them correctly according to established academic citation rules.

Göttingen, 14. January 2022

Acknowledgment

I first want to thank my doctoral advisor Prof. Dr.-Ing. Marcus Baum for his support, feedback, insights, and many discussions. I also want to thank Prof. Dr. Jens Grabowski for supporting me in my advisory committee and providing advise over these past years, as well as Prof. Dr. Gustaf Hendeby for refereeing my thesis. Furthermore, I thank the other members of my examination board, Prof. Dr. Dieter Hogrefe, Prof. Dr. Florin Manea, and Prof. Dr. Bernhard Schmitzer for attending my thesis defense.

Furthermore, I want to thank my colleagues and fellow students for providing feedback, as well as insightful and fun discussions over the course of my time as a PhD student. Namely, I want to thank Jaya Shradha Fowdur, Dr. Hauke Kaulbersch, Fabian Sigges, Laura Wolf, and Dr. Shishan Yang.

Additionally, I give my thanks to Dr. Jens Honer for being a co-author on some of my papers and providing insightful discussions as well.

Finally, I want to thank my family and friends for their moral support and for being great people.

Abstract

The improvements in sensor technology, e.g., the development of automotive Radio Detection and Ranging (RADAR) or Light Detection and Ranging (LIDAR), which are able to provide a higher detail of the sensor's environment, have introduced new opportunities but also new challenges to target tracking. In classic target tracking, targets are assumed as points. However, this assumption is no longer valid if targets occupy more than one sensor resolution cell, creating the need for extended targets, modeling the shape in addition to the kinematic parameters. Different shape models are possible and this thesis focuses on an elliptical shape, parameterized with center, orientation, and semi-axes lengths. This parameterization can be used to model rectangles as well. Furthermore, this thesis is concerned with multi-sensor fusion for extended targets, which can be used to improve the target tracking by providing information gathered from different sensors or perspectives. We also consider estimation of extended targets, i.e., to account for uncertainties, the target is modeled by a probability density, so we need to find a so-called point estimate.

Extended target tracking provides a variety of challenges due to the spatial extent, which need to be handled, even for basic shapes like ellipses and rectangles. Among these challenges are the choice of the target model, e.g., how the measurements are distributed across the shape. Additional challenges arise for sensor fusion, as it is unclear how to best consider the geometric properties when combining two extended targets. Finally, the extent needs to be involved in the estimation.

Traditional methods often use simple uniform distributions across the shape, which do not properly portray reality, while more complex methods require the use of optimization techniques or large amounts of data. In addition, for traditional estimation, metrics such as the Euclidean distance between state vectors are used. However, they might no longer be valid because they do not consider the geometric properties of the targets' shapes, e.g., rotating an ellipse by 180 degree results in the same ellipse, but the Euclidean distance between them is not 0. In multi-sensor fusion, the same holds, i.e., simply combining the corresponding elements of the state vectors can lead to counter-intuitive fusion results.

In this work, we compare different elliptic trackers and discuss more complex measurement distributions across the shape's surface or contour. Furthermore, we discuss the problems which can occur when fusing extended target estimates from different sensors and how to handle them by providing a transformation into a special density. We then proceed to discuss how a different metric, namely the Gaussian Wasserstein (GW) distance, can be used to improve target estimation. We define an estimator and propose an approximation based on an extension of the square root distance. It can be applied on the posterior densities of the aforementioned trackers to incorporate the unique properties of ellipses in the estimation process. We also discuss how this can be applied to rectangular targets as well. Finally, we evaluate and discuss our approaches. We show the benefits of more complex target models in simulations and on real data and we demonstrate our estimation and fusion approaches compared to classic methods on simulated data.

Contents

List of Figures

List of Tables

List of Abbreviations

CT	Coordinated Turn	17
DW-C	Discretized Wasserstein - Contour	64
DW-S	Discretized Wasserstein - Surface	66
EKF	Extended Kalman Filter	7
Ellipse-RHM-EKF	Ellipse Random Hypersurface Model Extended Kalman Filter	28
Ellipse-RHM-EKF-imp	Ellipse Random Hypersurface Model Extended Kalman Filter Implicit	28
EM	Expectation Maximization	24
EOT	Extended Object Tracking	1
ESR	Extended Square Root	57
ETT	Extended Target Tracking	1
FMCW	Frequency-Modulated Continuous-Wave	25
GM	Gaussian Mixture	5
GM-MEM-EKF*	Gaussian Mixture Multiplicative Error Model Extended Kalman Filter*	6
GP	Gaussian Process	24
GW	Gaussian Wasserstein	6
HTG-RM	Hierarchical Truncated Gaussian Random Matrix	75
IAE	Independent Axis Estimation	23
IEKF	Iterative Extended Kalman Filter	23
IOU	Intersection-over-Union	14
LIDAR	Light Detection and Ranging	1
MC	Monte Carlo	59
MC-MMGW	Monte Carlo Minimum Mean Gaussian Wasserstein	63
MEM	Multiplicative Error Model	31
MEM-EKF*	Multiplicative Error Model Extended Kalman Filter*	6

MEM-EKF*-D	Multiplicative Error Model Extended Kalman Filter* Doppler	6
MMDW	Minimum Mean Discretized Wasserstein	64
MMDW-C	Minimum Mean Discretized Wasserstein - Contour	64
MMDW-S	Minimum Mean Discretized Wasserstein - Surface	66
MMGW	Minimum Mean Gaussian Wasserstein	6
MMOSPA	Minimum Mean Optimal Sub-Pattern Assignment	50
MMSE	Minimum Mean Square Error	51
NCV	Nearly Constant Velocity	17
OSPA	Optimal Sub-Pattern Assignment	14
PDA	Probabilistic Data Association	25
PMHT	Probabilistic Multi-Hypothesis Tracker	25
RADAR	Radio Detection and Ranging	1
RED	Random Ellipse Density	6
RED-MMGW	Random Ellipse Density Minimum Mean Gaussian Wasserstein	62
RFS	Random Finite Set	54
RHM	Random Hypersurface Model	21
RM	Random Matrix	5
RM-D	Random Matrix Doppler	72
UKF	Unscented Kalman Filter	7

Nomenclature

a	Scalar
$\mathbf{a}$	Column vector
$\mathbf{A}$	Matrix
$\mathbb{X}$	Set
$a \mod (b)$	Modulo operation of dividing a by b
$\measuredangle(\mathbf{a})$	Angle of a 2D input vector $\mathbf{a} \in \mathbb{R}^2$
$\cos(a)$	Cosine of a
$\sin(a)$	Sine of a
$\cosh(a)$	Hyperbolic cosine of a
$\sinh(a)$	Hyperbolic sine of a
$\lvert a \rvert$	Absolute value of a
$\lvert\lvert \mathbf{a} \rvert\rvert_2$	Euclidean norm of $\mathbf{a}$
$\mathbf{a}^{\mathrm{T}}$	$\mathbf{a}$ transposed
$\mathbf{ab}$	Dot product between $\mathbf{a}$ and $\mathbf{b}$
$\mathbf{ab}^{\mathrm{T}}$	Outer product between $\mathbf{a}$ and $\mathbf{b}$

$\mathbf{a} \otimes \mathbf{b}$	Kronecker product between $\mathbf{a}$ and $\mathbf{b}$
$\mathbf{a} \times \mathbf{b}$	Cross product between $\mathbf{a}$ and $\mathbf{b}$
$\mathbf{0}_{a \times b}$	A matrix with a rows and b columns and each element as 0
$\mathbf{I}_a$	Identity matrix with a rows and columns
diag($\mathbf{a}$)	Diagonal matrix with main diagonal consisting of the elements of $\mathbf{a}$
$\mathbf{A}^{\frac{1}{2}}$	Matrix square root, i.e., $\mathbf{A} = \mathbf{A}^{\frac{1}{2}} \left(\mathbf{A}^{\frac{1}{2}}\right)^{\mathrm{T}}$
tr($\mathbf{A}$)	Trace of matrix $\mathbf{A}$
det($\mathbf{A}$)	Determinant of matrix $\mathbf{A}$
$\mathbb{Z}$	Set of integers
$\mathbb{R}$	Set of real numbers
$[a, b]$	Interval between a and b, including both
$[a, b[$	Interval between a and b, excluding b
$]a, b]$	Interval between a and b, excluding a
$]a, b[$	Interval between a and b, excluding both
$\{a \ldots b\}$	Set of integers from a to b
$p(a)$	Probability density function of a
$p(a\|b)$	Probability density function of a given b
$\hat{a}$	Mean of random variable a
σ_a^2	Variance of random variable a
$\mathbf{C}^{\mathbf{a}}$	Covariance matrix of random vector $\mathbf{a}$
$\mathcal{N}(\hat{a}, \sigma_a^2)$	Gaussian distribution with mean $\hat{a}$ and variance σ_a^2
$\mathcal{U}(a, b)$	Uniform distribution between a and b
E$[a]$	Expectation of a random variable a
$\delta(a)$	Delta function producing 1 for input 0 and 0 otherwise

T	Time difference for prediction step
e	Euler′s number
π	Pi
m	meter

1 Introduction

Contents

1.1 Motivation

A topic gaining more relevance over the past years concerns the challenges and opportunities involving improvements in sensor technology. Classic point target tracking assumes that sensors like Radio Detection and Ranging (RADAR) are tracking far away targets, which generate only a single measurement per time step originating from a single point, i.e., the target extent is smaller than a sensor resolution cell. Now, due to sensors with higher resolution tracking targets in closer range, e.g., automotive RADAR or Light Detection and Ranging (LIDAR), this point assumption is no longer valid. Instead, a target can generate multiple detection points per time step scattered across its contour or surface. In this case, not just the kinematic parameters, but also the shape of the target need to be considered and estimated, see Figure 1.1 and Figure 1.2. Such targets are called extended targets and the problem is named Extended Target Tracking (ETT) or Extended Object Tracking (EOT) [MCS$^+$14, GBR17]. The goal of ETT is to find a shape model which fits the data and to derive an appropriate measurement equation which can be used for iterative tracking over several time steps.

Different sensors can be used for tracking, but we focus on RADAR sensors. While LIDAR sensors produce clearly visible edges and camera provides color images, they are both susceptible to certain weather conditions such as heavy rain. RADAR sensors in turn can handle those challenges

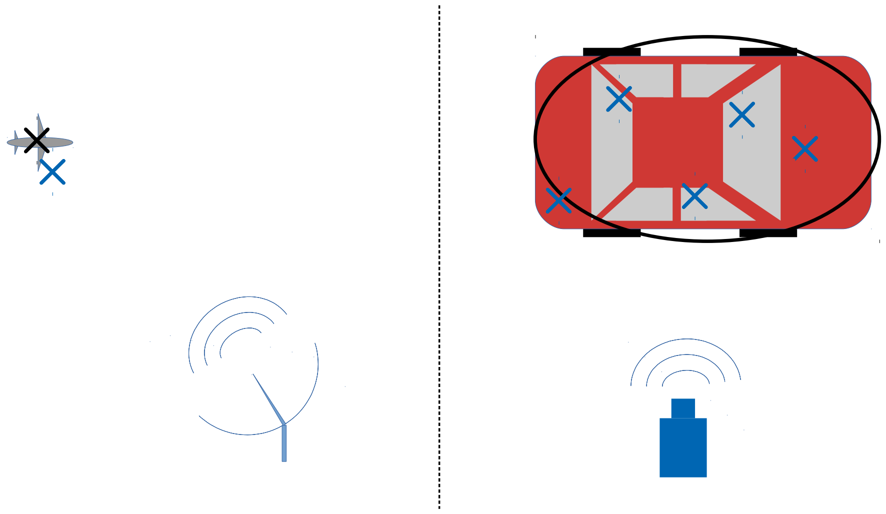


Figure 1.1: Comparison of point target tracking (left) and ETT (right). Measurements are blue crosses. The estimated targets are a black cross and an ellipse respectively.

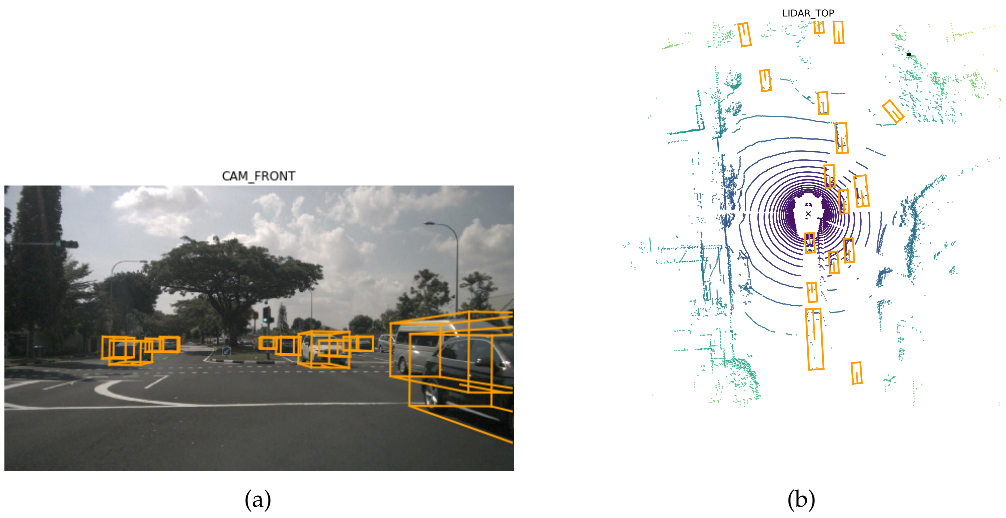


Figure 1.2: Example from the nuScenes dataset [CBL+20], showing a camera view with orange bounding boxes in Figure 1.2a and the corresponding LIDAR point cloud with bounding boxes in top down view in Figure 1.2b

and can produce range rate measurements, i.e., the velocity in measurement direction. They still have their own challenges, often producing noisier measurements, scattered across the target, as can be seen in, e.g., the nuScenes dataset [CBL+20]. Often times basic shapes like an ellipse or a rectangle are used. However, there are several different algorithms using such shapes.

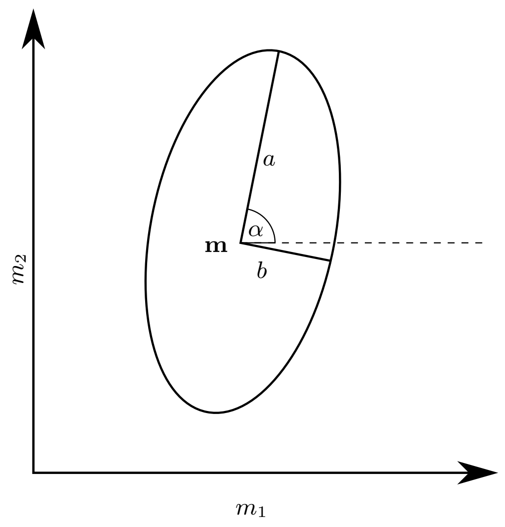


Figure 1.3: Ellipse parameterized with a center $\mathbf{m} = [m_1 \quad m_2]^{\mathrm{T}}$, an orientation α, and semi-axes lengths a and b.

This thesis focuses on those methods parameterizing an ellipse (or a rectangle) with a center $\mathbf{m} = \begin{bmatrix} m_1 & m_2 \end{bmatrix}^{\mathrm{T}}$, an orientation α, and semi-axes lengths a and b, see Figure 1.3, as it is intuitive to track the uncertainties of the different elements of the shape state. For example, a car usually does not change its dimensions, but it can rotate, so process noise on the semi-axes lengths is kept to a minimum while the noise on the orientation can be set higher. There are several different options for trackers with such a parameterization, so they will be compared in simulations in this thesis. Furthermore, this thesis discusses the usage of the range rate measurement in these approaches and evaluates it in simulations as well. In real automotive data, the detection points are often biased towards the visible side, which is called self-occlusion, or stem mainly from certain reflection areas such as the wheels. Therefore, we investigate and develop methods which apply the aforementioned parameterization but use a more complex measurement distribution across the surface as well.

Independent of the automotive case, ETT and this parameterization in particular lead to interesting aspects which need to be considered in estimation and fusion. Multi-sensor fusion is one approach to improve the quality of the estimation by incorporating different sensor estimates which can provide different information. As an example, the aforementioned self-occlusion can lead to uncertainty in the occluded parts, which can be solved by involving estimates from sensors placed at different angles around the target. Figure 1.4 shows a scenario in which an elliptic target is tracked by two sensors. The red sensor sees the target from the side and thus, it has a good estimate of the major axis length as well as the m_1 position. Meanwhile, the blue sensor sees it from the front and has a good estimate of the minor axis length and the m_2 position. If the sensors also

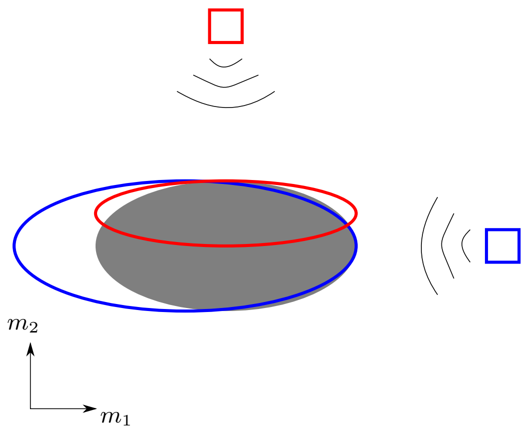


Figure 1.4: Example of two sensors in red and blue, tracking an ellipse. The ground truth is in gray and the sensor estimates are the ellipses in the respective color of the sensor.

provide covariances properly reflecting these uncertainties, an estimate close to the ground truth can be obtained. However, this can only be achieved if the semi-axes lengths (and the orientation) are used to model the shape, so that their individual uncertainties are available and they can be fused directly. A challenge in the fusion process using this parameterization are ambiguities, e.g., rotating an ellipse by $180°$ results in the same ellipse, but in a different state vector. Classic fusion approaches can therefore produce counter-intuitive results.

These ambiguities lead us to another challenge in ETT, which is the estimation. In Bayesian estimation (and tracking), the target state is modeled as a probability density to account for uncertainties stemming from, e.g., uncertain prior knowledge, as well as process and measurement noise. Therefore, after updating the density with measurements or after fusing it with sensor estimates, we end up with a posterior density. In practice, we often do not want an entire density as the output of the tracker, but instead a specific point in that density, a so-called point estimate (not to be confused with point targets). In the context of a density describing elliptic targets, the point estimate would be a specific ellipse. It is usually gained by calculating the expectation of the density. To do so, we integrate over the density and find the value which minimizes the average distance to it. The squared Euclidean distance between the state vectors is generally used as the distance measure. This approach is straightforward, but it does not consider the random variable as a representation of a shape, ignoring geometric properties of the state, i.e., the aforementioned ambiguities.

We will highlight conditions under which classic fusion and estimation would fail and focus on how to solve these fusion and estimation problems mainly considering elliptic targets, but comment on the applicability to rectangular targets as well.

1.2 Considered Problems and Research Questions

This thesis is based in the field of ETT. The focus of this work is particularly on elliptic extended targets. However, the applicability of the methods developed in this thesis on other shapes will be discussed as well. The considered problems and research questions are described in this section.

1.2.1 Tracking using Doppler RADAR

Several different approaches exist for tracking extended targets, which will be discussed in more detail in Chapter 3. For this thesis, we will focus on RADAR data and its unique properties, i.e., detections on the surface and the range rate measurement. The latter describes the velocity of the measured point in the direction of the measurement and is determined via the Doppler effect. While the incorporation of the range rate can introduce greater non-linearities, it provides information about the target's velocity. We compare elliptic ETT methods and discuss how they can be extended via the range rate measurement. Therefore, the first research question is:

RQ 1 How can elliptic extended target trackers, parameterized with orientation and semi-axes lengths and based on the standard spatial distribution model, be extended by the range rate measurement and what are its benefits?

However, these classic approaches assume the measurements to be uniformly distributed across the entire surface, while in reality, they are often biased towards the sensor or originate from certain reflectors, e.g., the wheels of a car. The current state of the art captures these more complex distributions using a Gaussian Mixture (GM) density, e.g., [SD18, KHB19, HK20] or, in case of the Random Matrix (RM) model, truncates the density modeling the shape [XWB$^+$21]. Our focus is on the shape parameterization via orientation and semi-axes lengths, which can be used in conjunction with the GM model. However, the aforementioned trackers using GM apply, e.g., optimization techniques or a particle filter. Therefore, our next research question is:

RQ 2 Can a closed-form measurement update be derived for a measurement model using additive sensor noise and a GM based measurement source distribution on an extended target?

1.2.2 Fusion and Estimation

Fusion of extended target state densities in multi-sensor scenarios can be more complex than fusion of point target densities due to the ambiguities in the shape parameterization. This introduces challenges for fusing ellipse (or rectangle) state densities parameterized with orientation and semi-axes lengths, which we will investigate. The corresponding research question is:

RQ 3 How can state densities of ellipses be fused with respect to their geometric properties?

The aforementioned challenges stemming from geometric properties of shape densities do occur in the estimation process as well, to be specific, in the metric which determines the cost function, i.e., the Euclidean distance. Rotating an ellipse by $180°$ results in the same ellipse, but the Euclidean distance between the state vectors is not 0. Therefore, the final research question is:

RQ 4 How can an estimator be derived which considers the geometric properties of ellipses?

1.3 Contributions

The contributions of this thesis are based on the publications [1–5] (other unrelated publications are [6–9]) and refer to the four research questions from Section 1.2. They can be summed up as

- a discussion and implementation of range rate usage in single ETT methods, which parameterize an elliptical shape with orientation and semi-axes lengths, including a discussion and comparison of state of the art methods in ETT using this parameterization,
- a closed-form measurement update for elliptic (or rectangular) shapes based on multiplicative noise modeled by a GM, the Gaussian Mixture Multiplicative Error Model Extended Kalman Filter* (GM-MEM-EKF*), along with an evaluation on real automotive RADAR data,
- a novel probability density for elliptical targets, a so-called Random Ellipse Density (RED), to handle challenges introduced by the geometric properties of ellipses in sensor fusion,
- an estimator on ellipse densities replacing the Euclidean distance with the Gaussian Wasserstein (GW) distance [GS84] as the metric in the cost function, the Minimum Mean Gaussian Wasserstein (MMGW) estimator, and an approximation of it using an extension of the square root distance [DKZ09] which can be calculated using particles.

1.4 Structure of this Thesis

The remainder of this thesis is structured as follows. In Chapter 2, the foundations for this thesis are laid. Next, elliptical extended target trackers parameterizing the ellipse with orientation and semi-axes lengths are discussed in Chapter 3. In this chapter, we also discuss the usage of the range rate, including the introduction of the Multiplicative Error Model Extended Kalman Filter* Doppler (MEM-EKF*-D) to incorporate the range rate measurement into the Multiplicative Error Model Extended Kalman Filter* (MEM-EKF*) [YB19]. The GM-MEM-EKF* to handle more complex distributions across the shape's surface is then presented in Chapter 4. After that, the fusion and estimation on state densities describing elliptic extended targets are handled in Chapter 5, introducing the RED and the MMGW estimator. The trackers and fusion and estimation methods from the previous three chapters are then evaluated in Chapter 6 and discussed in Chapter 7. This thesis is concluded and future work is proposed in Chapter 8.

2

Foundations for Extended Target Tracking

Contents

This chapter lays the foundation of the thesis by presenting the basic concept of filtering in Section 2.1. Special considerations necessary for ETT are then elaborated in Section 2.2 including extended target metrics. While the discussed ETT methods generally assume Gaussian densities, we require some special densities later on, which are discussed in Section 2.3. Dynamic models for the prediction part of the tracking process are presented in Section 2.4.

2.1 Filtering Methodology

The goal of filtering is to estimate a desired state based on observations. This thesis considers Bayesian filtering, which is explained in Section 2.1.1. For the special case of a linear dependency between the state and measurements as well as all densities being Gaussian, the Kalman filter is the optimal Bayesian filter and is presented in Section 2.1.2. If linearity is not given, extensions of the Kalman filter exist, which will be discussed afterwards, namely the Extended Kalman Filter (EKF) in Section 2.1.3 and the Unscented Kalman Filter (UKF) in Section 2.1.4. A discussion on fusing estimates from different sensors is provided in Section 2.1.5.

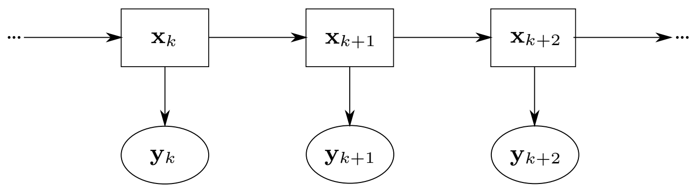


Figure 2.1: Hidden Markov model showing the evolution of the hidden state $\mathbf{x}_k$ over time and generating measurements $\mathbf{y}_k$.

2.1.1 Bayesian Filtering

The goal in Bayesian filtering [BSLK04] is to track a state $\mathbf{x}_k$ over time steps k. The state itself is unknown and not directly observable. Instead, a measurement $\mathbf{y}_k$ is received at time step k. To account for uncertainties, we model the state as a random variable with probability density function $p(\mathbf{x}_k)$. The tracking consists of two steps: predicting the state to future time steps and combining the state with measurements $\mathbf{y}_k$ observed at those time steps. To project the state to the next time step $k+1$, a transition model is used, described by a function

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{w}_k) \ , \tag{2.1}$$

which is corrupted by process noise $\mathbf{w}_k$. A measurement is generated from the state via a measurement function

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k) \ , \tag{2.2}$$

which is corrupted by sensor noise $\mathbf{v}_k$. The state evolving over time and generating measurements can be represented by a hidden Markov model as in Figure 2.1. We have the Markov property

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}\dots\mathbf{x}_0) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \ , \tag{2.3}$$

$$p(\mathbf{y}_k|\mathbf{x}_k\dots\mathbf{x}_0) \quad = p(\mathbf{y}_k|\mathbf{x}_k) \ , \tag{2.4}$$

i.e., the current state only depends on the previous state and the current measurement only depends on the current state.

To account for our initial knowledge or lack thereof about the state, we define a prior $p(\mathbf{x}_0)$ at time step 0. The lesser information is known beforehand, the more spread out the prior density can be chosen, i.e., if no prior information is available, the prior can be modeled by a uniform distribution across the entire state space.

Based on the transition function (2.1), we derive a transition density $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ to model the state changing over time and being influenced by process noise. Note that the transition density only depends on the current state and the state depends on all previous measurements. We get

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k) \cdot p(\mathbf{x}_k|\mathbf{y}_{1:k})\mathrm{d}\mathbf{x}_k \ , \tag{2.5}$$

by applying the Chapman-Kolmogorov equation. We denote $\mathbf{y}_{1:k}$ as all measurements from time steps 1 to k. This step is named prediction or time update.

We can relate the state and measurement via a likelihood function $p(\mathbf{y}_k|\mathbf{x}_k)$ based on the measurement function 2.2. Once measurement $\mathbf{y}_k$ is received, we want to fuse the state and measurement to calculate the so-called posterior density using Bayes formula. We multiply the likelihood with the prior density $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ at time step k,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \sim p(\mathbf{y}_k|\mathbf{x}_k) \cdot p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \ . \tag{2.6}$$

This is called the update or measurement update step.

After the update, we have a posterior density. However, we usually want to provide a specific realization of the state, not a density, as output. Therefore, we calculate a so-called point estimate using the squared Euclidean distance as a cost function

$$\hat{\mathbf{z}}_k = \underset{\mathbf{z}}{\operatorname{argmin}} \int ||\mathbf{z} - \mathbf{x}_k||_2^2 \cdot p(\mathbf{x}_k|\mathbf{y}_{1:k})\mathrm{d}\mathbf{x}_k \ . \tag{2.7}$$

2.1.2 Kalman Filter

The challenge in Bayesian tracking is that the calculation of the posterior (2.6) and the integrals in (2.7) and (2.5) require full knowledge of the likelihood, a closed-form solution does not necessarily exist, and the calculations might not be computationally feasible. The Kalman filter [Kal60, KB61, BSLK04] provides closed-form formulas and is the optimal filter if two conditions are met, the relation of the measurement and state and the relation of the state and future state need to be linear and all densities need to be Gaussian.

To ensure linearity, we define the measurement and transition functions (2.2) and (2.1) as

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \ , \tag{2.8}$$

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k \ , \tag{2.9}$$

with additive measurement noise $\mathbf{v}_k$, process noise $\mathbf{w}_k$, measurement matrix $\mathbf{H}$, and process matrix $\mathbf{F}$. For the Gaussian condition to hold, we define the prior as a Gaussian $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{C}_0^{\mathbf{x}})$. We define the noises as Gaussians $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_k^{\mathbf{v}})$ and $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_k^{\mathbf{w}})$, as well, which are usually assumed to be zero-mean and independent to each other and the state [BSLK04]. Given non zero-mean noise, the mean can be incorporated into the tracking procedure if it is known.

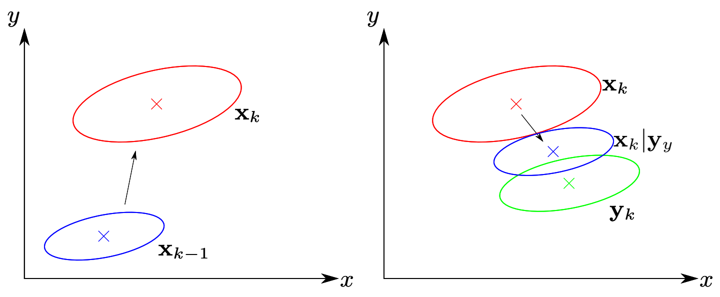


Figure 2.2: Kalman filter prediction and update step with state $\mathbf{x}_k$ being predicted in the left figure and updated using the measurement $\mathbf{y}_k$ in the right figure, resulting in a posterior $\mathbf{x}_k|\mathbf{y}_k$.

As all densities are Gaussian and they are combined in a linear fashion, the posterior will again be Gaussian, enabling an iterative tracking approach. Additionally, the point estimate (2.7) can simply be calculated by taking the posterior density's mean.

To differentiate the state before and after the measurement update, we use the notation $\mathbf{x}_{k|k-1}$ and $\mathbf{x}_{k|k}$ as the state at time k incorporating measurements up to time $k-1$ and up to time k respectively. The update is conducted using the formulas

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \ , && (2.10)\\
\mathbf{C}^{\mathbf{x}}_{k|k} &= \mathbf{C}^{\mathbf{x}}_{k|k-1} - \mathbf{K}_k\mathbf{C}^{\mathbf{y}}_k\mathbf{K}_k^{\mathrm{T}} \ , && (2.11)\\
\mathbf{K}_k &= \mathbf{C}^{\mathbf{x}}_{k|k-1}\mathbf{H}^{\mathrm{T}}(\mathbf{C}^{\mathbf{y}}_k)^{-1} \ , && (2.12)\\
\mathbf{C}^{\mathbf{y}}_k &= \mathbf{H}\mathbf{C}^{\mathbf{x}}_{k|k-1}\mathbf{H}^{\mathrm{T}} + \mathbf{C}^{\mathbf{v}}_k \ . && (2.13)
\end{aligned}$$

We name $\nu = \mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}$ the innovation.

To transition the state into the next time step in the prediction step, the Kalman filter provides the formulas

$$\begin{aligned}
\hat{\mathbf{x}}_{k+1|k} &= \mathbf{F}\hat{\mathbf{x}}_{k|k} \ , && (2.14)\\
\mathbf{C}^{\mathbf{x}}_{k+1|k} &= \mathbf{F}\mathbf{C}^{\mathbf{x}}_{k|k}\mathbf{F}^{\mathrm{T}} + \mathbf{C}^{\mathbf{w}}_k \ . && (2.15)
\end{aligned}$$

In summary, the two step process consists of a prediction step, which usually increases the covariance, i.e., the uncertainty of the estimate due to process noise and an update step, which combines prior and likelihood using Bayes formula to improve the estimate, usually decreasing the uncertainty. This is visualized in Figure 2.2.

2.1.3 Extended Kalman Filter

A disadvantage of the Kalman filter is the condition that the measurement and process model need to be linear. A modification of the Kalman filter able to deal with non-linear systems using linearization is provided by the EKF [BSFC90] formulas. The general assumption is that we have non-linear functions with additive noise

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k \ , \tag{2.16}$$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k \ . \tag{2.17}$$

The EKF uses Taylor-series expansion to linearize the functions around the current estimate as the linearization point, calculating the Jacobians

$$\tilde{\mathbf{H}}_k = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}_{k|k-1}} \ , \tag{2.18}$$

$$\tilde{\mathbf{F}}_k = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}_{k|k}} \ . \tag{2.19}$$

We then conduct the update and prediction as above, replacing $\mathbf{H}\hat{\mathbf{x}}_{k|k-1}$ with $h(\hat{\mathbf{x}}_{k|k-1})$ and $\mathbf{F}\hat{\mathbf{x}}_{k|k}$ with $f(\hat{\mathbf{x}}_{k|k})$ to calculate the innovation and the mean of the predicted state. Every other occurrence of $\mathbf{H}$ and $\mathbf{F}$ is replaced with the respective Jacobians $\tilde{\mathbf{H}}_k$ and $\tilde{\mathbf{F}}_k$.

The downsides of an EKF are the possibility of divergence in strong non-linearities, the dependence on the prior as a linearization point, the often large and complex Jacobians being a source for coding errors, and the potential loss of optimality.

2.1.4 Unscented Kalman Filter

Another option to handle non-linearities is the UKF [WVDM00,JU04]. The basic idea is to represent the densities as samples. This is similar to a particle filter [GSS93]. However, while a particle filter needs to draw a large number of samples, the UKF provides a solution to sample in a smart way and keep the number of samples low. It deterministically draws a small number of so-called sigma points, which are chosen in a way to capture the density up to its second order properties with a minimum number of points. After we calculate the sigma points, we transform them using the non-linear function. Then, we use the transformed points to approximate the transformed density.

During the prediction, we choose N sigma points $\mathbf{x}_{k|k}^{(i)}$ with $i \in \{1 \dots N\}$. N is set as $2d+1$ with state dimension d, i.e., $\mathbf{x}_{k|k} \in \mathbb{R}^d$. The sigma points are drawn based on the mean and the columns of a square root decomposition of the covariance matrix. The weights $w_{k|k}^{(i)}$ are determined under the condition

$$\sum_{i=1}^{N} w_{k|k}^{(i)} = 1 \ . \tag{2.20}$$

Each sigma point is transformed using

$$\mathbf{x}^{(i)}_{k+1|k} = f(\mathbf{x}^{(i)}_{k|k}) \ . \tag{2.21}$$

After each sigma point is transformed, the predicted state density can be approximated as

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=1}^{N} w^{(i)}_{k|k} \mathbf{x}^{(i)}_{k+1|k} \ , \tag{2.22}$$

$$\mathbf{C}^{\mathbf{x}}_{k+1|k} = \sum_{i=1}^{N} w^{(i)}_{k|k} (\mathbf{x}^{(i)}_{k+1|k} - \hat{\mathbf{x}}_{k+1|k})(\mathbf{x}^{(i)}_{k+1|k} - \hat{\mathbf{x}}_{k+1|k})^{\mathrm{T}} + \mathbf{C}^{\mathbf{w}}_{k} \ , \tag{2.23}$$

with $\mathbf{C}^{\mathbf{w}}_{k}$ being the covariance of the zero-mean process noise. The measurement update is conducted analogously. For more details, see [WVDM00, JU04].

2.1.5 Multi-Sensor Track to Track Fusion

One way to improve tracking is the combination of the information from multiple sensors. Knowledge can be shared among neighboring sensors in a sensor network, e.g., [LJMD15] or it can be combined in a central fusion unit [DWH08]. One has to consider the amount of data shared as well. One way to reduce communication is to share the state densities instead of the entire point clouds [DWH08]. The estimates must then be fused either locally at the sensor or in the central fusion unit. As different sensors have different quality and can detect different parts of the target, it is important to consider the uncertainty in the fusion, e.g., [GK12], in other words, not just the point estimates, but the entire state densities need to be fused.

In general, we assume n sensors and for each a local state $p(\mathbf{x}_s|\mathbf{y}_s)$ depending on the local measurement $\mathbf{y}_s$ with $s \in \{1 \dots n\}$. Each state is modeled as a Gaussian, i.e., $\mathbf{x}_s \sim \mathcal{N}(\hat{\mathbf{x}}_s, \mathbf{C}^{\mathbf{x}}_s)$. We now want to combine the states to get a global estimate such that $p(\mathbf{x}|\{\mathbf{y}_1, \dots, \mathbf{y}_s\})$ [GK12]. Different strategies exist like fusing sensor estimates directly, for which they would have to be synchronized, or fuse them with a global estimate [ASKB12]. The easiest way to do so is to fuse the local sensor estimates with a global estimate as if they were measurements in a Kalman filter. This can pose problems if the sensor estimates are correlated. A correlation can occur due to a common process model or common prior. If the correlations are not considered properly, the covariance can be underestimated, e.g., it is reduced by the information it already had [CAM02]. To handle unknown correlations, covariance intersection can be used, e.g., [JU97, CAM02, RNAH15], which estimates the fused covariance as one encompassing the intersection of the covariances of the densities which are fused. Another solution providing less conservative results for the fused covariance is the inverse-covariance intersection [NSRH17, NSH17]. If the information form of the Kalman filter is used, the redundant information can be subtracted [Mut98, ASKB12]. Alternatively, known correlations from, e.g., a common dynamic model, can be tracked [DWH08, BS81], with more recent approaches suggesting to track the correlation via square root decomposition [RNH19, RNH20].

2.2 Extended Target Tracking

Unlike point targets, extended targets are assumed to be bigger than a single sensor resolution cell. As a result, an extended target can generate multiple spatially distributed measurements per time step, distributed across its surface [GBR17]. This principle is introduced in Section 2.2.1. It is necessary to correctly model the shape as well as how the measurements are distributed across that shape to properly track extended targets. Due to the varying geometric properties of the different shapes, we need to make special consideration to find suitable metrics that incorporate those geometric properties, which is discussed in Section 2.2.2.

2.2.1 Extended Targets

In ETT, we assume a target can generate multiple measurements, i.e., at time step k we get $n_k \geq 0$ measurements $\mathbf{y}_k^{(i)}$ with $i \in \{1 \dots n_k\}$. A common assumption is that n_k is Poisson distributed [GBR17]. We further assume that the target is not a point, but possesses an extent from which measurements are generated. This mean the target state consists of the position, the kinematic parts, and the extent [GBR17]. [GBR17] describe three levels of complexity to model the extent. It can not be modeled at all, i.e., we still assume a point, it can be modeled by a simple shape like an ellipse or rectangle, and it can be modeled by a more complex shape, e.g., a star-convex one. Another differentiation is given by a model's assumption of the measurement distribution on the shape. [GBR17] differentiate two settings, measurements from the boundary of the shape or measurements from the entire surface. Another option would be to assume specific reflection points as in, e.g., [HSSS12, BWBS+17].

In the standard spatial distribution model [GS05], we assume that a measurement $\mathbf{y}_k^{(i)}$ originated from a measurement source $\mathbf{z}_k^{(i)}$, which in turn is drawn from a density $p(\mathbf{z}_k^{(i)}|\mathbf{x}_k)$ modeling the distribution of measurement sources across the extent of the target $\mathbf{x}_k$. To get our likelihood, we need to integrate over all measurement sources

$$p(\mathbf{y}_k^{(i)}|\mathbf{x}_k) = \int p(\mathbf{y}_k^{(i)}|\mathbf{z}_k^{(i)})p(\mathbf{z}_k^{(i)}|\mathbf{x}_k)\mathrm{d}\mathbf{z}_k^{(i)} \quad . \tag{2.24}$$

While some approaches try to model that distribution to identify specific measurement sources for each measurement, e.g., [BH14, WÖ15, HSRD16, TLTK19, KBW17], others apply different models, e.g., utilizing the mean and spread of the measurement point clouds as in the RM approach [Koc08, FFK11]. An overview of ETT models can be found in [MCS+14, GBR17].

2.2.2 Extended Target Metrics

Metrics for extended targets are an ongoing research topic. For point targets, it is often enough to simply use the Euclidean distance between the state vectors to model the error as it fully covers the distance between two points. However, for extended targets, the direct comparison of individual state parameters can be counter-intuitive. Depending on the parameterization, different state vectors can describe the same shape, even though their Euclidean distance is not zero. Additionally, it is unclear how a difference in shape should be punished, i.e., if two different but overlapping shapes are better or worse than two similar, but not overlapping shapes or if a difference in orientation should be punished more or less than a difference in semi-axes lengths.

A metric considering the actual area of the estimated shape is Intersection-over-Union (IOU) [GLO11], which is often used in computer vision. The metric divides the overlapping area by the area of the two shapes' union, resulting in a single value between 0 and 1. The advantage of this metric is the incorporation of the actual area. The downsides are, however, that all non-overlapping shapes produce a value of 0, independent of the differences in their extents. Additionally, calculating the area of arbitrary shapes is not always simple and computationally feasible. Another metric is the Hausdorff distance, e.g., [VH00], which does not consider the entire area, but instead finds the point on one shape which has the maximum minimal distance to a point on the other shape. This metric works for non-overlapping shapes but is harder to interpret and can result in counter-intuitive behavior, e.g., the Hausdorff distance of a circle to the same circle with one spike is the same as the distance of a circle to a circle with several similar spikes.

A different approach is to model the shapes as probability densities and to then apply a metric for measuring the difference between probability densities on them. A straightforward approach is to use a uniform distribution across the shape's surface. Given two shapes described by the densities $f(\mathbf{z})$ and $g(\mathbf{y})$, the p-Wasserstein distance can be used to describe their difference [HM02]

$$W_p(f,g) = \Big(\inf_h \int ||\mathbf{z}-\mathbf{y}||_2^p h(\mathbf{z},\mathbf{y}) \mathrm{d}\mathbf{z}\mathrm{d}\mathbf{y}\Big)^{\frac{1}{p}} \quad , \tag{2.25}$$

with $h(\mathbf{z},\mathbf{y})$ describing any joint density with marginals f and g. Depending on the shape, the calculation of the integral might be infeasible. As an approximation, the continuous densities can be replaced with empirical densities $f_n(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^n \delta(\mathbf{z}_i - \mathbf{z})$ and $g_m(\mathbf{y})$ analogously, to get the form [HM02]

$$W_p(f,g) = \Big(\inf_{\mathbf{C}} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{C}_{ij}||\mathbf{z}_i - \mathbf{y}_j||_2^p\Big)^{\frac{1}{p}} \quad , \tag{2.26}$$

with transport matrix $\mathbf{C}$ so that $\sum_{i=1}^n \mathbf{C}_{ij} = \frac{1}{m}$ and $\sum_{j=1}^m \mathbf{C}_{ij} = \frac{1}{n}$. If we have $m = n$ in Equation (2.26), the formula can be simplified to the Optimal Sub-Pattern Assignment (OSPA) distance [SVV08] between the discrete densities

$$W_p(f,g) = \min_{\pi\in\Pi_n}\Big(\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{z}_i - \mathbf{y}_{\pi(i)}||_2^p\Big)^{\frac{1}{p}} \quad , \tag{2.27}$$

with Π_n being the set of all permutations of the numbers 1 to n. This metric provides a straight-forward scalar value, but depending on the shape, it can be difficult to find a proper distribution of the points on the surface. Its easier to only use the contour, but then the weight of the density would only be on the contour as well.

A covariance matrix can be visualized as an ellipse in 2D. Therefore, elliptic shapes can be modeled as 2D Gaussian densities, using the center as the mean and reconstructing the covariance matrix from the orientation and semi-axes lengths of the ellipse. One metric to model the difference between Gaussians is the Kullback-Leibler divergence, see, e.g., [YBG16]. It measures the information lost in the approximation of one density with the other. As the Kullback-Leibler divergence is not a metric because it lacks symmetry, it can be turned into one by taking the sum of it in both directions. A different metric for Gaussian densities is the GW distance [GS84]

$$\text{GW}(\mathbf{m_z}, \mathbf{Z}, \mathbf{m_y}, \mathbf{Y}) = \sqrt{||\mathbf{m_z} - \mathbf{m_y}||_2^2 + \text{Tr}[\mathbf{Z} + \mathbf{Y} - 2(\mathbf{Z}^{\frac{1}{2}}\mathbf{Y}\mathbf{Z}^{\frac{1}{2}})^{\frac{1}{2}}]} \ , \tag{2.28}$$

with mean $\mathbf{m_y}$ and covariance $\mathbf{Y}$ of the Gaussian describing the ellipse $\mathbf{y}$ and $\mathbf{z}$ analogously. Unlike the Wasserstein distance based on the discretization of a uniform distribution as described above, the Wasserstein distance between two Gaussians can be calculated in closed form. However, modeling the shape as a Gaussian concentrates a larger amount of the probability mass in the center than a uniform distribution would do. [YBG16] demonstrate, however, that the GW and the Wasserstein distance based on the discretization of a uniform distribution across the contour behave similarly for ellipses, while the Kullback-Leibler divergence shows some counter-intuitive behavior, e.g., quadratic growth of the error in relation to a linearly growing distance between the centers and the error of the centers depending on the shape matrix.

Due to the same parameterization, the difference between rectangles can be measured using the GW distance as well. However, the density of the shape would be weighted less in the corners. This leads to a limitation pointed out by [YBG16], i.e., the GW distance finds no difference in squares only differing in their rotation (as two circles in this case would be the same independent of their orientation).

2.3 Special Densities

We will employ different densities in addition to Gaussian densities in this thesis to handle the unique challenges of ETT. More details can be achieved using a GM density, described in Section 2.3.1. If a parameter of the density is periodic, e.g., the orientation, the concept of wrapped distributions can be applied, which is presented in Section 2.3.2.

2.3.1 Gaussian Mixture

Sometimes, a Gaussian density is not enough to model a part of the system, e.g., the state or noise. To handle this, the corresponding density can instead be modeled using a GM density. A GM with n components is distributed according to

$$\mathbf{x}|\mathbb{X}_n \sim \sum_{i=1}^{n} w_i \mathcal{N}(\hat{\mathbf{x}}_i, \mathbf{C}_i^{\mathbf{x}}) \ , \tag{2.29}$$

with $\mathbb{X}_n$ being a set of n components each consisting of a mean $\hat{\mathbf{x}}_i$, a covariance $\mathbf{C}_i^{\mathbf{x}}$, and a weight w_i, with the condition that $\sum_{i=1}^{n} w_i = 1$ and $w_i \geq 0 \; \forall i \in \{1 \ldots n\}$.

GMs can be used to represent the state of a target to handle unclear associations, e.g., [Sal09]. For example, if an extended target has multiple possible measurement sources, then all possible associations of the measurement to the sources need to be considered. Similarly, if we have multiple targets, then all possible associations between measurements and targets need to be considered. Each of these associations leads to a new GM component and from each component, new components can arise. This can lead to the number of components growing exponentially over time, making it computationally infeasible to keep track of all of them. To counter this behavior, mixture reduction techniques can be applied, as discussed in, e.g., [CWPS11]. In general, the goal is to find a second mixture $\mathbb{X}_{\tilde{n}}$ with $\tilde{n} < n$ components $\tilde{w}_j$, $\tilde{\mathbf{x}}_j$, and $\tilde{\mathbf{C}}_j^{\mathbf{x}}$ ($j \in \{1 \ldots \tilde{n}\}$), which minimizes a distance measure, e.g., the integral square error, to the original mixture, i.e.,

$$\mathbb{X}_{\tilde{n}} = \underset{\mathbb{X}_{\tilde{n}}}{\operatorname{argmin}} \int (p(\mathbf{x}|\mathbb{X}_n) - p(\mathbf{x}|\mathbb{X}_{\tilde{n}}))^2 \mathrm{d}\mathbf{x} \ . \tag{2.30}$$

Other options for the distance measure include the normalized integral square error or the Kullback-Leibler divergence.

A simple approach is to prune the components starting with those that have a lower weight until the desired number of components is reached. This approach can, however, be less precise than merging of the components. There are different merging based approaches, further discussed in [CWPS11]. The general idea is to use a greedy merging approach to find an initial reduced GM and then use optimization to find a reduced GM by merging components in a way that minimizes a distance measure, e.g., the integral square error, to the original GM. For the greedy approach, components can be merged in pairs until the desired number of components is reached or clustered and then the clusters are merged [Sal09]. Examples are the GM reduction via clustering [SH09], constrained optimized weight adaption [CS10], an approach based on sparse modeling [ZHL12], or a method based on fuzzy adaptive resonance theory [ZJ14]. [AOÖ15] propose to use the Kullback-Leibler divergence from the reduced mixture to the original one, not the other way around, and include pruning in addition to the merging used by the aforementioned approaches. An approach consisting of pruning, clustering, and merging via covariance intersection based on an optimization criterion is described in [XFPW19].

The merging process itself can be conducted via moment matching, finding a single component Gaussian which captures the mean and covariance of the GM [HH08]. Assume we have a cluster of $\overline{n} \leq n$ components with indices $|\mathbb{I}| = \overline{n}$ and respective means $\hat{\mathbf{x}}_i$, covariances $\mathbf{C}_i^{\mathbf{x}}$, and weights w_i $(i \in \mathbb{I})$. According to [CWPS11], a single component Gaussian can be calculated using

$$w_{\text{merged}} = \sum_{i \in \mathbb{I}} w_i \ , \tag{2.31}$$

$$\hat{\mathbf{x}}_{\text{merged}} = \sum_{i \in \mathbb{I}} \frac{w_i}{w_{\text{merged}}} \hat{\mathbf{x}}_i \ , \tag{2.32}$$

$$\mathbf{C}^{\mathbf{x}}_{\text{merged}} = \sum_{i \in \mathbb{I}} \frac{w_i}{w_{\text{merged}}} (\mathbf{C}_i^{\mathbf{x}} + \hat{\mathbf{x}}_i \hat{\mathbf{x}}_i^{\mathrm{T}}) - \hat{\mathbf{x}}_{\text{merged}} \hat{\mathbf{x}}_{\text{merged}}^{\mathrm{T}} \ . \tag{2.33}$$

2.3.2 Wrapped Distribution

A wrapped distribution [MJ09,KGH14] is a modification of a probability density to account for parameters lying on a unit sphere. Given, e.g., a probability density of an angle θ, which by definition is 2π periodic, the density $p(\theta)$ can be wrapped

$$p_{\mathrm{w}}(\theta) = \sum_{k=-\infty}^{\infty} p(\theta + k2\pi) \ , \tag{2.34}$$

with $-\pi \leq \theta < \pi$. In Kalman filtering, the 2π periodicity of the orientation can be handled in a similar way by wrapping the innovation between $-\pi$ and π, e.g., [MĆP16]. The formula to restrict an innovation angle ν_θ is simply

$$\nu_{\theta,\mathrm{w}} = \big((\nu_\theta + \pi) \mod (2\pi)\big) - \pi \ . \tag{2.35}$$

2.4 Dynamic Models

To realize the prediction of the target state, a suitable process model is required. Different models exist depending on the target type. We will present two, the Nearly Constant Velocity (NCV) model because of its linearity in Section 2.4.1 and the Coordinated Turn (CT) model because of its incorporation of the yaw rate in Section 2.4.2. In the following, the time difference between time steps k and $k + 1$ is denoted by T.

2.4.1 Nearly Constant Velocity Model

For the NCV model, e.g. [LJ03], the kinematic part of the state is modeled using a position $\mathbf{m} = \begin{bmatrix} m_1 & m_2 \end{bmatrix}^{\mathrm{T}}$, velocity $\dot{\mathbf{m}} = \begin{bmatrix} \dot{m}_1 & \dot{m}_2 \end{bmatrix}^{\mathrm{T}}$, and sometimes an acceleration. As an example, we provide a model in 2D using position and velocity, with the acceleration modeled as zero-mean white process noise $\begin{bmatrix} a_1 & a_2 \end{bmatrix}^{\mathrm{T}}$, i.e., we have state vector and discretized process model

$$\mathbf{x}_k^{\text{kin}} = \begin{bmatrix} m_1 & m_2 & \dot{m}_1 & \dot{m}_2 \end{bmatrix}^{\mathrm{T}} , \tag{2.36}$$

$$\mathbf{x}_{k+1}^{\text{kin}} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k^{\text{kin}} + \mathbf{w}_k \ . \tag{2.37}$$

If we assume discrete acceleration process noise (resulting in the model also being called a discrete white noise acceleration model), i.e., the noise is constant during the sampling period and the noises drawn in different time steps are uncorrelated, we get [BSLK04]

$$\mathbf{w} = \underbrace{\begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{G} \begin{bmatrix} \sigma_{a_1}^2 & 0 \\ 0 & \sigma_{a_2}^2 \end{bmatrix} \mathbf{G}^{\mathrm{T}}) \ . \tag{2.38}$$

Sometimes, a polar velocity model is more appropriate. We define the velocity using a scalar velocity v and orientation α with acceleration noise on the velocity a and noise on the orientation ω. This changes state, process model, and noise to

$$\mathbf{x}_k^{\text{kin}} = \begin{bmatrix} m_1 & m_2 & \alpha & v \end{bmatrix}^{\mathrm{T}} , \tag{2.39}$$

$$\mathbf{x}_{k+1}^{\text{kin}} = \begin{bmatrix} 1 & 0 & 0 & T\cos(\alpha) \\ 0 & 1 & 0 & T\sin(\alpha) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k^{\text{kin}} + \mathbf{w}_k \ , \tag{2.40}$$

$$\mathbf{w} = \underbrace{\begin{bmatrix} 0.5T^2\cos(\alpha) & 0 \\ 0.5T^2\sin(\alpha) & 0 \\ 0 & T \\ T & 0 \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} a \\ \omega \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{G} \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} \mathbf{G}^{\mathrm{T}}) \ . \tag{2.41}$$

As the transition matrix now contains the orientation from the state vector, the process model is not linear anymore. The nonlinear Kalman filter prediction can be applied using, e.g., EKF or UKF.

2.4.2 Coordinated Turn Model

To model turning motions more accurately, the yaw rate ω can be added to the state, resulting in a Coordinated Turn (CT) or (nearly) constant turn model [RHG14]. Approximating the acceleration on velocity and yaw rate as 0, a CT model with polar velocity can be derived as

$$\mathbf{x}_k^{\text{kin}} = \begin{bmatrix} m_1 & m_2 & \alpha & v & \omega \end{bmatrix}^{\mathrm{T}} \ , \tag{2.42}$$

$$\mathbf{x}_{k+1}^{\text{kin}} = \begin{bmatrix} m_1 + \frac{2v}{\omega}\sin(0.5\omega T)\cos(\alpha + 0.5\omega T) \\ m_2 + \frac{2v}{\omega}\sin(0.5\omega T)\sin(\alpha + 0.5\omega T) \\ \alpha + \omega T \\ v \\ \omega \end{bmatrix} + \mathbf{w}_k \ . \tag{2.43}$$

Assuming constant noise input while approximating the orientation as constant between sampling times, the acceleration noise can be calculated as

$$\mathbf{w} = \underbrace{\begin{bmatrix} 0.5T^2\cos(\alpha) & 0 \\ 0.5T^2\sin(\alpha) & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} a_v \\ a_\omega \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{G} \begin{bmatrix} \sigma_{a_v}^2 & 0 \\ 0 & \sigma_{a_\omega}^2 \end{bmatrix} \mathbf{G}^{\mathrm{T}}) \ . \tag{2.44}$$

Here, a_v describes the acceleration noise in the direction of the heading, while a_ω is the acceleration noise on the yaw rate. For Cartesian velocity, the model can be converted to

$$\mathbf{x}_k^{\text{kin}} = \begin{bmatrix} m_1 & m_2 & \dot{m}_1 & \dot{m}_2 & \omega \end{bmatrix}^{\mathrm{T}} , \tag{2.45}$$

$$\mathbf{x}_{k+1}^{\text{kin}} = \begin{bmatrix} m_1 + \frac{\dot{m}_1}{\omega}\sin(\omega T) - \frac{\dot{m}_2}{\omega}(1-\cos(\omega T)) \\ m_2 + \frac{\dot{m}_1}{\omega}(1-\cos(\omega T)) + \frac{\dot{m}_2}{\omega}\sin(\omega T) \\ \dot{m}_1\cos(\omega T) - \dot{m}_2\sin(\omega T) \\ \dot{m}_1\sin(\omega T) + \dot{m}_2\cos(\omega T) \\ \omega \end{bmatrix} + \mathbf{w}_k , \tag{2.46}$$

with the noise under constant noise assumption being approximated as

$$\mathbf{w} = \underbrace{\begin{bmatrix} 0.5T^2\cos(\alpha) & 0 \\ 0.5T^2\sin(\alpha) & 0 \\ T\cos(\alpha) & 0 \\ T\sin(\alpha) & 0 \\ 0 & T \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} a_v \\ a_\omega \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{G} \begin{bmatrix} \sigma_{a_v}^2 & 0 \\ 0 & \sigma_{a_\omega}^2 \end{bmatrix} \mathbf{G}^{\mathrm{T}}) , \tag{2.47}$$

with $\alpha = \measuredangle(\begin{bmatrix} \dot{m}_1 & \dot{m}_2 \end{bmatrix}^{\mathrm{T}})$. Again, an appropriate model needs to be chosen to deal with the non-linearities, e.g., EKF or UKF. For more details, we refer to [RHG14].

3

Elliptical Target Tracking on Doppler RADAR

Contents

Due to a higher resolution resulting from higher frequency, automotive RADAR can provide more information about its surroundings than classic RADAR used for tracking, e.g., distant airplanes [GSD$^+$17]. Multiple RADARs can be utilized to realize a surround view of the host. Unlike camera and LIDAR, RADAR is more robust against bad lighting or weather conditions. On the other hand, RADAR often has lower measurement rate and higher noise than LIDAR. An additional feature of RADAR is the usage of the Doppler effect to measure the velocity of a measurement point in measurement direction, the so-called Doppler velocity or range rate.

This chapter deals with trackers using elliptic shapes to track extended targets. The exact problem setting is described in Section 3.1. Several elliptical extended target trackers exist in literature, all with their own advantages and disadvantages. To better differentiate them, we focus on approaches, which model the shape using orientation and semi-axes lengths and are based on the common spatial distribution model [GS05], assuming a uniform distribution of the measurements across the surface. We utilize a variant of the Random Hypersurface Model (RHM) [BH14, BH09, BNH10] using the polar ellipse equation [ZXWA13] and provide an EKF formulation using the aforementioned ellipse parameterization in Section 3.2. Another method of interest is the MEM-EKF* [YB19], which is a problem-tailored EKF for ellipses. It has been demonstrated to be able to provide better results during turning maneuvers than the RM model [Koc08, FFK11]. The latter models the shape as a positive definite matrix distributed by an inverse Wishart density. The MEM-EKF* is used in literature, e.g., for tracking on image data in conjunction with a convolutional neural network in [XN21]. It is presented in Section 3.3. For both trackers, we provide a discussion on the incorporation of the range rate measurements. For the MEM-EKF*, we also present an implementation using range rate.

Remark. *This chapter is based on the conference publication [1], which compares several elliptic extended target trackers parameterized with center, orientation, and semi-axes lengths. The Doppler extension for MEM-EKF*, MEM-EKF*-D, is adapted from the conference publication [5].*

3.1 Problem Setting

We consider the problem of single ETT, i.e., a target's shape is bigger than a single sensor resolution cell. Unlike a point target, which can generate a single or no measurement per time step, an extended target can also generate more than one measurement per time step, distributed across its shape. See Section 2.2 for more details on the foundations of ETT. We describe our assumptions for extended targets in Section 3.1.1. We assume the measurements stem from a RADAR sensor, which can produce detections not just from the contour, but from across the entire surface. Related works in ETT are presented in Section 3.1.2 and properties of RADAR are discussed along with datasets and tracking methods incorporating the range rate in Section 3.1.3.

3.1.1 Target Model Assumptions

This work focuses on elliptical shapes as they are convenient due to their simplicity, their suitability for high noise scenarios, in which the target's shape is hard to estimate, and them having the same parameterization as bounding boxes. We parameterize the ellipse using the center $\mathbf{m} \in \mathbb{R}^2$, orientation $\alpha \in [-\pi, \pi[$, and semi-axes lengths $a \in \mathbb{R}^+$ and $b \in \mathbb{R}^+$. Unlike shape matrices, this representation can consider different noise values on the individual shape parameters, e.g., when an elliptical target has fixed semi-axes lengths, but the orientation can change, the noise on the semi-axes can be chosen much lower than on the orientation. This has been demonstrated in [YB19]. In addition, kinematic parameters are tracked. Different models are possible, e.g., a Cartesian velocity, a polar velocity, or a polar velocity in the direction of the shape's orientation. It must be noted here that, while using the same parameter for the velocity and shape orientation can be intuitive, in some situation, it is not always appropriate, e.g., due to the drifting motion of ships [FBH19, SRW15]. In the following, we have the change in position $\dot{\mathbf{m}}$ along with optional additional parameters like the yaw rate, which are represented as dots our state equation. In summary, our considered state at time step k, $\mathbf{x}_k$, is modeled as

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{m}_k^{\mathrm{T}} & \dot{\mathbf{m}}_k^{\mathrm{T}} & \dots & \alpha_k & a_k & b_k \end{bmatrix}^{\mathrm{T}} . \tag{3.1}$$

Each time step k, the state is represented by a Gaussian random variable $\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_k, \mathbf{C}_k^{\mathbf{x}})$. We assume that the sensor measures n_k 2D reflection points or sources $\mathbf{z}_k^{(i)}$ in Cartesian space

($i \in \{1, \dots, n_k\}$), which are distributed uniformly across the surface and are mutually independent [GGMS05, GS05]. Each measurement source i can be located anywhere on the surface of the ellipse, so we can describe it using the following formula which holds true for all points $\mathbf{z}_k^{(i)}$ on the surface of the ellipse described by the matrix $\mathbf{A}_k$

$$(\mathbf{z}_k^{(i)} - \mathbf{m}_k)^{\mathrm{T}} \mathbf{A}_k (\mathbf{z}_k^{(i)} - \mathbf{m}_k) \leq 1 \ , \tag{3.2}$$

with

$$\mathbf{A}_k = \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix} \begin{bmatrix} a_k^2 & 0 \\ 0 & b_k^2 \end{bmatrix} \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix}^{\mathrm{T}} \ . \tag{3.3}$$

The sources are then corrupted by independent Gaussian white noise $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_k^{\mathbf{v}})$. This gives us the measurement equation of measurement i at time k

$$\mathbf{y}_k^{(i)} = \mathbf{z}_k^{(i)} + \mathbf{v}_k^{(i)} \ . \tag{3.4}$$

For the sake of simplicity, the measurement and time indices are omitted in the remainder of this chapter.

3.1.2 Related Work

In general, different shape models exist, such as elliptic shapes [Koc08, FFK11, BNH10, YB19, BDD17, BH14, GBR17, LL16, BH09, BFF+10, YSD20, ZXWA13, Gov19, VBGs+15, DWS11, AGS19, TÖ21, LLL20, AMPG13, GLO11] or rectangles [GLO11, SWBH12]. There are also approaches modeling more detailed distributions on a simple shape, e.g., [SD18, KHB18, KHB19, HK20, ZL21, XWB+21, CLLL21, YWB+21]. More complex shapes can be achieved via multiple ellipses [LL14, TÖO21], multiple reflector points [BWBS+17, HSSS12], star-convex shapes via RHM [BH14, WÖ15, HSRD16, TLTK19, KBW17], splines [KHB18, NBW19], or via an extension deformation approach [CLL20]. [LGYBC21, LGB+19] track banana-shaped clusters based on shape matrices and Lie groups using an Iterative Extended Kalman Filter (IEKF), e.g., [Zha97].

An elliptic extended target approach which fulfills the criteria set at the beginning of this chapter is the Independent Axis Estimation (IAE) by [Gov19], which splits the state into a kinematic state and two independent Gaussian random variables to model the semi-axis lengths. They approximate the orientation of the shape via the orientation of the velocity vector, similar to [BFH12]. The kinematic part is updated using Kalman filter formulas and they derive closed-form formulas for the update of the semi-axis lengths using the eigenvalues of the spread matrix, approximating its rotation with the velocity vector as well.

Other elliptical extended target trackers which we do not consider here are the aforementioned, widely-used RM approaches [Koc08, FFK11, LL16, VBGs+15], pioneered by Koch [Koc08] and further developed by Feldmann et al. [FFK11]. They model the shape as a shape matrix instead of using explicit parameters as in Equation (3.1). In this approach, the kinematic and shape states are decoupled, modeling the kinematic state as a Gaussian density, while the shape state is realized using an inverse Wishart distributed shape matrix. Here, the shape state is updated using the spread matrix of the measurements.

A method which uses the random matrix likelihood, but models the shape parameters as orientation and semi-axes is presented in [TÖ21]. They do, however, model the parameters as independent random variables, using Gamma distributions for the semi-axes, and apply a variational approach for the update. An approach proposed by [DWS11] is not based on the common spatial distribution model, modeling the measurements' principal components as Gaussians. An ellipse is tracked in [AMPG13] using a convolutional particle filter and approximates the orientation by the velocity vector. In [YSD20], an ellipse is modeled via image moments and then tracked using an RHM along with an UKF. The approach by [LLL20] constructs a measurement equation using the polar ellipse equation and greedy association, but they model the orientation using a vector consisting of $\begin{bmatrix}\cos(\alpha) & \sin(\alpha)\end{bmatrix}^{\mathrm{T}}$ instead of just α, requiring a pseudo-measurement to restrain it. They handle non-linearities using an UKF for the kinematic update and apply optimization to estimate the semi-axes.

Other approaches which can deal with a variety of star-convex shapes apply an RHM [BH14, WÖ15, HSRD16, TLTK19, KBW17], which defines a function to describe the contour of the target. There exists a version which utilizes the implicit ellipse equation to create a measurement equation for ellipses [BNH10], but often a radial function is defined yielding the distance of the center to the contour for a given angle. This is possible due to the star-convex assumption. The radial function can be used in combination with a scaling parameter to allow a distribution across the surface and not just the contour. Various implementations exist realizing the radial function using the polar ellipse equation for elliptic shapes [ZXWA13], [3] or using Fourier coefficients [BH11] or a Gaussian Process (GP) [WÖ15] for arbitrary star-convex shapes. A GP is described by a mean and a kernel function and [WÖ15] provide a periodic kernel, while a modification in [HSRD16] uses an axis-symmetric kernel. For these approaches, a greedy association model is used to determine the expected measurement source, i.e., the distribution of the source is approximated by a Dirac distribution. This can, however, lead to a bias in the estimation of the target's size. More details on that follow in the description of the RHM in Section 3.2. Discussions and approaches to handle the bias can be found in, e.g., [BH14, BH09, BFF+10, ZXWA13, FZH14, FDZH16, FZBH15].

An approach sampling equidistantly distributed measurement sources on a star-convex shape defined by a radial function and associating measurements to them via an Expectation Maximization (EM) approach can be found in [KBW17]. A different association for a GP based RHM is proposed in [MBL+20], which samples expected measurement sources based on the sensor resolution and uses global nearest neighbor to associate the measurements to them. A method defining a shape function based on splines using a running parameter can be found in [KHB18] and a method modeling more complex shapes via level-set RHMs is presented in [ZFBH16]. Tracking of star-convex shapes using GPs via a convolutional particle filter has been done in [ADFAM17]. For more complex shapes, a method which uses multiple elliptical sub-objects and handles the measurement update via a variational Bayesian approach is presented in [TÖO21]. Another recent implementation using variational Bayes to track a GP based RHM can be found in [KKÖ21].

A different approach is to model the shape using reflection points, e.g., [HSSS12, BWBS^{+}17], handling the association via Probabilistic Data Association (PDA) [BSDH09] as in [HSSS12] or via an EM strategy [BWBS^{+}17]. The Probabilistic Multi-Hypothesis Tracker (PMHT) [SL95] is used to handle multiple extended targets and point targets at the same time in [TLTK19], using a GP based target model and dynamically estimating the number of contour points via the expected measurement rate. An implementation of the histogram-PMHT for extended targets using RMs can be found in [WD14]. The work in [PMGA11] defines a likelihood for arbitrary shapes by discretizing the visible part of the contour and applies a particle filter.

3.1.3 Automotive Doppler RADAR

A RADAR measurement point usually consists of range and angle, but can be converted to Cartesian coordinates as is discussed in Section 3.3.1. In general, RADAR sensors can use the Doppler effect to determine the velocity of a measured point in the direction of the measured angle. This can be considered as the change rate of the range and is often called range rate. The nuScenes dataset [CBL^{+}20] is an example of a dataset providing RADAR data with range rate measurements. They utilize five 77 GHz Frequency-Modulated Continuous-Wave (FMCW) RADAR sensors positioned at the corners and the front of the vehicle. They have a range of up to 250 m, a capture frequency of 13 Hz, and an accuracy in the range rate measurement of ± 0.1 km/h. The dataset contains scenes of about 20 s long from urban scenarios.

Other datasets include CARRADA [ONR^{+}21], which provides range-Doppler and range-angle measurements of cars, pedestrians, and cyclists, the Oxford Radar RobotCar dataset [BGM^{+}20, MPLN17], which provides range-angle measurements, the Astyx dataset [MK19], which provides a smaller but higher resolution set of RADAR point clouds, and the Epan dataset [YSKR20], which uses a RADAR in some of its scenes. Furthermore, [MWRH20] present a high resolution RADAR dataset using a synthetic aperture RADAR providing range, azimuth, and Doppler information. The PixSet dataset [DMT^{+}21] presents annotated data with focus on full-waveform LIDAR data, but also contains detections from a RADAR sensor.

Incorporation of the range rate into the measurement equation of tracking algorithms can be found, e.g., in [DHL04] for point targets. An approach for extended targets is presented in [KSD16], defining a likelihood for a particle filter. They apply it on multiple targets in [SKRD16]. An approach using multiple reflection points can be found in [BWBS+17] based on EM and an EKF. While [HSSS12] use reflectors as well, they cluster them, use PDA for the association, and apply an UKF. In [SRW15], Jacobians are used to incorporate the Doppler measurement into the RM tracker by transforming the shape matrix and updating the kinematics in conjunction with an UKF. Based on the RHM approach which uses a GP for the radial function from [WÖ15], the model was extended by the range rate using an EKF in [6]. Another approach presented in [KBK+16] calculates the mean of the measurements to avoid modeling a shape, applying an offset to account for the fact that the mean is not always at the vehicle's center of rotation, and incorporates the range rate by calculating the velocity profile and applying an UKF. Learned models via variational approaches [SD18] which handle more complex distributions can be found as well. In the volcanormal model [BDD17], which models a density around the contour of an ellipse, the range rate measurement is integrated and the update conducted via batch optimization techniques. In [GM20], they directly use the extent of a measurement cluster for the shape update and calculate the velocity of the cluster via recursive least squares. [CLLL21] provide a measurement equation for a more complex distribution of the measurement sources and incorporate the range rate, handling non-linearities with an UKF. A grid based approach to enhance measurement information is provided in [BML+20], which calculates occupancy evidence.

3.2 Ellipse-RHM-EKF

The RHM [BH14, BH09, BNH10] approach has been developed to work for star-convex shapes. [BH14, BH09, BFF+10] provide an (implicit) ellipse parameterization, but they model the shape using the elements of the shape matrix' square root obtained via a Cholesky decomposition and apply an UKF. However, the Cholesky decomposition does not have an intuitive meaning and similar to using a shape matrix, individual ellipse parameters cannot be tuned. Therefore, we suggest to model the shape using orientation and semi-axes lengths.

Furthermore, we adapt the usage of the polar ellipse equation for the radial function of the RHM from [ZXWA13]. This results in an explicit measurement equation. However, due to approximations regarding the actual measurement source, this method introduces a bias. [BH11] suggest an implicit measurement equation to reduce the bias for their RHM based on Fourier coefficients and [ZXWA13] adapted this idea for their polar ellipse equation. Both approaches handle the non-linearities using an UKF. In the following, we will present compact closed-form expressions using an EKF implementation for both the explicit measurement equation in Section 3.2.1 and the implicit one in Section 3.2.2.

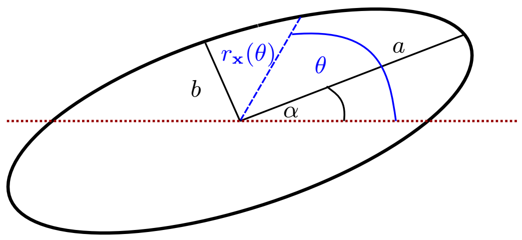


Figure 3.1: Visualization of the radial function for an ellipse, with the radial function providing the length of the dashed line between the center and the contour point at angle θ.

3.2.1 Explicit Equation

The polar ellipse equation of our state $\mathbf{x}$

$$r_{\mathbf{x}}(\theta) = \frac{ab}{\sqrt{(b \cdot \cos(\theta - \alpha))^2 + (a \cdot \sin(\theta - \alpha))^2}} \quad , \tag{3.5}$$

describes the distance of the ellipse's center to its surface for a given angle θ and is visualized in Figure 3.1. To model a distribution across the entire surface, we introduce a scaling factor s and model the measurement source as

$$\mathbf{z} = \mathbf{m} + s r_{\mathbf{x}}(\theta)\mathbf{e}_\theta \quad , \tag{3.6}$$

with $\mathbf{e}_\theta = \begin{bmatrix}\cos(\theta) & \sin(\theta)\end{bmatrix}^{\mathrm{T}}$ describing a unit vector in direction θ. Inserting (3.6) into (3.4), we get the measurement equation

$$\boxed{\mathbf{y} = h(\mathbf{x}) = \mathbf{m} + s r_{\mathbf{x}}(\theta)\mathbf{e}_\theta + \mathbf{v} \quad .} \tag{3.7}$$

This leaves us with two parameters to model the actual distribution, the input angle θ of the radial function and the scaling factor s. In [BH14], they demonstrated that a uniform distribution on the surface of the ellipse can be realized for a given angle θ if the squared scaling factor is uniformly distributed, i.e.,

$$s^2 \sim \mathcal{U}(0, 1) \quad . \tag{3.8}$$

As a Kalman filter assumes all random variables to be Gaussian distributed, s is approximated by a Gaussian with mean $\hat{s} = \frac{2}{3}$ and variance $\sigma_s^2 = \frac{1}{18}$. To deal with the resulting multiplicative noise term, [WÖ15] proposed to shift the zero-mean portion of s into an additive noise term. This replaces s in (3.7) with $\hat{s}$ and the noise $\mathbf{v}$ with

$$\overline{\mathbf{v}} = (s - \hat{s}) r_{\mathbf{x}}(\theta)\mathbf{e}_\theta + \mathbf{v} \quad . \tag{3.9}$$

The angle θ on the other hand is commonly approximated by a Dirac delta function, which in turn is determined based on the target to measurement geometry, i.e., the orientation of the vector $(\mathbf{y}-\mathbf{m})$

$$\theta \approx \measuredangle(\mathbf{y}-\mathbf{m}) \ . \tag{3.10}$$

This approach to approximate θ is called a greedy approximation. Associating a specific measurement to an angle θ can be seen as associating the measurement to a measurement source, which is defined by θ. Therefore, this approach is also called greedy association. While this method works well enough in most cases, it is sensitive to the prior estimate and it introduces a bias in high noise scenarios, i.e., the size of the shape is overestimated. Other more complex methods exist as well, such as the partial likelihood [FZBH15], but they are excluded as we want to focus on compact closed-form solutions.

The algorithm is named Ellipse Random Hypersurface Model Extended Kalman Filter (Ellipse-RHM-EKF). It processes the measurements sequentially. The state estimate at time step k after the first i measurements have been processed is denoted as $\hat{\mathbf{x}}_k^{(i)}$ and its covariance matrix is named $\mathbf{C}_k^{\mathbf{x}(i)}$. Given m_k measurements, the measurement update at time step k is summarized in Algorithm 1. The time update can be conducted using any appropriate dynamic model.

3.2.2 Implicit Equation

[BH11,ZXWA13] state that the influence of the greedy association and thus the bias can be reduced by reformulating the measurement equation into an implicit form

$$\boxed{0 = g(\mathbf{x},\mathbf{y}) = \underbrace{s^2 \cdot \|r_{\mathbf{x}}(\theta)\|^2}_{g_1} - \underbrace{\|\mathbf{y}-\mathbf{m}\|^2}_{g_2} + \underbrace{2\hat{s}r_{\mathbf{x}}(\theta)\cdot \mathbf{e}_\theta^{\mathrm{T}} \cdot \mathbf{v} + \|\mathbf{v}\|^2}_{g_3} \ .} \tag{3.11}$$

The multiplicative noise from the scaling factor s in g_1 is handled similarly to the explicit version and results in the modified g_1 and g_4 in the following equation (keep in mind that $s^2 \sim \mathcal{U}(0,1)$, so that $\mathrm{E}[s^2] = 0.5$)

$$g(\mathbf{x},\mathbf{y}) \approx \underbrace{0.5 \cdot \|r_{\mathbf{x}}(\theta)\|^2}_{g_1} - \underbrace{\|\mathbf{y}-\mathbf{m}\|^2}_{g_2} + \underbrace{2\hat{s}r_x \cdot \mathbf{e}_\theta^{\mathrm{T}} \cdot \mathbf{v} + \|\mathbf{v}\|^2}_{g_3} + \underbrace{(s^2-0.5)\cdot \|r_{\mathbf{x}}(\theta)\|^2}_{g_4} \ . \tag{3.12}$$

The general structure of the algorithm, named Ellipse Random Hypersurface Model Extended Kalman Filter Implicit (Ellipse-RHM-EKF-imp), is the same as for the explicit version and can be found in Algorithm 2. The analytic approximations dealing with the measurement error are based on [Bau13].

Iterative Measurement Update
$i = 1, ..., m_k$

$\mathbf{y}_k^{(i)}$

$$\hat{\mathbf{y}}_k = h(\hat{\mathbf{x}}_k^{(i-1)}, \mathbf{y}_k^{(i)})$$
$$\mathbf{C}_k^{\mathbf{x}(i-1)\mathbf{y}(i)} = \mathbf{C}_k^{\mathbf{x}(i-1)}\mathbf{H}^{\mathrm{T}}$$
$$\mathbf{C}_k^{\mathbf{y}(i)} = \mathbf{H}\mathbf{C}_k^{\mathbf{x}(i-1)}\mathbf{H}^{\mathrm{T}} + \mathbf{C}_k^{\mathbf{v}} + \mathbf{C}^s$$

$$\hat{\mathbf{x}}_k^{(i)} = \hat{\mathbf{x}}_k^{(i-1)} + \mathbf{C}_k^{\mathbf{x}(i-1)\mathbf{y}(i)}\left(\mathbf{C}_k^{\mathbf{y}(i)}\right)^{-1}\left(\mathbf{y}_k^{(i)} - \hat{\mathbf{y}}_k\right)$$
$$\mathbf{C}_k^{\mathbf{x}(i)} = \mathbf{C}_k^{\mathbf{x}(i-1)} - \mathbf{C}_k^{\mathbf{x}(i-1)\mathbf{y}(i)}\left(\mathbf{C}_k^{\mathbf{y}(i)}\right)^{-1}\left(\mathbf{C}_k^{\mathbf{x}(i-1)\mathbf{y}(i)}\right)^{\mathrm{T}}$$

$h(\mathbf{x}, \mathbf{y}) = \mathbf{m} + r_{\mathbf{x}}(\theta) \cdot \mathbf{e}_\theta \cdot \hat{s}$ $\quad$ $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{C}^x)$ $\quad$ $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_k^{\mathbf{v}})$

$\mathbf{x} = [\mathbf{m}^{\mathrm{T}} \quad \dots \quad \alpha \quad a \quad b]^{\mathrm{T}}$ $\quad$ $r_{\mathbf{x}}(\theta) = \frac{ab}{\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}}$

$s^2 \sim \mathcal{N}(0.5, \sigma_{s^2}^2); \sigma_{s^2}^2 = \frac{1}{12}$ $\quad$ $\mathbf{e}_\theta = (\cos(\theta) \quad \sin(\theta))^{\mathrm{T}}$ $\quad$ $\theta = \measuredangle(\mathbf{y} - \mathbf{m})$

$$\mathbf{H} = \left(\frac{\partial h}{\partial \mathbf{m}} \quad \dots \quad \frac{\partial h}{\partial \alpha} \quad \frac{\partial h}{\partial a} \quad \frac{\partial h}{\partial b} \quad \frac{\partial h}{\partial \mathbf{r}}\right)$$
$$\frac{\partial h}{\partial \mathbf{m}} = \mathbf{I}_2 + \mu_s \cdot \left(r_{\mathbf{x}}(\theta)\begin{pmatrix}-\sin(\theta)\\ \cos(\theta)\end{pmatrix} \cdot \frac{\partial \theta}{\partial \mathbf{m}} + \mathbf{e}_\theta \cdot \frac{\partial r_{\mathbf{x}}}{\partial \mathbf{m}}\right)$$
$$\frac{\partial \theta}{\partial \mathbf{m}} = \frac{1}{(y_1-m_1)^2 + (y_2-m_2)^2} \cdot (y_2 - m_2 \quad -(y_1 - m_1))$$
$$\frac{\partial r_{\mathbf{x}}}{\partial \mathbf{m}} = \frac{\partial r_{\mathbf{x}}}{\partial \theta} \cdot \frac{\partial \theta}{\partial \mathbf{m}}$$
$$\frac{\partial r_{\mathbf{x}}}{\partial \theta} = \frac{0.5 \cdot (ab^3 - a^3 b) \cdot \sin(2(\theta-\alpha))}{\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}^3}$$
$$\frac{\partial h}{\partial \alpha} = -\frac{\partial r_{\mathbf{x}}}{\partial \theta} \cdot \hat{s} \cdot \mathbf{e}_\theta$$
$$\frac{\partial h}{\partial a} = \hat{s} \cdot \mathbf{e}_\theta \cdot \left(\frac{b}{\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}} - \frac{a^2 b \sin(\theta-\alpha)^2}{\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}^3}\right)$$
$$\frac{\partial h}{\partial b} = \hat{s} \cdot \mathbf{e}_\theta \cdot \left(\frac{a}{\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}} - \frac{ab^2 \sin(\theta-\alpha)^2}{\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}^3}\right)$$
$$\mathbf{C}^s = \sigma_{s^2}^2 \cdot \|r_{\mathbf{x}}(\theta)\|^4$$

Time Update

$$\hat{\mathbf{x}}_{k+1}^{(0)} = \mathbf{F}\hat{\mathbf{x}}_k^{(n_k)}$$
$$\mathbf{C}_{k+1}^{\mathbf{x}(0)} = \mathbf{F}\mathbf{C}_k^{\mathbf{x}(n_k)}\mathbf{F}^{\mathrm{T}} + \mathbf{C}_k^{\mathbf{w}}$$

Algorithm 1: The Ellipse-RHM-EKF algorithm iterative measurement update and time update (given a linear process model). At time step k, m_k measurements $\mathbf{y}_k^{(i)}$ with $i \in \{1 \dots m_k\}$ are sequentially processed. The upper rounded box prepares the values for the Kalman filter update and the second one describes the actual update. Below, variables and functions are defined and the rounded box below describes how the Jacobian and the noise term depending on the scaling factor are calculated. The derivatives of the kinematic part represented by dots is 0. Note that the most recent estimate $\hat{\mathbf{x}}_k^{(i-1)}$ is used to realize the Jacobian $\mathbf{H}$ and the noise covariance $\mathbf{C}^s$. The time update is outlined in the lower box. Based on [1].

Iterative Measurement Update

$i = 1, \cdots, m_k$

$\mathbf{y}_k^{(i)}$

$$\hat{y}_p = g_1(\hat{\mathbf{x}}_k^{(i-1)}, \mathbf{y}_k^{(i)}) - g_2(\hat{\mathbf{x}}_k^{(i-1)}, \mathbf{y}_k^{(i)}) + \text{tr}(\mathbf{C}^{\mathbf{v}})$$

$$\mathbf{C}_k^{\mathbf{x}(i-1)y_p(i)} = \mathbf{C}_k^{\mathbf{x}(i-1)}(\mathbf{H}_1 - \mathbf{H}_2)^{\mathrm{T}}$$

$$\mathbf{C}_k^{y_p(i)} = (\mathbf{H}_1 - \mathbf{H}_2)\mathbf{C}_k^{\mathbf{x}(i-1)}(\mathbf{H}_1 - \mathbf{H}_2)^{\mathrm{T}} + \mathbf{C}^{\overline{\mathbf{v}}} + \mathbf{C}^s$$

$$\hat{\mathbf{x}}_k^{(i)} = \hat{\mathbf{x}}_k^{(i-1)} + \mathbf{C}_k^{\mathbf{x}(i-1)y_p(i)} \left(\mathbf{C}_k^{y_p(i)}\right)^{-1} (0 - \hat{y}_p)$$

$$\mathbf{C}_k^{\mathbf{x}(i)} = \mathbf{C}_k^{\mathbf{x}(i-1)} - \mathbf{C}_k^{\mathbf{x}(i-1)y_p(i)} \left(\mathbf{C}_k^{y_p(i)}\right)^{-1} \left(\mathbf{C}_k^{\mathbf{x}(i-1)y_p(i)}\right)^{\mathrm{T}}$$

$$g(\mathbf{x}, \mathbf{y}) = \underbrace{0.5 \cdot \|r_{\mathbf{x}}(\theta)\|^2}_{g_1} - \underbrace{\|\mathbf{y} - \mathbf{m}\|^2}_{g_2} + \underbrace{2\hat{s}r_{\mathbf{x}}(\theta) \cdot \mathbf{e}_\theta^{\mathrm{T}} \cdot \mathbf{v} + \|\mathbf{v}\|^2}_{g_3} + \underbrace{(s^2 - 0.5) \cdot \|r_{\mathbf{x}}(\theta)\|^2}_{g_4}$$

$$r_{\mathbf{x}}(\theta) = \frac{ab}{\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}} \qquad \mathbf{x} = [\mathbf{m}^{\mathrm{T}} \quad \dots \quad \alpha \quad a \quad b]^{\mathrm{T}} \qquad \mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{C}^x)$$

$$\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}^{\mathbf{v}})$$

$$s^2 \sim \mathcal{N}(0.5, \sigma_{s^2}^2);\ \sigma_{s^2}^2 = \frac{1}{12} \qquad \mathbf{e}_\theta = (\cos(\theta) \quad \sin(\theta))^{\mathrm{T}}\ \theta = \measuredangle(\mathbf{y} - \mathbf{m})$$

$$\mathbf{H}_1 = \left(\frac{\partial g_1}{\partial \mathbf{m}} \quad \dots \quad \frac{\partial g_1}{\partial \alpha} \quad \frac{\partial g_1}{\partial a} \quad \frac{\partial g_1}{\partial b} \quad \frac{\partial g_1}{\partial \mathbf{r}}\right) \qquad \mathbf{H}_2 = \left(\frac{\partial g_2}{\partial \mathbf{m}} \quad \dots \quad \frac{\partial g_2}{\partial \alpha} \quad \frac{\partial g_2}{\partial a} \quad \frac{\partial g_2}{\partial b} \quad \frac{\partial g_2}{\partial \mathbf{r}}\right)$$

$$\frac{\partial g_1}{\partial \mathbf{m}} = \frac{\partial g_1}{\partial \theta} \cdot \frac{\partial \theta}{\partial \mathbf{m}} \qquad \frac{\partial g_2}{\partial \mathbf{m}} = (-2(y_1 - m_1) \quad -2(y_2 - m_2))$$

$$\frac{\partial g_1}{\partial \theta} = 0.5\frac{(a^2b^4 - a^4b^2)\sin(2(\theta-\alpha))}{((b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2)^2} \qquad \frac{\partial g_2}{\partial \alpha} = \frac{\partial g_2}{\partial a} = \frac{\partial g_2}{\partial b} = 0$$

$$\frac{\partial \theta}{\partial \mathbf{m}} = \frac{1}{(y_1 - m_1)^2 + (y_2 - m_2)^2} \cdot (y_2 - m_2 \quad -(y_1 - m_1))$$

$$\frac{\partial g_1}{\partial \alpha} = -\frac{\partial g_1}{\partial \theta}$$

$$\frac{\partial g_1}{\partial a} = 0.5\left(\frac{2ab^2}{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2} - \frac{2a^3b^2\sin(\theta-\alpha)^2}{((b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2)^2}\right)$$

$$\frac{\partial g_1}{\partial b} = 0.5\left(\frac{2a^2b}{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2} - \frac{2a^2b^3\cos(\theta-\alpha)^2}{((b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2)^2}\right)$$

$$\mathbf{C}^{\overline{\mathbf{v}}} = \text{tr}(\mathbf{t}\mathbf{t}^{\mathrm{T}}\mathbf{C}^{\mathbf{v}}) + \text{tr}((\mathbf{C}^{\mathbf{v}})^2) + \text{tr}(\mathbf{C}^{\mathbf{v}})^2 \qquad \mathbf{C}^s = \sigma_{s^2}^2 \cdot \|r_{\mathbf{x}}(\theta)\|^4$$

$$\mathbf{t} = 2\hat{s}r_{\mathbf{x}}(\theta)\mathbf{e}_\theta^{\mathrm{T}}$$

Time Update

$$\hat{\mathbf{x}}_{k+1}^{(0)} = \mathbf{F}\hat{\mathbf{x}}_k^{(n_k)}$$

$$\mathbf{C}_{k+1}^{\mathbf{x}(0)} = \mathbf{F}\mathbf{C}_k^{\mathbf{x}(n_k)}\mathbf{F}^{\mathrm{T}} + \mathbf{C}_k^{\mathbf{w}}$$

Algorithm 2: Ellipse-RHM-EKF-imp iterative measurement update using m_k measurements from time step k and the update with a specific measurement $\mathbf{y}_k^{(i)}$ is shown. The upper rounded box prepares the values for the Kalman filter update and the second one describes the actual update. Below, variables and functions are defined and the rounded box below describes how the Jacobians and the noise terms are calculated. The derivatives of the kinematic part represented by dots is 0. Note that the most recent estimate $\hat{\mathbf{x}}_k^{(i-1)}$ is used to realize the Jacobians $\mathbf{H}_1$ and $\mathbf{H}_2$ and the noise covariances $\mathbf{C}^{\overline{\mathbf{v}}}$ and $\mathbf{C}^s$. Based on [1].

An advantage of both the explicit and implicit version is that due to the joint state vector, it is easy to use the shape's orientation α as the direction of the velocity and realize the kinematics as a scalar velocity v, e.g.,

$$\mathbf{x}_{comb} = \begin{bmatrix} \mathbf{m}^{\mathrm{T}} & v & \dots & \alpha & a & b \end{bmatrix} , \tag{3.13}$$

As this results in a non-linear process model, an EKF or UKF can be used here to handle the non-linearities.

3.2.3 Range Rate usage

The formulas of an EKF RHM can be extended by the range rate as was shown in [6], estimating the range rate at the measurement determined via greedy association. While [6] uses a GP to calculate the radial function, the corresponding formulas can be swapped for the polar ellipse equation. Please note that the original work assumes a fix shape as well as contour measurements. Further research is required to determine the effectiveness of the Doppler given uncertain shape parameters or different measurement distributions.

3.3 MEM-EKF*

A different approach to handle the measurement distribution is the introduction of a Multiplicative Error Model (MEM) [BFH12, YB16, YB17, YB19]. It uses a multiplicative noise term $\mathbf{h} = \begin{bmatrix} h_1 & h_2 \end{bmatrix}^{\mathrm{T}}$ to create a measurement equation without the need for a greedy association. The resulting explicit measurement equation has the form

$$\boxed{\mathbf{y} = \mathbf{m} + \underbrace{\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}}_{\mathbf{S}} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \mathbf{v} \ .} \tag{3.14}$$

The multiplicative noise term $\mathbf{h}$ can be interpreted as describing a distribution across the unit circle, which is then transformed using the shape parameters in the matrix $\mathbf{S}$. While $\mathbf{h}$ would ideally describe a uniform distribution across the surface, it must be approximated as a Gaussian to work in the Kalman filter framework. Therefore, we have $\mathbf{h} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}^{\mathbf{h}})$ with $\mathbf{C}^{\mathbf{h}} = \text{diag}\Big(\begin{bmatrix} 0.25 & 0.25 \end{bmatrix} \Big)$ to approximate a uniform distribution across an ellipse. The variance can be set to $\frac{1}{3}$ to approximate a uniform distribution across a rectangle instead.

To handle the multiplicative noise, [YB19] developed a problem-tailored Kalman filter, the MEM-EKF*. To achieve a closed form solution, they decouple the kinematic and shape state, i.e., they use the state $\mathbf{x} = \begin{bmatrix}\mathbf{r}^{\mathrm{T}} & \mathbf{p}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ with

$$\mathbf{r} = \begin{bmatrix}\mathbf{m}^{\mathrm{T}} & \dot{\mathbf{m}}^{\mathrm{T}} & \dots\end{bmatrix}^{\mathrm{T}} \ , \tag{3.15}$$

$$\mathbf{p} = \begin{bmatrix}\alpha & a & b\end{bmatrix}^{\mathrm{T}} \ , \tag{3.16}$$

which are both modeled as Gaussian random variables with means $\hat{\mathbf{r}}$ and $\hat{\mathbf{p}}$ and covariances $\mathbf{C}^{\mathbf{r}}$ and $\mathbf{C}^{\mathbf{p}}$, respectively. This decoupling is not always justified in practical applications, but the formulas still consider correlations between the two state vectors based on the measurement equation. The variance of the multiplicative noise is used to incorporate the shape parameters' mean and variance into the kinematic update via analytic approximations. This is justified as the distribution of the measurement sources is described by mean and variance of the multiplicative noise. The measurement equation is linearized as

$$\mathbf{y} \approx \mathbf{H}\mathbf{r} + \mathbf{S}\mathbf{h} + \begin{bmatrix}\mathbf{h}^{\mathrm{T}}\mathbf{J}_1 \\ \mathbf{h}^{\mathrm{T}}\mathbf{J}_2\end{bmatrix}(\mathbf{p} - \hat{\mathbf{p}}) + \mathbf{v} \ , \tag{3.17}$$

with the exact realization of $\mathbf{H}$ depending on $\mathbf{r}$ so that $\mathbf{H}\mathbf{r} = \mathbf{m}$. $\mathbf{S}$ is realized using the values of $\hat{\mathbf{p}}$ and Jacobians $\mathbf{J}_1$ and $\mathbf{J}_2$ can be found in [YB19]. This results in a measurement noise covariance

$$\mathbf{C}^{\mathbf{y}} = \mathbf{H}\mathbf{C}^{\mathbf{r}}\mathbf{H}^{\mathrm{T}} + \underbrace{\mathbf{S}\mathbf{C}^{\mathbf{h}}\mathbf{S}^{\mathrm{T}}}_{\mathbf{C}^{\mathrm{I}}} + \mathbf{C}^{\mathrm{II}} + \mathbf{C}^{\mathbf{v}} \ , \tag{3.18}$$

with the elements of $\mathbf{C}^{\mathrm{II}}$ calculated as

$$\mathbf{C}^{\mathrm{II}}_{mn} = \mathrm{tr}(\mathbf{C}^{\mathbf{p}}(\mathbf{J}_n)^{\mathrm{T}}\mathbf{C}^{\mathbf{h}}\mathbf{J}_m) \ , \tag{3.19}$$

and $\mathbf{C}^{\mathrm{II}}_{mn}$ referring to the element at the mth row and nth column of $\mathbf{C}^{\mathrm{II}}$. The update is then conducted using the Kalman filter formulas.

For the shape update a pseudo-measurement is defined based on the innovation of the kinematic update. This is necessary, as with the ordinary measurement, the shape state could not be fully observed, as is proven for axis-aligned extended targets in [BFH12]. For the calculation of the pseudo-measurement, we utilize the Kronecker product $\otimes$, which is defined as $\mathbf{y} \otimes \mathbf{y} = \begin{bmatrix}y_1^2 & y_1y_2 & y_2y_1 & y_2^2\end{bmatrix}^{\mathrm{T}}$. It can be seen that the second and third element are the same. The pseudo-measurement is then calculated as the Kronecker product of the innovation with the duplicate elements removed via the matrix multiplication

$$\mathbf{y}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \Big((\mathbf{y} - \hat{\mathbf{y}}) \otimes (\mathbf{y} - \hat{\mathbf{y}}) \Big) \ . \tag{3.20}$$

The expected pseudo-measurement is then

$$\hat{\mathbf{y}}_p = \begin{bmatrix} \mathbf{C}^{\mathbf{y}}_{11} & \mathbf{C}^{\mathbf{y}}_{22} & \mathbf{C}^{\mathbf{y}}_{12} \end{bmatrix}^{\mathrm{T}} \ , \tag{3.21}$$

with $\mathbf{C}^{\mathbf{y}}_{mn}$ referring to the element at the mth row and nth column of $\mathbf{C}^{\mathbf{y}}$. Linearizing the corresponding measurement equation around $\mathbf{p}$, we get the Jacobian

$$\mathbf{M} = \begin{bmatrix} 2\mathbf{S}_1\mathbf{C}^{\mathbf{h}}\mathbf{J}_1 \\ 2\mathbf{S}_2\mathbf{C}^{\mathbf{h}}\mathbf{J}_2 \\ \mathbf{S}_1\mathbf{C}^{\mathbf{h}}\mathbf{J}_1 + \mathbf{S}_2\mathbf{C}^{\mathbf{h}}\mathbf{J}_2 \end{bmatrix} \ , \tag{3.22}$$

with $\mathbf{S}_1$ and $\mathbf{S}_2$ being the first and second row of $\mathbf{S}$ respectively. The derivation of $\mathbf{M}$ as well as the innovation covariance matrix can be found in [YB19].

As the MEM-EKF* does not use a greedy association model, it does not suffer from a bias in the shape estimation or a prior dependency for the measurement to target association. Therefore, it is more robust. However, it must still be noted that the prior needs to be chosen carefully, because a prior which provides a significant portion of the probability mass to negative semi-axes lengths can lead to unintuitive results.

In general, a Cartesian velocity is used for the kinematic state, but [YB19] present a modification to extend $\mathbf{r}$ by a yaw rate ω and incorporate its influence on the shape during the time update.

3.3.1 Range Rate Usage

The following section is based on our work [5]. We discuss how to make use of the unique range rate property of Doppler RADAR in the MEM-EKF*. We consider the same state dimensions as before, however, as the velocity is relevant for the measurement update, we explicitly define here that it consists of a Cartesian velocity $\dot{\mathbf{m}} = \begin{bmatrix} \dot{m}_1 & \dot{m}_2 \end{bmatrix}^{\mathrm{T}}$ and a yaw rate ω, i.e., we have

$$\mathbf{r} = \begin{bmatrix} m_1 & m_2 & \dot{m}_1 & \dot{m}_2 & \omega \end{bmatrix}^{\mathrm{T}} \ , \tag{3.23}$$

$$\mathbf{p} = \begin{bmatrix} \alpha & a & b \end{bmatrix}^{\mathrm{T}} \ . \tag{3.24}$$

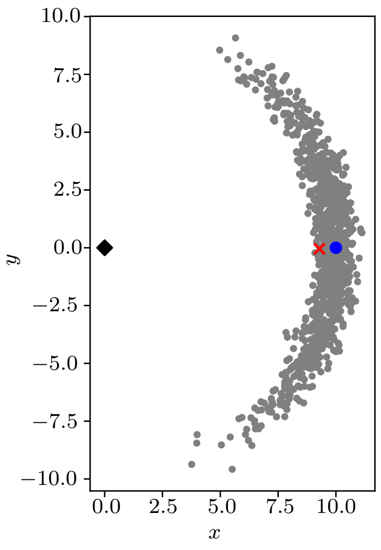


Figure 3.2: Example of the transformation bias from polar to Cartesian coordinates. The original mean is displayed as a blue circle in Cartesian coordinates. Samples drawn from the original distribution in polar space and transformed into Cartesian space are marked with gray dots. The mean of the samples in Cartesian space is drawn as a red cross, showing a visible bias to the actual mean. The sensor position is marked with a black diamond.

Previously, we assumed the measurements to be in Cartesian coordinates. In reality, the RADAR measurements consist of range, angle, and range rate $\tilde{\mathbf{y}} = \begin{bmatrix} r & \theta & \dot{r} \end{bmatrix}$ with additive zero-mean Gaussian measurement noise on those coordinates $\tilde{\mathbf{v}} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}^{\tilde{\mathbf{v}}})$, as mentioned in Section 3.1.3. In the remainder of this work, we assume the elements of the measurement noise to be uncorrelated in polar coordinates, i.e., $\mathbf{C}^{\tilde{\mathbf{v}}} = \mathrm{diag}\Big(\begin{bmatrix} \sigma_r^2 & \sigma_\theta^2 & \sigma_{\dot{r}}^2 \end{bmatrix}\Big)$. The range and angle measurements can be transformed into Cartesian measurements

$$\mathbf{y}_b^c = r \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \quad . \tag{3.25}$$

However, the conversion into Cartesian space under Gaussian noise in polar space leads to the introduction of a bias. Figure 3.2 shows an example in which 1000 samples in polar space with mean $\begin{bmatrix} 10.0 & 0.0 \end{bmatrix}^{\mathrm{T}}$ and covariance $\mathrm{diag}\Big(\begin{bmatrix} 0.2 & 0.05\pi \end{bmatrix}\Big)$ are drawn, transformed into Cartesian space and plotted as gray dots along with the true and sample mean, revealing a clear bias. This bias can be removed and the measurement noise covariance of the transformed noise $\mathbf{v}$ be adequately estimated using the formulas from [LBS93]

$$\mathbf{y}^c \quad = r \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \frac{1}{e^{-0.5\sigma_\theta^2}} \ , \tag{3.26}$$

$$\begin{aligned} \mathbf{C}^{\mathbf{v}}_{11} &= r^2 e^{-2\sigma_\theta^2}(\cos(\theta)^2(\cosh(2\sigma_\theta^2) - \cosh(\sigma_\theta^2)) + \sin(\theta)^2(\sinh(2\sigma_\theta^2) - \sinh(\sigma_\theta^2))) \\ &\quad + \sigma_r^2 e^{-2\sigma_\theta^2}(\cos(\theta)^2(2\cosh(2\sigma_\theta^2) - \cosh(\sigma_\theta^2)) + \sin(\theta)^2(2\sinh(2\sigma_\theta^2) - \sinh(\sigma_\theta^2))) \ , \end{aligned} \tag{3.27}$$

$$\begin{aligned} \mathbf{C}^{\mathbf{v}}_{11} &= r^2 e^{-2\sigma_\theta^2}(\sin(\theta)^2(\cosh(2\sigma_\theta^2) - \cosh(\sigma_\theta^2)) + \cos(\theta)^2(\sinh(2\sigma_\theta^2) - \sinh(\sigma_\theta^2))) \\ &\quad + \sigma_r^2 e^{-2\sigma_\theta^2}(\sin(\theta)^2(2\cosh(2\sigma_\theta^2) - \cosh(\sigma_\theta^2)) + \cos(\theta)^2(2\sinh(2\sigma_\theta^2) - \sinh(\sigma_\theta^2))) \ , \end{aligned} \tag{3.28}$$

$$\mathbf{C}^{\mathbf{v}}_{12} = \mathbf{C}^{\mathbf{v}}_{21} = \sin(\theta)\cos(\theta)e^{-4\sigma_\theta^2}(\sigma_r^2 + (r^2 + \sigma_r^2)(1 - e^{\sigma_\theta^2})) \ , \tag{3.29}$$

with $\mathbf{C}^{\mathbf{v}}_{nm}$ referring to the element of $\mathbf{C}^{\mathbf{v}}$ in row n and column m. They derive the formulas analytically, approximating the true measurement in the formulas by taking the expectation and conditioning it on the measurement. Note that this transforms the measurements into Cartesian coordinates in sensor space. To transform it into global coordinates, the data needs to be rotated and shifted by the sensor's orientation and position. Please note that it is still valid to assume a Cartesian measurement, as we can assume that the transformation and removal of the bias has been done in a preprocessing step.

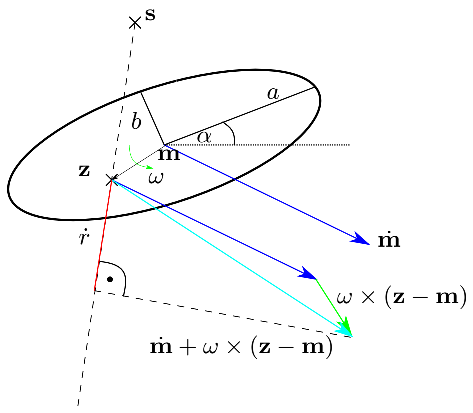
For the calculation of the range rate, we define the sensor position s $\in \mathbb{R}^2$ and the true Cartesian measurement source in local sensor coordinates $\mathbf{z}_l = \begin{bmatrix} z_{1,l} & z_{2,l} \end{bmatrix}^{\mathrm{T}} = \mathbf{z} - \mathbf{s}$ as well as the target's center in local sensor coordinates $\mathbf{m}_l = \begin{bmatrix} m_{1,l} & m_{2,l} \end{bmatrix}^{\mathrm{T}} = \mathbf{m} - \mathbf{s}$. The formula for the range rate is then

$$\dot{r} = \frac{\mathbf{z}_l}{||\mathbf{z}_l||_2}\Big(\begin{bmatrix} \dot{m}_1 \\ \dot{m}_2 \end{bmatrix} + \omega \begin{bmatrix} -(z_{l,2} - m_{l,2}) \\ z_{l,1} - m_{l,1} \end{bmatrix} \Big) \ . \tag{3.30}$$

The velocity at a point on the target consists of the velocity of the target and an influence of the yaw rate, which depends on the direction and length of the vector from the target's center to the detected point. The velocity is then projected onto the measurement direction. This is visualized in Figure 3.3. To reduce the non-linearities, we use a pseudo-measurement for the range rate as suggested in [DHL04] by multiplying the original range measurement with the range rate measurement, resulting in

$$\dot{r}_p = \dot{r}r = \mathbf{z}_l\Big(\begin{bmatrix} \dot{m}_1 \\ \dot{m}_2 \end{bmatrix} + \omega \begin{bmatrix} -(z_{l,2} - m_{l,2}) \\ z_{l,1} - m_{l,1} \end{bmatrix} \Big) \ . \tag{3.31}$$

Due to the incorporation of the range measurement into the range rate measurement, the noise covariance needs to be changed as follows [DHL04]



$\mathbf{s}$
a
b
α
$\mathbf{m}$
$\mathbf{z}$
ω
$\dot{r}$
$\dot{\mathbf{m}}$
$\omega \times (\mathbf{z}-\mathbf{m})$
$\dot{\mathbf{m}} + \omega \times (\mathbf{z}-\mathbf{m})$

Figure 3.3: Calculation of the range rate measurement $\dot{r}$ derived at a measurement source $\mathbf{z}$. $\times$ denotes the cross product, the calculation is done as in 3.30. The actual range rate measurement $\dot{r}$ is the length of the red line.

$$\mathbf{C}^{\mathbf{v}}_{33} = \sigma_r^2\dot{r}^2 + \sigma_{\dot{r}}^2 r^2 + 3\sigma_r^2\sigma_{\dot{r}}^2 \ , \tag{3.32}$$

$$\mathbf{C}^{\mathbf{v}}_{31} = \mathbf{C}^{\mathbf{v}}_{13} = \cos(\theta)\sigma_r^2\dot{r} \ , \tag{3.33}$$

$$\mathbf{C}^{\mathbf{v}}_{32} = \mathbf{C}^{\mathbf{v}}_{23} = \sin(\theta)\sigma_r^2\dot{r} \ . \tag{3.34}$$

Note that this transformation holds only if the range and range rate are independent. If they are correlated by some factor ρ, this needs to be reflected in the formulas. We refer to [DHL04] for the corresponding formulas and more information on this topic. For this thesis, we assume $\rho = 0$.

To incorporate the pseudo range rate measurement, we define the following measurement equation

$$h(\mathbf{r},\mathbf{p}) = \left[h^c(\mathbf{r},\mathbf{p})^{\mathrm{T}} \quad h^{\dot{r}}(\mathbf{r},\mathbf{p})\right]^{\mathrm{T}} + \mathbf{v} \ , \tag{3.35}$$

$$h^c(\mathbf{r},\mathbf{p}) = \mathbf{m} + \mathbf{S}\mathbf{h} \ , \tag{3.36}$$

$$h^{\dot{r}}(\mathbf{r},\mathbf{p}) = \dot{m}_1 m_{1,l} + \dot{m}_2 m_{2,l} + (\dot{m}_1\mathbf{S}_1 + \dot{m}_2\mathbf{S}_2 - \omega m_{1,l}\mathbf{S}_2 + \omega m_{2,l}\mathbf{S}_1)\mathbf{h} \ , \tag{3.37}$$

with $\mathbf{S}_1$ and $\mathbf{S}_2$ as the rows of $\mathbf{S}$ as defined in (3.14).

As in the original MEM-EKF*, we linearize around $\mathbf{r}$ and $\mathbf{p}$

$$\mathbf{y} \approx \begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \\ \hat{\dot{m}}_1\hat{m}_{1,l} + \hat{\dot{m}}_2\hat{m}_{2,l} \end{bmatrix} + \mathbf{H}(\mathbf{r}-\hat{\mathbf{r}}) + \mathbf{S}_d\mathbf{h} + \begin{bmatrix} \mathbf{h}^{\mathrm{T}}\mathbf{J}^{\mathbf{r}}_1 \\ \mathbf{h}^{\mathrm{T}}\mathbf{J}^{\mathbf{r}}_2 \\ \mathbf{h}^{\mathrm{T}}\mathbf{J}^{\mathbf{r}}_3 \end{bmatrix} (\mathbf{r}-\hat{\mathbf{r}}) + \begin{bmatrix} \mathbf{h}^{\mathrm{T}}\mathbf{J}^{\mathbf{p}}_1 \\ \mathbf{h}^{\mathrm{T}}\mathbf{J}^{\mathbf{p}}_2 \\ \mathbf{h}^{\mathrm{T}}\mathbf{J}^{\mathbf{p}}_3 \end{bmatrix} (\mathbf{p}-\hat{\mathbf{p}}) + \mathbf{v} \ , \tag{3.38}$$

with $\mathbf{h} = \begin{bmatrix} h_1 & h_2 \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{J}_1^{\mathbf{p}}, \mathbf{J}_2^{\mathbf{p}}$ being the same as $\mathbf{J}_1, \mathbf{J}_2$ from the original formulas [YB19] and $\mathbf{S}_d$ and the Jacobians realized using the values of $\hat{\mathbf{p}}$ and $\hat{\mathbf{r}}$ and constructed as

$$\mathbf{S}_d = \begin{bmatrix} \mathbf{S} \\ \dot{m}_1\mathbf{S}_1 + \dot{m}_2\mathbf{S}_2 - \omega m_{1,l}\mathbf{S}_2 + \omega m_{2,l}\mathbf{S}_1 \end{bmatrix} , \tag{3.39}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \dot{m}_1 & \dot{m}_2 & m_{1,l} & m_{2,l} & 0 \end{bmatrix} , \tag{3.40}$$

$$\mathbf{J}_1^{\mathbf{r}} = \mathbf{J}_2^{\mathbf{r}} = \mathbf{0}_{2\times 5} , \tag{3.41}$$

$$\mathbf{J}_3^{\mathbf{r}} = \begin{bmatrix} -\omega\mathbf{S}_2^{\mathrm{T}} & \omega\mathbf{S}_1^{\mathrm{T}} & \mathbf{S}_1^{\mathrm{T}} & \mathbf{S}_2^{\mathrm{T}} & -m_{1,l}\mathbf{S}_2^{\mathrm{T}} + m_{2_l}\mathbf{S}_1^{\mathrm{T}} \end{bmatrix} , \tag{3.42}$$

$$\mathbf{J}_3^{\mathbf{p}} = \begin{bmatrix} \mathbf{J}_{31}^{\mathbf{p}} & \mathbf{J}_{32}^{\mathbf{p}} & \mathbf{J}_{33}^{\mathbf{p}} \end{bmatrix} , \tag{3.43}$$

$$\mathbf{J}_{31}^{\mathbf{p}} = \begin{bmatrix} l(\sin(\alpha)(-\dot{m}_1 - \omega m_{2,l}) + \cos(\alpha)(\dot{m}_2 - \omega m_{1,l})) \\ w(\cos(\alpha)(-\dot{m}_1 - \omega m_{2,l}) + \sin(\alpha)(-\dot{m}_2 + \omega m_{1,l})) \end{bmatrix} , \tag{3.44}$$

$$\mathbf{J}_{32}^{\mathbf{p}} = \begin{bmatrix} \sin(\alpha)(\dot{m}_2 - \omega m_{1,l}) + \cos(\alpha)(\dot{m}_1 + \omega m_{2,l}) \\ 0 \end{bmatrix} , \tag{3.45}$$

$$\mathbf{J}_{33}^{\mathbf{p}} = \begin{bmatrix} 0 \\ \sin(\alpha)(-\dot{m}_1 - \omega m_{2,l}) + \cos(\alpha)(\dot{m}_2 - \omega m_{1,l}) \end{bmatrix} . \tag{3.46}$$

We exclude the range rate measurement from the shape update, because the shape update contains greater non-linearities and the range rate is mainly influenced by the kinematic properties. As the algorithm uses a multiplicative error and linearizations based on the MEM-EKF* and extends the measurement equation by the range rate of Doppler RADAR, we call it the MEM-EKF*-D.

4 Extended Target Tracking with Self-occlusion

Contents

A limitation of the classic elliptic (or rectangular) extended target trackers described in the previous chapter is that they assume that measurements originate from the entire surface or, in case of the RHM based approaches, it is also possible to assume a distribution on the contour. On real RADAR or LIDAR data, however, the detection points are rather biased towards the visible side of the contour, e.g., due to self-occlusion, or originate from specific reflection points such as the wheels of a car. An example can be seen in Figure 4.1. Therefore, we present a tracker for more complex distributions. Based on the good performance of the MEM-EKF*, we use the multiplicative noise measurement model and the corresponding linearizations from it as basis. We describe our assumptions about the target and setting in Section 4.1. The tracker we propose is then presented in Section 4.2, given a simple rectangular model. Then, it is modified for the more complex distribution of measurements in the nuScenes dataset [CBL$^+$20] in Section 4.3. Just as in Chapter 3, we omit time and measurement indices for readability in this chapter.

Remark. *This chapter is based on the workshop publication [4]. It extends the publication by a discussion and implementation of the model for real data.*

4.1 Problem Setting

The main challenge in this chapter is the distribution of measurement sources for RADAR data. Usually, due to so-called self-occlusion, only the parts of the target on the sensor side are visible.

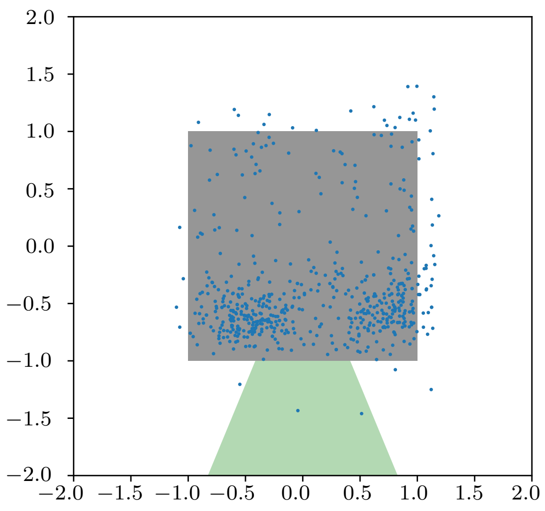


Figure 4.1: Example of a RADAR measurement distribution in unit coordinates, i.e., transformed into a coordinate system with the target's center at $\begin{bmatrix}0 & 0\end{bmatrix}$ and the shape being axis-aligned with semi-axes lengths equal to 1, as was done in [SD18], accumulated over a nuScenes [CBL$^+$20] scene. The ground truth scaled to unit coordinates is in gray and the measurements are blue dots. To show the self-occlusion, measurements were only considered if the sensor was in the green area, i.e., on the right side of the target.

In addition, the different parts of the target have different reflectivity, meaning the distribution is usually more complex than a simple uniform distribution. In Figure 4.1, which shows an instance of the nuScenes dataset [CBL$^+$20], it can be seen clearly that a simple uniform distribution does not capture the measurements distribution well. Instead, a GM representing all areas of the surface which are likely to generate measurements can be a more precise approximation for the multiplicative noise. We present the assumed target and measurement model in Section 4.1.1 and related works in Section 4.1.2.

4.1.1 State and Measurement Model

Regarding the target model in state of the art automotive tracking, other road users are often described by bounding boxes rather than ellipses. However, we can easily change the representation from ellipses to rectangles as they use the same parameterization: center, orientation, and semi-axes lengths. A uniform distribution on a rectangle instead of on an ellipse can be approximated by a Gaussian multiplicative noise [YB19] as well, changing the variance from $\frac{1}{4}$ to $\frac{1}{3}$. Therefore, we use the same decoupled kinematic and shape state as before, i.e.,

$$\mathbf{r} = \begin{bmatrix}\mathbf{m}^{\mathrm{T}} & \dot{\mathbf{m}}^{\mathrm{T}} & \dots\end{bmatrix}^{\mathrm{T}} , \tag{4.1}$$

$$\mathbf{p} = \begin{bmatrix}\alpha & a & b\end{bmatrix}^{\mathrm{T}} . \tag{4.2}$$

This target model and the aforementioned description of the measurement model result in the measurement equation

$$\mathbf{y} = \mathbf{H}\mathbf{r} + \mathbf{S}\mathbf{h} + \mathbf{v} \ , \tag{4.3}$$

$$\mathbf{S} = \mathbf{R}_\alpha \mathrm{diag}\Big(\begin{bmatrix} a & b \end{bmatrix} \Big) \ , \tag{4.4}$$

$$\mathbf{R}_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \ , \tag{4.5}$$

$$\mathbf{h} \sim \sum_{j=1}^{n} w_j \mathcal{N}(\hat{\mathbf{h}}_j, \mathbf{C}_j^{\mathbf{h}}) \ , \tag{4.6}$$

with $\mathbf{H}$ depending on the number of kinematic parameters so that $\mathbf{H}\mathbf{r} = \mathbf{m}$. The multiplicative noise $\mathbf{h}$ is modeled by a GM and each mixture component j consists of a mean $\hat{\mathbf{h}}_j$ and covariance $\mathbf{C}_j^{\mathbf{h}}$, both describing the area on the target in unit coordinates which generates detections, and a weight w_j, describing how likely the respective area produces a reflection point. Based on this formulation, $\mathbf{h}$ can be seen as distributed in unit coordinates, applicable to different semi-axes lengths and orientations of the target, similar to [SD18, KHB19, HK20].

4.1.2 Related Work

In [ZL20], a bias is introduced into the RM density. They provide a different method using a GM representation by providing different shifting and scaling parameters of the original RM in [ZL21], handling the update using a variational approach. In [XWB$^+$20a, XWB$^+$20b, XWB$^+$21], the RM density is truncated to model a distribution along the visible side.

The approach from [BRAD16] models a density with most of the probability mass on the contour and in [BDD17], they present the so-called volcanormal model, which models the distribution as a density across the contour of an ellipse, utilizing optimization techniques to update the target.

The method from [CLLL21] models the distribution across a rectangular shape via segments between the corners or across the surface which they realize as uniform distributions via a multiplicative scaling factor. They handle association via PDA and non-linearities via UKF. While their proposed method is made for an extent modeled by corners of a rectangle, they also provide a version using orientation, length, and width.

[GLO11] propose a different approach for measurements distributed across the contour of an ellipse or rectangle, analyzing the measurement distribution to determine how many sides are visible and assigning the measurements to equally spaced measurement sources. [AGS19] assumes a distribution across the contour of an ellipse, discretizes it, updates with each individual measurement source to create a GM posterior, and then uses moment matching to reduce it back to a single component. [YWB$^+$21] model the distribution via ellipses which are located on a contour described by a B-spline and handle association via PMHT.

The RHM based approaches, e.g., [BH14, WÖ15, HSRD16, TLTK19, KBW17] can handle distribution across the contour and due to their greedy association model, they can handle only part of the contour to be visible. The greedy association model comes with other downsides, however, such as a bias in high noise and a strong dependency on the prior estimate.

Approaches using a parameterization suitable for ellipses, but modeling a more complex distribution across the shape include [SD18, KHB19, HK20]. They model the distribution across the shape via a GM. For the update, [SD18] define a likelihood and use a particle filter. The other two approaches define a closed-form update via association of the measurements to components of the GM. They handle this via EM in [KHB19] or via cluster processes in [HK20]. These approaches do, however, learn their measurement distribution, requiring large amounts of data. In addition, they learn the sensor noise as part of the GM, while in reality, it can vary depending on the sensor to target geometry. On the other hand, we use the linearization from the MEM-EKF* to provide a model which keeps the distribution across the shape and the sensor noise separate.

4.2 Proposed Tracker

We use the linearizations from the MEM-EKF* [YB19] as a basis for the filter to deal with the multiplicative noise, with the difference that we consider it to be a GM and not a single zero-mean Gaussian. In addition, this formulation keeps the measurement noise separate from the multiplicative noise. This can be important, as the sensor noise can depend on the sensor to target geometry, e.g., when transforming it from polar to Cartesian coordinates as described in Section 3.3.1.

For our tracker, we have to consider three additional challenges here

- the *weights* of the GM need to be chosen appropriately,
- an expectation of a component of $\mathbf{h}$ that is non-zero must be considered in the *update*, and
- the posterior will be a *mixture*.

Weights The weights represent how likely a component generates a measurement. This depends on the target and sensor properties. In general, one can assume measurements are more likely to stem from the visible side of the target. Therefore, the weight should depend on the sensor position relative to the target, see, e.g., Figure 4.1 or the sensor-dependent distributions in [KHB19]. Here, we propose a straightforward approach. We find the weight for a component j by first transforming the sensor position $\mathbf{s}$ relative to the target into unit coordinates. Then, we find the angle difference θ_j between it and the component's mean $\hat{\mathbf{h}}_j$ to get the weight

$$w_j = \begin{cases} 1.0 & \text{if } \theta_j < 0.4\pi \ , \\ 0.5 & \text{if } 0.4\pi \leq \theta_j < 0.7\pi \ , \\ 0.001 & \text{otherwise} \ , \end{cases} \tag{4.7}$$

$$\theta_j = |(\psi_s - \measuredangle(\hat{\mathbf{h}}_j) + \pi \mod (2\pi)) - \pi| \ , \tag{4.8}$$

$$\psi_{\mathbf{s}} = \measuredangle(\mathbf{s}_{\mathbf{x}}) \ , \tag{4.9}$$

$$\mathbf{s}_{\mathbf{x}} = \text{diag}\Big(\begin{bmatrix} \frac{1}{a} & \frac{1}{b} \end{bmatrix} \Big) \mathbf{R}_\alpha^{\mathrm{T}} (\mathbf{s} - \mathbf{m}) \ , \tag{4.10}$$

with $\mathbf{s}_{\mathbf{x}}$ describing the sensor position transformed into target unit coordinates, $\psi_{\mathbf{s}}$ being the angle from the target's center to the sensor in unit coordinates, and θ_j representing the absolute difference between the angle from the center to the sensor and the angle from the center to the mean of component j. This is also visualized later in this section in Figure 4.2. The absolute value is taken as the direction does not matter, only the difference. Components with a small angle difference θ_j have a high weight and those with large angle difference have a low weight as we assume they are occluded by the rest of the target. Components in between get a relatively high weight as well. There are two reasons: the first is that it is hard to find when exactly a component stops generating measurements and the other more important reason is that the true state is not known. Therefore, the equations above must be approximated with the state estimate. If the estimate is off, the wrong side can get a higher weight than the correct one. Therefore, neighboring components should get a descent weight and the rest can be handled via the predicted likelihood. Our initial experiments revealed that this approach produces better updates than giving the neighboring components a low weight. Note that for more complex distributions, a different strategy might be necessary, e.g., choosing different set of weights or even different component means and covariances based on $\psi_{\mathbf{s}}$.

Update The original MEM-EKF* [YB19] demonstrated how the covariance of the multiplicative noise can be incorporated into the shape update. This is important as it essentially describes the spread of the measurement sources. But as the multiplicative noise does not need to be zero-mean anymore, the resulting bias needs to be incorporated into the expected measurements and into the covariance as well (as similar to the covariance of the multiplicative noise, its mean influences the innovation covariance originating from the shape uncertainty, i.e., $\mathbf{C}_{nm}^{\mathrm{II}}$ from (3.19), which needs to be modified in the following). Ignoring the measurement and time indices for readability, the update formulas for a mixture component j are as follows

- the new expected measurement is

$$\bar{\mathbf{y}}_j = \mathbf{H}\hat{\mathbf{r}} + \mathbf{S}\hat{\mathbf{h}}_j \ , \tag{4.11}$$

with $\hat{\mathbf{h}}_j$ being the mean of the jth GM component, $\mathbf{S}$ being realized using $\hat{\mathbf{p}}$, and $\hat{\mathbf{r}}$ and $\hat{\mathbf{p}}$ being the latest estimates of the kinematic and shape states.

- the covariance $\mathbf{C}^{\mathbf{h}}$ in $\mathbf{C}^{\mathrm{I}}$ from equation (3.18) and $\mathbf{M}$ from equation (3.22) is replaced with $\mathbf{C}^{\mathbf{h}}_j$.
- every occurrence of $\mathbf{C}^{\mathbf{h}}$ in $\mathbf{C}^{\mathrm{II}}$ from equation (3.19) is replaced by $\mathbf{C}^{\mathbf{h}}_j + \hat{\mathbf{h}}_j\hat{\mathbf{h}}_j^{\mathrm{T}}$. For example, the mth row and nth column $\mathbf{C}^{\mathrm{II}}_{mn}$ of covariance $\mathbf{C}^{\mathrm{II}}$ becomes

$$\mathbf{C}^{\mathrm{II}}_{mn} = \mathrm{tr}\left(\mathbf{C}^{\mathbf{p}}\mathbf{J}_n^{\mathrm{T}}\left(\mathbf{C}^{\mathbf{h}}_j + \hat{\mathbf{h}}_j\hat{\mathbf{h}}_j^{\mathrm{T}}\right)\mathbf{J}_m\right) \tag{4.12}$$

for $m, n = 1, 2$.

According to [4], details on the changes in the derivations of $\mathbf{C}^{\mathrm{II}}$ and $\mathbf{M}$ are as follows. Recap that for $\mathbf{C}^{\mathrm{II}}$, we have [YB19]

$$\mathbf{C}^{\mathrm{II}} = \mathrm{cov}\left\{\begin{bmatrix}\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_1\\ \mathbf{h}_j^{\mathrm{T}}\mathbf{J}_2\end{bmatrix}(\mathbf{p}-\hat{\mathbf{p}}), \begin{bmatrix}\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_1\\ \mathbf{h}_j^{\mathrm{T}}\mathbf{J}_2\end{bmatrix}(\mathbf{p}-\hat{\mathbf{p}})\right\} ,$$

with $\mathbf{J}_1, \mathbf{J}_2$ referring to the Jacobians used in (3.17). If $\mathbf{h}_j$ is not zero-mean, the formula (3.19) changes to

$$\begin{aligned}\mathbf{C}^{\mathrm{II}}_{mn} &= \mathrm{E}\big[\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_m(\mathbf{p}-\hat{\mathbf{p}})(\mathbf{p}-\hat{\mathbf{p}})^{\mathrm{T}}\mathbf{J}_n^{\mathrm{T}}\mathbf{h}_j\big]\\ &= \mathrm{E}\big[\mathrm{tr}\big(\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_m(\mathbf{p}-\hat{\mathbf{p}})(\mathbf{p}-\hat{\mathbf{p}})^{\mathrm{T}}\mathbf{J}_n^{\mathrm{T}}\mathbf{h}_j\big)\big]\\ &= \mathrm{E}\big[\mathrm{tr}\big((\mathbf{p}-\hat{\mathbf{p}})(\mathbf{p}-\hat{\mathbf{p}})^{\mathrm{T}}\mathbf{J}_n^{\mathrm{T}}\mathbf{h}_j\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_m\big)\big]\\ &= \mathrm{tr}\big(\mathrm{E}\big[(\mathbf{p}-\hat{\mathbf{p}})(\mathbf{p}-\hat{\mathbf{p}})^{\mathrm{T}}\mathbf{J}_n^{\mathrm{T}}\mathbf{h}_j\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_m\big]\big)\\ &= \mathrm{tr}\big(\mathbf{C}^{\mathbf{p}}\mathbf{J}_n^{\mathrm{T}}(\mathbf{C}^{\mathbf{h}}_j + \mathbf{h}_j\hat{\mathbf{h}}_j^{\mathrm{T}})\mathbf{J}_m\big) \ .\end{aligned} \tag{4.13}$$

Keep in mind that $\mathbf{h}_j$ and $\mathbf{p}$ are assumed to be independent. The formula shows that not just the covariance, but also the mean of $\mathbf{h}_j$ takes a role in determining the influence of the shape covariance on the innovation covariance.

To calculate $\mathbf{M}$, we take the measurement equation for the pseudo-measurement [YB19]

$$h_p(\mathbf{r},\mathbf{p}) \quad = \left[h_p^{(11)} \quad h_p^{(22)} \quad h_p^{(12)}\right]^{\mathrm{T}} \ , \tag{4.14}$$

$$h_p^{(nm)}(\mathbf{r},\mathbf{p}) = (\mathbf{H}_n\mathbf{r} + \mathbf{S}_n\mathbf{h}_j + v_n - \overline{y}_n)(\mathbf{H}_m\mathbf{r} + \mathbf{S}_m\mathbf{h}_j + v_m - \overline{y}_m) \ , \tag{4.15}$$

with the subscripts n, m referring to the corresponding row of the respective vector or matrix and $\overline{\mathbf{y}}$ from (4.11). We now linearize this equation while considering a non zero-mean $\mathbf{h}_j$ and take the expectation, i.e., $\mathbf{M} = \mathrm{E}\big[\frac{\partial h_p(\mathbf{r},\mathbf{p})}{\partial \mathbf{p}}\big]$. For the first row, we have

$$\begin{aligned}
&\mathrm{E}\big[2(\mathbf{H}_1\mathbf{r}+\mathbf{S}_1\mathbf{h}_j+v_1-\overline{y}_1)\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_1\big]\\
&=2\mathrm{E}\big[(\mathbf{H}_1\mathbf{r}-\mathbf{H}_1\hat{\mathbf{r}})\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_1\big] && (4.16)\\
&\quad+2\mathrm{E}\big[(\mathbf{S}_1\mathbf{h}_j\mathbf{h}_j^{\mathrm{T}}-\widehat{\mathbf{S}}_1\hat{\mathbf{h}}_j\mathbf{h}_j^{\mathrm{T}})\mathbf{J}_1\big] && (4.17)\\
&\quad+2\mathrm{E}\big[v_1\mathbf{h}_j^{\mathrm{T}}\mathbf{J}_1\big] && (4.18)\\
&=2\widehat{\mathbf{S}}_1(\mathrm{E}[\mathbf{h}_j\mathbf{h}_j^{\mathrm{T}}]-\hat{\mathbf{h}}_j\mathrm{E}[\mathbf{h}_j]^{\mathrm{T}})\mathbf{J}_1 && (4.19)\\
&=2\widehat{\mathbf{S}}_1\mathbf{C}_j^{\mathbf{h}}\mathbf{J}_1\ \ . && (4.20)
\end{aligned}$$

Note that $\mathbf{r}$ and $\mathbf{h}$ as well as $\mathbf{v}$ and $\mathbf{h}$ are independent, resulting in (4.16) and (4.18) being 0. The second and third row in $\mathbf{M}$ are derived analogously. The formula shows that $\mathbf{M}$ stays the same.

Mixture The updates with a measurement $\mathbf{y}$ using each of the n components leaves us with a resulting GM of n components

$$\mathbf{r}|\mathbf{y}\sim\sum_{j=1}^{n}w_j^{+}\mathcal{N}(\hat{\mathbf{r}}_j,\mathbf{C}_j^{\mathbf{r}})\ \ , \tag{4.21}$$

$$\mathbf{p}|\mathbf{y}\sim\sum_{j=1}^{n}w_j^{+}\mathcal{N}(\hat{\mathbf{p}}_j,\mathbf{C}_j^{\mathbf{p}})\ \ , \tag{4.22}$$

with $w_j^{+}=w_j\cdot l_j^{(\mathbf{y})}$ and $l_j^{(\mathbf{y})}$ as the predicted likelihood of the update with the jth component of $\mathbf{h}$

$$l_j^{(\mathbf{y})}=\frac{1}{2\pi\mathrm{det}(\mathbf{C}^{\mathbf{y}})}e^{(\mathbf{y}-\overline{\mathbf{y}}_j)^{\mathrm{T}}(\mathbf{C}^{\mathbf{y}})^{-1}(\mathbf{y}-\overline{\mathbf{y}}_j)}\ \ , \tag{4.23}$$

with $\overline{\mathbf{y}}_j$ from (4.11) and $\mathbf{C}^{\mathbf{y}}$ from (3.18), but using $\mathbf{C}^{\mathrm{I}}$ and $\mathbf{C}^{\mathrm{II}}$ as described in this section. After each sequential measurement update, the GM is reduced to a single component via moment matching as in, e.g., [BNH11]. The corresponding formulas are [GO12])

$$\hat{\mathbf{r}}=\sum_{j=1}^{n}w_j^{+}\hat{\mathbf{r}}_j\ \ , \tag{4.24}$$

$$\mathbf{C}^{\mathbf{r}}=\sum_{j=1}^{n}w_j^{+}(\mathbf{C}_j^{\mathbf{r}}+(\hat{\mathbf{r}}_j-\hat{\mathbf{r}})(\hat{\mathbf{r}}_j-\hat{\mathbf{r}})^{\mathrm{T}})\ \ , \tag{4.25}$$

$$\hat{\mathbf{p}}=\sum_{j=1}^{n}w_j^{+}\hat{\mathbf{p}}_j\ \ , \tag{4.26}$$

$$\mathbf{C}^{\mathbf{p}}=\sum_{j=1}^{n}w_j^{+}(\mathbf{C}_j^{\mathbf{p}}+(\hat{\mathbf{p}}_j-\hat{\mathbf{p}})(\hat{\mathbf{p}}_j-\hat{\mathbf{p}})^{\mathrm{T}})\ \ . \tag{4.27}$$

```
input  : prior kinematic and shape mean and covariance $\hat{\mathbf{r}}^{(0)}$, $\mathbf{C}^{\mathbf{r},(0)}$, $\hat{\mathbf{p}}^{(0)}$, and $\mathbf{C}^{\mathbf{p},(0)}$, $n$ multiplicative noise
         means and covariances $\hat{\mathbf{h}}$ and $\mathbf{C}^{\mathbf{h}}$ as defined in, e.g., (4.28) and (4.29), $m$ measurements $\mathbf{y}$, and global
         sensor position $\mathbf{s}$
output : posterior kinematic and shape mean and covariances $\hat{\mathbf{r}}^{(m)}$, $\mathbf{C}^{\mathbf{r},(m)}$, $\hat{\mathbf{p}}^{(m)}$, and $\mathbf{C}^{\mathbf{p},(m)}$
$\psi_{\mathbf{s}} \leftarrow$ (4.9), (4.10) using $\hat{\mathbf{r}}^{(0)}$, $\hat{\mathbf{p}}^{(0)}$, and $\mathbf{s}$
for $j \in \{1 \dots n\}$ do
    $w_j \leftarrow$ (4.7), (4.8) using $\psi_{\mathbf{s}}$ and $\hat{\mathbf{h}}_j$
end
for $i \in \{1 \dots m\}$ do
    for $j \in \{1 \dots n\}$ do
        $\hat{\mathbf{r}}_j^{(i)}, \mathbf{C}_j^{\mathbf{r},(i)}, \hat{\mathbf{p}}_j^{(i)}, \mathbf{C}_j^{\mathbf{p},(i)}, l_j \leftarrow$ memekfstar($\hat{\mathbf{r}}^{(i-1)}, \mathbf{C}^{\mathbf{r},(i-1)}, \hat{\mathbf{p}}^{(i-1)}, \mathbf{C}^{\mathbf{p},(i-1)}, \hat{\mathbf{h}}_j, \mathbf{C}_j^{\mathbf{h}}$)
        $w_j^+ \leftarrow w_j \cdot l_j$
    end
    $w^+ \leftarrow$ normalize($w^+$)
    $\hat{\mathbf{r}}^{(i)} \leftarrow$ (4.24) using all $w_j^+$ and all $\hat{\mathbf{r}}_j^{(i)}$
    $\hat{\mathbf{p}}^{(i)} \leftarrow$ (4.26) using all $w_j^+$ and all $\hat{\mathbf{p}}_j^{(i)}$
    $\mathbf{C}^{\mathbf{r},(i)} \leftarrow$ (4.25) using all $w_j^+$ and all $\mathbf{C}_j^{\mathbf{r},(i)}$
    $\mathbf{C}^{\mathbf{p},(i)} \leftarrow$ (4.27) using all $w_j^+$ and all $\mathbf{C}_j^{\mathbf{p},(i)}$
end
return $\hat{\mathbf{r}}^{(m)}, \mathbf{C}^{\mathbf{r},(m)}, \hat{\mathbf{p}}^{(m)}, \mathbf{C}^{\mathbf{p},(m)}$
```

Algorithm 3: Measurement update procedure using GM-MEM-EKF*. The function `memekfstar()` takes kinematic and shape mean and covariance as well as a multiplicative noise term, updates according to the GM-MEM-EKF* formulas modified as described in Chapter 4, and returns the update's kinematic and shape mean and covariance as well as the predicted likelihood. The function `normalize()` normalizes the input weights. Based on [4].

Finally, to provide an example, we assume a rectangular target generating measurements from its contour. Only the visible side of the rectangle can generate measurements. To describe this model, we chose $n = 4$ components, one for each side, as follows

$$\hat{\mathbf{h}} = \begin{bmatrix} 0.9 & 0.0 & -0.9 & 0.0 \\ 0.0 & 0.9 & 0.0 & -0.9 \end{bmatrix} , \tag{4.28}$$

$$\mathbf{C}^{\mathbf{h}} = \begin{bmatrix} \mathrm{diag}\left(\begin{bmatrix} \frac{1}{300} & \frac{1}{3} \end{bmatrix} \right) \\ \mathrm{diag}\left(\begin{bmatrix} \frac{1}{3} & \frac{1}{300} \end{bmatrix} \right) \\ \mathrm{diag}\left(\begin{bmatrix} \frac{1}{300} & \frac{1}{3} \end{bmatrix} \right) \\ \mathrm{diag}\left(\begin{bmatrix} \frac{1}{3} & \frac{1}{300} \end{bmatrix} \right) \end{bmatrix} , \tag{4.29}$$

so that the component approximates a uniform distribution over the respective side and assumes a uniform distribution within the first 10% of the target's depth to account for uncertainties. The tracker is called GM-MEM-EKF* and we summarize it in Algorithm 3 and an example setup and how the weights are chosen can be found in Figure 4.2. Note that the choice of components and

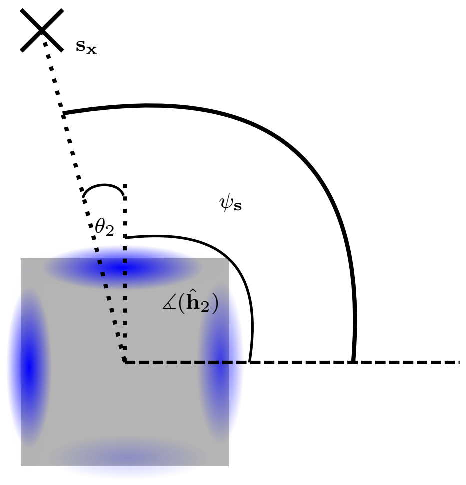


Figure 4.2: Setup of 4 GM components in unit coordinates with the origin at the gray rectangle's center and the local sensor in unit coordinates at $\mathbf{s_x}$. The orientation of the upper component, $\mathbf{h}_2$, is displayed. The target is in gray, the GM components in blue and their weights are represented by the intensity of the components' color. [4]©2021IEEE

the weighting scheme depend on the application scenario. They could be replaced by different models, e.g., learned models from a real data set [KHB19, SD18, HK20].

4.3 Modifications for Real Data

To demonstrate our approach on real data, we add a hand-tuned version of the model for nuScenes [CBL+20]. The dataset provides only few RADAR detections per target and time step, so we looked into four selected targets with higher measurement rate (while nuScenes offers an accumulation of measurements, this can lead to smears of the RADAR data for moving targets). For more technical details on the nuScenes dataset and the sensor setup, we refer to Section 3.1.3. As can be seen in Figure 4.1, the measurements are distributed on the visible side, but can also originate from the inside of the target. Therefore, the GM and the rules for the weights are modified as follows

$$\hat{\mathbf{h}} = \begin{bmatrix} 0.0 & 0.85 & 0.0 & -0.85 & 0.0 \\ 0.0 & 0.0 & 0.85 & 0.0 & -0.85 \end{bmatrix} , \tag{4.30}$$

$$\mathbf{C^h} = \begin{bmatrix} \mathrm{diag}\Big(\begin{bmatrix}\frac{49}{300} & \frac{49}{300}\end{bmatrix}\Big) \\ \mathrm{diag}\Big(\begin{bmatrix}\frac{3}{400} & \frac{1}{3}\end{bmatrix}\Big) \\ \mathrm{diag}\Big(\begin{bmatrix}\frac{1}{3} & \frac{3}{400}\end{bmatrix}\Big) \\ \mathrm{diag}\Big(\begin{bmatrix}\frac{3}{400} & \frac{1}{3}\end{bmatrix}\Big) \\ \mathrm{diag}\Big(\begin{bmatrix}\frac{1}{3} & \frac{3}{400}\end{bmatrix}\Big) \end{bmatrix} , \tag{4.31}$$

$$w_j = \begin{cases} 0.4 & \text{if } j == 1 \ , \\ 1.0 & \text{if } \theta_j < 0.45\pi \ , \\ 0.05 & \text{otherwise} \ . \end{cases} \tag{4.32}$$

The intention was to allow a wider distribution across the contour. The model assumes that to a certain probability, measurements can originate from the inside as well. For actual learned distributions, e.g., [KHB19, HK20, SD18], the number of components is usually much larger. For the hand-tuned model, we found that a simpler model keeps the estimate more stable using the moment matching approach. We comment more on this later in the discussion in Chapter 7.

5

Fusion and Estimation on Ellipse Densities

Contents

Until now, we described the tracking of single extended targets in single sensor scenarios. In the last chapter, we considered the challenge of self-occlusion and how it affects the distribution of measurement sources. While a more complex model for the distribution helps to keep track of the target, we are still missing information if, e.g., one side of the target is fully occluded. One way to combat this is the use of multiple cooperating sensors, each of which tracks the target locally and then provides its estimate along with the covariance. Different topologies for such sensor networks are possible, but our main concern in this chapter is the fusion process itself, so they will not be discussed in detail here. If we have multiple state densities in the current state of the art, we can combine them using the Kalman filter formulas. To then get a point estimate on the fused posterior density, in classic estimation, an estimate is gained by calculating its expectation. In the case of a Gaussian density, this means simply taking its mean. However, if our state is an ellipse described by a center, orientation, semi-axes lengths, and kinematic parameters, counter-intuitive estimates or fusion results can arise using these classic methods as the geometric properties of ellipses are not considered in them. This problem is described in detail in Section 5.1. We propose a special density, an RED, for the fusion of ellipses in Section 5.2 and an MMGW estimator for ellipses in Section 5.3, which replaces the Euclidean with the GW distance. Details on the methods' implementations are provided in Section 5.4 and their applicability to rectangles, which can be parameterized in the same way as ellipses, is discussed in Section 5.5.

To the best of our knowledge, there is no other method in literature fusing Gaussian densities of ellipses parameterized by orientation and semi-axes lengths while considering their geometric properties. There are only methods fusing their point estimates, i.e., the means of the Gaussian densities, as in [RX20]. We also know of no approach replacing the cost function of the estimation of extended targets with a metric on shapes. This idea is, however, similar and inspired by the Minimum Mean Optimal Sub-Pattern Assignment (MMOSPA) estimators [GSSW10, BWH15], which replaced the cost function in the estimation of multiple point targets [VMBS$^+$15] by the OSPA distance [SVV08]. As both of these concepts essentially model the shape/targets as densities and find a density minimizing a Wasserstein distance, they are related to the concept of a Wasserstein barycenter [AC11, JPDM11, COO15, CD14].

Remark. *This chapter is based on the conference publication [2], which describes fusion and estimation on ellipse densities, as well as the journal publication [3], in which the MMGW estimation is presented and evaluated in greater detail and which introduces the concept of an RED for fusion. In addition, this thesis extends these publications via a discussion and implementation of estimators on rectangles.*

5.1 Problem Setting

We consider sensors providing estimates for elliptic extended targets. For this chapter, the sensors are handled as black-boxes, i.e., they track targets internally and only provide the mean and covariance of the target, not the entire point cloud. Examples for trackers they can use internally are the ones from Chapter 3 or Chapter 4. In either case, we assume the ellipses we track are modeled as in Section 5.1.1. We further assume that we have a central fusion unit and the sensors provide their estimates along with their covariances to it. This is described in Section 5.1.2. The challenges of fusing the local state densities, provided by the sensors, with the global state density in the central fusion unit and of gaining an intuitive point estimate from the density is elaborated in Section 5.1.3. Related works in this area are discussed in Section 5.1.4.

5.1.1 State

Just as in the previous chapter, we model the ellipse using a center $\mathbf{m} = \begin{bmatrix} m_1 & m_2 \end{bmatrix}^{\mathrm{T}}$, an orientation α, and semi-axis lengths a and b (see Figure 5.1a). Kinematics are modeled by a velocity $\dot{\mathbf{m}}$ and optional other parameters, such as the yaw rate. Based on (3.1), we get

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{m}_{\mathbf{x},k}^{\mathrm{T}} & \dot{\mathbf{m}}_{\mathbf{x},k}^{\mathrm{T}} & \dots & \alpha_{\mathbf{x},k} & a_{\mathbf{x},k} & b_{\mathbf{x},k} \end{bmatrix}^{\mathrm{T}} . \tag{5.1}$$

Each sensor keeps track of the entire shape, either because it can detect the entire shape or it can handle only part of the shape being visible (see Chapter 4 for more details).

5.1.2 Fusion Model

As the focus is on the unique properties of fusing elliptic targets, we will only consider a single target scenario with multiple sensors. As stated above, we get an estimate $\hat{\mathbf{x}}_{s,k}$ for each sensor s at time k, parameterized as described in Equation (5.1). The sensors also provide the uncertainties of the estimates via a covariance matrix $\mathbf{C}_{s,k}$. These local estimates and covariances are sent to a central fusion unit to be fused with the global estimate $\mathbf{x}_k$, which is modeled by a Gaussian density. The fusion can be conducted using the Kalman filter formulas, e.g., [ASKB12] by relating the estimates and global state as follows

$$\mathbf{x}_{s,k} = \mathbf{H}_s \mathbf{x}_k + \mathbf{v}_{s,k} \ , \tag{5.2}$$

with $\mathbf{v}_{s,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{s,k})$ and $\mathbf{H}_s = \mathbf{I}_d$ with $\mathbf{I}_d$ being the d-dimensional identity matrix and d the dimension of $\mathbf{x}$. If not all of the state elements are provided by the sensor, the corresponding rows of $\mathbf{H}_s$ can be deleted.

This method provides an easy fusion approach, which does, however, ignore possible correlations between the estimates [ASKB12]. As the focus of this work is on handling the unique geometric properties of ellipses during the fusion process, we assume $\mathbf{x}_k$ and all $\mathbf{v}_{s,k}$ are independent for all s. This is not always given in practice and methods described in Section 2.1.5 can be used to handle correlations. How to incorporate them into the methods introduced in this chapter will be commented on in Section 5.4.1.

5.1.3 Problem Analysis

For brevity's sake, we will omit the time index k from here on. Given the aforementioned global estimate $\mathbf{x}$ with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{C}$ and the local sensor estimate $\hat{\mathbf{x}}_s$ with sensor noise modeled by the covariance $\mathbf{C}_s$, they can be combined with the Kalman filter formulas. This naive approach is based on Bayesian fusion

$$p(\mathbf{x}|\hat{\mathbf{x}}_s) \sim p(\hat{\mathbf{x}}_s|\mathbf{x}) \cdot p(\mathbf{x}) \ , \tag{5.3}$$

with prior $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{C})$ and Gaussian likelihood $\hat{\mathbf{x}}_s|\mathbf{x} \sim \mathcal{N}(\mathbf{x}, \mathbf{C}_s)$. An estimate is calculated by finding the Minimum Mean Square Error (MMSE) estimate using the squared Euclidean distance as cost function

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\text{argmin}} \int ||\mathbf{z} - \mathbf{x}||_2^2 \cdot p(\mathbf{x}|\hat{\mathbf{x}}_s) \mathrm{d}\mathbf{x} = \mathrm{E}[\mathbf{x}|\hat{\mathbf{x}}_s] \ . \tag{5.4}$$

However, when fusing elliptic targets, this approach can yield counter-intuitive results. To give an example, consider a global estimate as in Figure 5.1a and a local sensor estimate as in Figure 5.1b.

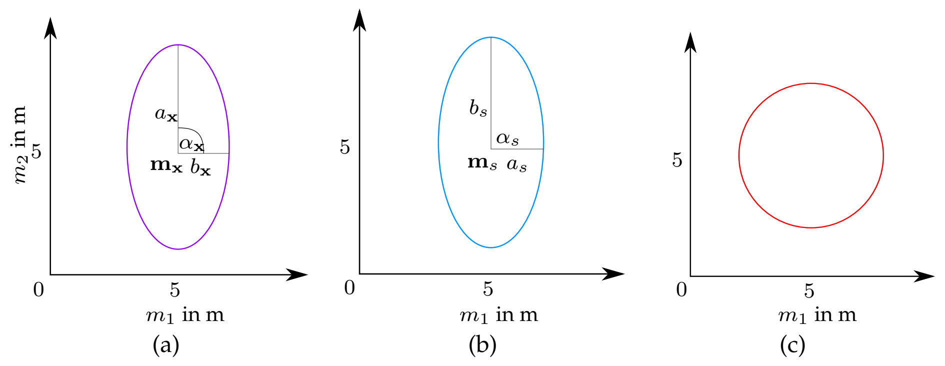


Figure 5.1: Ellipses with parameters $\mathbf{m}_{\mathbf{x}} = [5 \quad 5]^{\mathrm{T}}$, $\alpha_{\mathbf{x}} = \frac{\pi}{2}$, $a_{\mathbf{x}} = 4$, and $b_{\mathbf{x}} = 2$ in Figure 5.1a and $\mathbf{m}_s = [5 \quad 5]^{\mathrm{T}}$, $\alpha_s = 0$, $a_s = 2$, and $b_s = 4$ in Figure 5.1b and their MMSE estimate using Euclidean distance in Figure 5.1c. [3]©2021IEEE

If both the global estimate and the sensor noise had the same covariance, the Kalman filter would provide the fused estimate as the average of the two means (see the formulas in Section 2.1.2), i.e.,

$$\hat{\mathbf{z}} = \frac{1}{2}(\hat{\mathbf{x}} + \hat{\mathbf{x}}_s) \ , \tag{5.5}$$

which can be found in Figure 5.1c. It is clearly visible that the global estimate and the local sensor estimate describe the same ellipse, yet the fusion looks different. This is due to ambiguities in the ellipse parameterization. Both ellipses are geometrically identical, but the sensor estimate is rotated by 0.5π and semi-axes lengths are swapped. As a result, the naive fusion would average the wrong axes with each other ($a_{\mathbf{x}}$ with a_s and $b_{\mathbf{x}}$ with b_s in Figures 5.1a and 5.1b).

Examples for the origins of these ambiguities are

- sensor placement, i.e., sensors initializing a target with the orientation facing away from the sensor,
- high process noise, i.e., due to a strong orientation shift, the estimate might rotate too strongly, or
- different inner workings of the trackers, i.e., an orientation shift of the ground truth of 0.5π can lead a tracker to also shift the orientation or to exchange the semi-axes lengths.

In addition to the same ellipse parameterized in an ambiguous way, it can be unclear how different ellipses should be associated. As an example, we consider an ellipse and the same ellipse rotated $90°$ to the first one. It is unclear whether it is better to align the orientations or the semi-axes lengths or to find a different combination (see Figure 5.2).

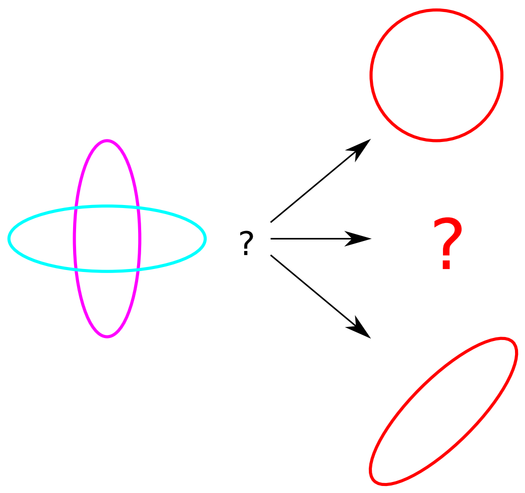


Figure 5.2: Illustration of the possibilities of combining the magenta and cyan ellipse into one of the red ellipses. For the upper one, we assume the orientation is the same and average the overlapping semi-axes, while for the lower one, we pair the semi-axes of the same lengths and average the orientations.

The main reason for these counter-intuitive results is that the naive approach considers the state as elements of a vector, ignoring the geometric dependencies of the described ellipses. Looking at Equation (5.4) we can split the issue with the naive approach into two parts

(1) the fusion by means of Equation (5.3) and

(2) the squared Euclidean distance $||\mathbf{z} - \mathbf{x}||_2^2$ as cost function.

While one solution would be to transform the state into shape matrices and combine them, this would ignore the uncertainties. We do, however, want to fuse the entire state density, not just combine the estimates. We handle the two challenges that we identified above in the following two sections by providing

(1) a novel probability density for elliptic shapes in Section 5.2, which considers the ambiguities of ellipses and allows for Bayesian fusion and

(2) an MMGW estimator which uses the squared GW distance as a cost function to obtain intuitive point estimates in Section 5.3.

As the focus lies on the shape state, we will omit the kinematic part in the remainder of this chapter for readability, only using center, orientation, and semi-axes lengths.

5.1.4 Related Work

For elliptic targets, many fusion approaches exist based on RMs. [GO13] propose to combine RMs using their Poisson rates, creating a shape that would have been created by the measurements creating the input shapes, which is more a combination of the shapes than a fusion of the densities. Another approach in [LJMD15] proposes to minimize the Kullback-Leibler divergence between the fused density and the input densities, all as inverse Wishart densities, by taking the weighted average of the shape matrices (and the degrees of freedom). The fusion method from [VGBW17] uses the point clouds directly. However, one of their approaches uses a particle filter with particles drawn from an importance distribution and weighted using the RM likelihood. For the estimate, the weighted average of the particle shape matrices is calculated. That approach was extended in [LG19] for asynchronous sensors, which approximates the local particle densities as a GM, sends only the GM, and then fuses them via geometric mean densities. A different approach presented in [HL19] fuses the estimates of different sensors using the statistics of noise-free latent variables gained from the RMs at each sensor. They combine this with a variational Bayesian approach.

An approach using ellipses parameterized with orientation and semi-axes lengths in a distributed sensor network based on diffusion Kalman filter can be found in [RX20]. However, they only combine the estimates without considering the covariances and share the point clouds between the sensors as well.

For other shapes, there is a method fusing rectangular estimates based on the covariances of their corners in [NK16] and another using one corner as a reference point in [HMSB19]. [DGN+16] presents a fusion method for segments, i.e., points, lines, or L-shapes. A combination of star-convex shapes modeled by GPs using measurements from different sensors directly can be found in [MBMW17] and an approach fusing by finding a new center and radial function is presented in [MBL+19]. For multi-target tracking using a Random Finite Set (RFS), the RFSs provided by different sensors are combined by minimizing the Kullback-Leibler divergence in [FGW20]. Another central fusion approach sharing detections directly to update a rectangular target can be found in [SKRD16].

Fusion of bounding boxes can also be found in computer vision. In [SWG21], a method is presented which clusters and fuses bounding boxes received from different detection methods or from the original and augmented versions of the same input image. The fusion is conducted using the confidence scores of the bounding boxes. Another approach by [BPS14] finds the best fusion result of a set of bounding boxes by defining a distance measure based on position and size dimensions, which they call attraction, and then optimize using gradient ascent. Optionally, weights can be provided based on how well the algorithms perform if that knowledge is available. In their approach in [LHGD18], the fusion of bounding boxes is conducted via weighted averages, with the weights being determined via the distances to other boxes. Averaging was also done in [BYML16]. In the hierarchical Bayesian data fusion approach [RM17], the trackers to be fused are given weights as well based on the Mahalanobis distance to the measurement and the Euclidean distance to the trackers. The bounding boxes are then fused using Kalman filter formulas, with the weights determining the variances.

5.2 Random Ellipse Densities

In this section, we model a density for ellipses incorporating their geometric properties. While the center poses no problem, the orientation and semi-axes lengths, however, can be the source for ambiguities and unclear meanings.

We start with the semi-axes lengths. They need to be positive, so we turn the state density into a truncated Gaussian density with lower bounds of 0 for a and b

$$p_t(\mathbf{x}) = \begin{cases} 0 & \text{if } a < 0 \vee b < 0 \ , \\ c \cdot p(\mathbf{x}) & \text{otherwise} \ , \end{cases} \tag{5.6}$$

with normalizing constant c.

The orientation poses a greater challenge. Due to the 2π periodicity, every shift of the orientation by a multiple of 2π results in the same ellipse. A wrapped distribution is a common solution for this type of problem [MJ09, KGH14]. However, due to the symmetric properties of ellipses, a shift of the orientation by a multiple of π results in the same ellipse as well. Additionally, rotating the ellipse by 0.5π and swapping the semi-axes lengths also results in the same ellipse (see, e.g., Figure 5.1) and even without swapping the semi-axes lengths, it is unclear how the optimal fusion would look like (see, e.g., Figure 5.2). In other words, the semi-axes lengths need to be included in the wrapping, which in turn means the period is smaller, i.e., the wrapped distribution must restrict the orientation between 0 and 0.5π. To formulate the wrapped distribution, we define a transformation for equivalent ellipses, which can handle the swapping of semi-axes lengths

$$K_k(\mathbf{x}) = \begin{bmatrix} \mathbf{m}_{\mathbf{x}}^{\mathrm{T}} & \alpha_{\mathbf{x},k} & v_k(l_{\mathbf{x}}, w_{\mathbf{x}}) & v_{k+1}(l_{\mathbf{x}}, w_{\mathbf{x}}) \end{bmatrix}^{\mathrm{T}} \ , \tag{5.7}$$

with

$$v_k(l, w) = \begin{cases} l & \text{if } k \text{ is even} \ , \\ w & \text{if } k \text{ is odd} \ , \end{cases} \tag{5.8}$$

$$\alpha_{\mathbf{x},k} = \alpha_{\mathbf{x}} + k \cdot \frac{\pi}{2} \ , \tag{5.9}$$

and $k \in \mathbb{Z}$. The above transformation incorporates all representations of the ellipse by including all shifts by a multiple of 0.5π. The function in (5.8) ensures that for shifts by a multiple of π, nothing else changes, while for all other shifts, the semi-axes lengths are swapped as well. To incorporate the entire state density, all k from $-\infty$ to ∞ need to be considered, resulting in a density

$$\tilde{p}(\mathbf{x}) = \begin{cases} \sum_{k=-\infty}^{\infty} p_t(K_k(\mathbf{x})) & \text{if } 0 \leq \alpha_{\mathbf{x}} < \frac{\pi}{2} \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{5.10}$$

The density avoids ambiguities by incorporating the geometric properties of ellipses. We call it a Random Ellipse Density (RED).

Note that the only exception regarding the ambiguities occurs in the corner case of a circle. If both semi-axes lengths are the same, the orientation does not matter. As this is a zero-probability event mathematically speaking, we do not consider this special case.

Also, note that taking a point estimate from this density is not straightforward. In fact, the estimate minimizing the Euclidean distance as a cost function would not have a meaning for this density, as it does not consider the geometric properties of ellipses. In the next section, we provide an estimator which considers these properties and which can be used to get an intuitive point estimate from an RED.

5.3 Minimum Mean Gaussian Wasserstein Estimator

To find a meaningful point estimate given an ellipse density, we need to incorporate the geometric properties of ellipses. Therefore, we need to replace the Euclidean distance with a distance on ellipses. Extended target metrics are presented in Section 2.2.2. [YBG16] discuss metrics for elliptic extended targets including IOU [GLO11], the GW distance [GS84], Kullback-Leibler Divergence, and the Hausdorff distance, e.g., [VH00]. According to their evaluation, the GW distance provides an intuitive, scalar value that can be calculated in closed form. It is originally used as a measure of distance between Gaussian densities. However, using the center of the ellipse as mean and the shape matrix as covariance, we can use the GW distance as a distance measure between ellipses. The shape matrix of a state vector $\mathbf{x}$ can be calculated from the orientation and semi-axes lengths using

$$\mathbf{X} = \mathbf{R}_{\alpha_\mathbf{x}} \cdot \begin{bmatrix} a_\mathbf{x}^2 & 0 \\ 0 & b_\mathbf{x}^2 \end{bmatrix} \cdot \mathbf{R}_{\alpha_\mathbf{x}}^\mathrm{T} \ , \tag{5.11}$$

$$\mathbf{R}_\alpha = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \ . \tag{5.12}$$

The squared GW distance [GS84] between two ellipses (modeled as Gaussians with means $\mathbf{m_x}$ and $\mathbf{m_z}$ and covariances $\mathbf{X}$ and $\mathbf{Z}$ respectively) is then defined as

$$\mathrm{GW}(\mathbf{m_z}, \mathbf{Z}, \mathbf{m_x}, \mathbf{X})^2 = ||\mathbf{m_z} - \mathbf{m_x}||_2^2 + \mathrm{tr}(\mathbf{Z} + \mathbf{X} - 2(\mathbf{Z}^{\frac{1}{2}}\mathbf{X}\mathbf{Z}^{\frac{1}{2}})^{\frac{1}{2}}) \ , \tag{5.13}$$

with the matrix square root defined as

$$\mathbf{X} = \mathbf{X}^{\frac{1}{2}} \left(\mathbf{X}^{\frac{1}{2}}\right)^\mathrm{T} \ , \tag{5.14}$$

which in the case of positive definite matrices can be calculated from their eigenvalues and eigenvectors as

$$\mathbf{X}^{\frac{1}{2}} = \mathbf{R}_{\alpha_\mathbf{x}} \cdot \begin{bmatrix} a_\mathbf{x} & 0 \\ 0 & b_\mathbf{x} \end{bmatrix} \cdot \mathbf{R}_{\alpha_\mathbf{x}}^{\mathrm{T}} \quad . \tag{5.15}$$

Note that the squared semi-axes lengths are the eigenvalues of the shape matrix and the columns of the rotation matrix are the eigenvectors. Using the GW distance does not weigh each part of the ellipse equally, concentrating more of the weight on the center than on the edges. Still, it is a good candidate for estimation.

Using this distance measure instead of the Euclidean distance, we can get an estimate using

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\text{argmin}} \int \text{GW}(\mathbf{m}_\mathbf{z}, \mathbf{Z}, \mathbf{m}_\mathbf{x}, \mathbf{X})^2 \cdot \tilde{p}(\mathbf{x}) \mathrm{d}\mathbf{x} \quad , \tag{5.16}$$

which finds the ellipse $\hat{\mathbf{z}}$ which minimizes the GW distance to the density $\tilde{p}(\mathbf{x})$. We call this estimator a Minimum Mean Gaussian Wasserstein (MMGW) estimator. This estimator can be used to gain a meaningful estimate on the RED, but also on the original density $p(\mathbf{x})$, as it considers the geometric properties and, therefore, the ambiguities of ellipses.

However, due to the GW distance, the estimator is difficult to calculate. Even if only samples of the density were used, an iterative optimization needs to be conducted [PRV20]. As we wish to find a closed-form solution, we advocate the use of the square root distance [DKZ09]

$$\text{SR}(\mathbf{Z}, \mathbf{X}) = ||\mathbf{Z}^{\frac{1}{2}} - \mathbf{X}^{\frac{1}{2}}||_F \quad , \tag{5.17}$$

with the squared Frobenius norm of an $n \times m$ matrix $\mathbf{A}$ defined as

$$||\mathbf{A}||_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{A}_{ij}^2 \quad , \tag{5.18}$$

with $\mathbf{A}_{ij}$ referring to the element of $\mathbf{A}$ at the ith row and jth column. As the GW distance also considers the center, we need to extend the square root distance by the center as well, naming the result the Extended Square Root (ESR) distance. The squared ESR distance is defined as

$$\text{ESR}(\mathbf{m}_\mathbf{z}, \mathbf{Z}, \mathbf{m}_\mathbf{x}, \mathbf{X})^2 = ||\mathbf{m}_\mathbf{z} - \mathbf{m}_\mathbf{x}||_2^2 + \text{SR}(\mathbf{Z}, \mathbf{X})^2 \quad . \tag{5.19}$$

To explain the reason for using the ESR distance, we have to define commuting matrices. Two matrices $\mathbf{A}$ and $\mathbf{B}$ commute if $\mathbf{AB} = \mathbf{BA}$. For positive definite matrices, this happens when the axes of the corresponding ellipses align. For matrices that commute, the GW distance can be transformed into the ESR distance, i.e., the two metrics are equivalent for commuting matrices [GS84, AP18]. Therefore, we use the ESR distance as an approximation for matrices that do not commute. We will provide an evaluation of this approximation in Section 6.3. Given this approximation, we can replace the GW distance with the ESR distance

$$\begin{aligned}\mathrm{GW}(\mathbf{m}_{\mathbf{z}}, \mathbf{Z}, \mathbf{m}_{\mathbf{x}}, \mathbf{X})^2 &\approx ||\mathbf{m}_{\mathbf{z}} - \mathbf{m}_{\mathbf{x}}||_2^2 + \mathrm{tr}((\mathbf{Z}^{\frac{1}{2}} - \mathbf{X}^{\frac{1}{2}})(\mathbf{Z}^{\frac{1}{2}} - \mathbf{X}^{\frac{1}{2}})) \\ &= ||\mathbf{m}_{\mathbf{z}} - \mathbf{m}_{\mathbf{x}}||_2^2 + ||\mathbf{Z}^{\frac{1}{2}} - \mathbf{X}^{\frac{1}{2}}||_F^2 \\ &= \mathrm{ESR}(\mathbf{m}_{\mathbf{z}}, \mathbf{Z}, \mathbf{m}_{\mathbf{x}}, \mathbf{X})^2 \ , \end{aligned} \tag{5.20}$$

creating an approximated MMGW estimator. Using this approximation, we can apply a transformation

$$T(\mathbf{x}) = \begin{bmatrix} \mathbf{m}_{\mathbf{x}}^{\mathrm{T}} & s_{\mathbf{x}}^{(11)} & s_{\mathbf{x}}^{(12)} & s_{\mathbf{x}}^{(22)} \end{bmatrix}^{\mathrm{T}} \ , \tag{5.21}$$

with $s^{(nm)}$ being cells of the symmetric square root matrix

$$\begin{bmatrix} s_{\mathbf{x}}^{(11)} & s_{\mathbf{x}}^{(12)} \\ s_{\mathbf{x}}^{(21)} & s_{\mathbf{x}}^{(22)} \end{bmatrix} = \mathbf{X}^{\frac{1}{2}} \ , \tag{5.22}$$

and $s_{\mathbf{x}}^{(12)} = s_{\mathbf{x}}^{(21)}$, to get

$$\begin{aligned}\hat{\mathbf{z}} &= \underset{\mathbf{z}}{\mathrm{argmin}} \int \mathrm{GW}(\mathbf{m}_{\mathbf{z}}, \mathbf{Z}, \mathbf{m}_{\mathbf{x}}, \mathbf{X})^2 \cdot \tilde{p}(\mathbf{x})\mathrm{d}\mathbf{x} \\ &\approx \underset{\mathbf{z}}{\mathrm{argmin}} \int ||T(\mathbf{z}) - T(\mathbf{x})||_2^2 \cdot \tilde{p}(\mathbf{x})\mathrm{d}\mathbf{x} \\ &= T^{-1}(\mathrm{E}[T(\mathbf{x})]) \ . \end{aligned} \tag{5.23}$$

In other words, if we sample enough particles from the density, transform, and average them, we can approximate the MMGW estimate. In [AP18], it was also argued in the context of GM reduction, that a single Gaussian minimizing the GW distance to a GM can be approximated by averaging the square roots of the GM's covariances. In the following, we denote the space consisting of the center and elements of the square root matrix as “square root space" while we name the original state space the “ellipse parameter space". To include kinematic parts, the ESR distance in (5.19) and by that the transformation in (5.21) need to be extended by the kinematic vector analogous to $\mathbf{m}$.

Note here that we use the matrix square root as defined for positive definite matrices in (5.15). There are other ways to get a square root, e.g., the Cholesky decomposition. The elements of the Cholesky decomposition have been used to represent the state vector of an ellipse in other works as well [BH14, ZXWA13]. We prefer the described method, however, since calculating the square root matrix based on the orientation and semi-axes lengths of the ellipse is straightforward.

Figure 5.3 provides an overview of how the methods are related. The REDs from the previous section are defined in the original ellipse parameter space and used for fusion. To approximate

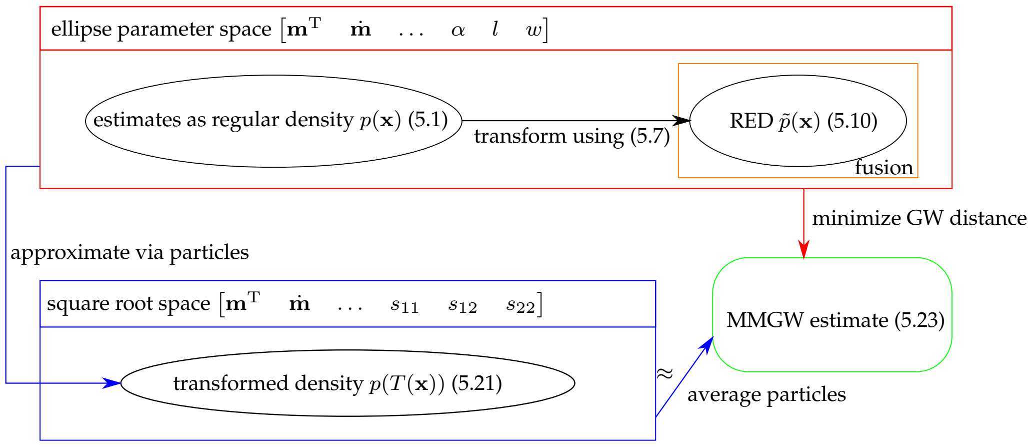

ellipse parameter space $[\mathbf{m}^{\mathrm{T}} \quad \dot{\mathbf{m}} \quad \dots \quad \alpha \quad l \quad w]$
estimates as regular density $p(\mathbf{x})$ (5.1)
transform using (5.7)
RED $\tilde{p}(\mathbf{x})$ (5.10)
fusion
minimize GW distance
approximate via particles
square root space $[\mathbf{m}^{\mathrm{T}} \quad \dot{\mathbf{m}} \quad \dots \quad s_{11} \quad s_{12} \quad s_{22}]$
MMGW estimate (5.23)
transformed density $p(T(\mathbf{x}))$ (5.21)
$\approx$
average particles

Figure 5.3: A summary of the two spaces and the densities we consider and how the MMGW estimate can be gained from them. [3]©2021IEEE

the MMGW estimate (red arrow), we approximate the density transformed into square root space via particles (first blue arrow). This can also be done with an ordinary ellipse density to find the MMGW estimate. The estimate itself can then be approximated by calculating the mean in square root space, i.e., averaging the particles (second blue arrow).

5.4 Implementation

In this section, we present our implementations of the algorithms proposed in this chapter. Section 5.4.1 shows how an algorithm using an RED for fusion in ellipse parameter space can be implemented. As the square root space does not have the issues of ambiguity present in ellipse parameter space, an alternative fusion approach is to transform the density before the fusion and then fuse in square root space. A practical implementation of this approach using Monte Carlo (MC) simulation is presented in Section 5.4.2.

5.4.1 Ellipse Parameter Space

Revisiting the example from Figure 5.2, which shows the issue of which fusion is better, the RED offers a solution by being a multi-modal Gaussian density, considering all combinations of ellipse representations. To fuse two REDs using the Kalman filter formulas, each component of the prior needs to be combined with each component of the likelihood, i.e.,

$$
\begin{aligned}
\tilde{p}(\mathbf{x}|\hat{\mathbf{x}}_s) &\sim \tilde{p}(\hat{\mathbf{x}}_s|\mathbf{x}) \cdot \tilde{p}(\mathbf{x}) \\
&= \sum_{k=-\infty}^{\infty} p_t(K_k(\hat{\mathbf{x}}_s)|\mathbf{x}) \cdot \sum_{j=-\infty}^{\infty} p_t(K_j(\mathbf{x})) \\
&= \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} p_t(K_k(\hat{\mathbf{x}}_s)|\mathbf{x}) \cdot p_t(K_j(\mathbf{x})) \ , \qquad (5.24)
\end{aligned}
$$

with the orientations $\alpha_{\mathbf{x}}$ and $\alpha_{\hat{\mathbf{x}}_s}$ restricted as in (5.10). This ensures that all possible combinations are considered.

As the number of components is infinite, we need to reduce (5.24). To do so, we consider that the orientation is still per definition 2π periodic, so as an approximation, we can apply the trick from wrapped distributions to restrict the orientation innovation between $-\pi$ and π. That means the update of a component $K_j(\mathbf{x})$ with another component $K_k(\hat{\mathbf{x}}_s)$ would be the same as the update with component $K_{k+4}(\hat{\mathbf{x}}_s)$. We can then reduce the updates to only 4 consecutive components of the likelihood for each prior component. This also works the other way around. Updating $K_j(\mathbf{x})$ with $K_k(\hat{\mathbf{x}}_s)$ would provide the same result as updating $K_{j+4}(\mathbf{x})$ with $K_k(\hat{\mathbf{x}}_s)$. Therefore, we can restrict both prior and likelihood RED to 4 consecutive components and apply the orientation innovation restriction between $-\pi$ and π for each update. Applying these two approximations gives us the density

$$
\tilde{p}(\mathbf{x}|\hat{\mathbf{x}}_s) \approx c_1 \cdot \sum_{j=n}^{n+4} \sum_{k=m}^{m+4} p_t(K_k(\hat{\mathbf{x}}_s)|\mathbf{x}) p_t(K_j(\mathbf{x})) \ , \qquad (5.25)
$$

with normalizing constant c_1 and $n, m \in \mathbb{Z}$. This way, all four orientations are used. The innovation in α is restricted between $-\pi$ and π in the Kalman filter update as suggested in, e.g., [MĆP16]. To use the Kalman filter formulas properly, we assume most of the probability mass for the semi-axes lengths is on the positive axis, so that we can approximate the truncated densities as regular Gaussians

$$
\begin{aligned}
p_t(K_k(\hat{\mathbf{x}}_s)|\mathbf{x}) &\approx p(K_k(\hat{\mathbf{x}}_s)|\mathbf{x}) \ , \qquad (5.26) \\
p_t(K_j(\mathbf{x})) &\approx p(K_j(\mathbf{x})) \ . \qquad (5.27)
\end{aligned}
$$

Each component consists of a mean $\hat{\mathbf{x}}^{(i)}$, a covariance $\mathbf{C}^{\mathbf{x}(i)}$, and a weight w_i. The initial RED components have equal weights, but after each fusion, the weight of a new component is calculated as the product of the old weight and the predicted likelihood $l_{j,k}$ of the fusion of component j from the global prior at the central fusion unit and component k from the local estimate, i.e.,

$$l_{j,k} = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{C}^l_{j,k})}} e^{-0.5(K_k(\hat{\mathbf{x}}_s)-K_j(\hat{\mathbf{x}}))^{\mathrm{T}}(\mathbf{C}^l_{j,k})^{-1}(K_k(\hat{\mathbf{x}}_s)-K_j(\hat{\mathbf{x}}))} \quad , \tag{5.28}$$

$$\mathbf{C}^l_{j,k} = K_k(\mathbf{C}^{\mathbf{x}_s}) + K_j(\mathbf{C}^{\mathbf{x}}) \quad , \tag{5.29}$$

with d being the dimension of the state, i.e., $\mathbf{x} \in \mathbb{R}^d$ and $K_j(\mathbf{C}^{\mathbf{x}})$ and $K_k(\mathbf{C}^{\mathbf{x}_s})$ producing the covariances with the rows and columns corresponding to a and b shifted accordingly if j or k is odd respectively.

After the fusion, the 4 components of the prior grow to 16. To counter this exponential increase in component numbers, we apply a mixture reduction. A brief discussion on GM reduction techniques can be found in Section 2.3.1. In the original article, we used a more heuristic mixture reduction. However, to improve the results, we now apply the clustering algorithm from [Sal09], as it provides an algorithm with shorter runtime at the cost of possibly merging more components as is necessary compared to other pair-wise merging approaches. As certain combinations of ellipse representations do not reflect a likely semi-axes association, we start with a pruning step before using the merging algorithm. The algorithm itself assumes components with higher weights have higher importance, so they start by selecting the highest weighted component, denoted with index c, then calculate the distance to all other components which are not in a cluster yet, denoted with index i. The distance measure uses the weights to make low weight components more likely to be merged and considers the central component's covariance. It is calculated as

$$d_{c,i} = \frac{w_i w_c}{w_i + w_c}(\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(c)})^{\mathrm{T}}\big(\mathbf{C}^{\mathbf{x}(c)}\big)^{-1}(\hat{\mathbf{x}}^{(i)} - \hat{\mathbf{x}}^{(c)}) \quad . \tag{5.30}$$

All components with $d_{c,i}$ below a threshold are combined with the central component c into a cluster. The process is then repeated for the remaining components until all belong to a cluster. Note that a cluster can also consist of only a single component. The components in each cluster are then merged by minimizing the Kullback-Leibler divergence, using (2.31), (2.32), and (2.33). If the reduced mixture still has a number of components above a threshold, the distance threshold for the clustering is increased and the process repeated. A more thorough study of mixture reduction methods for REDs and how they compare in terms of accuracy versus runtime needs to be studied in future work.

To incorporate correlation between the sensor estimates as described in Section 2.1.5, stemming from, e.g., a common prior or process model, several aspects need to be considered, as we are not fusing Gaussian densities, but GM densities. Covariance intersection is a straightforward solution as it can be considered for the fusion of each individual mixture component. Another idea could be to use information form. Approaches to track the correlation between sensor estimates can also be considered, given a common process model. The exact realization depends on the type of network

```
input  : multi-modal global RED with means $\hat{\mathbf{x}}$, covariances $\mathbf{C}$, and weights $\mathbf{w}$, local sensor estimate's mean $\hat{\mathbf{x}}_s$
         and covariance $\mathbf{C}_s$
output : updated RED means $\hat{\mathbf{x}}^+$, covariances $\mathbf{C}^+$, and weights $\mathbf{w}^+$
for $k \in \{0 \dots 3\}$ do
    $\hat{\mathbf{x}}_s.k \leftarrow$ K($\hat{\mathbf{x}}_s$, $k$)
    $\mathbf{C}_s.k \leftarrow$ K($\mathbf{C}_s$, $k$)
end
for $n \in \{0 \dots$ len($\mathbf{w}$) $-1\}$ do
    for $k \in \{0 \dots 3\}$ do
        $\hat{\mathbf{x}}^+.(4n+k)$, $\mathbf{C}^+.(4n+k)$, $\mathbf{w}^+.(4n+k) \leftarrow$kalman($\hat{\mathbf{x}}.n$, $\mathbf{C}.n$, $\hat{\mathbf{x}}_s.k$, $\mathbf{C}_s.k$)
        $\mathbf{w}^+.(4n+k) \leftarrow \mathbf{w}^+.(4n+k) \cdot \mathbf{w}.n$
    end
end
$\hat{\mathbf{x}}^+$, $\mathbf{C}^+$, $\mathbf{w}^+ \leftarrow$reduce_mixture($\hat{\mathbf{x}}^+$, $\mathbf{C}^+$, $\mathbf{w}^+$)
return $\hat{\mathbf{x}}^+$, $\mathbf{C}^+$, $\mathbf{w}^+$
```

Algorithm 4: The RED fusion algorithm relies on the following functions: `K()` switches according to (5.7) with the last input as k when given state means and switches covariances accordingly as well, `len()` provides the length of the input vector, `kalman()` is the Kalman filter update from Section 2.1.2 (restricting the innovation in α between $-\pi$ and π) providing updated mean, covariance, and predicted likelihood, and `reduce_mixture()` is a mixture reduction function also providing normalized weights. $\hat{\mathbf{x}}_s.k$ refers to the kth element in the array $\hat{\mathbf{x}}_s$. Based on [3].

used as well, i.e., is the RED kept in a central unit and updated with the estimates of the sensors or does each sensor keep an RED and update with neighboring sensors in a distributed fashion? As noted in Section 5.1.2, we assume independence between the sensor estimates to focus on the geometric properties, but these challenges should be considered in future work.

To retrieve a point estimate from the posterior, we sample particles, transform them using (5.21), and average them, following the blue arrows in Figure 5.3. This method is called Random Ellipse Density Minimum Mean Gaussian Wasserstein (RED-MMGW) and pseudo-code can be found in Algorithm 4.

5.4.2 Square Root Space

The square root space possesses the quality of not having ambiguities in the ellipse representation, i.e., given the transformation (5.21), we have

$$T(K_0(\mathbf{x})) = T(K_k(\mathbf{x})) \quad \forall k \in \mathbb{Z} \ . \tag{5.31}$$

This includes the property that all representations of a circle result in the same transformed value, i.e.,

$$T(\hat{\mathbf{x}}_1) = T(\hat{\mathbf{x}}_2) \quad \text{if } a_1 = b_1 = a_2 = b_2 \land \mathbf{m}_1 = \mathbf{m}_2 \ . \tag{5.32}$$

Using this quality, we can transform the densities first and then fuse them in square root space. Due to the non-linearities of the transformation, we approximate the transformed density as a Gaussian using MC simulations. We also tried a linearization using Jacobians, but our previous investigation revealed that approach to not be promising [2].

Given an ellipse density, we draw m particles,

$$\mathbf{p}^{(j)} \sim \mathcal{N}(\hat{\mathbf{x}}, \mathbf{C}) \quad j \in \{1 \dots m\} \ , \tag{5.33}$$

apply the transformation (5.21), and approximate the density in square root space as a Gaussian with mean $\hat{\mathbf{y}}$ and covariance $\mathbf{D}$

$$\hat{\mathbf{y}} \approx \frac{1}{m} \sum_{j=1}^{m} T(\mathbf{p}^{(j)}) \ , \tag{5.34}$$

$$\mathbf{D} \approx \frac{1}{m} \sum_{j=1}^{m} (T(\mathbf{p}^{(j)}) - \hat{\mathbf{y}})(T(\mathbf{p}^{(j)}) - \hat{\mathbf{y}})^{\mathrm{T}} \ . \tag{5.35}$$

Transforming prior and local estimate in this way, we conduct the fusion using Kalman filter formulas. The mean of the resulting density can be used for the MMGW estimate as discussed previously. The posterior is kept as a transformed density and each incoming local estimate is transformed as described above before the fusion to iterate over multiple time steps.

While this method makes some approximations, it still preserves some information on the uncertainties of the different parameters, which generally provides better results than simply averaging the shape matrices. Additionally, it uses the square roots of the shape matrices, which provides additional advantages regarding the GW distance as will be demonstrated in Section 6.4.1 and discussed in Section 7.3.2.

This method is called Monte Carlo Minimum Mean Gaussian Wasserstein (MC-MMGW) with pseudo-code provided in Algorithm 5.

5.5 Applicability to Rectangles

Due to rectangles being parameterized with the same parameters as ellipses, one can argue to use the MMGW estimator and RED concepts for rectangular shapes as well. We can create a covariance matrix corresponding to the orientation and shape of a rectangle. However, a Gaussian concentrates more probability mass in the center than on the contour. For ellipses, we could argue that the shape of the Gaussian was still elliptical, but for a rectangle, the corners would get much less probability weight than the sides. Therefore, a different metric can be more applicable. In this section, we argue for an appropriate metric for rectangles, which we then use to define an estimator for rectangle densities. We will then later compare it to the MMGW estimate for rectangles.

```
input  : Gaussian approximated global mean $\hat{\mathbf{y}}$ and covariance $\mathbf{D}$ in square root space, local sensor estimate's
         mean $\hat{\mathbf{x}}_s$ and covariance $\mathbf{C}_s$ in ellipse parameter space, and the number of particles $m$
output : updated mean $\hat{\mathbf{y}}^+$ and covariance $\mathbf{D}^+$
for $j \in \{1 \dots m\}$ do
    $p.j \leftarrow$ mvn($\hat{\mathbf{x}}_s$, $\mathbf{C}_s$)
end
$\hat{\mathbf{y}}_s \leftarrow$mean(T($p$))
$\mathbf{D}_s \leftarrow$mean(outer(T($p$)$-\hat{\mathbf{y}}_s$, T($p$)$-\hat{\mathbf{y}}_s$))
$\hat{\mathbf{y}}^+, \mathbf{D}^+ \leftarrow$kalman($\hat{\mathbf{y}}, \mathbf{D}, \hat{\mathbf{y}}_s, \mathbf{D}_s$)
return $\hat{\mathbf{y}}^+, \mathbf{D}^+$
```

Algorithm 5: The MC-MMGW algorithm relies on the following functions: `mvn()` provides a sample from a multivariate normal distribution, `T()` describes the transformation from (5.21), `mean()` provides a mean, `outer()` describes the outer product, and `kalman()` is a regular Kalman update. Based on [3].

We presented extended target metrics in Section 2.2.2. There, we argued that an appropriate metric can be derived by modeling the shapes as densities and then calculating a distance on densities between them. A general approach would be to model the shape as a uniform distribution and apply the Wasserstein distance. To make the calculations more feasible, we use discrete densities and apply the corresponding formulas [HM02]. The cost function can then be replaced by the Wasserstein distance between the discrete densities to create a Minimum Mean Discretized Wasserstein (MMDW) estimator. The question is then how to discretize, e.g., the entire surface or only the contour. Examples for rectangles modeled by a Gaussian or discretized uniform distribution are presented in Figure 5.4.

The first option we consider is to discretize only the contour. For rectangles, this makes it easier to calculate a fixed number of equally spaced discretization points across the contour so that each rectangle is discretized by the same number of points. Using a discrete density with the same number of points, we can calculate the Wasserstein distance as the OSPA distance [SVV08]. We define a function $f_n(\mathbf{x})$, which takes a rectangle and transforms it into a set of n points distributed equally across its contour. In this notation, $f_n(\mathbf{x})_i$ refers to a specific point $i \in \{1 \dots n\}$. Given two rectangles $\mathbf{x}$ and $\mathbf{z}$, we can calculate the 2-Wasserstein distance between them using

$$W_2^{(n)}(\mathbf{x}, \mathbf{z}) = \min_{\pi \in \Pi_n} \left(\frac{1}{n} \sum_{i=1}^{n} ||f_n(\mathbf{x})_i - f_n(\mathbf{z})_{\pi(i)}||_2^2 \right)^{\frac{1}{2}} , \tag{5.36}$$

with Π_n being the set of all permutations of the numbers 1 to n and for a specific permutation $\pi \in \Pi_n$, $\pi(i)$ refers to the index of the element assigned to the element with index i. Similar to the MMGW estimator, we now replace the cost function in the classic estimator with the squared $W_2^{(n)}$ distance calculated using Equation (5.36), which we call Discretized Wasserstein - Contour (DW-C), to get a Minimum Mean Discretized Wasserstein - Contour (MMDW-C) estimator. Given a state density $p(\mathbf{x})$, we get an estimate using

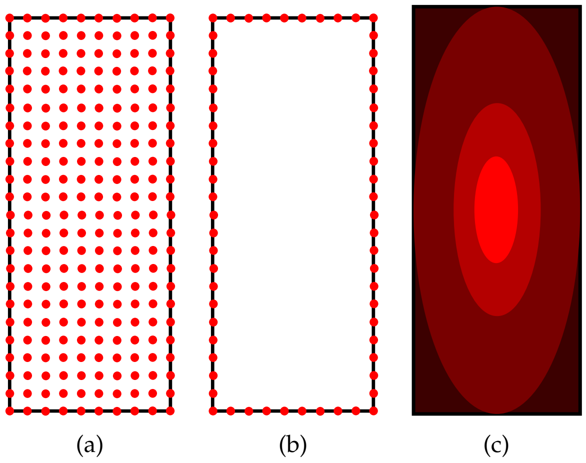


Figure 5.4: A uniform distribution is used to model a rectangle and approximated via a discrete density visualized by red points discretizing the entire surface (a) or only the contour (b). Alternatively, the uniform distribution is approximated by a Gaussian with brighter color representing higher probability (c).

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmin}} \int W_2^{(n)}(\mathbf{x}, \mathbf{z})^2 \cdot p(\mathbf{x}) \mathrm{d}\mathbf{x} \ . \tag{5.37}$$

Unlike with the estimation using the GW distance, the above estimate can not be calculated by simply finding the mean of the density in a transformed space. However, as stated above, minimizing the Wasserstein distance in this setting means minimizing the OSPA distance, which can be handled using the MMOSPA estimator as suggested in [GSSW10]. To calculate the integral, we approximate the density using m particles, i.e.,

$$\hat{\mathbf{z}} \approx \underset{\mathbf{z}}{\operatorname{argmin}} \sum_{j=1}^{m} \frac{1}{m} W_2^{(n)}(\mathbf{p}^{(j)}, \mathbf{z})^2 \ , \tag{5.38}$$

with each $\mathbf{p}^{(j)}$ drawn from $p(\mathbf{x})$ as in (5.33). Then, we iteratively find the estimate which minimizes the average OSPA distance to each particle (remember that each particle and the estimate are rectangles, which are represented by a discrete set of points along their contour, e.g., Figure 5.4b, to model them as discrete densities, meaning the estimate we find can be considered as the Wasserstein barycenter to the particles [AC11]). This process is illustrated in Algorithm 6. Note that using the MMOSPA estimator in this way provides an estimate in the shape of a point cloud. To actually get an estimate parameterized by center, orientation, and semi-axes lengths, a rectangle has to be fit onto the points or additional restrictions need to be incorporated into the process. We further elaborate on this later in this section.

```
input  : a rectangle state density with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{C}^{\mathbf{x}}$, a number of discretization points $n$, and a
         number of particles to be drawn $m$
output : point cloud $\mathbf{z}$ of $n$ points representing the MMDW-C estimate
$\mathbf{z} \leftarrow$ get_contour ($\hat{\mathbf{x}}$, $n$)
$\mathbf{p} \leftarrow$ draw_particles ($\hat{\mathbf{x}}$, $\mathbf{C}^{\mathbf{x}}$, $m$)
for $j \in \{1 \dots m\}$ do
    $\mathbf{q}.j \leftarrow$ get_contour ($\mathbf{p}.j$, $n$)
end
while not converged do
    for $j \in \{1 \dots m\}$ do
        $\pi.j \leftarrow \underset{\pi \in \Pi_n}{argmin} \sum_{i=1}^{n} ||\mathbf{q}.j(\pi(i)) - \mathbf{z}(i)||_2^2$
    end
    for $i \in \{1 \dots n\}$ do
        $\mathbf{z}(i) \leftarrow \sum_{j=1}^{m} \frac{1}{m} \mathbf{q}.j(\pi.j(i))$
    end
end
return $\mathbf{z}$
```

Algorithm 6: The MMDW-C algorithm uses the following functions. `get_contour()` provides an array of points on the contour of a rectangle, with the first argument describing the shape and the second being the number of discretization points. `draw_particles()` draws particles from the provided Gaussian density with the mean and covariance described by the first two elements and the third element being the number of particles. $\mathbf{p}.j$ refers to the jth particle in $\mathbf{p}$ and $\mathbf{z}(i)$ refers to the ith point in the array $\mathbf{z}$. Π_n refers to the set of permutations over n. Possible convergence criteria are, e.g., a maximum number of iterations and a threshold for the difference between two consecutive estimates.

Another option is to discretize the entire surface, but it is more difficult to find an equally spaced set of points for that setting, especially for a given number of points. Therefore, we suggest to discretize it as best as possible, risking an unequal number of points, and then calculating the 2-Wasserstein distance between them. Using surface discretization, we call it the Discretized Wasserstein - Surface (DW-S). For the estimator, we would once again discretize the original density $p(\mathbf{x})$. Each particle represents a rectangle, which we model as a uniform distribution. The best estimate with respect to the 2-Wasserstein distance would then be the Wasserstein barycenter of the set of uniform distributions, which can be calculated using [HM02]. We applied the function *free_support_barycenter()* of the lp module of the python optimal transport library [FCG+21]. The resulting estimator is called the Minimum Mean Discretized Wasserstein - Surface (MMDW-S) estimator.

For both MMDW estimators, it must be noted that the estimate is not a rectangle but a discrete density that does not necessarily have a rectangular shape. Therefore, as a second approximation step, a rectangle needs to be fitted onto the estimated density. This can be done by finding the bounding box that minimizes the squared distances from the points to the box via an optimization algorithm. Finding the minimum enclosing bounding box [FS75] can be an initial guess. If the entire surface is used, minimizing the distance between the bounding box and all points might return counter-intuitive results. Instead, the convex hull can be used to find the points which form the contour.

A third option is to use a Gaussian to approximate the rectangles, as the parameters are the same. While the distribution is not uniform, the advantage is that the MMGW estimator could be used here. It provides an estimate consisting of rectangle parameters and not of a point cloud onto which a rectangle needs to be fit as with the other two approaches. We will also demonstrate in Section 6.4.2 that the MMGW estimator is much faster.

We will analyze the behavior of the three approaches, the two metrics for rectangles and their relation to the GW distance for rectangles, in Section 6.3 and compare the estimates of the different estimators under varying noise as well in Section 6.4.2. Theoretical differences are the concentration of the density mass, either in the center for GW distance or on the contour or the entire surface, depending on the discretization. That means the main difference occurs in the corner case of equal semi-axes lengths.

Regarding the REDs, ambiguities can occur in the same way in rectangles as they do in ellipses, making it applicable to rectangles. The main difference between rectangles and ellipses regarding ambiguities lies in the corner case of equal semi-axes lengths as well. For ellipses, this represents a circle, for which the orientation does not matter anymore. This was one of the limitations of the REDs. For squares, however, the orientation still matters, so one can argue that the REDs fit rectangles better than ellipses. Regarding the need for an appropriate estimator, we will show in the evaluation that the MMGW estimator can be sufficient for rectangles.

6 Evaluation and Analysis

Contents

We provide a comparison of the different trackers from Chapter 3 as well as an evaluation of the MEM-EKF*-D in Section 6.1. The proposed approach for non-uniform distributed noise, the GM-MEM-EKF* from Chapter 4, is then evaluated in Section 6.2. To support our findings regarding estimation and fusion from Chapter 5, we provide an analysis of possible metrics to justify the use of the ESR distance in Section 6.3 along with experiments for a comparison of the MMGW estimator with state of the art methods in Section 6.4. We also examine the GW distances and the estimator's applicability on rectangle densities in these sections. To evaluate the RED-MMGW and the MC-MMGW we conduct a simulation with a moving ellipse in Section 6.5 and compare them to a fusion approach for RMs.

6.1 Tracker Comparison in Simulations

The simulations of this section are based on [1] and [5]. The goal is to get an initial comparison of existing elliptic target trackers based on an extent modeled by center, orientation, and semi-axes lengths and on the common spatial distribution model. We consider the trackers described in Chapter 3, i.e., the MEM-EKF* [YB19], the Ellipse-RHM-EKF and the Ellipse-RHM-EKF-imp. As it uses the same parameterization and the same assumptions regarding the measurement source distribution, we add the IAE [Gov19] to the comparison. For the MEM-EKF*, we have the version

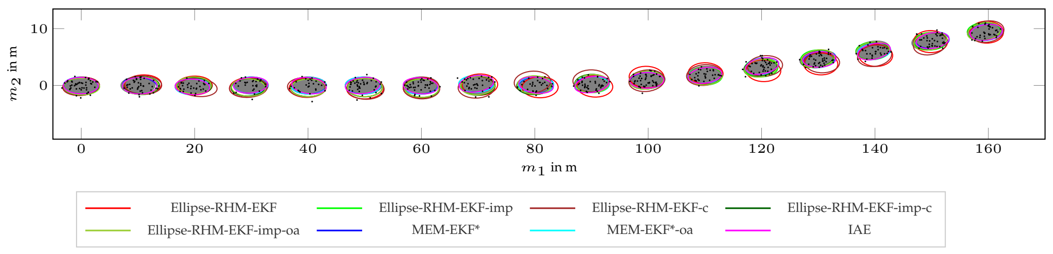


Figure 6.1: Example run of trajectory A for *low* noise with gray ground truth and measurements as black dots. [1]©2020IEEE

as described in Section 3.3 and one with the orientation approximated by the velocity vector, marked as -oa (orientation approximation), inspired by [BFH12, Gov19]. For the RHMs, we have the versions from Section 3.2 and for both a version using a single state variable for the shape and velocity orientation, denoted -c (combined), and in addition, a variant of the implicit version using an approximation of the orientation by the velocity vector, denoted -oa. All methods use an NCV model for the prediction, along with a Cartesian velocity, except for the trackers with combined orientation for velocity and shape (denoted with -c) which use a polar velocity.

We evaluate the trackers on a simple trajectory (trajectory A) in Section 6.1.1 and on a more difficult one with stronger turns (trajectory B) in Section 6.1.2. The prior is set as

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} 0 & 0 & 10 & 0 & 0 & 3 & 1.5 \end{bmatrix}^{\mathrm{T}} , \tag{6.1}$$

$$\mathbf{C}_0^{\mathbf{x}} = \mathrm{diag}\Big(\begin{bmatrix} 0.5 & 0.5 & 0.1 & 0.1 & 0.05\pi & 0.5 & 0.2 \end{bmatrix} \Big)^2 . \tag{6.2}$$

In each time step, the number of measurements is drawn from a Poisson distribution with rate 40, the measurement sources are sampled uniformly across the ground truth's surface area and corrupted by zero mean Gaussian noise. There are two settings with low and high noise respectively

$$\mathbf{C}_{low}^{\mathbf{v}} = \mathrm{diag}\Big(\begin{bmatrix} 0.5^2 & 0.5^2 \end{bmatrix} \Big) , \tag{6.3}$$

$$\mathbf{C}_{high}^{\mathbf{v}} = \mathrm{diag}\Big(\begin{bmatrix} 2.0^2 & 2.0^2 \end{bmatrix} \Big) . \tag{6.4}$$

The squared GW distance is used for the error calculation and its mean is determined over 1000 MC runs.

Finally, to evaluate the benefits of using the range rate measurement, we evaluate the range rate extension of the MEM-EKF* in Section 6.1.3.

6.1.1 Trajectory A

The target moves straight, then does a slight turn from time steps 9 to 13 with yaw rate 0.01π (see Figure 6.1). The results are presented in Figure 6.2. Due to the bias resulting from the

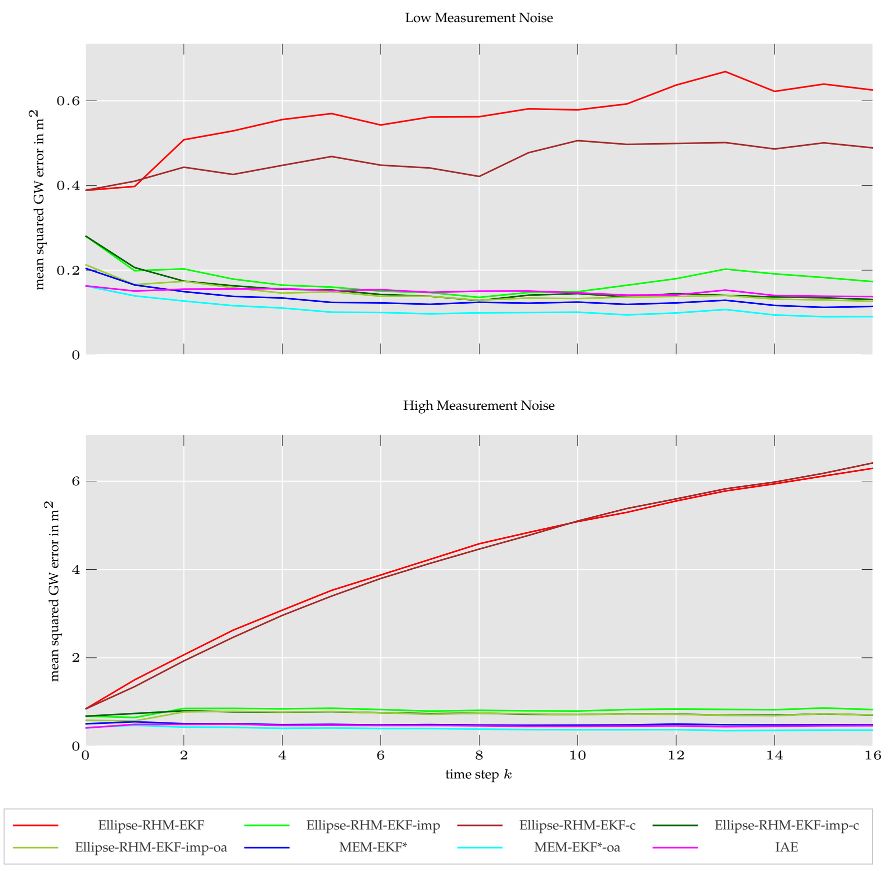


Figure 6.2: Mean squared GW error in trajectory A. [1]©2020IEEE

greedy association, the explicit RHM versions do not perform well, especially in high noise. The implicit equation improves the quality, but a bias is still visible when compared to the MEM-EKF* and IAE. The IAE is better than the MEM-EKF* in high noise, but it must be noted that its approximation of the shape orientation via the velocity vector is true in the simulation. Using the same approximation, the MEM-EKF*-oa performs best in our scenarios.

6.1.2 Trajectory B

The target moves straight, then turns from steps 6 to 9 with yaw rate 0.05π, slows down later from time steps 13 to 15 from 10 to 7 meters per second, and then turns more strongly from time steps 19 to 21 with yaw rate 0.1π (see Figure 6.3). The results are presented in Figure 6.4. Again, the explicit versions do not work well. The combination of the shape and velocity orientation as well as the orientation approximation improve the Ellipse-RHM-EKF-imp to be better than the MEM-EKF* in turns, with the orientation approximation working better in low noise and the combination of the

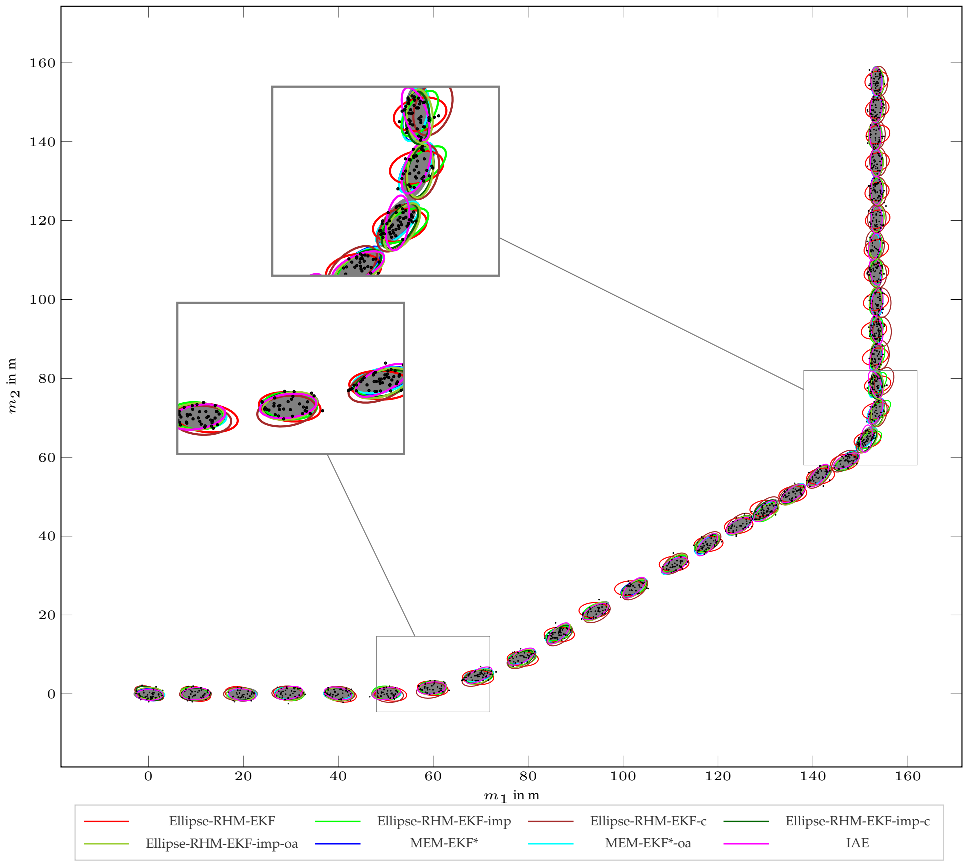


Figure 6.3: Example run of trajectory B for *low* noise with gray ground truth and measurements as black dots. [1]©2020IEEE

orientations working better in high noise. Overall, the IAE provides again good results, especially in high noise, only beaten by the MEM-EKF*-oa.

6.1.3 Usage of Range Rate

To evaluate the usage of range rate measurements, we compare the MEM-EKF*-D from Section 3.3.1 to the MEM-EKF* not using the range rate. We demonstrated in the previous section that the MEM-EKF* provides better results for elliptic targets than the RHM based trackers, so we will not consider an elliptic RHM with range rate measurements here. For the IAE, a range rate extension is not available. However, an extension of the RM approach for polar measurements and range rate is presented in [SRW15], which we name Random Matrix Doppler (RM-D) here. We add it to the comparison as well as the RM approach from [FFK11] not using range rate. All trackers use a CT model with Cartesian velocity and a yaw rate as fifth kinematic state dimension, handling the non-linearities in the time update using an UKF.

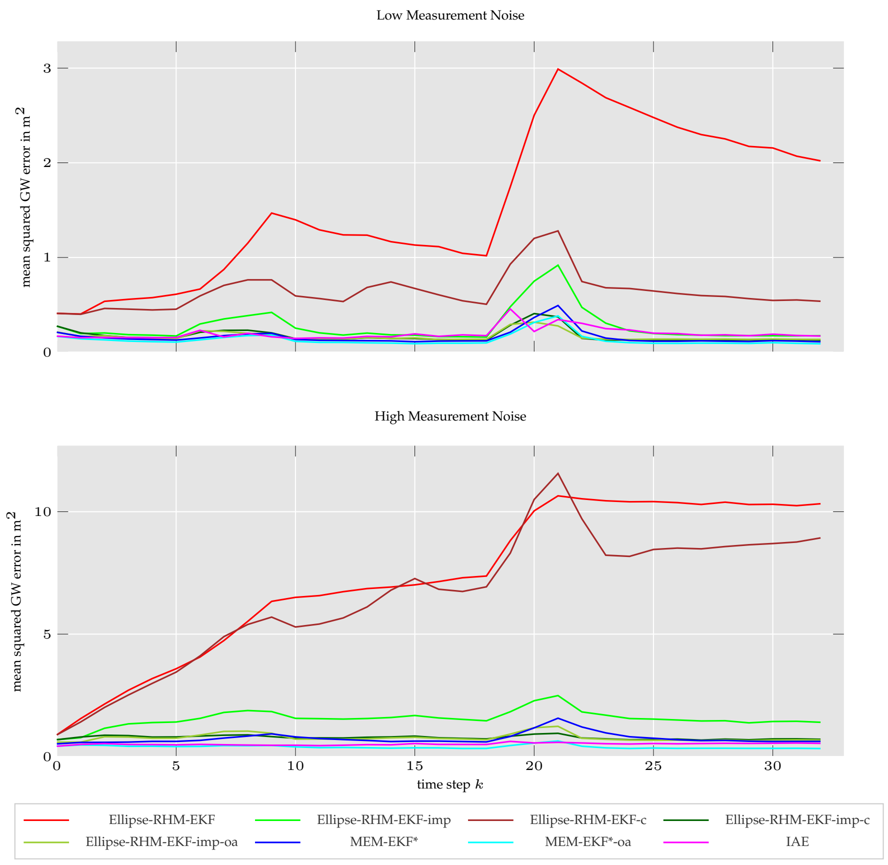


Figure 6.4: Mean squared GW error in trajectory B. [1]©2020IEEE

We simulate 2 scenarios with the aim of providing challenges regarding higher process noise. The first scenario models a turning trajectory. The process noise is $\sigma_v = 0.2$ on the velocity and $\sigma_\omega = 0.01\pi$ on the yaw rate. The target moves for 70 time steps and from time steps 20 to 30, the yaw rate is set to 0.05π, otherwise to 0 before adding noise, to ensure a turning motion in the simulation. The prior has means and covariances

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} -30.0 & -40.0 & 2.5 & 0.0 & 0.0 & 0.0\pi & 2.0 & 1.0 \end{bmatrix} , \tag{6.5}$$

$$\mathbf{C}_0^{\mathbf{x}} = \operatorname{diag}\left(\begin{bmatrix} 0.5 & 0.5 & 0.1 & 0.1 & 0.001\pi & 0.01\pi & 0.01 & 0.01 \end{bmatrix} \right)^2 . \tag{6.6}$$

For the second scenario, we simulated stronger process noise, using the trajectory from, e.g., [FFK11, YB19]. We again set the process noise on the velocity to $\sigma_v = 0.2$, but on the yaw rate to $\sigma_\omega = 0.001\pi$ this time to enforce a similar motion in each run. The trajectory is 60 time steps long and sets the yaw rate to 0.08π during time steps 14 to 17, to 0.1π during time steps 26 to 31 and 38 to 43, and to 0 otherwise. The prior means and covariances are

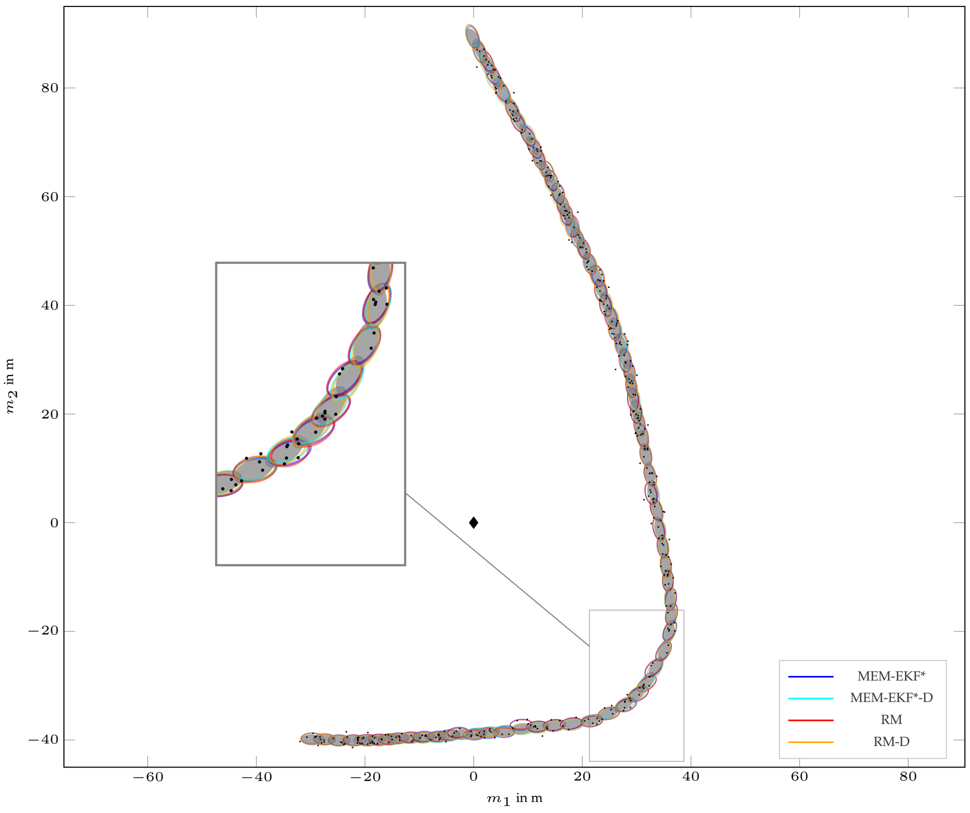


Figure 6.5: Example run of the first scenario with ground truth in gray, measurements as black dots, sensor position as a black diamond, and estimates as colored ellipses. [5]©2021IEEE

$$\mathbf{x}_0 = \begin{bmatrix} -50.0 & 10.0 & 3.64 & -3.42 & 0.0 & -0.24\pi & 2.0 & 1.0 \end{bmatrix} , \tag{6.7}$$

$$\mathbf{C}_0^{\mathbf{x}} = \operatorname{diag}\Big(\begin{bmatrix} 0.5 & 0.5 & 0.01 & 0.01 & 0.001\pi & 0.001\pi & 0.01 & 0.01 \end{bmatrix} \Big)^2 . \tag{6.8}$$

For both scenarios, the measurements are sampled uniformly across the surface of the ground truth, turned into polar measurements with the sensor position always at $\begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathrm{T}}$ and then corrupted by zero-mean Gaussian noise with standard deviations $\sigma_r = 0.5$, $\sigma_\theta = 0.002\pi$, and $\sigma_{\dot{r}} = 0.1$. The number of measurements is drawn from a Poisson distribution with rate 5 to better simulate RADAR measurements. We conducted 100 MC runs and used the GW distance as error measure.

An example of the first scenario can be found in Figure 6.5 and the results are displayed in Figure 6.6. For both MEM-EKF* and RM, the inclusion of the range rate improves the estimation, with MEM-EKF*-D outperforming RM-D. However, for all trackers, the error spikes at the beginning and end of the turning motion.

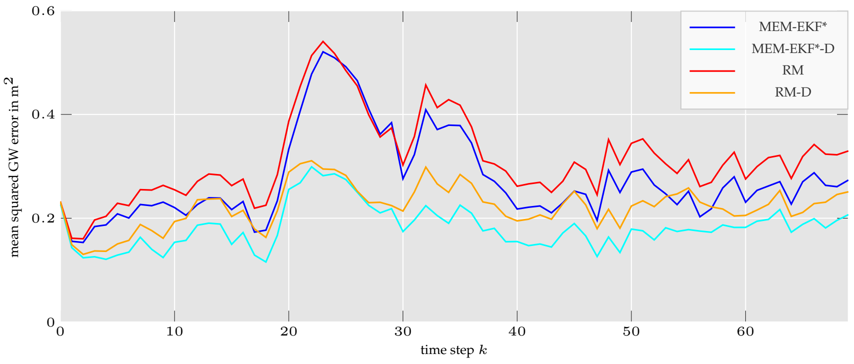


Figure 6.6: Results of the first scenario. [5]©2021IEEE

An example of the second scenario can be found in Figure 6.7 and the results are displayed in Figure 6.8. Similar to the first scenario, there are spikes during the turns, with the one at the beginning of the turning motion usually higher than the one at the end, but overall, the order of the algorithms in terms of precision is the same as in the first scenario.

6.2 Evaluation of the GM-MEM-EKF* Algorithm

This section is based on [4]. As we simply hand-tuned the GM of our GM-MEM-EKF* filter, we exclude learned models from the comparison. Furthermore, we exclude trackers which only consider contour measurements, as our approach can handle more complex distributions. The method from [CLLL21] has different assumptions, which is discussed in more detail in Section 7.2. Furthermore, we will consider the Hierarchical Truncated Gaussian Random Matrix (HTG-RM) [XWB$^+$21], which can set a variable depth for the contour measurements by truncating the RM density and provides a closed-form solution. We will use hand-tuned truncation bounds as well. In addition, the standard MEM-EKF* [YB19] and the RM approach from [FFK11] are used. As the following simulation focuses on measurements situated on the contour, we apply rectangular targets. To handle this in the RM based approach, we scale the matrix using factor $\frac{1}{3}$ instead if $\frac{1}{4}$, which improved the results. Similarly, we replaced the variance of the multiplicative noise in the MEM-EKF* in the same way.

All trackers used a CT model, handling the non-linearities in the time update via an UKF. As the parameterization of rectangles and ellipses are the same, the GW distance can be used for rectangles, e.g., [XWB$^+$21], which we also demonstrate in Section 6.3. We evaluate the proposed tracker on simulated data in Section 6.2.1. In addition, this thesis extends the original publication via an evaluation on real data in Section 6.2.2.

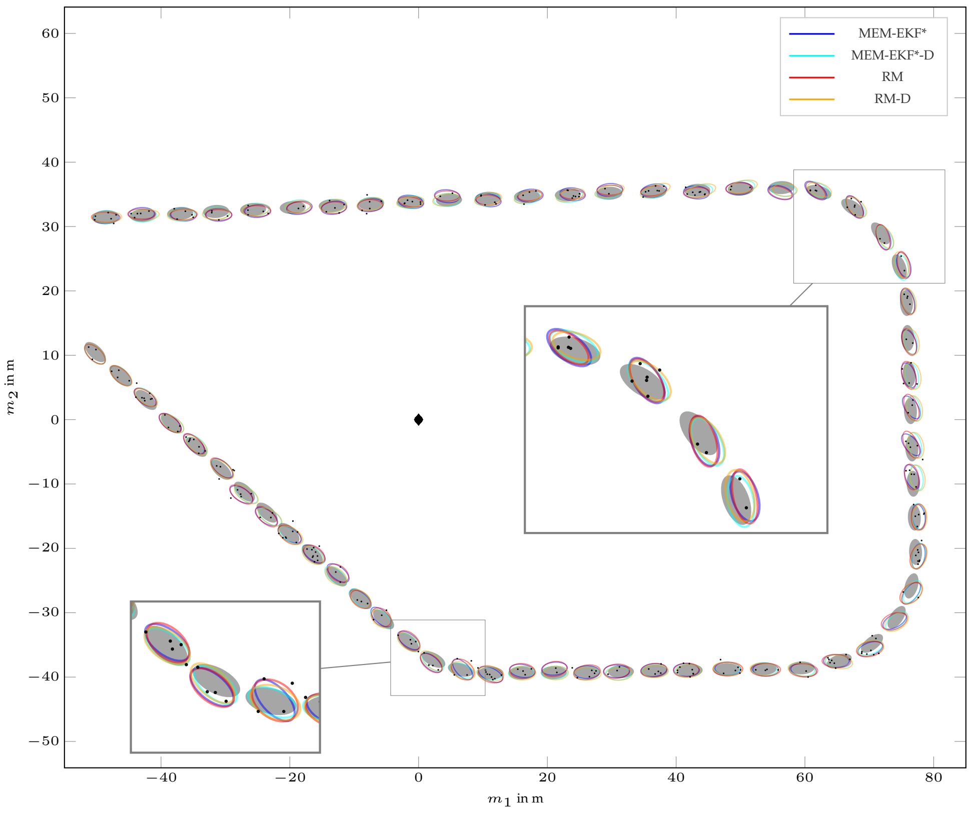


Figure 6.7: Example run of the second scenario with ground truth in gray, measurements as black dots, sensor position as a black diamond, and estimates as colored ellipses. [5]©2021IEEE

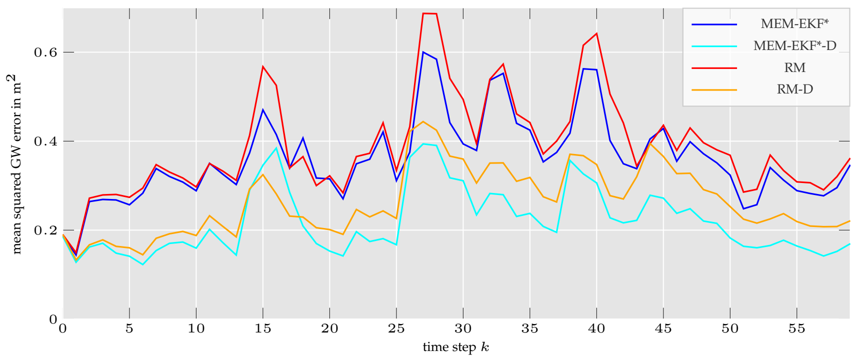


Figure 6.8: Results of the second scenario. [5]©2021IEEE

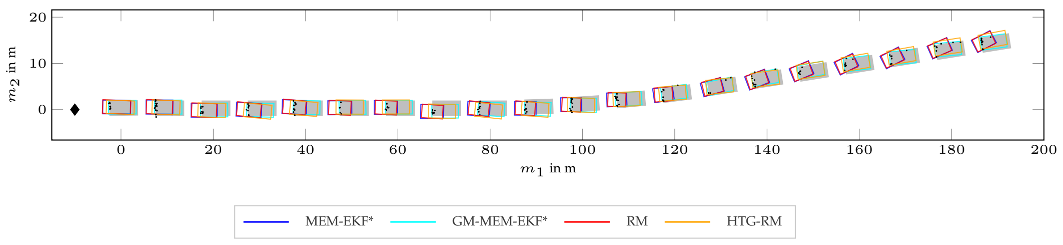


Figure 6.9: Example run of trajectory A with gray ground truth and measurements as black dots. The sensor position is marked with a black diamond. The track starts from the sensor. Based on [4].

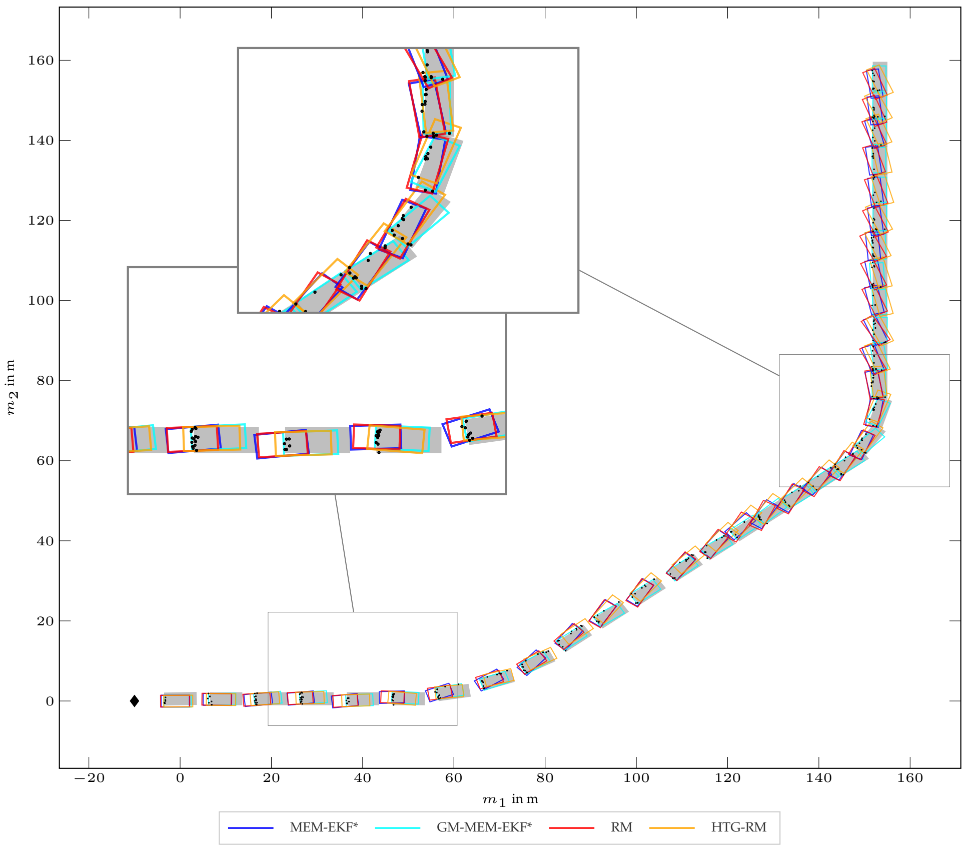


Figure 6.10: Example run of trajectory B with gray ground truth and measurements as black dots. The sensor position is marked with a black diamond. The track starts at the sensor. Based on [4].

6.2.1 Simulated data

We used the same 2 trajectories described in Section 6.1. The only difference is that to better highlight the convergence of the algorithm, we extended trajectory A to 19 time steps. Figures 6.9 and 6.10 show example runs using the proposed trackers. The prior state is

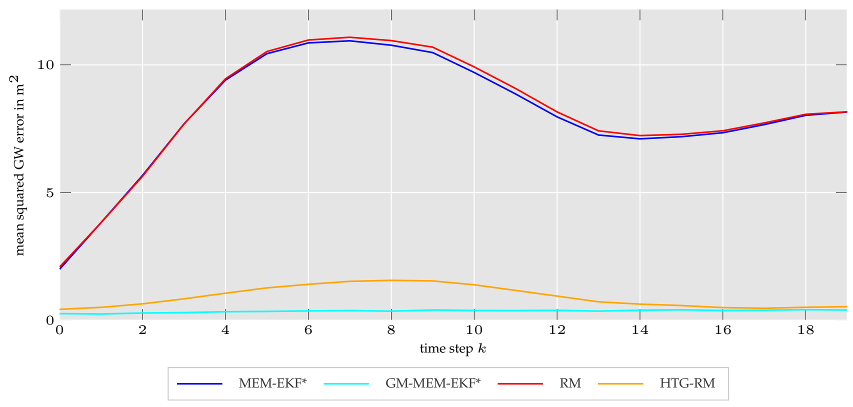


Figure 6.11: Mean squared GW error in trajectory A. Based on [4].

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} 0 & 0 & 10 & 0 & 0 & 0 & 3 & 1.5 \end{bmatrix}^{\mathrm{T}} , \tag{6.9}$$

$$\mathbf{C}_0^{\mathbf{x}} = \operatorname{diag}\left(\begin{bmatrix} 0.5 & 0.5 & 0.1 & 0.1 & 0.001\pi & 0.01\pi & 0.2 & 0.1 \end{bmatrix} \right)^2 , \tag{6.10}$$

with the dimensions corresponding to 2D center, 2D velocity, yaw rate, shape orientation, and semi-axes lengths. The RM based approaches turn the shape parameters into a shape matrix and the initial degree of the inverse Wishart density is set to 64. The process noise and other time update parameters are tuned to achieve the best performance to the best of our ability. In each time step, measurements are generated by determining the left most and right most corner of the ground truth visible from the sensor. The number of measurements is drawn from a Poisson distribution with rate 10. That number of angles between the two corners is then drawn uniformly. For each angle, the intersection with the visible contour is determined and then corrupted with zero mean Gaussian noise with covariance $\mathbf{C}^{\mathbf{v}} = \operatorname{diag}\left(\begin{bmatrix} 0.04 & 0.04 \end{bmatrix} \right)$. We calculated the mean squared error over 100 MC runs.

The results of trajectory A can be found in Figure 6.11. As expected, the algorithms that consider a uniform distribution across the entire surface are slowly diverging until they end up only tracking the visible part of the shape. The GM-MEM-EKF* and HTG-RM both keep track of the shape the entire time, keeping an overall small GW error, with GM-MEM-EKF* being better at first, but HTG-RM improving after the left side of the ground truth gets visible.

The results of scenario B can be found in Figure 6.12, leading to a similar conclusion. A notable difference is that all algorithms struggle more with the increased process noise, but GM-MEM-EKF* keeps the lowest error over most of the time steps.

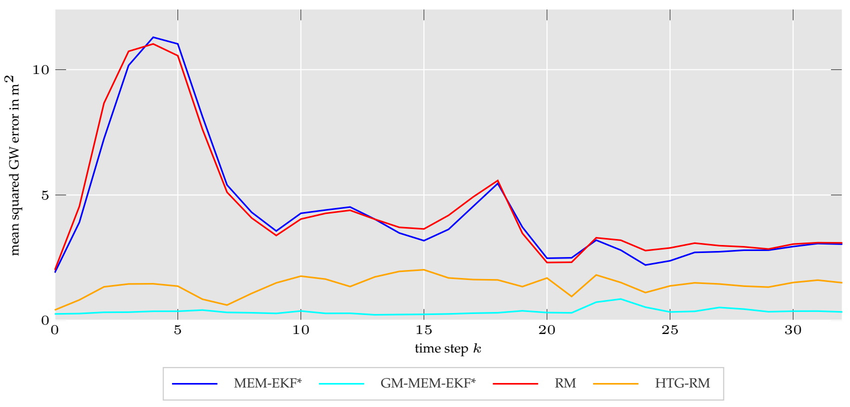


Figure 6.12: Mean squared GW error in trajectory B. Based on [4].

6.2.2 Real Data

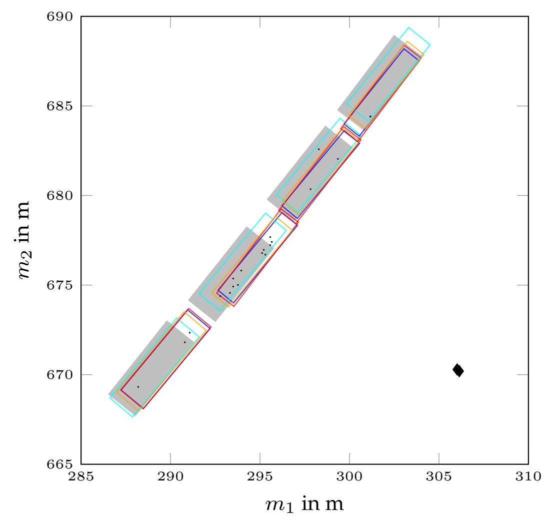
The algorithms were modified to work with the targets picked from nuScenes [CBL+20]. The goal was to find a setting which worked with all targets. See Section 4.3 for more details on the implementation of the GM-MEM-EKF* for nuScenes data. We kept the MEM-EKF* and the RM approach for the comparison. As the measurements are more distributed towards the inside of the target and not just the contour, the truncation in the HTG-RM was modified to assume detections on the entire visible half. For the evaluation, we picked four targets which provide a relatively higher number of measurements per time step. To include randomness, the tracks are initialized with the true position (except for the velocity, which was not given and was thus manually estimated) corrupted with Gaussian noise, which is used as the prior. The prior covariance is

$$\mathbf{C}_0^{\mathbf{x}} = \operatorname{diag}\left(\begin{bmatrix}1.0 & 1.0 & 0.1 & 0.1 & 0.001\pi & 0.01\pi & 0.2 & 0.1\end{bmatrix}\right)^2 . \tag{6.11}$$

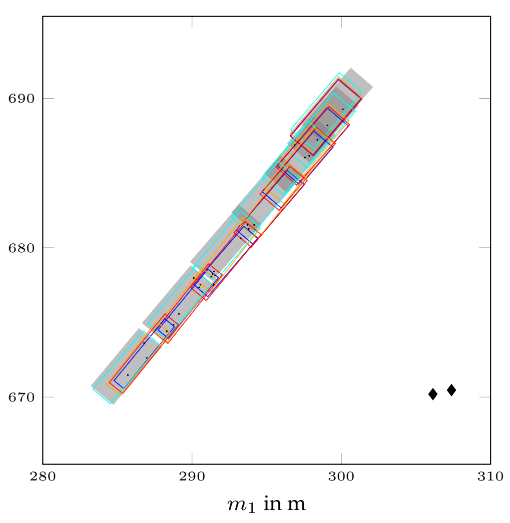
The four tracks can be seen in Figure 6.13 and the corresponding results over 100 MC runs in Figure 6.14. It can be seen that the GM-MEM-EKF* works well for all scenarios. It shows overall good performance especially when the target passes the sensor. Only at the end and sometimes the beginning of trajectories, the HTG-RM provides similar results. Please note that tuning the algorithms for even a few tracks of the nuScenes data is a difficult task and a large scale evaluation on the entire dataset would require an analysis of the entire dataset. Based on that, an evaluation against the methods using learned models on nuScenes could be conducted in future work.

6.3 Metric Analysis

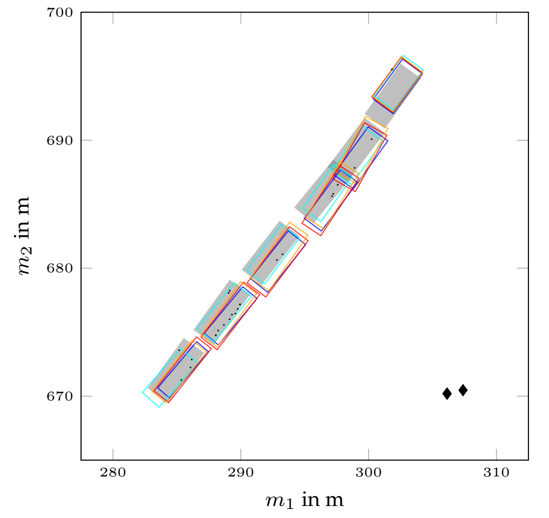
This analysis is based on publication [3]. Based on our discussion of the metric behavior, we did some modifications to the original experiments to better demonstrate the quality of approximating the GW distance using the ESR distance. We decided to compare the squared distances directly,



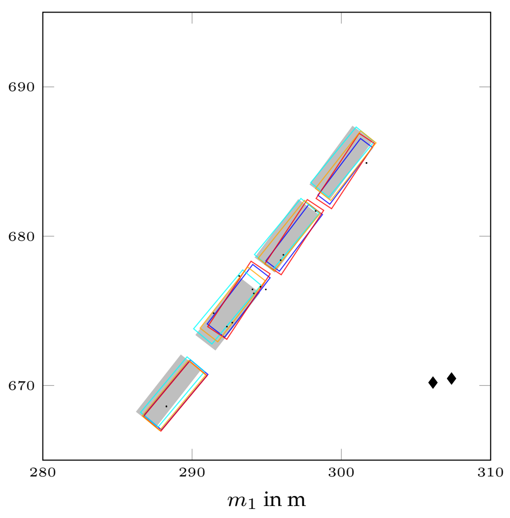
(a) Example run of first nuScenes trajectory. The track starts at the bottom left.



(b) Example run of second nuScenes trajectory. The track starts at the top right.



(c) Example run of third nuScenes trajectory. The track starts at the top right.



(d) Example run of fourth nuScenes trajectory. The track starts at the bottom left.

Figure 6.13: Experiments on real data from the nuScenes dataset [CBL+20] with gray ground truth and RADAR measurements as black dots. The positions of all involved sensors are marked with black diamonds. Only every tenth time step is plotted for visibility.

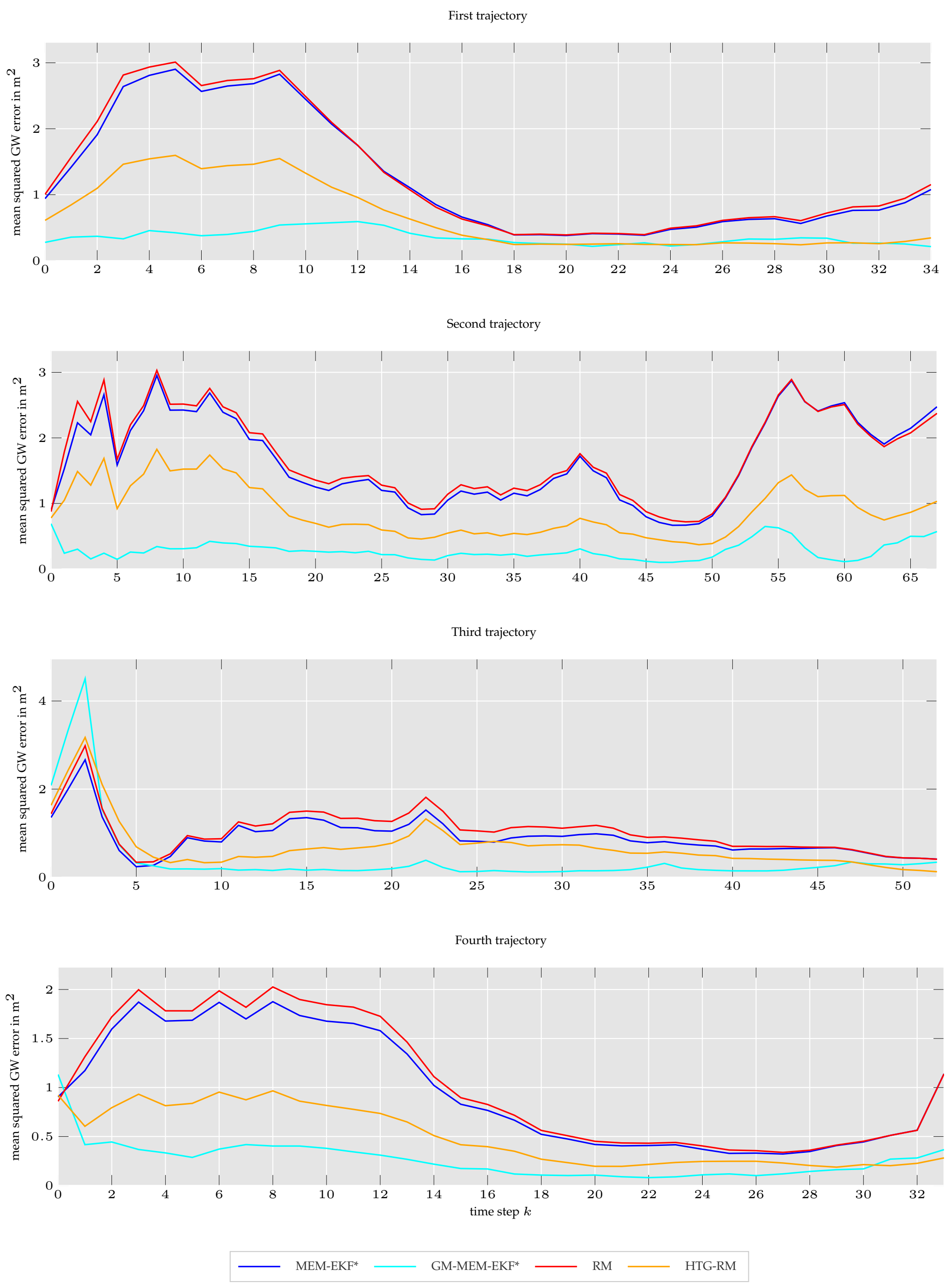


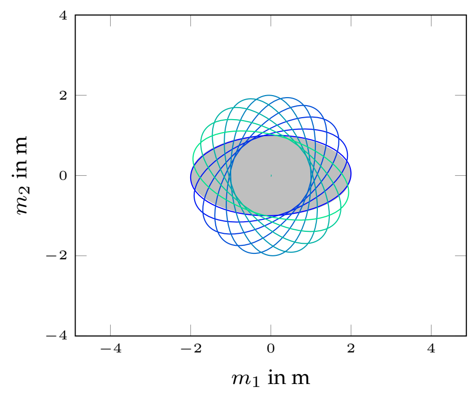
Figure 6.14: GW error in the four nuScenes trajectories.

as they are used in the estimation process. The basis of the experiments is still from [YBG16]. In our derivation of the approximation, we noted that the two metrics would be the same if the shape matrices commute (see Section 5.3 for more details). The matrices commute if the ellipses' axes align. Therefore, a first experiment compares the metric outputs for the distance between an ellipse with itself, rotating in small intervals until it rotated by π (see Figure 6.15a). It can be seen in Figure 6.15b that for a difference between the ellipses' orientations of 0.25π shifted by a multiple of 0.5π, the difference between the metrics is highest and for a difference of 0 shifted by a multiple of 0.5π, it is 0. This confirms that for commuting matrices, the metrics are the same, while for an orientation difference exactly in between two commuting states, it is highest. Still, the difference is minor if the other parameters are the same.

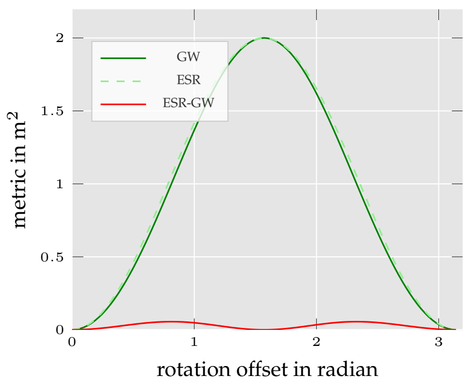
As the approximation error is highest for an orientation difference of 0.25π, we modified the remaining experiments from [YBG16] to consider this difference. The motivation here is that we want to investigate how this highest error is influenced by the other parameters. In Figure 6.15c, the distance between the ellipses' centers slowly increases. It can be seen in Figure 6.15d that the difference between the metrics is constant if only the center changes, which is to be expected as the center has the same influence for both metrics. In Figure 6.15e, one ellipse starts out smaller in length than the other and then increases. Interestingly, Figure 6.15f shows that the lowest metric difference is not at an axis length ratio of 1, but around 0.5, i.e., when the changing semi-axis of the scaling ellipse is 1. This can be explained as the scaling ellipse's fixed axis is 1 as well, meaning the ellipse is a circle at this point. For a circle, the orientation does not matter, i.e., the circle and the base ellipse commute. For an increasing axis ratio, the difference increases. The ESR distance punishes greater differences in the axis dimensions more severely than the GW distance, but otherwise, the general behavior of the metrics is the same.

These experiments demonstrate that the difference between the metrics is greatest if the two shapes are exactly in between two commuting states. Increases in positional difference between the ellipses do not increase the metric difference, while an increasingly different axis ratio does. In all cases, the difference was minor and the general behavior of the metrics was the same.

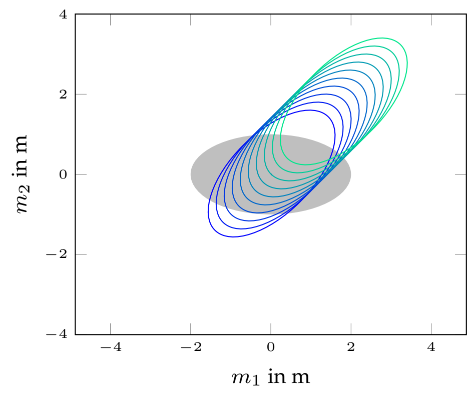
In addition, we conducted the same experiments, using the squared GW distance, the squared Euclidean distance, and the squared Wasserstein distance on rectangles modeled as uniform distributions as described in Section 5.5. The Wasserstein distance assumes a rectangular shape, sampling 100 equally spaced discretization points to discretize the contour and using optimization to find at least 400 equally spaced points to discretize the surface. The results can be found in Figure 6.16. As expected, the Euclidean distance works differently than the other metrics, increasing with an increasing angle offset in the first scenario. The outputs of the GW based approaches and the approaches using a uniform distribution to model the shape behave similarly, with the latter producing smaller values. In the second scenario, only the position changes, leading to a similar behavior of all approaches. The third scenario shows again the same results, similar behavior and higher values from the GW based approaches to the discretized Wasserstein distance



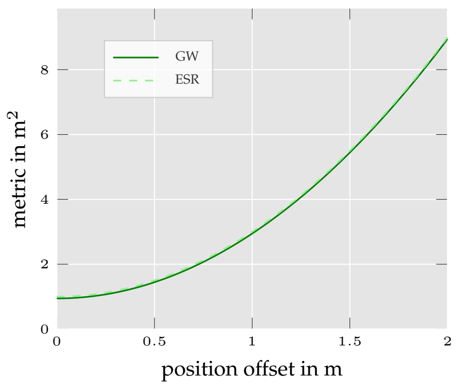
(a) Same ellipse rotating.



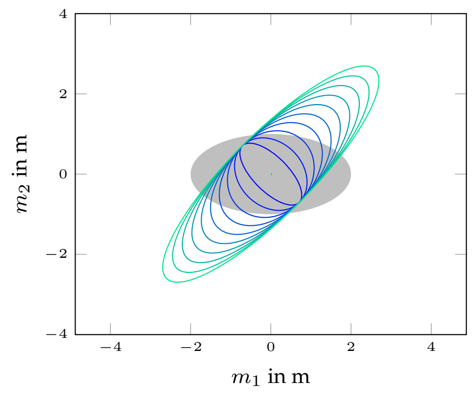
(b) Metric outputs for same ellipse rotating.



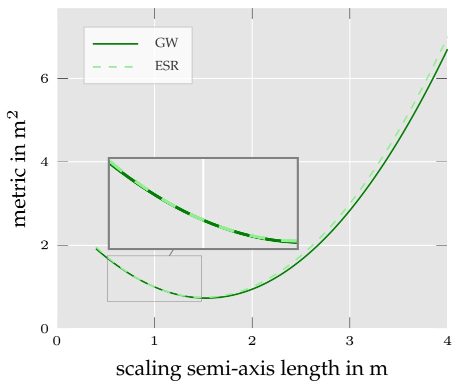
(c) Tilted ellipse moving.



(d) Metric outputs for tilted ellipse moving.



(e) Tilted ellipse scaling.



(f) Metric outputs for tilted ellipse scaling.

Figure 6.15: Comparison of squared GW and squared ESR distance with ground truth in gray, the estimates changing from blue to green, and the metric outputs over the different estimates in their respective color.

based approaches, with the difference getting higher for greater differences in the semi-axes. The minimum is similar for all approaches except for the Euclidean mean, which attains its minimum when both rectangles have the same length. Please note that the uneven behavior of the surface discretization Wasserstein metric in the scaling scenario is due to the change of the number and position of the discretization points on the surface as the shape's scale changes.

As the greatest difference between ellipses and rectangles occurs in the case of equal semi-axes lengths, i.e., the orientation is irrelevant for a circle but not for a square, we repeat the experiments with a square as ground truth. We modify the third scenario to scale the estimate equally so that it remains a square. The results can be found in Figure 6.17. Most notably is that the GW distance differs more greatly here. As expected, it considers the distance between all squares as 0 as long as only the rotation differs. In the second scenario, the GW distance is more similar to the discretized Wasserstein distances, but in the third, the difference gets greater for greater scaling. Another interesting point from the third scenario is that for squares, all approaches have their minimum when the squares have the same size.

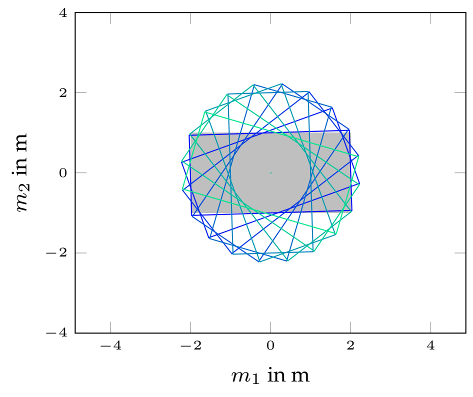
6.4 Estimation via MMGW

The goal in this section is to evaluate the quality of our estimator, the approximated MMGW estimator using the ESR distance, compared to the actual MMGW estimator and to the estimate using the Euclidean distance in Section 6.4.1. These experiments are based on the publication [3]. Furthermore, we extend the publication by providing two more experiments. The first is an evaluation on rectangles in Section 6.4.2, comparing the MMGW estimate to the estimate minimizing the Wasserstein distance over the discretized surface, as described in Section 5.5. For the second new experiment, we compare the MMGW estimator to the regular Euclidean estimate on the posterior density of the MEM-EKF* in different tracking scenarios in Section 6.4.3.

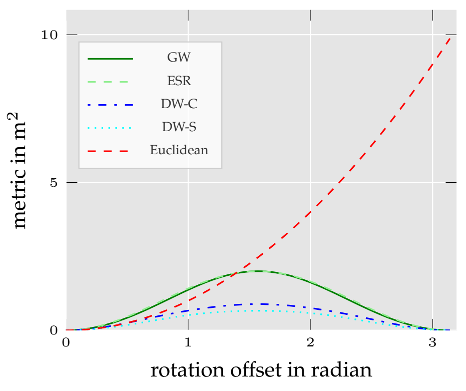
6.4.1 MMGW on Ellipses

Based on our analysis, we found that the orientation plays a major role when it comes to the difference between the estimators. In addition, the GW distance between two ellipses with the same orientation boils down to the Euclidean estimate, which is further elaborated in Section 7.3.1. Therefore, we focus on comparing the estimation for different orientation noise values. We provide three Gaussian distributed ellipse densities, all with mean

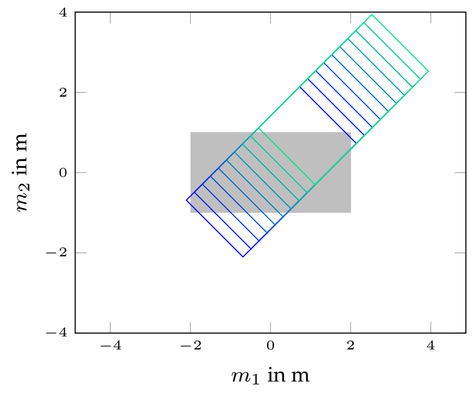
$$\hat{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & 8 & 3 \end{bmatrix}^{\mathrm{T}} , \tag{6.12}$$



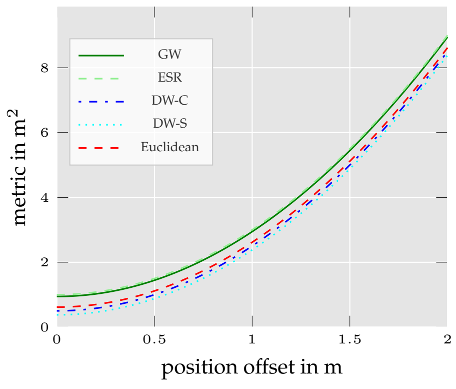
(a) Same rectangle rotating.



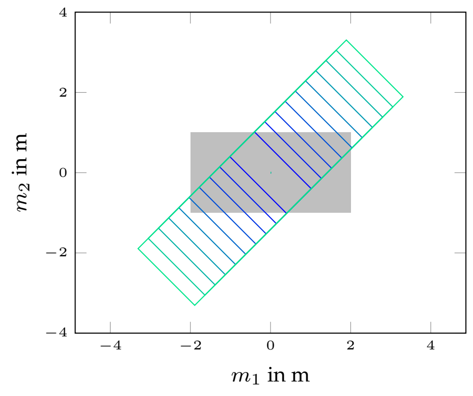
(b) Metric outputs for same rectangle rotating.



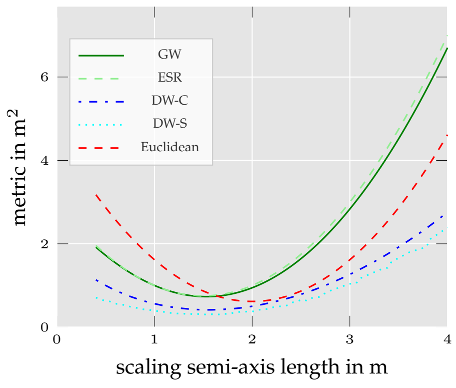
(c) Tilted rectangle moving.



(d) Metric outputs for tilted rectangle moving.

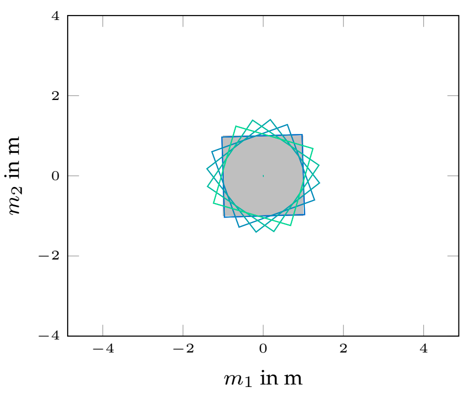


(e) Tilted rectangle scaling.

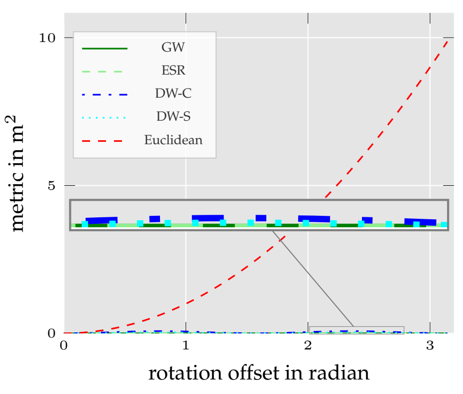


(f) Metric outputs for tilted rectangle scaling.

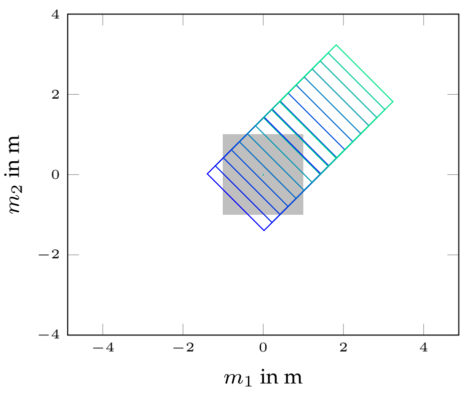
Figure 6.16: Comparison of squared GW distance, ESR distance, Euclidean distance, and Wasserstein distance with 100 contour discretization points (DW-C) and with at least 400 surface discretization points (DW-S) on rectangles with the ground truth in gray, the estimates changing from blue to green, and the metric outputs over the different estimates in their respective color.



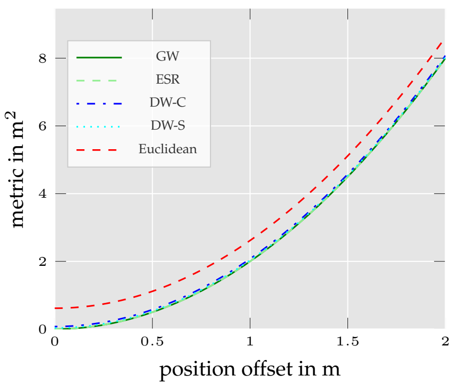
(a) Same square rotating.



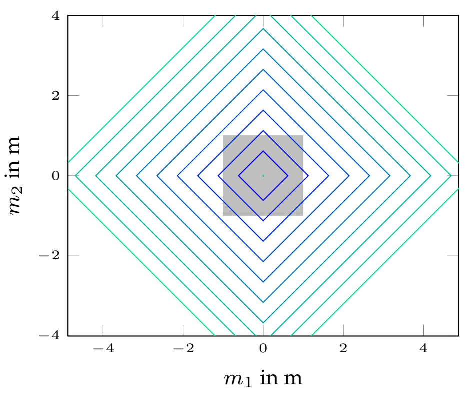
(b) Metric outputs for same square rotating.



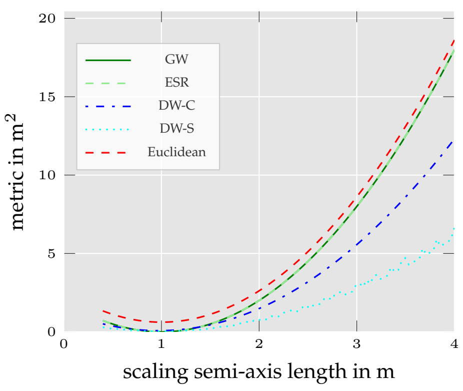
(c) Tilted square moving.



(d) Metric outputs for tilted square moving.



(e) Tilted square scaling.



(f) Metric outputs for tilted square scaling.

Figure 6.17: Comparison of squared GW distance, ESR distance, Euclidean distance, and Wasserstein distance with 100 contour discretization points (DW-C) and with at least 400 surface discretization points (DW-S) on squares with the ground truth in gray, the estimates changing from blue to green, and the metric outputs over the different estimates in their respective color.

consisting of center, orientation, and semi-axes lengths. The kinematics are omitted as the focus is on the shape estimate (see Section 5.1.3). The first density has the covariance

$$\mathbf{C} = \mathrm{diag}\Big(\begin{bmatrix} 0.5 & 0.5 & 0.01\pi & 0.5 & 0.5 \end{bmatrix} \Big) \ , \tag{6.13}$$

representing low noise on the orientation. The second increases the orientation variance to 0.2π and the third to 0.5π. To calculate the approximated MMGW estimate, we draw 1000 particles from the density described above, transform them into square root matrix space using (5.21), and average them. We compare our results with three other estimators. One uses the Euclidean distance as a cost function and can be obtained as the mean of the Gaussian density, i.e, (6.12). Another is the MMGW estimator minimizing the GW distance to the particles using an optimization method from [PRV20], which we initialize with equal weights and the approximated MMGW estimate as initial guess. Finally, as the approximated MMGW estimate is gained by averaging the square root matrices, we are interested in the behavior of the average of the shape matrices, i.e., the estimate minimizing the Euclidean distance in shape matrix space.

The 1000 particles we draw above can be used to represent the density described by (6.12) and (6.13). We use them to approximate the mean squared GW distance of each respective estimate to the density. Table 6.1 shows the metric outputs and Figure 6.18 shows the estimates given each metric, in both cases compared to the squared GW distance and its estimate.

The approximated MMGW estimator provides similar results to the MMGW estimator calculated via optimization, which confirms the observations from Section 6.3. Furthermore, the theoretical similarity of the Euclidean distance to the GW distance for commuting ellipses as discussed in Section 7.3.1 is shown here experimentally in the case of low orientation noise. It can be seen visually as well as in the values of the mean squared GW distance that the difference between the estimates is minimal for low orientation noise and increases for higher variances in orientation. We can also observe that the semi-axes lengths of the estimate minimizing the GW distance grow closer for higher orientation noise, resulting in a more circular estimate. The averaging of the shape matrices produces a slightly larger estimate than that of the MMGW estimator, being even worse than the Euclidean estimate for the density with low orientation noise. We show theoretically in Section 7.3.2 for commuting matrices why this occurs.

6.4.2 MMGW on Rectangles

We use the same density and covariance settings as described in the previous section. For the MMDW estimators, we also sample 1000 particles from the prior and apply the discretization of the entire surface as well as of only the contour. For the contour, we simply calculate a set of 100 equally spaced points. For the surface, we use optimization to find a set of equally spaced

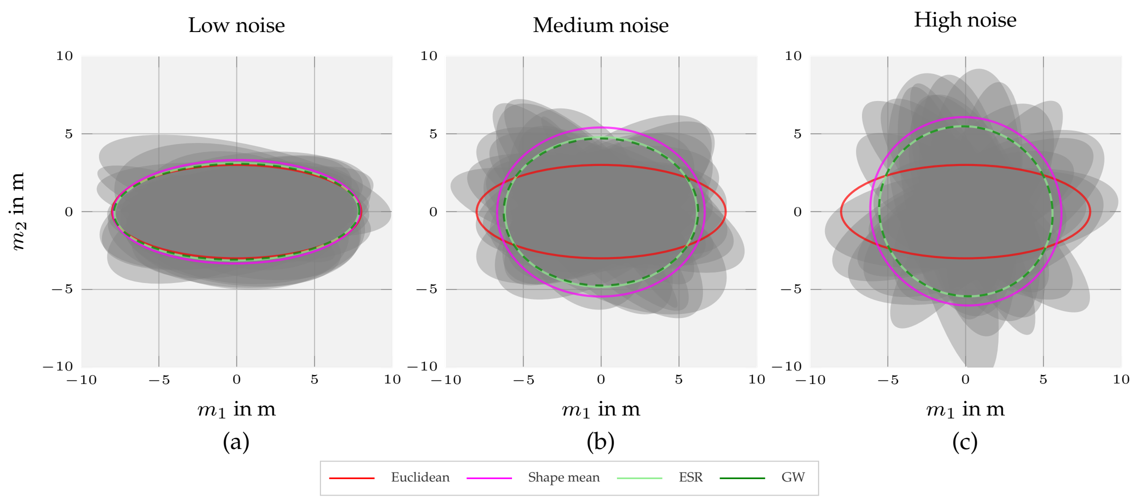


Figure 6.18: MMSE estimates on ellipses based on Euclidean distance, shape matrix mean, ESR distance, and GW distance for different orientation noises. 20 sample particles to highlight the uncertainty are drawn in gray. Based on [3].

	Euclidean	shape mean	ESR	GW
Test *low*	1.8065	1.8169	1.8013	1.8001
Test *medium*	4.3642	3.7792	3.6913	3.6897
Test *high*	5.0107	3.8869	3.7994	3.7994

Table 6.1: Squared GW error of MMSE estimate using Euclidean distance, the shape matrix mean, the approximated MMGW estimate using ESR distance, and the MMGW estimate using GW distance. The results are presented for different orientation noises. Based on [3].

points that best fit the area of the rectangle and which contain a minimum of 400 points. Similar to the squared GW distance before, we calculate the distance between two rectangles as the mean squared Wasserstein distance between their discretization points.

The results can be found in Figure 6.19 and the error based on the Wasserstein distance using discretization points with a discretization of the contour as well as of the surface can be found in Table 6.2 and Table 6.3 respectively. We visualized the Euclidean and the MMGW estimates as rectangles and plotted the actual point clouds for the MMDW estimates (as mentioned in Section 5.5, we would need to fit a rectangle onto them, which would introduce the additional challenge of finding an appropriate algorithm, which is not part of this analysis). An important observation here is that the higher the orientation noise is, the less the MMDW estimate looks like a rectangle (see, e.g., Figure 6.19b and 6.19c).

The results show that, as expected, the MMDW point cloud regarding contour or surface discretization is always best in respect to their respective metric. The main problem with MMDW-S and MMDW-C however is their runtime, as can be seen in Table 6.4. The approximated MMGW is

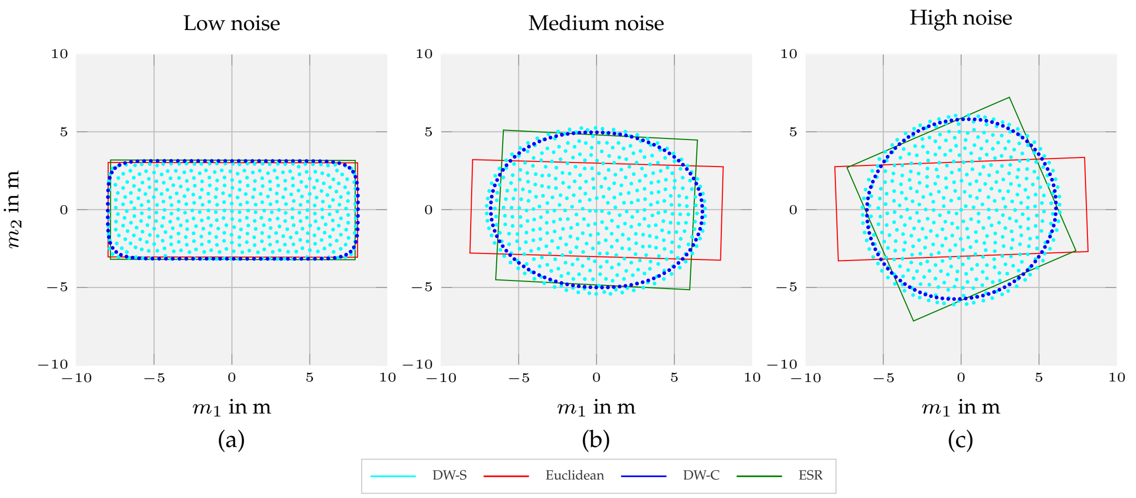


Figure 6.19: MMSE estimates on rectangles for different orientation noises given Euclidean distance, ESR distance, DW-C, and DW-S.

	Euclidean	ESR	DW-C
Test *low*	0.9313	0.9313	0.9304
Test *medium*	2.5013	2.3416	2.2223
Test *high*	3.3248	2.6395	2.4915

Table 6.2: Wasserstein error based on contour discretization (100 points) of the MMSE estimate using Euclidean distance, the approximated MMGW estimate using ESR distance, and the MMDW-C estimate using DW-C. The first test uses low, the second medium, and the third high orientation noise.

	Euclidean	ESR	DW-S
Test *low*	0.8971	0.8953	0.8733
Test *medium*	2.2536	2.0895	2.0345
Test *high*	2.9926	2.3331	2.2814

Table 6.3: Wasserstein error based on surface discretization (at least 400 points) of the MMSE estimate using Euclidean distance, the approximated MMGW estimate using ESR distance, and the MMDW-S estimate using DW-S. The first test uses low, the second medium, and the third high orientation noise.

	MMGW	MMDW-C	MMDW-S
Test *low*	0.0342	408.29	2793.85
Test *medium*	0.0432	2573.94	3739.63
Test *high*	0.0264	2007.84	2895.39

Table 6.4: Runtime in seconds of the estimators in scenarios with low, medium, and high noise on the orientation.

much faster while still giving a good approximation, especially for the MMDW-S. This is expected as the GW distance focuses the probability mass on the center and less on the contour, so it better approximates a uniform density on the entire surface and not the contour. It is also expected that due to the concentration on the center, the MMGW estimates are slightly smaller than the MMDW-S estimates. Another interesting observation here is that the contour discretization of the MMDW-C prefers slightly smaller shapes under high orientation noise than the MMDW-S (see Figure 6.19c). The high error scenario also shows that the MMGW estimates disregards the actual mean orientation. However, the MMDW estimates prefer circular estimates here, which would make it difficult to fit a proper rectangular estimate onto them. Overall, it seems that an ellipse might be a better estimate in this situation, which can be easily obtained from using the MMGW estimator. In summary, we demonstrated that the MMGW estimate is a well working approximation in respect to the discretized Wasserstein distance and can provide reasonable, if only slightly too small, estimates on rectangles as well.

6.4.3 MMGW in Extended Target Tracking

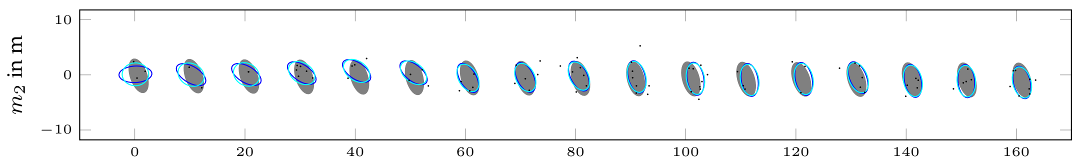
To demonstrate the benefit of using the MMGW estimate on the posterior density in ETT, we provide a simulated example. In Section 6.4.1, we demonstrated that the Euclidean estimate is a good approximation of the MMGW estimate if orientation noise is low. Therefore, we provide a simulation with an ellipse turning independently of the velocity vector. The ellipse is tracked using the MEM-EKF* and the Euclidean estimate as well as the MMGW estimate are provided.

The prior density is

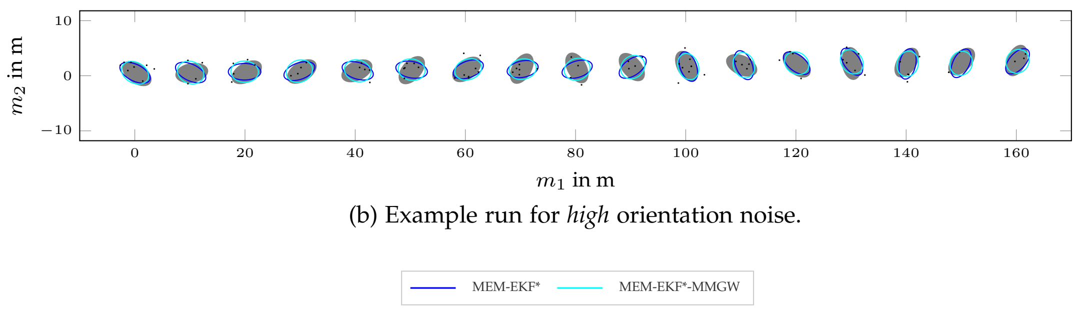
$$\hat{\mathbf{x}}_0 = \begin{bmatrix} 0 & 0 & 10 & 0 & 0 & 3 & 1.5 \end{bmatrix}^{\mathrm{T}} , \tag{6.14}$$

$$\mathbf{C}_0^{\mathbf{x}} = \operatorname{diag}\Big(\begin{bmatrix} 0.25 & 0.25 & 0.01 & 0.01 & 0.5\pi & 0.04 & 0.01 \end{bmatrix} \Big) , \tag{6.15}$$

to ensure a high uncertainty on the initial orientation. It consists of center, Cartesian velocity, orientation, and semi-axes lengths. As the focus is on the shape estimation, we only consider a Cartesian velocity and propagate the state in time using an NCV model. The velocity is corrupted by process noise $\sigma_{\dot{m}_1} = 0.1$ and $\sigma_{\dot{m}_2} = 0.1$. In addition, we assume a turning of the shape independent of the kinematics, simulated by a random process noise added to the orientation in each time step. For the variance of the orientation process noise, we have two setting. For *low*, it is 0.01π and for *high*, it is 0.2π. In each time step, the ellipse produce a number of measurements drawn from a Poisson distribution with mean 5. They are distributed uniformly across the surface and corrupted by zero-mean measurement noise with covariance $\mathbf{C}^{\mathbf{v}} = \operatorname{diag}(\begin{bmatrix} 1 & 1 \end{bmatrix})$. The important factors are high noise on the orientation and low measurement rate under high noise to ensure the orientation uncertainty is still high after the update. To account for the high noise, we conducted 1000 MC runs and calculated the mean squared GW error over the time steps. Example trajectories can be found in Figure 6.20 and the results are in Figure 6.21. For *low* noise, we see that



(a) Example run for *low* orientation noise.



(b) Example run for *high* orientation noise.

Figure 6.20: Tracking with Euclidean vs MMGW estimation with ground truths in gray and measurements as black dots.

the MMGW estimate is better in the first time step due to the high orientation noise in the prior, but as more measurements are generated and the estimate grows more certain in the orientation, the MMGW error of the MMGW and the Euclidean estimate become similar. This is expected given the results of Section 6.4.1. In *high* noise, the uncertainty on the orientation parameter stays high. As a result, the MMGW estimate produces a lower error than the Euclidean estimate over all time steps.

6.5 Fusion using REDs

This evaluation is based on [3]. The goal is to compare the iterative fusion via REDs and via the approximated fusion in square root space from Section 5.4. We consider the fusion using the Kalman filter formulas on the original densities, i.e., using the Euclidean distance, as state of the art. Other fusion methods in recent literature focus on RMs and thus inverse Wishart densities. The one in [RX20] considers orientation and semi-axes lengths and combines the shape matrices. It does, however, share the original measurement point clouds, so we will not consider it here. To have a baseline, we will however apply an averaging of the shape matrices, which we call “shape mean”. To consider a shape change, e.g., through orientation changes, we model the shapes as inverse Wishart densities and combine them using the method from [LJMD15], as well as the forget model described in [GsFS16] for the prediction step.

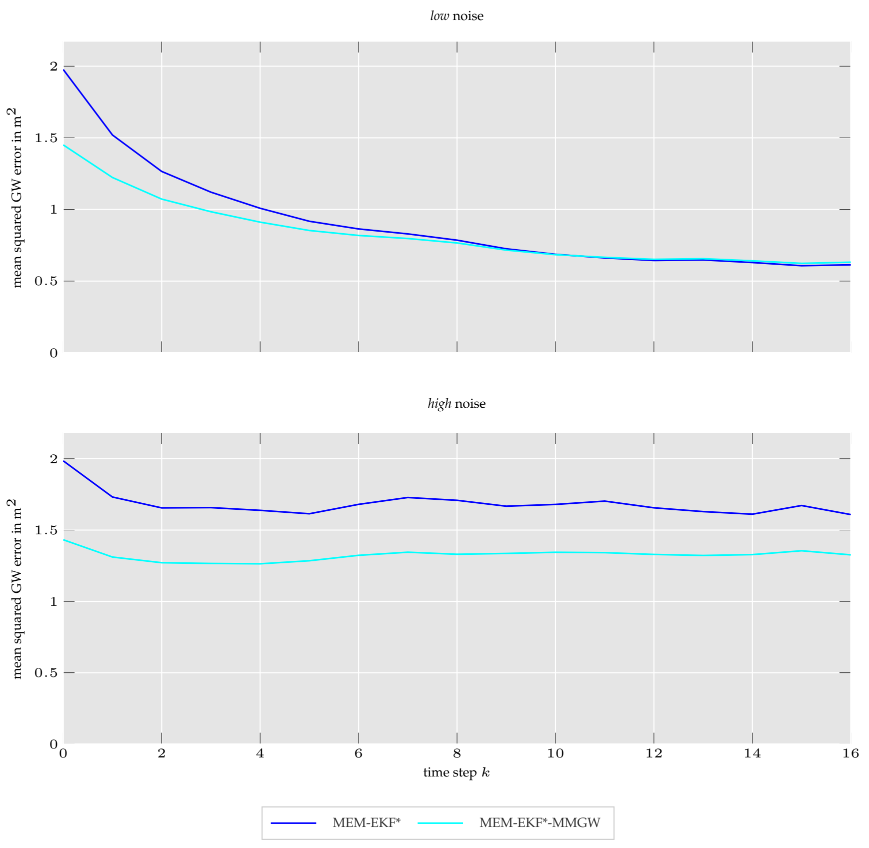


Figure 6.21: Mean squared GW error for Euclidean vs MMGW estimation.

We use 1000 particles for the approximation of the MMGW estimate in the RED-MMGW and the approximation of the state as a Gaussian in square root space for the MC-MMGW. In addition to the Kalman filter based fusion, here named “Regular”, we provide the Kalman filter based fusion using the MMGW estimator on the posterior, named “Regular-MMGW”. For the “shape mean”, each inverse Wishart density gets a degree of 6. We consider a moving ellipse and apply an NCV model with Cartesian velocity.

We conduct three experiments, each with prior mean and covariance

$$\hat{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 10 & 0 & 0 & 8 & 3 \end{bmatrix}^{\mathrm{T}} , \tag{6.16}$$

$$\mathbf{C} = \mathrm{diag}\Big(\begin{bmatrix} 0.5 & 0.5 & 10.0 & 10.0 & 0.5\pi & 0.5 & 0.5 \end{bmatrix} \Big) , \tag{6.17}$$

consisting of 2D center, 2D velocity, orientation, and semi-axes lengths. The orientation noise is chosen high enough to provide low initial knowledge about the orientation. We conducted

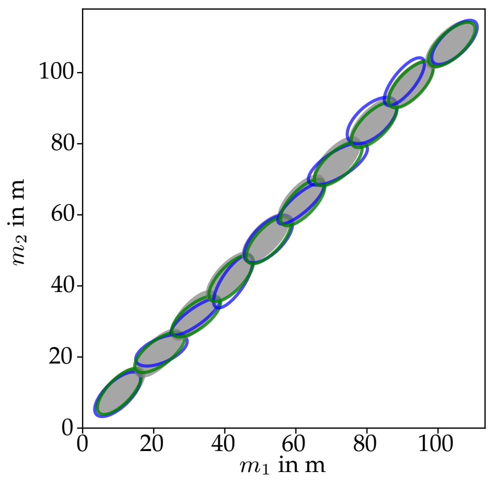


Figure 6.22: Sample run under low orientation noise with ground truth in gray, sensor estimates in blue, and the centrally fused RED-MMGW estimate in green for every other time step. [3]©2021IEEE

1000 MC simulations. For each, the ground truth is sampled from the prior and then propagated over 20 time steps using random zero-mean process noise on the velocities with standard deviations $\sigma_{\dot{m}_1} = 0.1$ and $\sigma_{\dot{m}_2} = 0.1$. To provoke a turning motion, we align the shape orientation with the velocity vector. In each time step, a measurement is generated by taking the ground truth, adding ambiguity by transforming it according to (5.7) with a random integer k, and then adding measurement noise. For the first experiment, titled *low*, the noise covariance is $\mathbf{C}_s = \text{diag}\left(\begin{bmatrix} 0.5 & 0.5 & 0.05 & 0.05 & 0.01\pi & 0.5 & 0.1 \end{bmatrix}\right)$. The second experiment, *medium*, uses the same covariance and only changes the orientation variance to 0.2π, and the third one, *high*, uses 0.5π. The three settings can be compared to the noise settings in Figure 6.18. In addition to the orientation noises, the semi-axes lengths have different uncertainties. Due to the switching of the axes to model ambiguities, not all measurements of the same axis have the same variance, simulating different axis uncertainties by different sensors. To handle the shape process and measurement noise in the "shape mean" approach, we used a forget parameter of 0.2 in *low*, 5.0 in *medium*, and 10.0 in *high*, which provided the best results in our experiments. Figure 6.22 shows an example run using the RED-MMGW, plotting only every other time step for better visibility. We use the squared GW distance as an error function. The results over 1000 MC runs with respect to the root mean squared error can be found in Figure 6.23.

Figure 6.23a shows that the regular fusion fails even under *low* orientation uncertainty due to the ambiguities. The shape mean converges, but RED-MMGW and MC-MMGW provide the best results. As demonstrated before, the combination of shape matrices instead of their square roots can produce estimates that are too large (considering the GW distance). A second reason for the worse estimate of the shape mean is that it does not consider the covariances.

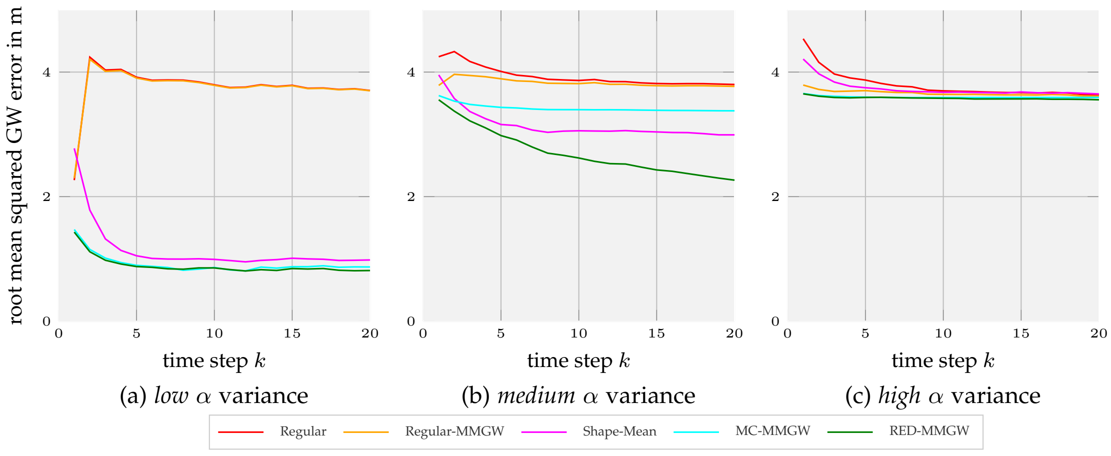


Figure 6.23: Tests comparing different ellipse fusion approaches under different sensor noise. Based on [3].

For *medium* noise, the advantage of the RED-MMGW over the MC-MMGW can be seen in Figure 6.23b. It seems that the transformed density is less Gaussian the higher the noise is in the orientation, which can be a possible explanation of the decrease in estimation quality of the Gaussian approximation of the MC-MMGW.

Figure 6.23c reveals that under *high* noise, the methods seem similar again. We demonstrated before that under high uncertainty in the orientation, a circular estimate is preferred by the GW distance. However, due to the averaging of the wrong semi-axes lengths in the regular fusion, a circular estimate is produced as well, e.g., Figure 5.1, meaning the better results are only achieved by coincidence. The Gaussian approximation also uses a circular mean for all measurements in MC-MMGW, so the information loss from the Gaussian approximation is less severe.

The experiments demonstrate the effectiveness of the RED in dealing with ambiguities. Note that if no ambiguity would exist, the regular Kalman filter based fusion would of course be optimal. However, as demonstrated in Section 6.4, using the MMGW estimator on the posterior of the fusion can still improve the results with respect to the mean squared GW distance under high orientation uncertainty.

7 Discussion of the Results

Contents

In this chapter, the methods we compared, proposed, and evaluated in this thesis are discussed. We first discuss the results of the comparison of elliptic trackers in Section 7.1, followed by a discussion of the GM-MEM-EKF* in Section 7.2. After that, we examine the MMGW in Section 7.3 and finally the fusion via REDs in Section 7.4.

7.1 Discussion of Elliptic Target Trackers

In this section, general advantages and disadvantages of the algorithms are discussed as well as the benefits of the range rate usage. It is based on the publications [1] and [5].

Table 7.1 summarizes Ellipse-RHM-EKF, which has the general issues of a bias in high sensor noise and a strong sensitivity to the prior because of the greedy association. While the Ellipse-RHM-EKF-imp, summarized in Table 7.2, reduces the bias, it is not gone completely and the prior dependency is still present. This is demonstrated by the results of Section 6.1. However, both RHMs are still flexible and can be changed to a different star-convex shape given the circumstances. Depending on the underlying model, more complex parameters to describe the shape can be chosen, e.g., by incorporating the support points of the GP based RHM [WÖ15] into the state vector. Also, if the prior is good, they can even deal with measurements originating only from the visible side. The RHM in general can offer tight coupling of kinematic and shape parameters. The joint state vector makes it possible to use the same parameter for the orientation of the shape

Pros	Cons
straightforward measurement model	bias in high noise
coupling of shape and kinematic parameters	sensitive to prior
extension via range rate straightforward	

Table 7.1: Pros and Cons of Ellipse-RHM-EKF. Based on [1].

Pros	Cons
reduced bias in high noise	complicated formulas
coupling of shape and kinematic parameters	sensitive to prior
extension via range rate straightforward	

Table 7.2: Pros and Cons of Ellipse-RHM-EKF-imp. Based on [1].

and velocity while keeping correlations to the kinematic and shape parameters. Extending the measurement equation via the range rate is straightforward due to the greedy association [6]. Its usability for different shapes, however, needs to be confirmed in future work.

As the IAE uses the spread matrix, a sufficient number of measurements are necessary. But it is possible to skip one semi-axis length during the update if necessary. Furthermore, it is unclear how to handle different noise covariances for the individual measurements, e.g., if the original measurement noise is in polar coordinates, the approximated Cartesian noise can differ for each measurement. However, the IAE requires only few formulas and works similarly well than the MEM-EKF* if its conditions are met. The findings are summarized in Table 7.3.

The MEM-EKF* is summarized in Table 7.4. While the IAE and MEM-EKF* both provide the best results in our scenarios with respect to the GW distance, the MEM-EKF* is more general. It works if velocity and shape are not aligned, if the measurement rate is low, i.e., 1, or if a target is stationary. This can be expected as the MEM-EKF* is problem-tailored to this specific scenario. An extension for rectangular targets is straightforward by using a variance of $\frac{1}{3}$ instead of $\frac{1}{4}$ for the multiplicative noise to approximate a uniform distribution between -1 and 1. In addition, we have shown how the range rate can be incorporated into the MEM-EKF* and demonstrated its benefits in simulations.

7.2 Discussion of Gaussian Mixture Modeled Multiplicative Noise

This section is based on [4] and extends the publication via additional theoretical differentiation to other methods and discussion about range rate and the influence of the sequential update. As described in Chapter 4 and demonstrated in Section 6.2, a GM can be used as a multiplicative noise term to extend the MEM-EKF* to a more complex measurement source distribution. Unlike RHMs,

Pros	Cons
no bias in high noise	needs ≥ 3 measurements with same noise
can deal with bad prior	no coupling of axes and kinematics
works well even in strong turns	needs velocity orientation

Table 7.3: Pros and Cons of IAE. Based on [1].

Pros	Cons
no bias in high noise	no contour measurements
can deal with bad prior	no coupling of shape and kinematics
works well even in strong turns	sensitive to shape covariance
can be extended by range rate	

Table 7.4: Pros and Cons of MEM-EKF*. Based on [1].

the shape does not need to be star-convex and the distribution can be arbitrary as it essentially consists of a set of Gaussians. However, unlike the GP based RHM, the shape is less flexible during the update, as it only depends on the orientation and two semi-axes lengths.

The difference to other GM based methods is that [KHB19] handle the unknown associations via EM. In a different approach, [HK20] handle them via cluster processes by keeping association events using an RFS framework. Additionally, both methods learn the additive sensor noise as part of the GM. Our method allows to separate the distribution of the measurement sources and the additive sensor noise, which can be of advantage if the sensor noise depends strongly on the sensor to target geometry. Another method using a learned GM is [SD18], which uses a particle filter. It should be noted that tuning the distribution of the measurement sources requires either large amounts of data in low noise scenarios to learn them or expert knowledge to tune them by hand.

The method in [AGS19] uses GMs as well, with the difference to our method being that they model the distribution of the measurement sources as a discrete density, while we describe the measurement sources via a GM distribution across the shape. While we then end up with a GM posterior component for each of the GM components describing the distribution, they end up with a GM posterior component for each discrete measurement source. The only similarity is that they also reduce the posterior to one component using moment matching.

The RM based approach using GMs from [ZL21] also uses optimization. Furthermore, they only provide a shift and a scaling factor, no rotation of the original RM, making it less flexible, e.g., to represent the four sides of a vehicle.

While the method from [CLLL21] uses multiplicative scaling factors, similar to the multiplicative noise, and differentiates different scattering regions across the shape, the method has several differences, i.e., they model the rectangle as a set of corners and use constraints to keep the rectangular shape. Alternatively, orientation and semi-axes can be used but need to be transformed into the space of rectangle corners. They use the corners to model a uniform distribution for each line segment between two corners, which can be extended to different polynomials as well. In addition to the line segments, they also model a distribution over the surface of a rectangle using two scaling factors, assuming a uniform distribution. In summary, they have different assumption of the measurement source distribution than our approach. Therefore, a direct comparison is not feasible. Still, while they do not do this in their work, the two scaling factor can be used analogously to the multiplicative noise to model segments on the surface. Additionally, while they handle association via PDA instead of moment matching, they propose to use a sequential update, which would result in the same formulas as our moment matching approach. However, they do not weight the different regions depending on the aspect angle to the sensor, instead they use gating. It is unclear how the gating procedure would be modified if the surface is split into regions as in our approach. While it can be shown that the formulas for their model using orientation and axes could be transformed to the measurement equation (3.14) used in MEM-EKF*, they do not use a pseudo-measurement for the shape update and they apply an UKF to handle non-linearities and keep all state parameters in a combined state vector. A comparison between our proposed approach and a UKF implementation would therefore be interesting in future work.

We have demonstrated that the GM-MEM-EKF* works well in our simulations. However, due to the dependency on the prior, it is more vulnerable to strong process noise. We must also note that the order of measurements in the sequential update can influence the results. In future work, the exact impact and how to find an optimal order needs to be investigated.

An inclusion of the range rate measurement is possible for the kinematic update. However, while it produces good results in our simulations for a uniform measurement distribution as demonstrated for the MEM-EKF* in Section 6.1.3, initial tests on simulations for more complex GMs have concluded that the non-linearities are too strong for the range rate to provide a benefit. Depending on the process noise and sensor to target geometry, the inclusion of the range rate can even worsen the results. Therefore, a more thorough investigation of the inclusion of range rate into GM-MEM-EKF* needs to be conducted in future work. In addition, as with the greedy association, setting the weights based on the sensor position relative to the predicted target can lead to diverging tracks for highly uncertain priors, e.g., due to strong process noise on the yaw rate. In these cases, a combination with a more robust association method as was done in other trackers using a GM for the measurement distribution might be necessary.

7.3 Discussion of Minimum Mean Gaussian Wasserstein Estimation

This section is based on the discussion in the publication [3] and extends it via a discussion of the usage of MMGW on rectangles. The evaluation of the MMGW estimation provided insights we want to discuss further in this section. First, the MMGW estimate is compared to the classic estimate using Euclidean distance in Section 7.3.1. Then, the relation between the MMGW concept and fusion concepts of shape matrices, like RM fusion methods, are discussed in Section 7.3.2. Finally, we comment on the use of the MMGW estimator for rectangles compared to the MMDW estimators in Section 7.3.3.

7.3.1 GW vs Euclidean Distance on Ellipse Parameters

We have demonstrated that from a geometrical point of view, the influence of the orientation uncertainty is not considered properly in the classical estimation methods. A high uncertainty in the orientation means that the true ellipse can be rotated by nearly $90°$ to the mean of the posterior density. Especially for large ellipses, this can be a bad estimate based on a metric that actually considers the shape's area, like the GW distance. The MMGW estimator, on the other hand, considers this by increasing the estimate's minor axis length while decreasing its major axis length for increasing orientation noise. If the noise is high enough that essentially no information about the orientation is given, the estimator deals with it by providing circular estimates. It should be highlighted that these are just the point estimates. The mean of the density is preserved, as the semi-axes lengths might be certain and can be used if future updates decrease the orientation uncertainty.

If the orientation noise is low, however, the estimates are similar. To understand this relation, we need to look at the calculation of the ESR distance which we use to approximate the GW distance. If orientation noise is low, the majority of the density belongs to ellipses with almost overlapping semi-axes. If the axes overlap, the ESR distance boils down to

$$\begin{aligned}&\mathrm{ESR}(\mathbf{m_z},\mathbf{Z},\mathbf{m_x},\mathbf{X})^2 = \left|\left| \begin{bmatrix}(\mathbf{m_z}-\mathbf{m_x})^{\mathrm{T}} & v_k(l_\mathbf{z},w_\mathbf{z})-l_\mathbf{x} & v_{k+1}(l_\mathbf{z},w_\mathbf{z})-w_\mathbf{x}\end{bmatrix} \right|\right|_2^2\\ &\forall \alpha_\mathbf{z}, k \text{ with } (\alpha_\mathbf{z}=\alpha_\mathbf{x}+k\frac{\pi}{2}) \text{ and } k\in\mathbb{Z}\ ,\end{aligned} \tag{7.1}$$

with $v_k()$ from (5.8). As all orientations would be the same and thus, the axes are assigned correctly as well, the ESR distance and the Euclidean distance would be very similar for low orientation uncertainty. Additionally, the matrices would commute, which means that the approximation of the GW distance by the ESR distance is exact. Therefore, the Euclidean distance leads to a good estimate in regards to the GW distance for low orientation noise, which explains the results of Section 6.4.

This means that the Euclidean estimate can be used as a good approximation of the MMGW estimate if we know that orientation noise is low. In settings with high orientation noise, however, the MMGW estimate should be preferred to properly incorporate geometric information of the target as demonstrated in Section 6.4.3.

7.3.2 GW vs Euclidean Distance on Shape Matrices

To differentiate our estimation method from those used for RMs, e.g., for particle densities of RMs [VGBW17, RX20], we highlight that their estimate is gained as the mean of the shape matrices. However, the MMGW estimate is gained from the matrices' square roots, i.e., it finds the Euclidean mean in shape matrix square root space and not in shape matrix space. In the context of the RM based methods, their approach can be helpful, but for our purposes, the square roots provide better results. The reasons are for one that, as shown in Chapter 5, using the square roots provides a good approximation of the GW distance. Additionally, using square roots also provides better estimates if the original density is distributed as a Gaussian in the ellipse parameter space. For example, consider commuting ellipses, i.e., two ellipses with overlapping semi-axes, with 0 orientation uncertainty. For one of the semi-axes lengths, two equally likely estimates are given, 2 and 10. Given the formulas for the MMGW estimator, the averaging of the square root matrices would boil down to averaging the length estimates $\frac{2+10}{2} = 6$. Averaging the shape matrices, on the other hand, would boil down to averaging the squared lengths, i.e., $\sqrt{\frac{4+100}{2}} \approx 7$. The resulting length estimate would be bigger than expected given the original density in ellipse parameter space. This has been further demonstrated by the results of Section 6.4.

7.3.3 GW vs Discretized Wasserstein Distance

We have presented MMDW estimators for rectangles. They turn the shapes into discretized densities, either discretizing only the contour or the entire surface, and minimize the Wasserstein distance between them. We have demonstrated that these estimators have a high runtime and the MMGW estimator can provide a fast and precise approximation. We want to note here, however, that these MMDW estimators can be applied to other, more arbitrary shapes as well, for which an approximation as a Gaussian might not be appropriate anymore. The effect that the MMDW estimator has on different shapes under different noise is, therefore, something that should be investigated in the future.

7.4 Discussion of RED Fusion

This section is based on the publication [3]. To discuss the results of our evaluation of the fusion via REDs, we discuss the relation between RED fusion and classic fusion in Section 7.4.1 and differentiate RED fusion to RM fusion in Section 7.4.2.

7.4.1 RED vs Fusion on Ellipse Parameters

An RED is a special density tailored for the ambiguities of ellipse densities parameterized with orientation and semi-axes lengths. The density can handle unclear associations between axes and orientation. However, if the noise is low enough and the ellipse parameters are actually distributed as a Gaussian with no danger of switching in the parameterization due to ambiguities, the regular Kalman filter would of course be optimal. In practice, this is not necessarily given. Different initialization or high noise might lead to ambiguous parameterization, which can result in unintuitive estimates as demonstrated in Section 6.5. We like to note that while the RED provides clear advantages in high orientation noise, the MC-MMGW has a similar performance if the orientation noise is low and it does not require GM reduction. Depending on prior knowledge on the uncertainties, one or the other might be preferable.

The advantage of the RED can be applied to multi-target problems as well. We have not considered this challenge in the context of this thesis, but it would be a logical follow-up and we want to discuss it theoretically here. For multi-target fusion, the correct targets need to be associated with each other. Simply using Euclidean distance between two ellipses can lead to unintuitive associations here as well (remember the example from Chapter 5 Figure 5.1, which showed two ellipses which were the same but were described by different state vectors). The REDs can solve this problem while offering the option to incorporate the uncertainties of the parameters.

7.4.2 RED vs Fusion of Random Matrices

Literature provides several methods to fuse RM densities. As they consider shape matrices distributed via an inverse Wishart density, its applicability to ellipse densities parameterized with orientation and semi-axes lengths is limited and a clear distinction is given.

[GO13] provides a method to merge shapes, not their densities. [LJMD15] suggest to calculate the weighted average of shape matrices to minimize the Kullback-Leibler divergence, which would dismiss the covariance of the original ellipse parameters if given. Additionally, the difference in averaging shape matrices and their square roots was discussed in Section 7.3.2. [LG19] represent the state in ellipse parameter space using a GM calculated via weighted samples from an importance distribution with the parameters of the original RM as mean. These mixtures cannot be directly translated to our scenario, in which the uncertainties in ellipse parameter space are given from the beginning. [HL19] provide an approximation via latent variables. Again, due to the differences, a direct comparison here is not feasible.

In summary, while transforming the estimates' means to RMs and fusing them with one of the given fusion methods is an option to handle the ambiguities, the REDs provide an alternative fusion approach which preserves the estimates' covariances.

8 Conclusion and Outlook

This thesis considers ETT on RADAR data using elliptic extended targets parameterized with center, orientation, and semi-axes lengths. We answer our first two research questions as follows.

RQ 1 We discussed the usage of range rate measurements in elliptic target trackers assuming a uniform distribution across the surface. We also determined via a theoretical discussion and evaluation on simulated data that the best tracker in the scenarios and under the assumptions we considered is the MEM-EKF*, which is problem tailored to elliptic extended targets. It does not show the bias visible in the RHM based trackers and is more flexible regarding the orientation than the IAE. Should its assumption of aligned orientation and velocity be given, the IAE is a well working alternative with fewer formulas. Due to these results, we implemented an extension of the MEM-EKF* for the range rate measurement from Doppler RADARs, providing the MEM-EKF*-D and evaluated it on simulated data.

RQ 2 We applied linearizations based on the MEM-EKF* as well as weight approximation and moment matching to derive a closed-form solution for a tracker which uses a GM based measurement distribution on the surface of an extended target, creating the GM-MEM-EKF*, which can handle the complex measurement source distribution and additive sensor noise.

We discussed fusion and estimation of elliptical extended targets in this thesis as well, highlighting disadvantages of classical approaches and under which circumstances they would fail. We proposed methods specific for ellipses parameterized with orientation and semi-axes lengths and answer the last two research questions as follows.

RQ 3 Regarding the unique challenges for fusing state densities modeled with orientation and semi-axes lengths, we defined a GM based density combining all representations of the same ellipse, an RED, which can handle the geometric properties in the fusion process and provided an implementation of it for iterative fusion.

RQ 4 Considering the estimation on ellipse densities, we discussed issues stemming from using the classic Euclidean distance in the cost function and suggested to use a distance measure on ellipses instead, the GW distance, deriving the MMGW estimator and an approximation based on the ESR distance.

For all proposed methods, we conducted evaluations by comparing them to the state of the art and discussed the results. As rectangular targets, i.e., bounding boxes, can be parameterized using orientation and semi-axes lengths as well, we also investigated the applicability of the methods presented in this thesis for rectangles.

For future work, more complex distributions using GMs can be tested and compared to methods using optimization techniques. A comparison to learned models in scenarios with varying sensor noise can also be conducted. It should be further investigated what the limitations and possible benefits of using the range rate for the more complex measurement distribution of the GM-MEM-EKF* are. In addition, benefits of using more advanced methods to handle the association to the components need to be evaluated. As the proposed method depends on the prior, it would be interesting to combine the tracker with a method handling association via EM [KHB19] or which keeps track of the different association hypotheses [HK20]. In that regard, it must be investigated whether such methods decrease the influence of the order of measurements in the sequential update under high process noise and if there is a way to find an optimal order.

For the fusion, several aspects are of interest here as well. A fusion with two REDs is technically not an RED anymore, so it should be investigated if reducing the result back to an RED instead of keeping it and applying mixture reduction can improve the performance. Otherwise, trying different mixture reduction strategies might lead to improvements as well. Furthermore, it must be investigated for REDs how correlations between different sensor estimates can be handled best, e.g., via covariance intersection, usage of the information form, or tracking of the correlation, and how the REDs can be applied in different sensor network topologies.

Additionally, while the MMGW estimator provides an estimate incorporating the geometric properties of ellipses, it must be investigated how to best determine the uncertainties of the parameters of that estimate. The estimate is the mean in square root space transformed back into ellipse parameter space. For the uncertainty, the evaluation of the MC-MMGW has proven that a Gaussian approximation of the density in square root space is not enough to capture all of the transformed density under high orientation noise. The question is therefore how to better capture the uncertainty in square root space and how to transform it back into ellipse parameter space.

Furthermore, the usage of REDs and the GW distance for association in multi-target scenarios is a topic of interest for future work. For arbitrary shapes, the same concepts need to be investigated, too. The MMDW estimators presented for rectangles can be used for other shapes as well if an appropriate discretization can be found. Therefore, its usage on different types of extended targets and approximations for lower runtime should be investigated in future work.

Own Publications

[1] K. Thormann, S. Yang, and M. Baum, "A Comparison of Kalman Filter-based Approaches for Elliptic Extended Object Tracking," in *IEEE 23rd International Conference on Information Fusion (Fusion)*, pp. 1–8, 2020.

[2] K. Thormann and M. Baum, "Optimal Fusion of Elliptic Extended Target Estimates Based on the Wasserstein Distance," in *IEEE 22nd International Conference on Information Fusion (Fusion)*, pp. 1–6, 2019.

[3] K. Thormann and M. Baum, "Fusion of Elliptical Extended Object Estimates Parameterized With Orientation and Axes Lengths," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2369–2382, 2021.

[4] K. Thormann, S. Yang, and M. Baum, "Kalman Filter Based Extended Object Tracking with a Gaussian Mixture Spatial Distribution Model," in *IEEE 32nd Intelligent Vehicles Symposium (IV), Workshop*, pp. 1–1, 2021.

[5] K. Thormann and M. Baum, "Incorporating Range-Rate Measurements in EKF-based Elliptical Extended Object Tracking," in *IEEE International Conference on Multisensor Fusion and Integration (MFI)*, pp. 1–1, 2021.

[6] K. Thormann, J. Honer, and M. Baum, "Extended Target Tracking Using Gaussian Processes with High-Resolution Automotive Radar," in *IEEE 21st International Conference on Information Fusion (Fusion)*, pp. 1764–1770, 2018.

[7] K. Thormann, J. Honer, and M. Baum, "Fast Road Boundary Detection and Tracking in Occupancy Grids from Laser Scans," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 348–353, 2017.

[8] K. Thormann, F. Sigges, and M. Baum, "Learning an Object Tracker with a Random Forest and Simulated Measurements," in *IEEE 20th International Conference on Information Fusion (Fusion)*, pp. 1–4, 2017.

[9] S. Yang, K. Thormann, and M. Baum, "Linear-Time Joint Probabilistic Data Association for Multiple Extended Object Tracking," in *IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 6–10, 2018.

Bibliography

[AC11] Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein Space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[ADFAM17] Waqas Aftab, Allan De Freitas, Mahnaz Arvaneh, and Lyudmila Mihaylova. A Gaussian Process Approach for Extended Object Tracking with Random Shapes and for Dealing with Intractable Likelihoods. In *IEEE 22nd International Conference on Digital Signal Processing (DSP)*, pages 1–5, 2017.

[AGS19] Hosam Alqaderi, Felix Govaers, and Raymond Schulz. Spacial Elliptical Model for Extended Target Tracking Using Laser Measurements. In *IEEE Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6, 2019.

[AMPG13] Donka Angelova, Lyudmila Mihaylova, Nikolay Petrov, and Amadou Gning. A Convolution Particle Filtering Approach for Tracking Elliptical Extended Objects. In *IEEE 16th International Conference on Information Fusion (Fusion)*, pages 1542–1549, 2013.

[AOÖ15] Tohid Ardeshiri, Umut Orguner, and Emre Özkan. Gaussian Mixture Reduction Using Reverse Kullback-Leibler Divergence. *arXiv preprint arXiv:1508.05514*, 2015.

[AP18] Akbar Assa and Konstantinos N. Plataniotis. Wasserstein-distance-based Gaussian Mixture Reduction. *IEEE Signal Processing Letters*, 25(10):1465–1469, 2018.

[ASKB12] Michael Aeberhard, Stefan Schlichtharle, Nico Kaempchen, and Torsten Bertram. Track-to-track Fusion with Asynchronous Sensors Using Information Matrix Fusion for Surround Environment Perception. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1717–1726, 2012.

[Bau13] Marcus Baum. *Simultaneous Tracking and Shape Estimation of Extended Objects*, volume 13. KIT Scientific Publishing, 2013.

[BDD17] Peter Broßeit, Bharanidhar Duraisamy, and Jürgen Dickmann. The Volcanormal Density for Radar-based Extended Target Tracking. In *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[BFF$^+$10] Marcus Baum, Michael Feldmann, Dietrich Fränken, Uwe D. Hanebeck, and Wolfgang Koch. Extended Object and Group Tracking: A Comparison of Random Matrices and Random Hypersurface Models. In *IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Leipzig, Germany, October 2010.

[BFH12] Marcus Baum, Florian Faion, and Uwe D. Hanebeck. Modeling the Target Extent with Multiplicative Noise. In *IEEE 15th International Conference on Information Fusion (Fusion)*, Singapore, July 2012.

[BGM$^+$20] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, 2020.

[BH09] Marcus Baum and Uwe D. Hanebeck. Random Hypersurface Models for Extended Object Tracking. In *IEEE 9th International Symposium on Signal Processing and Information Technology (ISSPIT)*, Ajman, United Arab Emirates, December 2009.

[BH11] Marcus Baum and Uwe D. Hanebeck. Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs. In *IEEE 14th International Conference on Information Fusion (Fusion)*, Chicago, Illinois, USA, July 2011.

[BH14] Marcus Baum and Uwe D. Hanebeck. Extended Object Tracking with Random Hypersurface Models. *IEEE Transactions on Aerospace and Electronic Systems*, 50:149–159, January 2014.

[BML$^+$20] Philipp Berthold, Martin Michaelis, Thorsten Luettel, Daniel Meissner, and Hans-Joachim Wünsche. Deriving Spatial Occupancy Evidence from Radar Detection Data. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 831–836, 2020.

[BNH10] Marcus Baum, Benjamin Noack, and Uwe D. Hanebeck. Extended Object and Group Tracking with Elliptic Random Hypersurface Models. In *IEEE 13th International Conference on Information Fusion (Fusion)*, Edinburgh, United Kingdom, July 2010.

[BNH11] Marcus Baum, Benjamin Noack, and Uwe D. Hanebeck. Mixture Random Hypersurface Models for Tracking Multiple Extended Objects. In *IEEE 50th Conference on Decision and Control and European Control Conference*, pages 3166–3171, 2011.

[BPS14] Christian Bailer, Alain Pagani, and Didier Stricker. A Superior Tracking Approach: Building a Strong Tracker Through Fusion. In *European Conference on Computer Vision*, pages 170–185. Springer, 2014.

[BRAD16] Peter Broßeit, Matthias Rapp, Nils Appenrodt, and Jürgen Dickmann. Probabilistic Rectangular-shape Estimation for Extended Object Tracking. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 279–285, June 2016.

[BS81] Yaakov Bar-Shalom. On the Track-to-Track Correlation Problem. *IEEE Transactions on Automatic control*, 26(2):571–572, 1981.

[BSDH09] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. The Probabilistic Data Association Filter. *IEEE Control Systems Magazine*, 29(6):82–100, 2009.

[BSFC90] Yaakov Bar-Shalom, Thomas E. Fortmann, and Peter G. Cable. Tracking and Data Association, 1990.

[BSLK04] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2004.

[BWBS$^+$17] Steven Bordonaro, Peter Willett, Yaakov Bar-Shalom, Tod Luginbuhl, and Marcus Baum. Extended Object Tracking with Exploitation of Range Rate Measurements. *ISIF Journal of Advances in Information Fusion*, 12(2), December 2017.

[BWH15] Marcus Baum, Peter Willett, and Uwe D. Hanebeck. On Wasserstein Barycenters and MMOSPA Estimation. *IEEE Signal Processing Letters*, 22(10):1511–1515, October 2015.

[BYML16] Stefan Breuers, Shishan Yang, Markus Mathias, and Bastian Leibe. Exploring Bounding Box Context for Multi-object Tracker Fusion. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8, 2016.

[CAM02] Lingji Chen, Pablo O. Arambel, and Raman K. Mehra. Estimation Under Unknown Correlation: Covariance Intersection Revisited. *IEEE Transactions on Automatic Control*, 47(11):1879–1882, 2002.

[CBL$^+$20] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice E. Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[CD14] Marco Cuturi and Arnaud Doucet. Fast Computation of Wasserstein Barycenters. In *31st International Conference on Machine Learning (ICML-14)*, October 2014.

[CLL20] Xiaomeng Cao, Jian Lan, and X. Rong Li. Extension-Deformation Approach to Extended Object Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 57(2):866–881, 2020.

[CLLL21] Xiaomeng Cao, Jian Lan, X. Rong Li, and Yu Liu. Automotive Radar-Based Vehicle Tracking Using Data-Region Association. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[COO15] Guillaume Carlier, Adam Oberman, and Edouard Oudet. Numerical Methods for Matching for Teams and Wasserstein Barycenters. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1621–1642, 2015.

[CS10] Kuo-Chu Chang and Wei Sun. Scalable Fusion with Mixture Distributions in Sensor Networks. In *IEEE 11th International Conference on Control Automation Robotics & Vision*, pages 1251–1256, 2010.

[CWPS11] David F. Crouse, Peter Willett, Krishna Pattipati, and Lennart Svensson. A Look at Gaussian Mixture Reduction Algorithms. In *IEEE 14th International Conference on Information Fusion (Fusion)*, pages 1–8, 2011.

[DGN$^+$16] Bharanidhar Duraisamy, Michael Gabb, Aswin V. Nair, Tilo Schwarz, and Ting Yuan. Track Level Fusion of Extended Objects from Heterogeneous Sensors. In *IEEE 19th International Conference on Information Fusion (Fusion)*, pages 876–885, 2016.

[DHL04] Zhansheng Duan, Chongzhao Han, and X. Rong Li. Sequential Nonlinear Tracking Filter with Range-Rate Measurements in Spherical Coordinates. In *7th International Conference on Information Fusion (Fusion)*, volume 1, pages 599–605. Citeseer, 2004.

[DKZ09] Ian L. Dryden, Alexey Koloydenko, and Diwei Zhou. Non-Euclidean Statistics for Covariance Matrices, with Applications to Diffusion Tensor Imaging. *The Annals of Applied Statistics*, 3(3):1102–1123, 2009.

[DMT$^+$21] Jean-Luc Déziel, Pierre Merriaux, Francis Tremblay, Dave Lessard, Dominique Plourde, Julien Stanguennec, Pierre Goulet, and Pierre Olivier. PixSet: An Opportunity for 3D Computer Vision to Go Beyond Point Clouds With a Full-Waveform LiDAR Dataset. *arXiv preprint arXiv:2102.12010*, 2021.

[DWH08] Hugh Durrant-Whyte and Thomas C. Henderson. Multi Sensor Data Fusion. *Springer Handbook of Robotics*, pages 585–610, 2008.

[DWS11] Johan Degerman, Johannes Wintenby, and Daniel Svensson. Extended Target Tracking using Principal Components. In *IEEE 14th International Conference on Information Fusion (Fusion)*, Chicago, Illinois, USA, July 2011.

[FBH19] Jaya Shradha Fowdur, Marcus Baum, and Frank Heymann. Tracking Targets with Known Spatial Extent Using Experimental Marine Radar Data. In *IEEE 22nd International Conference on Information Fusion (Fusion)*, Ottawa, Canada, July 2019.

[FCG$^+$21] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[FDZH16] Florian Faion, Maxim Dolgov, Antonio Zea, and Uwe D. Hanebeck. Closed-form Bias Reduction for Shape Estimation with Polygon Models. In *IEEE 19th International Conference on Information Fusion (Fusion)*, pages 581–588, 2016.

[FFK11] Michael Feldmann, Dietrich Fränken, and Wolfgang Koch. Tracking of Extended Objects and Group Targets using Random Matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, 2011.

[FGW20] Markus Fröhle, Karl Granström, and Henk Wymeersch. Decentralized Poisson Multi-Bernoulli Filtering for Vehicle Tracking. *IEEE Access*, 8:126414–126427, 2020.

[FS75] Herbert Freeman and Ruth Shapira. Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve. *Communications of the ACM*, 18(7):409–413, 1975.

[FZBH15] Florian Faion, Antonio Zea, Marcus Baum, and Uwe D. Hanebeck. Partial Likelihood for Unbiased Extended Object Tracking. In *IEEE 18th International Conference on Information Fusion (Fusion)*, Washington, USA, July 2015.

[FZH14] Florian Faion, Antonio Zea, and Uwe D. Hanebeck. Reducing Bias in Bayesian Shape Estimation. In *IEEE 17th International Conference on Information Fusion (Fusion)*, pages 1–8, 2014.

[GBR17] Karl Granström, Marcus Baum, and Stephan Reuter. Extended Object Tracking: Introduction, Overview and Applications. *ISIF Journal of Advances in Information Fusion*, 12(2), December 2017.

[GGMS05] Kevin Gilholm, Simon Godsill, Simon Maskell, and David Salmond. Poisson Models for Extended Target and Group Tracking. In *SPIE: Signal and Data Processing of Small Targets*, 2005.

[GK12] Felix Govaers and Wolfgang Koch. An Exact Solution to Track-to-Track-Fusion at Arbitrary Communication Rates. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2718–2729, 2012.

[GLO11] Karl Granström, Christian Lundquist, and Umut Orguner. Tracking Rectangular and Elliptical Extended Targets Using Laser Measurements. In *IEEE 14th International Conference on Information Fusion (Fusion)*, Chicago, Illinois, USA, Jul. 2011.

[GM20] Nikhil B. Gosala and Xiaoli Meng. An RLS-Based Instantaneous Velocity Estimator for Extended Radar Tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2273–2280, 2020.

[GO12] Karl Granström and Umut Orguner. On the Reduction of Gaussian Inverse Wishart Mixtures. In *IEEE 15th International Conference on Information Fusion (Fusion)*, pages 2162–2169, 2012.

[GO13] Karl Granström and Umut Orguner. On Spawning and Combination of Extended/Group Targets Modeled With Random Matrices. *IEEE Transactions on Signal Processing*, 61(3):678–692, 2013.

[Gov19] Felix Govaers. On Independent Axes Estimation for Extended Target Tracking. In *IEEE Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6, 2019.

[GS84] Clark R. Givens and Rae M. Shortt. A Class of Wasserstein Metrics for Probability Distributions. *The Michigan Mathematical Journal*, 31(2):231–240, 1984.

[GS05] Kevin Gilholm and David Salmond. Spatial Distribution Model for Tracking Extended Objects. *IEE Proceedings on Radar, Sonar and Navigation*, 152(5):364–371, October 2005.

[GSD$^+$17] Davide Guermandi, Qixian Shi, Andy Dewilde, Veerle Derudder, Ubaid Ahmad, Annachiara Spagnolo, Ilja Ocket, André Bourdoux, Piet Wambacq, Jan Craninckx, and Wim V. Thillo. A 79-GHz 2 x 2 MIMO PMCW Radar SoC in 28-nm CMOS. *IEEE Journal of Solid-State Circuits*, 52(10):2613–2626, 2017.

[GsFS16] Karl Granström, Maryam Fatemi, and Lennart Svensson. Gamma Gaussian Inverse-Wishart Poisson Multi-Bernoulli Filter for Extended Target Tracking. In *IEEE 19th International Conference on Information Fusion (Fusion)*, Heidelberg, Germany, July 2016.

[GSS93] Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. Novel Approach to Nonlinear/non-Gaussian Bayesian State Estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

[GSSW10] Marco Guerriero, Lennart Svensson, Daniel Svensson, and Peter Willett. Shooting Two Birds with Two Bullets: How to Find Minimum Mean OSPA Estimates. *IEEE 13th International Conference on Information Fusion (Fusion)*, 2010.

[HH08] Marco F. Huber and Uwe D. Hanebeck. Progressive Gaussian Mixture Reduction. In *IEEE 11th International Conference on Information Fusion*, pages 1–8, 2008.

[HK20] Jens Honer and Hauke Kaulbersch. Bayesian Extended Target Tracking with Automotive Radar using Learned Spatial Distribution Models. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 316–322. IEEE, 2020.

[HL19] Junhao Hua and Chunguang Li. Distributed Variational Bayesian Algorithms for Extended Object Tracking. *arXiv preprint arXiv:1903.00182*, 2019.

[HM02] John R. Hoffman and Ronald P. S. Mahler. Multitarget Miss Distance and its Applications. In *IEEE 5th International Conference on Information Fusion*, volume 1, pages 149–155, July 2002.

[HMSB19] Martin Herrmann, Johannes Müller, Jan Strohbeck, and Michael Buchholz. Environment Modeling Based on Generic Infrastructure Sensor Interfaces Using a Centralized Labeled-Multi-Bernoulli Filter. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2414–2420, 2019.

[HSRD16] Tobias Hirscher, Alexander Scheel, Stephan Reuter, and Klaus Dietmayer. Multiple Extended Object Tracking Using Gaussian Processes. In *IEEE 19th International Conference on Information Fusion (Fusion)*, pages 868–875, July 2016.

[HSSS12] Lars Hammarstrand, Lennart Svensson, Fredrik Sandblom, and Joakim Sorstedt. Extended Object Tracking using a Radar Resolution Model. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2371–2386, 2012.

[JPDM11] Rabin Julien, Gabriel Peyré, Julie Delon, and Bernot Marc. Wasserstein Barycenter and its Application to Texture Mixing. In *SSVM'11*, pages 435–446, Israel, 2011. hal-00476064.

[JU97] Simon J. Julier and Jeffrey K. Uhlmann. A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations. In *IEEE American Control Conference (Cat. No. 97CH36041)*, volume 4, pages 2369–2373, 1997.

[JU04] Simon J. Julier and Jeffrey K. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[Kal60] Rudolph E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[KB61] Rudolph E. Kalman and Richard S. Bucy. New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83(1):95–108, 03 1961.

[KBK$^+$16] Dominik Kellner, Michael Barjenbruch, Jens Klappstein, Jürgen Dickmann, and Klaus Dietmayer. Tracking of Extended Objects with High-Resolution Doppler Radar. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1341–1353, 2016.

[KBW17] Hauke Kaulbersch, Marcus Baum, and Peter Willett. EM Approach for Tracking Star-Convex Extended Objects. In *IEEE 20th International Conference on Information Fusion (Fusion)*, Xi'an, P.R. China, July 2017.

[KGH14] Gerhard Kurz, Igor Gilitschenski, and Uwe D. Hanebeck. Efficient Evaluation of the Probability Density Function of a Wrapped Normal Distribution. In *IEEE Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–5, 2014.

[KHB18] Hauke Kaulbersch, Jens Honer, and Marcus Baum. A Cartesian B-Spline Vehicle Model for Extended Object Tracking. In *IEEE 21st International Conference on Information Fusion (Fusion)*, Cambridge, United Kingdom, July 2018.

[KHB19] Hauke Kaulbersch, Jens Honer, and Marcus Baum. EM-based Extended Target Tracking with Automotive Radar Using Learned Spatial Distribution Models. In *IEEE 22nd International Conference on Information Fusion (Fusion)*, Ottawa, Canada, July 2019.

[KKÖ21] Murat Kumru, Hilal Köksal, and Emre Özkan. Variational Measurement Update for Extended Object Tracking Using Gaussian Processes. *IEEE Signal Processing Letters*, 28:538–542, 2021.

[Koc08] Wolfgang Koch. Bayesian Approach to Extended Object and Cluster Tracking using Random Matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.

[KSD16] Christina Knill, Alexander Scheel, and Klaus Dietmayer. A Direct Scattering Model for Tracking Vehicles with High-resolution Radars. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 298–303, 2016.

[LBS93] Don Lerro and Yaakov Bar-Shalom. Tracking with Debiased Consistent Converted Measurements Versus EKF. *IEEE Transactions on Aerospace and Electronic Systems*, 29(3):1015–1022, 1993.

[LG19] Jinqi Liu and Ge Guo. Distributed Asynchronous Extended Target Tracking Using Random Matrix. *IEEE Sensors Journal*, 20(2):947–956, 2019.

[LGB$^+$19] Samy Labsir, Audrey Giremus, Guillaume Bourmaud, Brice Yver, and Thomas Benoudiba-Campanini. Tracking a Cluster of Space Debris in Low Orbit by Filtering on Lie Groups. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5481–5485, 2019.

[LGYBC21] Samy Labsir, Audrey Giremus, Brice Yver, and Thomas Benoudiba-Campanini. A Lie-group Based Modelling for Centroid and Shape Estimation of a Cluster of Space Debris. In *IEEE 28th European Signal Processing Conference (EUSIPCO)*, pages 960–964, 2021.

[LHGD18] Isabelle Leang, Stéphane Herbin, Benoît Girard, and Jacques Droulez. On-line Fusion of Trackers for Single-object Tracking. *Pattern Recognition*, 74:459–473, 2018.

[LJ03] X. Rong Li and Vesselin P. Jilkov. Survey of Maneuvering Target Tracking. Part I. Dynamic Models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, 2003.

[LJMD15] Wenling Li, Yingmin Jia, Deyuan Meng, and Junping Du. Distributed Tracking of Extended Targets Using Random Matrices. In *IEEE 54th Conference on Decision and Control (CDC)*, pages 3044–3049, 2015.

[LL14] Jian Lan and X. Rong Li. Tracking of Maneuvering Non-Ellipsoidal Extended Object or Target Group Using Random Matrix. *IEEE Transactions on Signal Processing*, 62(9):2450–2463, 2014.

[LL16] Jian Lan and X. Rong Li. Tracking of Extended Object or Target Group Using Random Matrix: New Model and Approach. *IEEE Transactions on Aerospace and Electrical Systems*, 52(6):2973–2988, Dec. 2016.

[LLL20] Mingkai Li, Jian Lan, and X. Rong Li. Tracking of Elliptical Extended Object with Unknown but Fixed Lengths of Axes. In *IEEE 23rd International Conference on Information Fusion (Fusion)*, pages 1–8, 2020.

[MBL$^+$19] Martin Michaelis, Philipp Berthold, Thorsten Luettel, Daniel Meissner, and Hans-Joachim Wünsche. A Merging Strategy for Gaussian Process Extended Target Estimates in Multi-Sensor Applications. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1803–1808, 2019.

[MBL$^+$20] Martin Michaelis, Philipp Berthold, Thorsten Luettel, Daniel Meissner, and Hans-Joachim Wünsche. Extended Object Tracking with an Improved Measurement-to-Contour Association. In *IEEE 23rd International Conference on Information Fusion (Fusion)*, pages 1–6, 2020.

[MBMW17] Martin Michaelis, Philipp Berthold, Daniel Meissner, and Hans-Joachim Wünsche. Heterogeneous Multi-Sensor Fusion for Extended Objects in Automotive Scenarios Using Gaussian Processes and a GMPHD-Filter. In *IEEE Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–6, 2017.

[MĆP16] Ivan Marković, Josip Ćesić, and Ivan Petrović. On Wrapping the Kalman Filter and Estimating with the SO (2) Group. In *19th International Conference on Information Fusion (Fusion)*, pages 2245–2250, 2016.

[MCS$^+$14] Lyudmila Mihaylova, Avishy Y. Carmi, François Septier, Amadou Gning, Sze K. Pang, and Simon Godsill. Overview of Bayesian Sequential Monte Carlo Methods for Group and Extended Object Tracking. *Digital Signal Processing*, 25:1–16, February 2014.

[MJ09] Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*, volume 494. John Wiley & Sons, 2009.

[MK19] Michael Meyer and Georg Kuschk. Automotive Radar Dataset for Deep Learning Based 3D Object Detection. In *IEEE 16th European Radar Conference (EuRAD)*, pages 129–132, 2019.

[MPLN17] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[Mut98] Arthur G. O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, Inc., 1998.

[MWRH20] Mohammadreza Mostajabi, Ching Ming Wang, Darsh Ranjan, and Gilbert Hsyu. High-Resolution Radar Dataset for Semi-Supervised Learning of Dynamic Objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 100–101, 2020.

[NBW19] Benjamin Naujoks, Patrick Burger, and Hans-Joachim Wünsche. Fast 3D Extended Target Tracking using NURBS Surfaces. In *IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1104–1109, 2019.

[NK16] Sofie Nilsson and Axel Klekamp. Object Level Fusion of Extended Dynamic Objects. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 251–258, Sept 2016.

[NSH17] Benjamin Noack, Joris Sijs, and Uwe D. Hanebeck. Inverse Covariance Intersection: New Insights and Properties. In *IEEE 20th International Conference on Information Fusion (Fusion)*, pages 1–8, 2017.

[NSRH17] Benjamin Noack, Joris Sijs, Marc Reinhardt, and Uwe D. Hanebeck. Decentralized Data Fusion with Inverse Covariance Intersection. *Automatica*, 79:35–41, 2017.

[ONR$^+$21] Arthur Ouaknine, Alasdair Newson, Julien Rebut, Florence Tupin, and Patrick Pérez. CARRADA Dataset: Camera and Automotive Radar with Range-Angle-Doppler Annotations. In *IEEE 25th International Conference on Pattern Recognition (ICPR)*, pages 5068–5075, 2021.

[PMGA11] Nikolay Petrov, Lyudmila Mihaylova, Amadou Gning, and Donka Angelova. A Novel Sequential Monte Carlo Approach for Extended Object Tracking Based on Border Parameterisation. In *IEEE 14th International Conference on Information Fusion (Fusion)*, pages 1–8, 2011.

[PRV20] Giovanni Puccetti, Ludger Rüschendorf, and Steven Vanduffel. On the Computation of Wasserstein Barycenters. *Journal of Multivariate Analysis*, 176:104581, 2020.

[RHG14] Michael Roth, Gustaf Hendeby, and Fredrik Gustafsson. EKF/UKF Maneuvering Target Tracking using Coordinated Turn Models with Polar/Cartesian Velocity. In *IEEE 17th International Conference on Information Fusion (Fusion)*, pages 1–8, 2014.

[RM17] Yevgeniy Reznichenko and Henry Medeiros. Improving Target Tracking Robustness with Bayesian Data Fusion. In *BMVC*, 2017.

[RNAH15] Marc Reinhardt, Benjamin Noack, Pablo O. Arambel, and Uwe D. Hanebeck. Minimum Covariance Bounds for the Fusion under Unknown Correlations. *IEEE Signal Processing Letters*, 22(9):1210–1214, 2015.

[RNH19] Susanne Radtke, Benjamin Noack, and Uwe D. Hanebeck. Distributed Estimation using Square Root Decompositions of Dependent Information. In *IEEE 22nd International Conference on Information Fusion (Fusion)*, pages 1–8, 2019.

[RNH20] Susanne Radtke, Benjamin Noack, and Uwe D. Hanebeck. Fully Decentralized Estimation using Square-root Decompositions. In *IEEE 23rd International Conference on Information Fusion (Fusion)*, pages 1–8, 2020.

[RX20] Yuanyuan Ren and Wei Xia. Distributed Extended Object Tracking Based on Diffusion Strategy. In *IEEE 28th European Signal Processing Conference (EUSIPCO)*, pages 2338–2342, 2020.

[Sal09] David J. Salmond. Mixture Reduction Algorithms for Point and Extended Object Tracking in Clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 45(2):667–686, 2009.

[SD18] Alexander Scheel and Klaus Dietmayer. Tracking Multiple Vehicles Using a Variational Radar Model. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3721–3736, 2018.

[SH09] Dennis Schieferdecker and Marco F. Huber. Gaussian Mixture Reduction via Clustering. In *IEEE 12th International Conference on Information Fusion (Fusion)*, pages 1536–1543, 2009.

[SKRD16] Alexander Scheel, Christina Knill, Stephan Reuter, and Klaus Dietmayer. Multi-Sensor Multi-Object Tracking of Vehicles Using High-Resolution Radars. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 558–565, 2016.

[SL95] Roy L. Streit and Tod E. Luginbuhl. Probabilistic Multi-hypothesis Tracking. Technical report, Naval Underwater Systems Center Newport RI, 1995.

[SRW15] Michael Schuster, Johannes Reuter, and Gerd Wanielik. Probabilistic Data Association for Tracking Extended Targets Under Clutter Using Random Matrices. In *IEEE 18th International Conference on Information Fusion (Fusion)*, pages 961–968, 2015.

[SVV08] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. A Consistent Metric for Performance Evaluation of Multi-Object Filters. *IEEE Transactions on Signal Processing*, 56(8):3447 –3457, August 2008.

[SWBH12] Kai Schueler, Tobias Weiherer, Essayed Bouzouraa, and Ulrich Hofmann. 360 Degree Multi Sensor Fusion for Static and Dynamic Obstacles. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 692–697, 2012.

[SWG21] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted Boxes Fusion: Ensembling Boxes from Different Object Detection Models. *Image and Vision Computing*, 107:104117, 2021.

[TLTK19] Xu Tang, Mingyan Li, Ratnasingham Tharmarasa, and Thiagalingam Kirubarajan. Seamless Tracking of Apparent Point and Extended Targets Using Gaussian Process PMHT. *IEEE Transactions on Signal Processing*, 67(18):4825–4838, 2019.

[TÖ21] Barkin Tuncer and Emre Özkan. Random Matrix Based Extended Target Tracking with Orientation: A New Model and Inference. *IEEE Transactions on Signal Processing*, 69:1910–1923, 2021.

[TÖO21] Barkin Tuncer, Emre Özkan, and Umut Orguner. Multi-Ellipsoidal Extended Target Tracking with Variational Bayes Inference. *TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.14178494.v1*, 2021.

[VBGs$^+$15] Gemine Vivone, Paolo Braca, Karl Granström, Antonio Natale, and Jocelyn Chanussot. Converted Measurements Random Matrix Approach to Extended Target Tracking Using X-band Marine Radar Data. In *IEEE 18th International Conference on Information Fusion (Fusion)*, pages 976–983, Washington, DC, USA, July 2015.

[VGBW17] Gemine Vivone, Karl Granström, Paolo Braca, and Peter Willett. Multiple Sensor Measurement Updates for the Extended Target Tracking Random Matrix Model. *IEEE Transactions on Aerospace and Electronic Systems*, 53(5):2544–2558, 2017.

[VH00] Remco C. Veltkamp and Michiel Hagedoorn. Shape Similarity Measures, Properties and Constructions. In *International Conference on Advances in Visual Information Systems*, pages 467–476. Springer, 2000.

[VMBS$^+$15] Ba-Ngu Vo, Mahendra Mallick, Yaakov Bar-Shalom, Stefano Coraluppi, Richard Osborne, Ronald Mahler, and Ba-Tuong Vo. Multitarget Tracking. *Wiley encyclopedia of electrical and electronics engineering*, 2015.

[WD14] Monika Wieneke and Sam Davey. Histogram-PMHT for Extended Targets and Target Groups in Images. *IEEE Transactions on Aerospace and Electronic Systems*, 50(3):2199–2217, 2014.

[WÖ15] Niklas Wahlström and Emre Özkan. Extended Target Tracking Using Gaussian Processes. *IEEE Transactions on Signal Processing*, 63(16):4165–4178, 2015.

[WVDM00] Eric A. Wan and Rudolph Van Der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158, 2000.

[XFPW19] Yang Xu, Yangwang Fang, Weishi Peng, and Youli Wu. An Efficient Gaussian Sum Filter Based on Prune-Cluster-Merge Scheme. *IEEE Access*, 7:150992–151005, 2019.

[XN21] Liang Xu and Ruixin Niu. Tracking Visual Object as an Extended Target. In *IEEE International Conference on Image Processing (ICIP)*, pages 664–668, 2021.

[XWB$^+$20a] Yuxuan Xia, Pu Wang, Karl Berntorp, Toshiaki Koike-Akino, Hassan Mansour, Milutin Pajovic, Petros Boufounos, and Philip V. Orlik. Extended Object Tracking Using Hierarchical Truncation Measurement Model with Automotive Radar. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4900–4904, 2020.

[XWB$^+$20b] Yuxuan Xia, Pu Wang, Karl Berntorp, Hassan Mansour, Petros Boufounos, and Philip V. Orlik. Extended Object Tracking Using Hierarchical Truncation Model with Partial-View Measurements. In *IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5, 2020.

[XWB$^+$21] Yuxuan Xia, Pu Wang, Karl Berntorp, Lennart Svensson, Karl Granström, Hassan Mansour, Petros Boufounos, and Philip V. Orlik. Learning-Based Extended Object Tracking Using Hierarchical Truncation Measurement Model With Automotive Radar. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):1013–1029, 2021.

[YB16] Shishan Yang and Marcus Baum. Second-Order Extended Kalman Filter for Extended Object and Group Tracking. In *IEEE 19th International Conference on Information Fusion (Fusion)*, Heidelberg, Germany, July 2016.

[YB17] Shishan Yang and Marcus Baum. Extended Kalman Filter for Extended Object Tracking. In *IEEE 42nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, New Orleans, USA, March 2017.

[YB19] Shishan Yang and Marcus Baum. Tracking the Orientation and Axes Lengths of an Elliptical Extended Object. *IEEE Transactions on Signal Processing*, 67(18):4720–4729, September 2019.

[YBG16] Shishan Yang, Marcus Baum, and Karl Granström. Metrics for Performance Evaluation of Elliptic Extended Object Tracking Methods. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Baden-Baden, Germany, September 2016.

[YSD20] Gang Yao, Ryan Saltus, and Ashwin Dani. Image Moment-Based Extended Object Tracking for Complex Motions. *IEEE Sensors Journal*, 20(12):6560–6572, 2020.

[YSKR20] Zhi Yan, Li Sun, Tomas Krajnik, and Yassine Ruichek. EU Long-term Dataset with Multiple Sensors for Autonomous Driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[YWB$^+$21] Gang Yao, Perry Wang, Karl Berntorp, Hassan Mansour, Petros Boufounos, and Philip V. Orlik. Extended Object Tracking With Automotive Radar Using B-Spline Chained Ellipses Model. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8408–8412, 2021.

[ZFBH16] Antonio Zea, Florian Faion, Marcus Baum, and Uwe D. Hanebeck. Level-set Random Hypersurface Models for Tracking Nonconvex Extended Objects. *IEEE Transactions on Aerospace and Electronic Systems*, 52(6):2990–3007, 2016.

[Zha97] Zhengyou Zhang. Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting. *Image and Vision Computing*, 15(1):59–76, 1997.

[ZHL12] Hongyan Zhu, Chongzhao Han, and Yan Lin. A Reduced Gaussian Mixture Representation Based on Sparse Modeling. In *IEEE 15th International Conference on Information Fusion (Fusion)*, pages 684–691, 2012.

[ZJ14] Yongquan Zhang and Hongbing Ji. Clustering Gaussian Mixture Reduction Algorithm based on Fuzzy Adaptive Resonance Theory for Extended Target Tracking. *IET Radar, Sonar & Navigation*, 8(5):536–546, 2014.

[ZL20] Le Zhang and Jian Lan. Extended Object Tracking Using Random Matrix With Skewness. *IEEE Transactions on Signal Processing*, 68:5107–5121, 2020.

[ZL21] Le Zhang and Jian Lan. Tracking of Extended Object Using Random Matrix with Non-Uniformly Distributed Measurements. *IEEE Transactions on Signal Processing*, 2021.

[ZXWA13] Hui Zhang, Hui Xu, Xue-Ying Wang, and Wei An. A PHD Filter for Tracking Closely Spaced Objects with Elliptic Random Hypersurface Models. In *IEEE 16th International Conference on Information Fusion (Fusion)*, pages 1558–1565, July 2013.