Extended Object Tracking of Pedestrians in Automotive Applications

Georgios Katsaounis

Master of Science Thesis

TU Delft Delft University of Technology

Department of Cognitive Robotics (CoR)

Extended Object Tracking of Pedestrians in Automotive Applications

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Georgios Katsaounis

June 10, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Cognitive Robotics (CoR)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Extended Object Tracking of Pedestrians in Automotive Applications

by

Georgios Katsaounis

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: June 10, 2019

Supervisor(s):

Dr. J. Alonso-Mora

Ir. J.F.M. Domhof

Ir. A. Tasoglou

Reader(s):

Prof.dr. D. Gavrila

Abstract

Recent advances in sensor technology have lead to increased resolution of novel sensors, while tracking applications where distance between sensors and objects of interest is very small have gained research interest recently. In these cases, it is possible that multiple sensor detections are generated by each object of interest. Extended Object Tracking (EOT) approaches consist of algorithms which make use of multiple sensor detections per object to jointly estimate their kinematic and shape extent attributes within the Bayesian tracking framework. In the last decade, various EOT algorithms have been proposed for different types of tracking applications.

This M.Sc. thesis project addresses the problem of extended tracking of a single pedestrian walking in the area of a stationary vehicle (referred as ego-vehicle in this report) during a real automotive scenario. The objective is to achieve accurate estimation of both the kinematic attributes (2D centroid position/velocity), as well as its shape extent in x-y plane. In more detail, PreScan software is enabled to design a simulation scenario that is very close to a real automotive application, in terms of motion characteristics of objects of interest and sensor data acquisition. In the considered scenario, different sensor modalities are mounted on the ego-vehicle, namely a Lidar sensor and a mono camera sensor. Moreover, OpenPose library is employed to to obtain pose detections of human body parts from obtained camera images.

Concerning shape extent representation, the simplest and most popular approach in previous studies, in general and especially for VRUs tracking, is to assume an elliptical shape. In fact, the Random Matrix Model (RMM), proposed originally bu Koch [8], is a state-of-the-art EOT state modeling approach that allows for joint estimation of centroid kinematics and physical extent for considered elliptical objects of interest. Based on that, a RMM-based filter using Lidar position measurements has been proposed by Feldmann in [7]. In this project, this algorithm is used as a baseline filter for comparison with our proposed algorithm.

In addition, an alternative tracking algorithm is proposed in this study, which has the following differences with respect to the baseline filter:

- State Initialization of the filter: In our proposed version of the tracking algorithm, human pose detections of shoulders and ankles are are associated with obtained Lidar position measurements in order to provide initial values for the kinematic state (2D position/velocity) and shape parameters (ellipse orientation and semi-axes lengths) of the pedestrian.
- Measurement Update step of the filter: In our proposed version of the tracking algorithm, camera-obtained pose detections of pedestrian shoulders are associated with obtained Lidar position measurements in order to create an extra measurement, for pedestrian heading angle. Subsequently, a nonlinear filtering update step fusing Lidar-obtained point cloud data for pedestrian position and human-pose-obtained measurement for pedestrian heading angle is implemented.

Both considered tracking algorithms are evaluated for the designed simulation scenario. In detail, the following performance metrics are used for evaluation of each filter:

- RMSE for estimated pedestrian 2D position and velocity, respectively.
- Modified Hausdorff distance for estimated pedestrian shape extent.

In more detail, Monte Carlo simulations with multiple runs are designed to evaluate performance of each state initialization approach and each tracking algorithm, where the following parameters change in each run:

- Additive zero-mean Gaussian measurement noise on obtained Lidar position detections.
- Initial simulation timestep.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The completion of this M.Sc. Thesis project denotes the end of a long, but productive period as a Master student in TU Delft. During this time, I had the chance to face new challenges in my field of interest, work together with fellow students from around the globe and gain experiences that I will never forget.

Fortunately, I had the pleasure to feel the support of really special people during this long period.

Firstly, I want to express my sincere gratitude to my daily supervisors during this M.Sc. thesis project, Joris Domhof and Athanasios Tasoglou. Without their constant support and the numerous hours that they devoted to my supervision, it would not be possible to design and complete this project. In many occasions, they had to take more responsibilities with respect to their academic position in order to help me and I will always be grateful for that.

In addition, I want to thank Professor Alonso Mora, who agreed to be my distant supervisor for the last two years.

However, it would not be possible to achieve anything without the ethical and practical support of my family. Special thanks to my parents, Maria and Ilias, for all personal sacrifices they have made over all those years to provide the best support, advice and opportunities to my sister and me. Thanks also to my sister Sofia and her boyfriend Moritz for their support during this stressful time.

A special thanks also to my girlfriend Eftychia. She knows that I could not have made it till the end of the road without her unconditional love and support during this stressful time. I promise that I will make it up to her in the very near future.

I also want to thank my fellow students in M.Sc. Systems & Control, Giannis, Gurol, Zhang, Shreyans, Shekhar, Shwetha and Dimitris. We spent numerous hours on campus working together on projects and supporting each other on difficult times.

Many thanks to my partners in crime, Panos, Apostolis, Lampros, Giannis, Matias, Giannis and Giorgos. Our long discussions in TU Delft library coffee was the best break from studying that anyone could imagine! It's about time to arrange another roadtrip, isn't it?

Last but not least, a big thanks to my best friends in Greece, Kostas, Gogo, Kostas, Dimos, Iris and Chrysa! They have always been there for me, even if they are miles away.

Delft, University of Technology Georgios Katsaounis
June 10, 2019

“We need to know where we are heading to. If the driver is not aware of the surrounding environment and the destination, it is not possible to succeed.”

Chapter 1

Introduction

1-1 Motivation

1-1-1 Why Extended Object Tracking is preferred than Conventional Object Tracking?

Conventional tracking algorithms, developed in the last 60 years, are tailored to applications were multiple objects of interest are far away from sensors, for example air traffic surveillance. Since each object may generate at most a single detection, these algorithms treat the objects of interest as point objects with no spatial extent. Thus, the objective is to estimate the number and kinematic attributes (e.g.: position, velocity, orientation, etc) of multiple point objects moving within the field of view FoV of available sensors.

Nevertheless, the recent advances in sensor technology have lead to increased resolution of novel sensors used for environment perception. In addition, research interest has been shifted to applications where the distance between sensors and detected objects is very small (e.g.: only a few meters), such as surveillance tracking of humans or tracking of vehicles in automotive applications. In these cases, it is possible that multiple sensor detections are generated by each object of interest. As a result, spatial extent of objects can also be taken into account. Extended Object Tracking (EOT) approaches consist of algorithms which make use of multiple sensor detections per object to jointly estimate their kinematic and shape extent attributes within the Bayesian tracking framework. In the last decade, various EOT algorithms have been proposed for different types of tracking applications.

1-1-2 Why Extended Object Tracking in automotive applications?

A typical example of environmental perception relevant to automotive applications is detection and tracking of moving objects around a vehicle with sensors mounted on

it. Other vehicles and vulnurable road users (VRUs), such as pedestrians or cyclists are typical examples of potential moving objects, while radar, Lidar or camera sensors can be used for object detection. In case of automated driving, detections or tracks of surrounding objects are required for tasks such as path-planning and collision avoidance.

The majority of object tracking problems linked to automotive applications enable point object representation of surrounding objects, meaning that kinematic attributes (such as position, velocity and/or orientation) are only estimated and shape extent is neglected. However, point representation of surrounding objects may not be enough in some automotive scenaria. In detail, false conclusions might be derived about the way that surrounding objects affect the planned motion of the ego-vehicle in case that their shape extent is not accounted for. For example, other vehicles or VRUs might be positioned in the path of the ego-vehicle, but this may not be concluded by the tracking algorithm in case that only its point state estimates are considered. As a result, estimation of shape and size for road users is crucial in order to alleviate limitations of traditional point object approaches, like the aforementioned.

In addition, novel sensors with increased resolution are developed and embedded on modern cars. This feature, combined with the small distance between sensors of the ego-vehicle and surrounding objects result in generation of multiple measurements per object per sensor scan. EOT algorithms make possible to use all available sensed data in order to jointly estimate the shape and size of surrounding objects.

1-1-3 Why Extended Object Tracking for pedestrians?

The vast majority of EOT approaches for automotive applications in recent studies involve tracking of kinematics and extent of one or multiple vehicles moving in the area of a considered ego-vehicle, while using Lidar position measurements to detect them. This problem is quite straightforward: To begin with, vehicle centroid position can be clearly defined (usually the center of rear-axis is selected). In addition, vehicles can be considered as rectangularly shaped objects with size that does not change over time. Thus, the objective is to estimate corresponding width and length, which are considered to be unknown but constant over time. As a result, measurement models involving just the two sides of the vehicle are appropriate to achieve accurate shape extent estimation. For all these reasons, various relevant examples can be found on previous studies, proposing measurement models and filter prediction/update steps for vehicles in automotive applications.

Nevertheless, this is not the case for VRUs, especially for pedestrians. Firstly, assuming a constant shape extent for pedestrians is not an appropriate choice. The reason is that their shape extent is expected to change frequently over time with respect to their motion, for instance while walking. As a result, differences in estimated pedestrian spatial extent are expected in each time instance of a real scenario. On top of that, due to sensor-to-object geometry, the possibility that some human body parts are not detected by the sensor is quite high. Meaning that popular measurement modeling approaches might not be efficient for practical cases of pedestrian tracking.

Secondly, it is found that recent studies only focus on scenaria with closely spaced or

occluded pedestrian. In more detail, the aim is to distinguish among multiple pedestrians, by applying corresponding sensor data association techniques, instead of trying to achieve (as much as possible) accurate shape extent tracking. In fact, to the best of the author's knowledge, no ground truth data for shape extent is available in recent studies or in benchmarks found online (KITTI, MOT, etc..). Thus, it is not possible to use corresponding evaluation criteria for spatial extent tracking performance.

To sum up, all aforementioned reasons have motivated us to investigate the way that state-of-the-art EOT algorithms, which are not tailored made for pedestrian tracking in automotive applications, perform in such a scenario. Also, an interesting aspect is to inspect whether an extra sensor modality (mono camera) can provide such information, which can alleviate some of the limitations of these algorithms in real conditions.

1-2 Problem Description

This M.Sc. thesis project addresses the problem of extended tracking of a single pedestrian walking in the area of a stationary vehicle (referred as ego-vehicle in this report) during a real automotive scenario. The objective is to achieve accurate estimation of both the kinematic attributes (2D centroid position/velocity), as well as its shape extent in x-y plane. The main issues of interest include the selection of an appropriate shape extent representation for the pedestrian, the implementation of an appropriate tracking algorithm involving fusion of sensor data, as well as the design of an automotive scenario that allows for shape extent tracking performance evaluation.

Concerning shape extent representation, the simplest and most popular approach in previous studies, in general and especially for VRUs tracking, is to assume an elliptical shape. In fact, the Random Matrix Model (RMM), proposed originally bu Koch [8], is a state-of-the-art EOT state modeling approach that allows for joint estimation of centroid kinematics and physical extent for considered elliptical objects of interest. Based on that, a RMM-based filter using Lidar position measurements has been proposed by Feldmann in [7]. In this project, this algorithm is used as a baseline filter for comparison with our proposed algorithm.

A major limitation of the RMM representation is the considered assumption of an almost uniform distribution of Lidar-obtained position measurements over the extent of objects of interest. This assumption might be valid for scenaria created in a simulation software like MATLAB, however this is not the case for practical examples,due to missing Lidar detections from some parts of human body in each simulation timestep. Moreover, it is shown that the initial state selected for the baseline filter is such, so that slow convergence of estimated centorid position and velocity to corresponding ground truth values might take place.

In this project, a monocular camera is added to the ego-vehicle to provide extra information about the pedestrian. In more detail, OpenPose library is used to obtain pose detections of human body parts from obtained camera images, which are then associated with obtained Lidar position measurements of the pedestrian. Since depth information is not available in mono camera images, an association method is proposed to map

pedestrian pose detections from 2D pixel coordinates to 2D world coordinates, where pedestrian tracking takes place. More specifically, the proposed tracking algorithm has the following differences with respect to the baseline filter:

- State Initialization of the filter: In our proposed version of the tracking algorithm, human pose detections of shoulders and ankles are are associated with obtained Lidar position measurements in order to provide initial values for the kinematic state (2D position/velocity) and shape parameters (ellipse orientation and semi-axes lengths) of the pedestrian.
- Measurement Update step of the filter: In our proposed version of the tracking algorithm, camera-obtained pose detections of pedestrian shoulders are associated with obtained Lidar position measurements in order to create an extra measurement, for pedestrian heading angle. Subsequently, a nonlinear filtering update step fusing Lidar-obtained point cloud data for pedestrian position and human-pose-obtained measurement for pedestrian heading angle is implemented.

In the analysis included in this report, it is shown that the accuracy of the calculated initial state, as well as the heading angle measurement for the pedestrian depends highly on the proposed association method between obtained pose detections and Lidar position measurements.

Last but not least, it is found that performance evaluation of estimated pedestrian shape extent does not take place in examples presented in recent studies. This trend is due to two main reasons, namely the unavailability of ground truth data for pedestrian shape in these examples and the absence of an appropriate performance metric for shape extent evaluation, respectively. In more detail, simulation scenarios examined in recent studies, as well as popular benchmarks found online (e.g.: KITTI, MOT) do not contain ground truth data for VRUs shape. As a result, comparing the estimated shape extent with a considered ground truth one is not possible. In fact, in most studies researchers perform a visual comparison of estimated shape extent with obtained Lidar position data to indirectly derive tracking accuracy of each algorithm.

Concerning the design of an automotive scenario for performance evaluation of the baseline and proposed EOT algorithms, it is clear that ground truth data for pedestrian shape extent in x-y plane must be available, in order to validate pedestrian shape tracking in an efficient way. As a result, PreScan software was employed to create an automotive simulation scenario.

The advantages of this decision are very significant. At first, PreScan software is tailored for creating simulations very similar to actual automotive applications,in terms of obtained sensor data and motion of objects of interest. An extra benefit is the availability of true values for position and velocity of all moving objects for all simulation timesteps, which can be used for filter performance evaluation. In addition, combining pedestrian Lidar point cloud data from multiple extra sensors, which are not mounted on the ego-vehicle, can lead to the definition of a ground truth shape for pedestrian extent in x-y plane. In terms of PreScan scenario examined in this project, the boundary of all Lidar-obtained points is considered to define it. Thus, this ground truth shape is not represented by a basic geometrical shape, but is treated as an arbitrary shape. Fortunately, the modified Hausdorff distance metric has been proposed for comparison

of two or more arbitrary shapes. In this project, this performance metric is used to compare the considered ground truth arbitrary shape and the estimated elliptical shape extent in each simulation timestep.

1-3 Research Questions

Based on the aforementioned problem description, the main research question addressed in this M.Sc. thesis is the following:

- What are the advantages and disadvantages of using multiple sensor modalities for extended object tracking of pedestrians?

Towards this direction, the following research questions should be investigated:

- Which steps should be followed for performance evaluation of pedestrian shape extent tracking in automotive simulation scenarios?
- What are the advantages and disadvantages of the proposed association method between Lidar position measurements, lying in world coordinates frame, and human pose detections, lying in image pixel coordinates frame?
- What is the effect of associating Lidar-obtained position measurements and camera-obtained body pose detections of the pedestrian on state initialization of an EOT filter?
- What is the effect of fusing Lidar-obtained position measurements and pose-obtained heading angle measurement of the pedestrian in the measurement update step of an EOT filter?
- What are the limitations of the RMM state representation for extended pedestrian tracking in automotive simulation scenarios?

1-4 Thesis Outline

In Chapter 2 of this M.Sc. thesis report, important definitions concerning the issue of extended object tracking are given. To begin with, a distinction is made between point, extended and group object, followed by a brief description of Bayesian tracking framework and an overview of widely used measurement and state modeling approaches. Two state-of-the-art state modeling approaches for single EOT are explained shortly: the Random Matrix Model (RMM) and the Random Hypersurface Model (RHM), tailored for elliptically and arbitrarily shaped objects, respectively. In addition, related work enabling EOT for VRUs, and especially pedestrians, are briefly presented.

Chapter 3 is devoted to the baseline tracking algorithm for a single object of interest, which is based on RMM and is proposed by Feldmann in [7]. In detail, chosen measurement and state modeling, together with corresponding assumptions for the object of interest are described in detail. On top of that, the prediction and measurement update steps of the baseline filter are explained step by step.

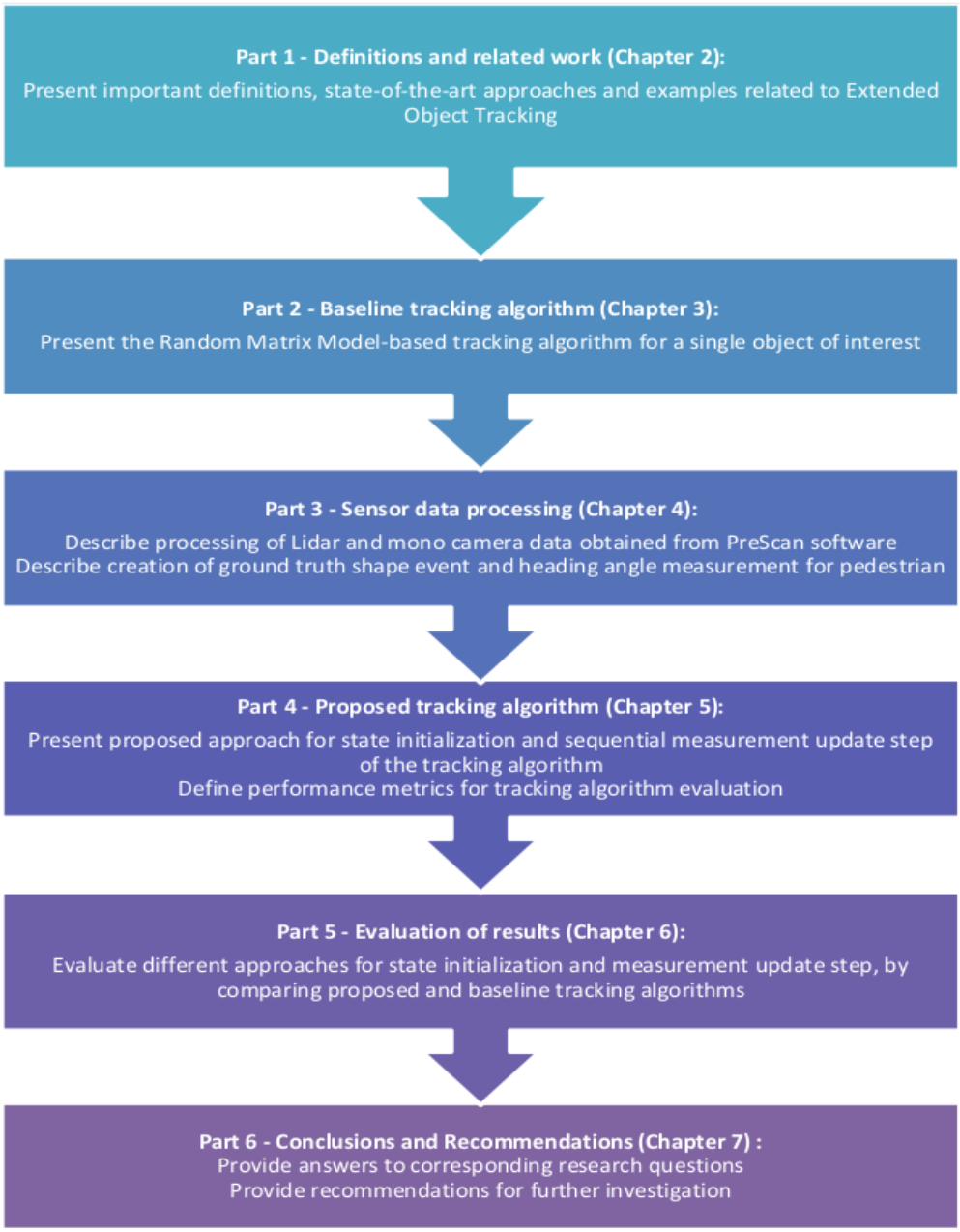


Figure 1-1: Thesis outline.

As already mentioned, PreScan software is used to create a simulation close to real automotive scenarios. Chapter 4 is devoted to processing of PreScan sensor data to be used for filter performance validation. Firstly, the design of considered simulation scenario in PreScan software is discussed. Also, creation of ground truth data for pedestrian shape extent in x-y plane is explained. Concerning the two different sensor modalities involved in created simulations, Lidar-obtained position measurements are defined in the x-y world coordinates frame, whereas camera-obtained human pose detections belong in

image plane coordinates. The steps applied to associate and fuse sensor data of different types is discussed in this Chapter as well. Associated sensor data is then used in state initialization and measurement update step of the proposed filter.

In Chapter 5, the proposed nonlinear tracking algorithm is discussed. Firstly, the approaches for state initialization definition and creation of pedestrian heading angle measurement are explained. Subsequently, the prediction and measurement update step of the proposed filter are presented, together with corresponding assumptions for pedestrian motion. Finally, the RMSE and mean Hausdorff distance, which are used as performance metrics for evaluation of estimated kinematic and shape extent state, are presented.

Finally, a comparison of the baseline and proposed tracking algorithms takes place in Chapter 6, where simulation results are presented. These results lead to specific conclusions which are presented in Chapter 7, leading to answers for corresponding research questions. Last but not least, recommendations for future work in this research topic are also presented in Chapter 7.

1-5 Contribution

In short, the contribution of this M.Sc. thesis project is the following:

- In this study, an attempt is made to create an arbitrary ground truth shape for the pedestrian, by combining obtained position detections from multiple Lidar sensors, employed in a simulation scenario designed in PreScan software. As a result, the considered ground truth pedestrian shape extent can be compared to corresponding estimated shape extents in each simulation timestep. The modified Hausdorff distance is a performance metric tailored for comparison of two arbitrary shapes and thus is used for evaluation of pedestrian shape extent tracking in this study.
- A state initialization approach is proposed, which associates obtained Lidar position measurements and body parts pose detections of the pedestrian to derive initial state values. Incorporation of this proposed state initialization approach to the baseline filter, results in decreased velocity RMSE of **33.3**% for the (randomly selected) first simulation timestep, in comparison to incorporation of the conventional state initialization approach. Moreover, incorporation of the proposed state initialization approach to the proposed tracking algorithm results in a smaller decrease in velocity RMSE of **6**% for the (randomly selected) first simulation timestep.
- The proposed tracking algorithm, which makes use of created pedestrian heading angle measurement, performs worse than the baseline filter, in terms of position and velocity RMSE. This behavior is not expected, but is justified by the inaccuracies in association method between Lidar detections and pose detections for the pedestrian, which results in an inaccurate calculated pedestrian heading angle.

Chapter 2

Definitions and Related Work

2-1 Definitions

2-1-1 Object Tracking Definitions

Object Tracking (OT) is defined as the processing of sensor measurements to determine the number and states of objects of interest [4]. Sensors can be any measuring devices that collect measurements from the environment, for example radar, laser scanner or camera. The objective of object tracking is to estimate the number and state of objects, such as position, velocity and shape information.

In its most general form, object tracking is a realistic version of dynamic estimation theory. In practice, the typical object tracking problem is a state estimation problem, where the object states are estimated from noisy and false sensor detections [9]. Moreover, in most object tracking problems, sensor measurements are obtained sequentially with respect to time. At each measurement cycle, new measurements are combined with current state estimates to form new state estimates. Subsequently, the latest estimates are updated by use of upcoming measurements in the next measurement cycle and the tracking process evolves in the same manner.

Object tracking examples are characterized by a wide variety of applications, such as aircrafts tracking in air traffic systems, moving objects tracking in automotive and robotics applications, vessels tracking in maritime applications or human tracking in ground surveillance systems [10],[11]. In each scenario, various aspects related to object tracking can differ. For instance, depending on the number of sensors used to acquire detections, object tracking problems are clustered in single and multiple sensor cases. Moreover, depending on the number of tracked objects, object tracking problems are clustered to single and multiple object tracking cases. Depending on sensor resolution and its relation to object size, a sensor might generate more than a single detection per object in each scan. Objects can be modeled as point objects, extended objects and group objects, as described in the next section.

2-1-2 Point Object, Extended Object and Group Object Tracking

The number of detections generated by an object in tracking applications depends mainly on two factors [4],[1]:

- Sensor spatial resolution (with respect to the detected object size)
- Sensor-to-object geometry

Concerning the former, a schematic representation of the comparison between low and high resolution sensors in automotive tracking applications is shown in Figure 2-1. In case of low resolution sensors setups, at most a single resolution cell may be occupied by any detected object within the surveillance area, resulting in at most a single detection per object per sensor scan (Figure 2-1a). On the contrary, multiple resolution cells may be occupied by each object when high resolution sensors are used, resulting in multiple detections per object per sensor scan (Figure 2-1b). Moreover, sensor-to-object geometry may refer to either the distance between them or the orientation of the detected object with respect to the sensor's field of view. Based on these characteristics, different approaches can be followed for object tracking, starting from choosing an appropriate object representation between point, extended and group objects, as shown in Figure 2-2.

(a) Low sensor resolution

(b) High sensor resolution

Figure 2-1: Schematic example of low and high sensor resolution with respect to object size in automotive tracking applications. Borrowed from Granström et.al. (2014) [1].

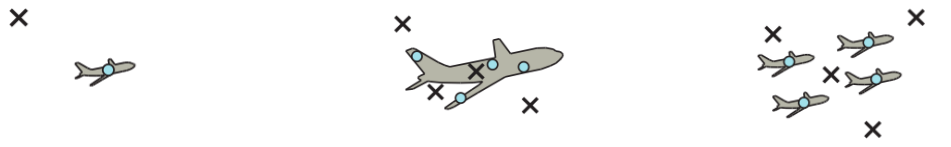
To begin with, object tracking is a well-studied field in applications where either tracked objects are far away from the sensor or sensor resolution is low in comparison to the object size [12]. A typical example is aircraft tracking in air surveillance systems. In that case, sensors lie on the ground, detecting flying objects in a distance magnitude of thousands of meters. In such scenarios, at most one sensor resolution cell is occupied by each object and the so-called "small object" assumptions hold [4]. In more detail, it is assumed that motion of each object is independent and at most a single detection per object per sensor scan is generated. In the **point object tracking** problem, each object is modeled as a point without any spatial extend. The objective is to estimate the number of objects and their kinematic state, which could consist of position, velocity, acceleration, heading and turn rate.

Nevertheless, the small-object assumptions are not valid in many tracking applications. Firstly, modern sensors with increased resolution, like radar, Velodyne LIDAR and laser scanner, provide multiple detections for each object per scan. In addition, novel scenarios consider tracking of objects in the sensors' near field. For example, this is common in automotive tracking applications, where vehicles and VRUs evolve in small distances far from each other [4]. In such tracking applications, multiple measurements per object

per sensor scan are provided, meaning that increased amount of information is available for the targets. In the **extended object tracking** (EOT) problem, each object is considered as a single entity that generates multiple detections per sensor scan per time step, which are spatially distributed around its extent. Hence, it is also possible to estimate the shape extend of objects, such as shape and size, alongside with their kinematic state. The main challenge is to design tracking algorithms that make use of these extra available information in an efficient way. A comprehensive overview of extended object tracking problem aspects and algorithms is included in [4].

A similar issue to the extended object tracking problem is the **group object tracking** (GOT) problem. In detail, a group object also generates multiple spatial distributed detections around its extend. However, it consists of a varying number of sub-objects, which share some common motion characteristics. Each sub-object is treated as a single entity. In other words, each group object occupies multiple sensor resolution cells, while each sub-object may occupy one or several of those cells [13]. In general, groups can spit and merge, as well as they can be close to each other or move independently from each other.

Moreover, group objects can be categorized into two main classes [13], namely the small and large group objects, respectively. Small group objects consist of a small number of sub-objects and usually it is possible to model the interactions and relationships between them. For example, a set of military aircrafts flying in a coordinated manner can be considered as a small group object. In such cases, the objective is to estimate the state of each particular object, as well as some common parameters, which characterize the size and volume of the group [13]. On the other hand, large group objects may consist of hundreds or thousands of sub-objects. An example is a crowded group of pedestrians in surveillance tracking applications. In such cases, it is not possible to distinguish and track the individual sub-objects of the group. Hence, the aggregated motion of the group is considered and interactions between its members are neglected. In other words, a large group is treated in a similar manner as an extended object, meaning that it is surrounded with a single shape and its kinematic and shape extend state is estimated [13]. A comprehensive overview of group object tracking problem aspects and algorithms is included in [13].



(a) Point Object **(b)** Extended Object **(c)** Group Object

Figure 2-2: Examples of point, extended and group objects, respectively. Borrowed from Baum et.al. (2011) [2] and Baum et.al. (2013) [3].

2-1-3 Bayesian State Estimation

As mentioned above, the objective of object tracking is to estimate the states of objects of interest. In terms of Bayesian estimation framework, a probabilistic approach is

used to deal with uncertainty. To be more specific, object states are considered as discrete random variables and knowledge about them is represented with probabilistic density functions (PDF). The recursive algorithm takes place between two consecutive discrete measurement time instances and is characterized by the prior distribution, the measurement likelihood function and the posterior distribution of the object state [10].

For instance, let us denote the states of interest at time step k, $\mathbf{x}_k$, and the sequence of measurements from time step $i = 0$ until time step $i = k$, $\mathbf{z}^k = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$. At first, the prior distribution $p(\mathbf{x}_k|\mathbf{z}^k)$ represents current knowledge about the states of interest $\mathbf{x}_k$, before a new set of measurements is observed. Secondly, the measurement likelihood function $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$ describes the probability of observing a new set of measurements when the states of interest are known. Based on this description, the Bayesian recursive algorithm aims to derive the posterior distribution $p(\mathbf{x}_{k+1}|\mathbf{z}^{k+1})$, which describes the updated knowledge of the states of interest when a new set of measurements has been observed [10].

The Bayesian recursive algorithm consists of two steps, namely the time (or prediction) update step and the measurement update step. Firstly, in the time (or prediction) update step, the motion of the object between two consecutive observations is predicted. For that purpose, a motion model $f(\cdot)$ is required to describe the expected current object state knowing the previous object state. It is accompanied with random process noise $\mathbf{v}$ to describe the uncertainty in object kinematics modeling [9]. The probabilistic representation of the selected motion model, namely the transition density $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$, is used alongside with the prior density to derive the predicted density of the object state $p(\mathbf{x}_{k+1}|\mathbf{z}^k)$, as shown in Equation 2-1. The later represents the prediction for the object state before a new measurement is observed.

Subsequently, in the time update step, the new measurement is used to update the predicted object state. A measurement model $h(\cdot)$, describing the sensor observations given the true object state, is required to achieve that. In addition, random measurement noise $\mathbf{e}$ is used to represent the imperfections in acquiring the data. The probabilistic representation of the selected measurement model, namely the measurement likelihood $p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})$ is used to derive the posterior density of the object state $p(\mathbf{x}_{k+1}|\mathbf{z}^{k+1})$, as shown in Equation 2-2. The aforementioned tracking process starts from a chosen initial object state and evolves accordingly for all operation cycles of the tracking system.

A schematic representation of the Bayesian recursive framework is demonstrated in Figure 2-3. [1]. The formal description of the recursive algorithm is shown in Equations 2-1 and 2-2. In the prediction update step, the Chapman-Kolmogorov equation (Equation 2-1) is used to obtain a prediction for the states of interest at time step $k+1$ when object kinematics are known. Subsequently, in the measurement update step, the prediction is updated with information from the latest set of measurements via Bayes rule (Equation 2-2) [14].

- Time update step - Chapman-Kolmogorov equation:

$$p(\mathbf{x}_{k+1}|\mathbf{z}^k) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}^k)d\mathbf{x}_k \tag{2-1}$$

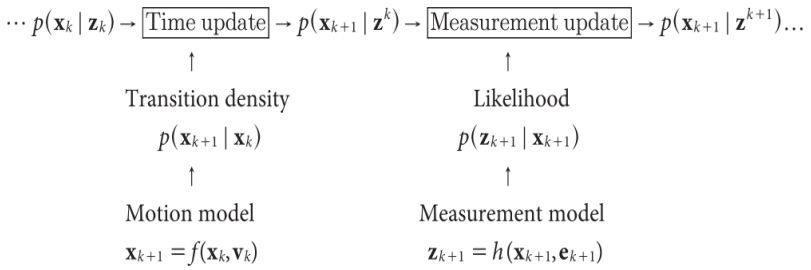


Figure 2-3: Bayesian recursive framework for state estimation. Borrowed from Granström et.al. (2014) [1].

- Measurement update step - Bayes rule:

$$p(\mathbf{x}_{k+1}|\mathbf{z}^{k+1}) = \frac{p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{z}^k)}{p(\mathbf{z}_{k+1}|\mathbf{Z}^k)} \tag{2-2}$$

2-1-4 Modeling Approaches for Extended Object Tracking

In order to apply the recursive Bayesian tracking algorithm to solve the extended object tracking problem, appropriate state and measurement model representations need to be formulated for extended objects. This section summarizes the state-of-the-art measurement and state modeling approaches, widely used in recent studies.

2-1-4-1 Measurement Modeling Approaches

To begin with, the measurement model should capture the number of sensor detections, as well as their spatial distribution around the object. Concerning measurement modeling, two main approaches have been proposed. Firstly, point source models assume that measurements are generated by a known number of specific points on each extended object, called reflection points. In this case, all sensor detections must be associated to one of the reflection points. As a result, required computational complexity to handle this modeling approach is increased for that reason [4].

The second approach concerns spatial distribution models. According to this, the measurements corresponding to an object are approximated as a spatial point process, meaning that their number is random and derived in each cycle by a selected distribution conditioned on the object state. Moreover, their spatial distribution is characterized by combining the individual single measurement likelihoods, resulting into an overall likelihood for the object. By this way, the measurement-to-point-source association problem, which was present in the reflection point approach, is alleviated.

A convenient spatial point process (or spatial model) is the inhomogeneous Poisson point process (PPP), proposed in [15]. According to this model, the number of measurements corresponding to each measurement source are Poisson distributed, with a rate $\gamma(x)$ that is a function of the object state [4]. The spatial distribution of measurements around each measurement source is described by taking into account the single measurement

likelihoods $p(z|x)$, each of them corresponds to a single measurement in the distribution. Each single measurement likelihood makes use of a model for the object extend and a model for the sensor noise [4], so its form depends mainly on the used sensor. A direct consequence of the Poisson assumption is that multiple measurements may originate from the same measurement source [15].

An exact expression for the overall measurement likelihood can be derived that does not require to define explicit associations between the measurement sources and the measurements, as presented in [4]:

$$p(\mathbf{Z}|\mathbf{x}) = \exp^{-\gamma(\mathbf{x})} \gamma(\mathbf{x})^{|\mathbf{Z}|} \prod_{\mathbf{z}\in\mathbf{Z}} p(\mathbf{z}|\mathbf{x}) \tag{2-3}$$

where the first term ($\exp^{-\gamma(\mathbf{x})} \gamma(\mathbf{x})^{|\mathbf{Z}|}$) describes the number of measurements and the second term ($\prod_{\mathbf{z}\in\mathbf{Z}} p(\mathbf{z}|\mathbf{x})$) their spatial distribution.

2-1-4-2 State Modeling Approaches

The exact state representation in each EOT problem depends on factors such as the types of object, sensor data and object motion [4]. Typical parameters of the kinematic state are the objects' (2D or 3D) position and velocity, as well as each heading, orientation and turn rate.

Concerning, shape extend state, parameter selection depends on the selected shape for object modeling. In most practical cases, shape extend models are chosen a priori depending on the type of tracked objects. In other words, object shape is usually assumed to be constant during the duration of a tracking application example. The different shape extend modeling approaches are clustered into three general levels of shape complexity [16], which are illustrated in Figure 2-4:

- Complexity Level 1: No shape modeling:

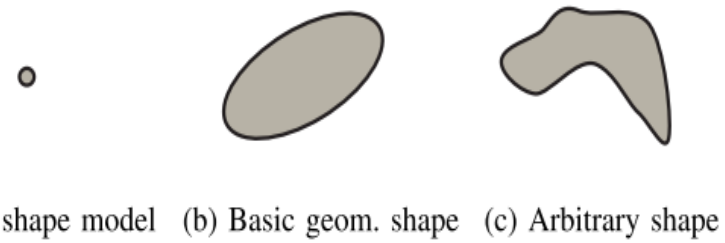
The simplest choice is to avoid shape modeling (Figure 2-4a). In this case, shape estimation is neglected, meaning that only the kinematic state of a selected centroid of the object (for example, the center of mass) is estimated. This approach is simplistic in terms of shape extend modeling and has very low computational complexity. Moreover, it is very flexible to track different kinds of objects, with a varying degree of accuracy in each case [16].

- Complexity Level 2: Simple geometric shape modeling:

The second complexity level enables approaches assuming a simple geometric shape for object extend modeling (Figure 2-4b). Typical examples of such geometric shapes are sticks, rectangles, ellipses and circles. This approach is rather simple and slightly more complex on comparison to the previous approach. Moreover, tracking accuracy is improved, since the object extend is also estimated in this approach.

- Complexity Level 3: Arbitrary shape modeling

Finally, the most advanced approach is to model the object extend with an arbitrary shape, which will be able to handle different object shapes (Figure 2-4c). Such an arbitrary shape can be represented with different ways, for example as a curve with some parameterization or as a combination of simpler geometric shapes. Computational complexity is increased in comparison to the aforementioned approaches, since more parameters are required for arbitrary shape modeling, while tracking accuracy is increased.



(a) No shape model (b) Basic geom. shape (c) Arbitrary shape

Figure 2-4: Shape extend modeling complexity levels.
Borrowed from Granström et.al. (2016) [4].

Depending on selected shape extent representation, different state modeling approaches can be followed for an extended object of interest. The simplest approach, in case of selecting simple geometric shapes such as ellipses, rectangles or sticks, is to incorporate the corresponding parameters in an augmented state vector, containing also the kinematic state components. This state parameterization allows for joint estimation of kinematic and shape extent state parameters.

An alternative approach tailored for elliptically shaped objects is the Random Matrix Model (RMM), where the extended object state is modeled as a combination of a kinematic state vector and a semi-positive definite shape extent matrix. In this case, the joint state density is considered as a product of a Gaussian and an Inverse Wishart distribution (GIW), which are recursively propagated in a linear Kalman-filter-like prediction and measurement update step.

Moreover, the Random Hypersurface Model (RHM) is proposed as a parametric representation of the object state extent, assuming that each measurement source lies in a scaled version of a randomly generated hypersurface. Hence, an augmented state vector is estimated, including parametric components of both kinematic state and shape extent state, by use of nonlinear stochastic filters. Note that RHM representations have been proposed for both elliptical and star-convex (e.g.:arbitrary) shaped objects. For instance, a RHM for star-convex shapes is specified by a radius function $r(\mathbf{p_k}, \phi_k)$, which calculates the distance between object centroid position $\mathbf{p_k}$ and the contour line. A schematic representation for a star-convex RHM is given in Figure 2-5.

Between the two aforementioned measurement models, the RMM is the simplest and most widely used approach in recent studies. In terms of this project, a tracking framework using the RMM for a single pedestrian is employed as our baseline filter, explained in detail in Chapter 3. On top of that, our proposed algorithm, described in Chapter 5, makes use of the RMM for state modeling as well. Note that the RMM can be extended to the multi-object case with a few modifications [4]. However, multi-object tracking

examples are out of the scope of this project, thus no further examination is made in this report.

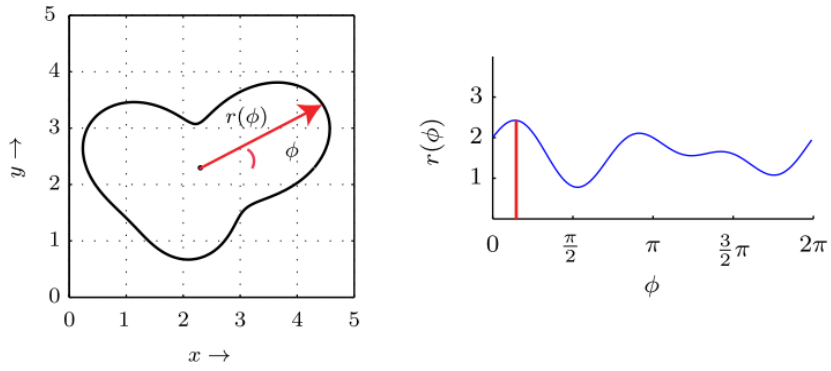


Figure 2-5: Star-convex Random Hypersurface Model (RHM) representation. Borrowed from Baum et.al. (2011) [2].

2-1-5 Random Matrix Model for Single Extended Object Tracking

In the RMM, the extended object state at time instance k is modeled as a combination of a kinematic state vector $\mathbf{x}_k$ and an extend matrix X_k. The probabilistic representation of the object state is given by the corresponding joint density $p(\mathbf{x}_k, X_k|\mathbf{Z}^k)$. In more detail, the kinematic state vector represents the kinematic components of the object state, for example position, velocity and orientation of the object. Moreover, the extend matrix represents the shape extend. X_k is a $d \times d$ matrix, where d is equal to the dimension of the position vector, which is incorporated into the kinematic state vector. In fact, $d = 2$ for 2D position tracking or $d = 3$ for 3D position tracking. The extend matrix is also symmetric and positive definite, meaning that the object extend is modeled as an ellipse [4]. For example, in [5] the RMM is applied to laser range data for state modeling of a cyclist and a pedestrian, as shown in Figure 2-6.

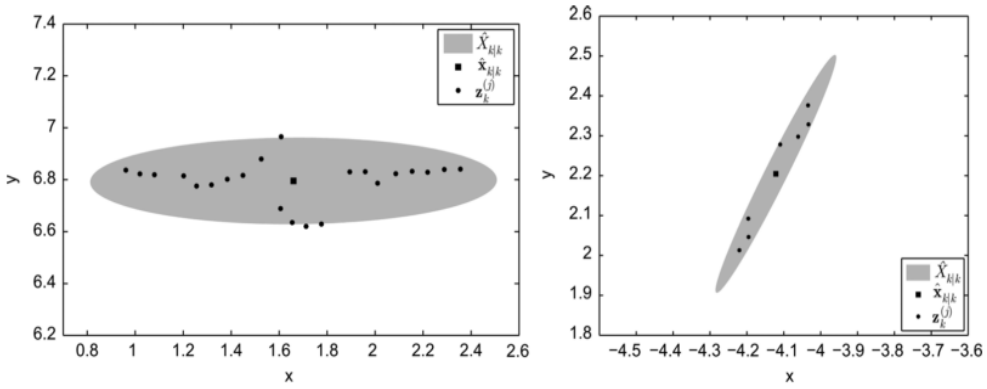


Figure 2-6: Random Matrix (RMM) model applied to laser range data for modeling of a cyclist and a pedestrian. Borrowed from Granström et.al. (2012) [5].

Bayesian Recursion for Single Extended Object Tracking using Random Matrix model
The RMM model for joint estimation of the kinematic and shape extend state within the Bayesian framework was initially proposed by Koch in [8] for single EOT applications. In a few words, RMM-based tracking is an extension of the standard Kalman filter approach for extended objects. Instead of a single measurement, multiple measurements are considered for each detected object. These measurements are assumed independent and conditioned on the object state $\mathbf{x}_k$,X_k, while measurement noise is assumed negligible with respect to the object extend. In detail, the single measurement likelihood is assumed as a Gaussian distribution:

$$p(\mathbf{z}_k \mid \mathbf{x}_k, X_k) \sim \mathcal{N}(\mathbf{z}_k; (H_k \otimes I_d)\mathbf{x}_k, X_k) \tag{2-4}$$

where $\otimes$ is the Kronecker product, the noise covariance matrix is the extend matrix X_k and $(H_k \otimes I_d)$ is a measurement model used to extract the Cartesian position from the kinematic state vector $\mathbf{x}_k$. Then, the measurement likelihood for the object is given by:

$$p(\mathbf{Z}_k \mid n_k, \mathbf{x}_k, X_k) = \prod_{j=1}^{n_k} p(\mathbf{z}_k^j \mid \mathbf{x}_k, X_k) \tag{2-5}$$

where n_k is the number of measurements at time instance k. The Bayesian recursion proposed using RMM state modeling characterizes the joint state density as a product of Gaussian- and Inverse-Wishart-related densities, as shown below:

$$p(\mathbf{x}_k, X_k \mid \mathbf{Z}^k) = p(\mathbf{x}_k \mid X_k, \mathbf{Z}^k)p(X_k \mid \mathbf{Z}^k) \tag{2-6}$$

$$\sim \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k} \otimes X_k) \times \mathcal{IW}_d(X_k; v_{k|k}, V_{k|k}) \tag{2-7}$$

As a result, the object kinematic state is modeled by a Gaussian distribution with mean $m_{k|k}$ and covariance $P_{k|k} \otimes X_k$, while the extend state is modeled by an inverse Wishart distribution with $v_{k|k}$ degrees of freedom and scale matrix $V_{k|k}$ [4].

In practice, instead of estimating the detailed joint posterior density of the object state, the aforementioned parameters of the Gaussian and the inverse Wishart distribution, respectively, are recursively propagated in a Kalman-filter-like prediction update and measurement update step. By this way, the object's centroid kinematic attributes, as well as the ellipse modeling the object extend are estimated [4]. A detailed description of the RMM prediction and measurement update steps in table form is presented in [4]. Note that in the initial RMM presented in [8], non-linear dynamics can not be included in the kinematic state vector. Instead, the kinematic state is limited to the object's position in 2D or 3D and its derivatives, such as velocity and acceleration.

2-2 Related Work

In this Section, a brief description of state-of-the-art EOT tracking algorithms found in recent studies is given. Section 2-2-1 refers to EOT filters tailored for a single elliptically shaped object. Note that in these papers none of corresponding tracking algorithms has been tested with real sensor data. Moreover, examples referring to extended tracking of pedestrians in real automotive scenarios are mentioned in 2-2-2. More focus is given to a specific case study, so that the limitations of state-of-the-art EOT algorithms in real automotive applications are pointed.

2-2-1 State-of-the-art Extended Object Tracking Filters

In terms of the RMM approach proposed by Koch in [8] (see Section 2-1-5), it is assumed that measurement sensor noise is neglected with respect to object extension. Each position measurement is considered as measurement of object centroid scattered over its extent. In other words, the spread of available measurements depends only on object extension and not on sensor noise. This may lead to overestimation of estimated shape extent, in case that this assumption is not valid in practice.

However, Feldmann et.al. proposed an alternative approach in [7], where both object extension and sensor noise is accounted for obtained position measurement spread. A zero-mean additive Gaussian measurement noise in 2D position is considered in the corresponding measurement model. In the so-called RMM filter, kinematic and shape extent representation follows the RMM. Joint estimation of kinematic state vector and extent semi-positive definite matrix takes place, by implementing a linear Kalman-like measurement update step. As a result, this is the simplest tracking approach in terms of implementation and computational complexity. For that reason, the RMM filter is selected as the baseline filter for this project and is explained in detail in this report, namely in Chapter 3.

An alternative tracking algorithm is found in [17], where different state representation and measurement models are selected. In more detail, an augmented state vector containing kinematic and shape extent attributes is defined. Moreover, a novel measurement model, involving a zero-mean multiplicative Gaussian noise (denoting the spread of obtained measurements on object extent) together with an additive Gaussian measurement noise (similarly to the RMM filter). Since a linear filter is not feasible for the multiplicative measurement noise, a second-order Extended Kalman Filter SOEKF is proposed. Since calculation of Jacobian and Hessians metrices is needed for implementation, the SOEKF has increased computational complexity in comparison to the RMM filter.

In addition, the exact same state representation and measurement model is also defined in [18]. The only difference is that a first-order EKF is implemented, where calculation of Hessians is not needed. As a result, the single EKF is less complex than the SOEKF, but still more complex than the RMM filter.

Since all three aforementioned filters are used for tracking of a single elliptical object, corresponding performance metrics for comparison of the estimated and the ground truth elliptical shape are used for filter evaluation. Note that in all three aforementioned studies, simulation scenarios are created in MATLAB software. As a result, creating ground truth data for kinematics and shape extent for the object of interest is a straightforward task. Then, the objective of considered tracking algorithms is to estimate an elliptic extent as close as possible to the considered elliptic ground truth extent in each simulation timestep.

A complete analysis of performance metrics for elliptically shaped objects is found in [19]. Based on this paper, the Gaussian Wasserstein Distance and the Uniform Wasserstein/OSPA distance are the most widely used metrics for ellipses. In fact, a comparison is made between the filters in [18] and results show a quite similar performance for them.

As a result, the low complexity of the RMM filter is the reason why it is selected as the baseline filter in terms of this project.

Nevertheless, in the examined scenario considered in this project, an arbitrary shape is created and used as ground truth shape for the pedestrian in x-y plane. Thus, the aforementioned performance metrics proposed for comparison of elliptical shapes are not suitable here. Instead, a modified version of the Hausdorff distance [20] is used as a performance metric for filter performance evaluation, comparing the arbitrary ground truth shape with the estimated elliptical shape for the pedestrian in 2D. The modified Hausdorff distance is explained in more detail in Section ..., together with corresponding performance measures for pedestrian kinematic state.

2-2-2 Extended Tracking of Pedestrians in Automotive Applications

Whilst various examples can be found in literature concerning extended tracking of vehicles in automotive applications, this is not the case for VRUs and especially pedestrians. In fact, this finding was the initial incentive that directed our focus in extended pedestrians tracking during this M.Sc. thesis project.

Nevertheless, recent studies concerning pedestrian EOT in real automotive scenarios ([21],[5],[22],[6],[23]) share some similarities. In detail, all examples involve data acquisition from Lidar or Laser range scanner (LRS) sensors, returning multiple position measurements of pedestrians. Moreover, the elliptical shape is a widely used choice for pedestrians shape extend, as projected in 2D (x-y plane). This can be justified by the fact that the cross-section of pedestrian body parts in the x-y plane can be considered to have an almost elliptical shape. For example, in applications involving sensors placed on waist level [5], a widely used assumption is to consider the cross-section of the human torso as an ellipsoid in the 2D-space and attempt to estimate its size. On the contrary, in scenarios where sensors are mounted on a vehicle's front bumper [22], the majority of measurements is generated by the legs of a pedestrian. Then an ellipsoid can be used to model the uncertainty of legs motion in 2D-space. In that case, ellipse axes would increase in time when the distance of the legs grows and decrease in the opposite case.

In addition, all found studies focus in scenarios involving multiple pedestrians moving within sensor FoV. Extensions of the aforementioned state-of-the-art EOT modeling approaches (such as the RMM) within the Random Finite Sets (RFS) tracking framework, together with data association methods tailored for extended objects, are employed to estimate the number of detected objects, as well as their kinematic and spatial extent attributes. Concerning shape extent tracking, the main objective in all found examples is to distinguish upon closely spaced and/or occluded pedestrians. In other words, researchers do not aim to estimate an as much as possible accurate shape extent in 2D for each pedestrian. Instead, designed filters aim to keep track of all involved pedestrians, even in time instances when they are moving very close to each other or one is occluded by another.

Last but not least, no ground truth data is available for pedestrian shape extent, thus it is not possible to directly compare the estimated pedestrian shape with a true shape. Instead, alternative performance metrics are applied for performance evaluation of shape

extent tracking. For example, in [21], the area of each extended object is computed and compared to a rough estimate of a cross section of the human torso, after the assumption that it is elliptically shaped. According to the authors of this study, it is assumed that an average person is roughly $50 - 60$ cm wide (torso and arms) and $25 - 30$ cm deep, meaning that the average area is assumed to be $0.1 - 0.15$ m^2. In case that estimated shape area is larger than this average value, it is concluded that the implemented filter has failed to distinguish between two or more closely spaced pedestrians. Moreover, in [6], no metric for extent performance evaluation is used. Instead, estimated pedestrian shape extent is only visually compared with the obtained sensor point cloud data.

2-2-2-1 Case Study

At this point, focus is given in the pedestrian tracking example found in [6] (Beard et.al., 2016), in order to get an insight into the difficulties state-of-the-art EOT algorithms face in real automotive scenarios. The problem discussed in this study is extended tracking of two pedestrians walking on a parking lot, which are recorded by three Ibeo Lux laser sensors mounted on the front bumper of a vehicle. A topview of the experiment is depicted in Figure 2-7a. Pedestrians start from different initial positions and move towards opposite directions. Ground truth and estimated pedestrian trajectories are depicted by dashed and solid lines, with different colour for each pedestrian. Note that ground truth trajectories were obtained by manually labelling the pedestrians in the raw laser scans. In addition, laser measurements (black points) and estimated shape extent for each pedestrian are depicted for selected simulation timesteps. Concerning measurement modeling, laser measurements are assumed to be normally distributed around the pedestrian extent in x-y plane.

A Random Finite Set -based extension of the RMM tailored for multiple object tracking applications is used for state and measurement modeling. In a few words, the unknown number of tracked pedestrians is modeled by a Gamma distribution. In addition, kinematic state (e.g.: 2D-position and velocity) is represented by a Gaussian distribution for each pedestrian, while its shape extent is represented by an Inverse Wishart distribution, similarly to the RMM. As a result, an elliptic shape is considered for each pedestrian. Subsequently, a Labelled Multi Bernoulli (LMB) filter is employed to estimate the aforementioned parameters (e.g.: number of pedestrians, kinematic state and shape extent state for each pedestrian, respectively).

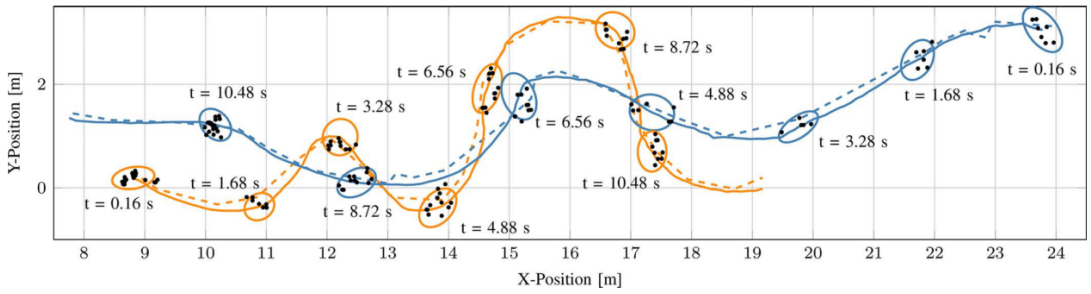
2-2-2-2 Limitations

The main difference between the RMM (as described in Section 2-1-5) and its extension presented in [6] (Beard et.al., 2016) refers to shape extent estimation. In RMM, shape extent tracking requires to estimate the degrees of freedom and scale matrix of the Inverse Wishart distribution modeling the shape extent, as shown in Equation 2-7. Instead, in this study, the two-sigma ellipses representing the uncertainty of 2D-position estimates are used as the estimated shape extent for each pedestrian. In other words, the case study presented in [6] is not an example of joint estimation of kinematics and extent

of pedestrians within the Bayesian framework, since shape extent for each pedestrian is implicitly estimated by employing the uncertainty measures of its kinematic state.

On top of that, no ground truth information is available for each pedestrian. As a result, none of the widely used performance metrics for shape extent tracking can be applied. As already mentioned, this is also the case for all found examples of extended pedestrian tracking in recent studies. To be more specific, in this case study, a visual inspection of estimated two-sigma ellipses and corresponding sensor measurements of the pedestrian takes place to validate shape extent tracking.

In fact, estimating the most accurate shape extent possible for each pedestrian is not the main objective in this case study. Instead, major focus is given on simulation time instances when the pedestrians are very close. As shown in Figure 2-7, this is the case for $t = 6.56$ seconds in the examined example. It is shown in Figure 2-7a that the proposed algorithm manages to distinguish between the two closely spaced targets, while also no overlap exists between the considered two-sigma ellipses representing pedestrian shape extent.



(a) Ground truth (dashed) and estimated (solid) pedestrian trajectories. Estimated two-sigma ellipses and corresponding laser measurements (black) are plotted for selected time steps.

(b) Pedestrian tracking scenario: The two pedestrians are getting close at $t = 6.56 \quad s$ seconds

Figure 2-7: Pedestrian tracking scenario with closely spaced pedestrians. Borrowed from Beard et.al. (2016) [6].

2-3 Pedestrian Pose Output

Based on the aforementioned analysis concerning state-of-the-art EOT algorithms, focusing either on general objects of interest or more specifically to pedestrians, it is concluded that the availability of only Lidar-obtained position measurements might not be enough for accurate pedestrian kinematics and shape tracking. In terms of this project, it is investigated whether an extra sensor modality, leading to the availability of pedestrian body parts detections, can improve tracking performance.

The OpenPose library [1] [24],[25] is the first real-time multi-person system to detect jointly human body, hand, facial and foot keypoints on single images. In the case examined in this report, where images of the pedestrian are obtained by the ego-vehicle Mono camera, 2D detections of 25 body/foot keypoints are calculated on image pixel coordinates by OpenPose (BODY_25 pose output format). In more detail, the set of 25 2D pose detections for the pedestrian body parts calculated by OpenPose in a camera scan at simulation time instance k is denoted as:

$$\mathbf{Z}_{pose,k} := \{\mathbf{z}^{(j)}_{pose,k}\}_{j=0}^{24} \tag{2-8}$$

In addition, each body part detection calculated at simulation time instance k is denoted as:

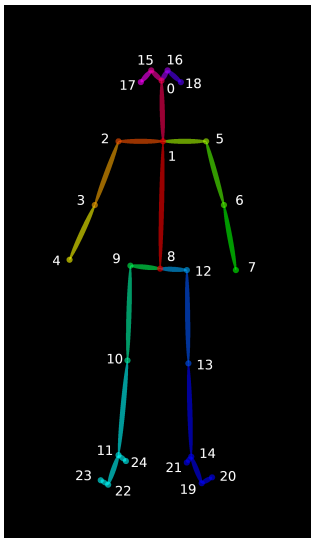
$$\mathbf{z}^{(j)}_{pose,k} = \begin{bmatrix} \mathbf{x}^{(j)}_{pose,k} \\ \mathbf{y}^{(j)}_{pose,k} \\ \mathbf{c}^{(j)}_{pose,k} \end{bmatrix} \tag{2-9}$$

where $\mathbf{x}^{(j)}_{pose,k}$, $\mathbf{y}^{(j)}_{pose,k}$ contain the location of body part j in 2D image pixel coordinates and $\mathbf{c}^{(j)}_{pose,k} \in [0, 1]$ is the detection confidence parameter of body part j. Note that an empty vector $\mathbf{z}^{(j)}_{pose,k}$ is returned in case that the corresponding body part is not detected. The exact locations of obtained pose detections are demonstrated in Figure 2-8a, while the mapping order of obtained pose detections with pedestrian body parts is shown in the list of Figure 2-8b.

Note that the implementation of an algorithm for human pose detection is out of the scope of this project. Thus, the OpenPose library is used for that purpose. The Mono camera images are fed as input to the pose detection algorithm and human pose detections as described in Equations 2-8 - 2-9.

[1] Source Code: https://github.com/CMU-Perceptual-Computing-Lab/openpose

[2] https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/output.md



(a) Location of pose detections on human body.

```
{0,  "Nose"},
{1,  "Neck"},
{2,  "RShoulder"},
{3,  "RElbow"},
{4,  "RWrist"},
{5,  "LShoulder"},
{6,  "LElbow"},
{7,  "LWrist"},
{8,  "MidHip"},
{9,  "RHip"},
{10, "RKnee"},
{11, "RAnkle"},
{12, "LHip"},
{13, "LKnee"},
{14, "LAnkle"},
{15, "REye"},
{16, "LEye"},
{17, "REar"},
{18, "LEar"},
{19, "LBigToe"},
{20, "LSmallToe"},
{21, "LHeel"},
{22, "RBigToe"},
{23, "RSmallToe"},
{24, "RHeel"},
```

(b) List mapping pose detections to corresponding pedestrian body parts.

Figure 2-8: OpenPose library output format (BODY_25). Borrowed from [2].

.

Chapter 3

Baseline EOT filter for single pedestrian tracking

This section presents in detail the RMM approach presented by Feldmann in [7], which is used to implement the Lidar-only based EOT filter. The objective is the following: Given a set of position measurements from a single Lidar sensor at each time step k, try to estimate the kinematic state and shape extent state of a single object of interest. The kinematic state is represented by the 2D centroid position vector $\mathbf{r}_k$ and the 2D centroid velocity vector $\dot{\mathbf{r}}_k$. In other words, position tracking takes place in x-y plane only. As a result, the kinematic state is defined as follows:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{r}_k \\ \dot{\mathbf{r}}_k \end{bmatrix} \tag{3-1}$$

with

$$\mathbf{r}_k = \begin{bmatrix} \mathbf{r}_{x,k} \\ \dot{\mathbf{r}}_{y,k} \end{bmatrix} \qquad \dot{\mathbf{r}}_k = \begin{bmatrix} \dot{\mathbf{r}}_{x,k} \\ \dot{\mathbf{r}}_{y,k} \end{bmatrix} \tag{3-2}$$

Moreover, the shape extent matrix $\mathbf{X}_k$ is a 2x2 SPD matrix, which accounts for the orientation and semi-axes lengths of the estimated ellipse for the object of interest. Concerning sensor detections, the Lidar sensor provides multiple position measurements in 2D per scan (x-,y- position coordinates).

3-1 Measurement modeling

Let us consider that the initial time step of the simulation is denoted as $\kappa = 0$. At each time step k of the simulation, with $k > 0$, a sensor scan takes place, returning

a random number of n_k position measurements $\mathbf{y}_k^j$. The set of n_k independent sensor measurements in a particular scan is denoted as [7]:

$$\mathbf{Y}_k := \{\mathbf{y}_k^j\}_{j=1}^{n_k} \tag{3-3}$$

while the sequence of sensor scans between the initial and current time step is denoted as [7]:

$$\mathcal{Y}_k := \{\mathbf{Y}_\kappa, n_\kappa\}_{\kappa=1}^{k} \tag{3-4}$$

The relation between each position measurement and the kinematic state vector is given by [7]:

$$\mathbf{y}_k^j = \mathbf{H}\boldsymbol{x}_k + \mathbf{w}_k^j \tag{3-5}$$

In detail, $\mathbf{H} = [\mathbf{I}_2, \mathbf{0}_2]$ is the matrix mapping kinematic states to position measurements, while $\mathbf{w}_k^j$ is additive noise to each position detection. Based on the analysis presented in [7], it is assumed that measurement noise is not negligible with respect to object extent. As a result, the variance of w_k^j, depends not only on measurement noise variance, but also on the size of the object of interest:

$$w_k^j \sim \mathcal{N}(w_k; 0, z\mathbf{X}_k + \mathbf{R}) \tag{3-6}$$

where $\mathbf{R}$ is sensor error covariance matrix and z is a scaling factor to the contribution of the object extent to the spread of obtained measurements [7]. Based on Equations 3-3 - 3-5, the measurement likelihood of set $\mathbf{Y}_k$, given the number of measurements, the kinematic state vector and shape extent matrix is given by:

$$p(\mathbf{Y}_k|n_k, \mathbf{x}_k, \mathbf{X}_k) = \prod_{j=1}^{n_k} \mathcal{N}(\mathbf{y}_k^j, \mathbf{H}\mathbf{x}_k, z\mathbf{X}_k + \mathbf{R}) \tag{3-7}$$

The intuition in the use of the scaling factor z can be understood by inspecting Figure 3-1, where three different cases of measurement spreads suited for the RMM applied on a considered elliptic object are demonstrated [7]. In Figure 3-1(a), additive sensor noise is assumed as normally distributed with variance equal to the shape extent matrix $\mathbf{X}_k$. In this case, each measurement y_k^j is considered as a measurement of the centroid scattered over object extension [4]. As a result, too many measurements lie outside the elliptic object area. Nevertheless, a more realistic assumption would be to consider a uniform distribution of scattering centers over the object extent, like the one shown in Figure 3-1(b). In [7],[26], it is proposed that the use of a scaling factor for the contribution of the object extent to the spread of obtained measurements can result in a Gaussian approximation of this uniform distribution (Equations 3-6-3-7), which is also convenient for the measurement update step of the filtering algorithm. An illustration of this is shown in Figure 3-1(c), where the scaling factor is set to $z = \frac{1}{4}$.

According to the analysis explained in detail in [27], it is shown that by introducing the mean measurement $\overline{\mathbf{y}}_k$ and the measurement spread $\overline{\mathbf{Y}}_k$:

$$\overline{\mathbf{y}}_k = \frac{1}{n_k}\sum_{j=1}^{n_k} \mathbf{y}_j^k \tag{3-8}$$

$$\overline{\mathbf{Y}}_k = \sum_{j=1}^{n_k} (\mathbf{y}_k^j - \overline{\mathbf{y}}_k)(\mathbf{y}_k^j - \overline{\mathbf{y}}_k)^T \tag{3-9}$$

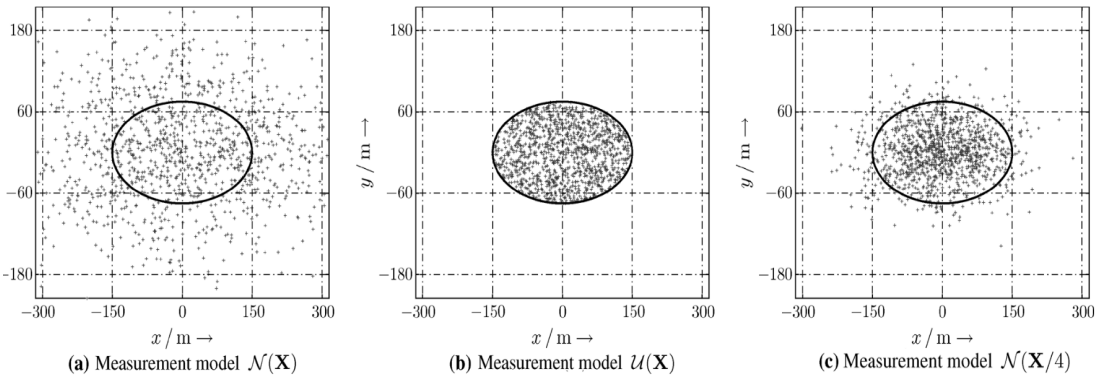


Figure 3-1: Examples of different measurement spreads concerning an elliptic object. The black ellipse denotes the true shape of the elliptic object. The black dots denote obtained measurements y_k^j at a selected time instance k. (a) Lidar measurements obtained from a single scattering center (e.g. object centroid) and additive Gaussian noise is considered with variance equal to the object extent $\mathbf{X}_k$. (b) Lidar measurements obtained from multiple scattering centers, which are uniformly distributed on object extent. (c) Gaussian approximation of scattering centers uniform distribution, with use of scaling factor z. Borrowed from Feldmann et.al. (2011) [7].

then, the measurement likelihood shown in Equation 3-7 can be written as:

$$p(\mathbf{Y}_k|n_k,\mathbf{x}_k,\mathbf{X}_k) \propto \mathcal{N}(\overline{\mathbf{y}}_k,\mathbf{H}\mathbf{x}_k,\frac{z\mathbf{X}_k+\mathbf{R}}{n_k}) \times \mathcal{W}(\overline{\mathbf{Y}}_k,n_k-1,\mathbf{X}_k) \tag{3-10}$$

In other words, the measurement likelihood can be seen as a combination of a Gaussian distribution, representing the obtained sensor measurements, and a Wishart distribution [28],[29], representing the spread of measurements at time instance k.

3-2 State modeling

Based on the form of the measurement likelihood shown in Equation 3-10, a factorized form is selected for the posterior state density [27],[7]:

$$p(\mathbf{x}_k,\mathbf{X}_k|\mathcal{Y}_k) = p(\mathbf{x}_k|\mathbf{X}_k,\mathcal{Y}_k)p(\mathbf{X}_k|\mathcal{Y}_k) \tag{3-11}$$

meaning that it is possible to estimate jointly, but in two separate steps, the kinematic and shape extent state of the object of interest, respectively.

In typical tracking applications, the concept of conjugate priors is applied to the measurement likelihood in order to derive the filtering update equations. This is done effectively in [8], where sensor noise is assumed to be negligible with respect to object extent. On the contrary, this is not the case in [7], which is the approach that is considered in this report. Instead, no analytic form of a conjugate prior of Equation 3-10 can be found, meaning that some approximations must be considered in order to derive the update equations of the filter. The approximations considered for the measurement update and prediction step of the filter are explained in detail in Sections 3-3 and 3-4, respectively.

3-3 Measurement Update Step

Update of Kinematic State The following assumptions are considered for the update step of the kinematic state [7]:

- (Assumption 1:) Assume independence in measurement update of kinematic state $\mathbf{x}_k$ and shape extent matrix $\mathbf{X}_k$. Then, the marginal density of kinematic state in Equation 3-11 becomes:
$$p(\mathbf{x}_k|\mathbf{X}_k, \mathcal{Y}_k) \approx p(\mathbf{x}_k|\mathcal{Y}_k) \tag{3-12}$$
- (Assumption 2:) Consider that the object extent matrix $\mathbf{X}_k$ is non-random and known. This consideration is taken to simplify the needed calculations. As a result, the predicted shape extent matrix $\mathbf{X}_{k|k-1}$ is used where needed.
- (Assumption 3:) Consider that predicted state density is normally distributed, given by:
$$p(\mathbf{x}_k|\mathcal{Y}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{3-13}$$

Then, posterior state density is assumed to be approximately normally distributed as well:
$$p(\mathbf{x}_k|\mathcal{Y}_k) \approx \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k}, \mathbf{P}_{k|k}) \tag{3-14}$$
and a version of the standard linear Kalman filtering measurement update equations can be used for the kinematic state:
$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_{k|k-1}(\overline{\mathbf{y}}_k - \mathbf{H}\mathbf{x}_{k|k-1}) \tag{3-15}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_{k|k-1}\mathbf{S}_{k|k-1}\mathbf{K}_{k|k-1}^T \tag{3-16}$$
where
$$\mathbf{S}_{k|k-1} = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \frac{\mathbf{Y}_{k|k-1}}{n_k} \tag{3-17}$$
$$\mathbf{K}_{k|k-1} = \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}_{k|k-1}^{-1} \tag{3-18}$$
$$\mathbf{Y}_{k|k-1} = z\mathbf{X}_{k|k-1} + \mathbf{R} \tag{3-19}$$
is the approximation of the true innovation covariance, the Kalman gain and the predicted variance of a single measurement, respectively.

Update of Shape Extent Matrix The following assumptions are considered for the update step of the shape extent matrix [7]:

- (Assumption 1:) same with previous paragraph (see Equation 3-12).
- (Assumption 4:) Consider that the predicted shape extent matrix is Inverse Wishart distributed [28],[29], given by:
$$p(\mathbf{X}_k|\mathcal{Y}_{k-1}) = \mathcal{IW}(\mathbf{X}_k, v_{k|k-1}, \alpha_{k|k-1}\mathbf{X}_{k|k-1}) \tag{3-20}$$
with $v_{k|k-1}$ degrees of freedom and $\alpha_{k|k-1}\mathbf{X}_{k|k-1}$ scale matrix.

Then, the posterior shape extent matrix density is also considered to be Inverse Wishart distributed:

$$p(\mathbf{X}_k|\mathcal{Y}_k) = \mathcal{IW}(\mathbf{X}_k, v_{k|k}, \alpha_{k|k}\mathbf{X}_{k|k}) \tag{3-21}$$

with $v_{k|k}$ degrees of freedom and $\alpha_{k|k}\mathbf{X}_{k|k}$ scale matrix. Thus, shape extent estimation consists of estimating the components of the scale matrix of the Inverse Wishart distribution, namely shape uncertainty parameter $\alpha_{k|k}$ and shape extent matrix $\mathbf{X}_{k|k}$.

The updated uncertainty parameter for the shape extent $\alpha_{k|k}$ is given by:

$$\alpha_{k|k} = \alpha_{k|k-1} + n_k \tag{3-22}$$

In addition, the updated shape extent matrix $\mathbf{X}_{k|k}$ is a weighted sum of three matrices:

$$\mathbf{X}_{k|k} = \frac{1}{\alpha_{k|k}}(\alpha_{k|k-1}\mathbf{X}_{k|k-1} + \hat{\mathbf{N}}_{k|k-1} + \hat{\mathbf{Y}}_{k|k-1}) \tag{3-23}$$

In detail:

- $\alpha_{k|k-1}\mathbf{X}_{k|k-1}$ is the predicted scaled matrix, representing the uncertainty of the predicted shape extent matrix
- $\hat{\mathbf{N}}_{k|k-1}$ is proportional to the spread of the true innovation $(\overline{\mathbf{y}}_k - \mathbf{H}\mathbf{x}_{k|k-1})$:

  $$\hat{\mathbf{N}}_{k|k-1} = \mathbf{X}_{k|k-1}^{\frac{1}{2}}\mathbf{S}_{k|k-1}^{-\frac{1}{2}}\mathbf{N}_{k|k-1}(\mathbf{S}_{k|k-1}^{-\frac{1}{2}})^T(\mathbf{X}_{k|k-1}^{\frac{1}{2}})^T \tag{3-24}$$

  with

  $$\mathbf{N}_{k|k-1} = (\overline{\mathbf{y}}_k - \mathbf{H}\mathbf{x}_{k|k-1})(\overline{\mathbf{y}}_k - \mathbf{H}\mathbf{x}_{k|k-1})^T \tag{3-25}$$

- $\hat{\mathbf{Y}}_{k|k-1}$ is proportional to the sum of spreads of measurements around the mean measurement:

  $$\hat{\mathbf{Y}}_{k|k-1} = \mathbf{X}_{k|k-1}^{\frac{1}{2}}\mathbf{Y}_{k|k-1}^{-\frac{1}{2}}\overline{\mathbf{Y}}_k(\mathbf{Y}_{k|k-1}^{-\frac{1}{2}})^T(\mathbf{X}_{k|k-1}^{\frac{1}{2}})^T \tag{3-26}$$

Note that the scaling with use of square root matrices for $\mathbf{X}_{k|k-1}$, $\mathbf{S}_{k|k-1}$, $\mathbf{Y}_{k|k-1}$ is performed during calculations to preserve the SPD structure of the updated shape extent matrix $\mathbf{X}_{k|k}$. In addition, note that despite the considered approximations and the independence assumption for the update of kinematic and shape extent state (Assumption 1), interdependency between kinematics and shape estimation exists, due to the presence of $\mathbf{S}_{k|k-1}$ and $\hat{\mathbf{N}}_{k|k-1}$.

3-4 Prediction Step

Prediction of Kinematic State The following assumptions are considered for the prediction step of the kinematic state [7]:

- (Assumption 5:) Consider independent dymanic models for kinematic state $\mathbf{x}_{k-1|k-1}$ and shape extent matrix $\mathbf{X}_{k-1|k-1}$
- (Assumption 6:) A nearly constant velocity model is considered to describe pedestrian dynamics.

Then, the standard Kalman filtering prediction equations can be used for the kinematic state:

$$\mathbf{x}_{k|k-1} = \mathbf{F}\mathbf{x}_{k-1|k-1} \tag{3-27}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q} \tag{3-28}$$

where $\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is the process matrix representing the evolution of kinematic state in time and $\mathbf{Q}$ is the process noise variance.

Prediction of Shape Extent Matrix The following assumptions are considered for the prediction step of the shape extent matrix [7]:

- (Assumption 7:) Object extent does not tend to change over time
- (Assumption 8:) The variance of the estimated extent matrix increases exponentially over time

Based on these assumptions, the predicted shape extent matrix and its predicted variance parameter are given by:

$$\mathbf{X}_{k|k-1} = \mathbf{X}_{k-1|k-1} \tag{3-29}$$

$$\alpha_{k|k-1} = 2 + exp(-T/\tau)(\alpha_{k-1|k-1} - 2) \tag{3-30}$$

where τ is a time constant modeling the agility with which the shape extent of the object may change over time and T is the prediction time interval.

3-5 Relation between shape extent matrix and parameters of estimated ellipsoid

Until this point, the steps for estimation of the shape extent matrix $\mathbf{X}_k$ for the object of interest have been explained in detail. As already mentioned, $\mathbf{X}_k$ is an SPD matrix, meaning that an elliptic shape is considered for the object. Nevertheless, it is not straightforward to observe the actual parameters of the estimated ellipse, e.g. orientation and lengths of semi-axes, just by inspecting the estimated shape extent matrix $\mathbf{X}_k$. In this paragraph, the mathematical relation between the orientation and semi-axes lengths of an ellipse and the corresponding SPD shape extent matrix is presented.

Definition of ellipse parameters according to RMM Let us denote a vector $\mathbf{s}_k$ containing the parameters of an ellipse at time instance k:

$$\mathbf{s}_k = \begin{bmatrix} \phi_k \\ l_{1,k} \\ l_{2,k} \end{bmatrix} \tag{3-31}$$

In detail:

- ϕ_k denotes ellipse orientation. According to the RMM, orientation is defined as the angle between x-axis of 2D world coordinates plane and the closest semi-axis of the ellipse in the counter-clockwise direction. As a result, $\phi_k \in [0°, 90°]$.
- $l_{1,k}$ denotes the semi-axis length of the ellipse that is closest to x-axis of the 2D world coordinates plane in the counter-clockwise direction
- $l_{2,k}$ denotes the semi-axis length of the ellipse that is furthest to x-axis of the 2D world coordinates plane in the counter-clockwise direction

The shape parameters vector $\mathbf{s}_k$ (Equation 3-31), together with the kinematic state vector $\mathbf{x}_k$ (Equation 3-1) provide a complete identification of an ellipsoid in 2D space (x-y plane). An illustration of shape parameters together with kinematic state parameters is shown in Figure 3-2. Note that ellipse orientation ϕ_k is considered to be different than pedestrian orientation, which is defined as the angle between x-axis and pedestrian velocity vector.

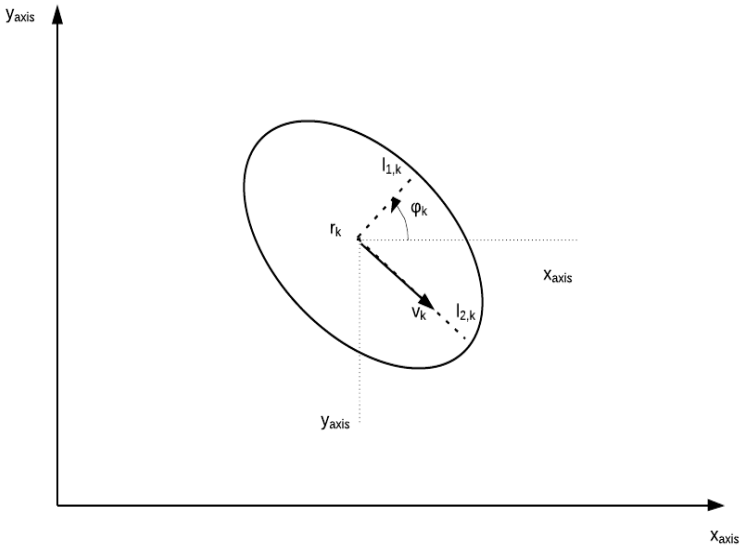


Figure 3-2: Estimated ellipse parameters and kinematic state variables in 2D world coordinates.

Obtain shape extent matrix given the shape parameters vector of ellipse Firstly, let us assume that the shape parameters vector $\mathbf{s}_k$ representing an ellipse is available (see Equation 3-31). Then, the corresponding SPD shape extent matrix $\mathbf{X}_k$ is given by [17]:

$$\mathbf{X}_k = \mathbf{R}_k \mathbf{D}_k \mathbf{R}_k^T \tag{3-32}$$

where

$$\mathbf{R}_k = \begin{bmatrix} cos(\phi_k) & -sin(\phi_k) \\ sin(\phi_k) & cos(\phi_k) \end{bmatrix} \tag{3-33}$$

$$\mathbf{D}_k = \begin{bmatrix} (l_{1,k})^2 & 0 \\ 0 & (l_{2,k})^2 \end{bmatrix} \tag{3-34}$$

is the rotation matrix and the diagonal matrix containing the squared values of semi-axes lengths, respectively.

Obtain shape parameters vector given the shape extent matrix of ellipse Secondly, let us assume that the shape extent matrix $\mathbf{X}_k$ representing an ellipse is available. Then, the orientation of the ellipse is given by:

$$\phi_k = \arctan(-\rho \pm \sqrt{1+\rho^2}) \tag{3-35}$$

with

$$\rho = \frac{\mathbf{X}_k(1,1) - \mathbf{X}_k(2,2)}{2\mathbf{X}_k(1,2)} \tag{3-36}$$

By using the calculated orientation from Equation 3-35, it is possible to calculate the rotation matrix $\mathbf{R}_k$ from Equation 3-33. Subsequently, given Equation 3-32, the diagonal matrix $\mathbf{D}_k$ is given by:

$$\mathbf{D}_k = \mathbf{R}_k^T \mathbf{X}_k \mathbf{R}_k \tag{3-37}$$

As a result, semi-axes lengths of the ellipse are equal to:

$$l_{1,k} = \sqrt{\mathbf{D}_k(1,1)} \tag{3-38}$$

$$l_{2,k} = \sqrt{\mathbf{D}_k(2,2)} \tag{3-39}$$

3-6 Selection of initial state and RMM-related parameters

Concerning state initialization for the RMM-based filter, it must be mentioned that most relevant studies avoid to justify the choices made for initial state values. In fact, justifications are missing both in papers referring to the baseline filter [7], as well as to previous studies regrading extended tracking of VRUs (see Section 2-2). The only useful reference is found in [8], where the following selections are made:

- Intial centroid position $\mathbf{r}_{k_{init}}^{(A)}$ is set equal to the mean of the first measurement set (Equation 3-8) with large covariance matrix, which denotes large uncertainty for initial position:

$$\mathbf{r}_{k_{init}}^{(A)} = \overline{\mathbf{y}}_{k_{init}} = \begin{bmatrix} \overline{\mathbf{y}}_{k_{init},x} \\ \overline{\mathbf{y}}_{k_{init},y} \end{bmatrix} \tag{3-40}$$

- Initial centroid velocity $\dot{\mathbf{r}}_{k_{init}}^{(A)}$ is set equal to zero with large variance, which denotes the maximum speed of the object of interest.

$$\dot{\mathbf{r}}_{k_{init}}^{(A)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \frac{m}{s} \tag{3-41}$$

- Initial ellipse orientation $\phi_{k_{init}}^{(A)}$ is set equal to zero and initial ellipse semi-axes lengths $l_{1,k_{init}}^{(A)}$ and $l_{2,k_{init}}^{(A)}$ are set to constant values, which are randomly selected:

$$\phi_{k_{init}}^{(A)} = 0^{\circ} \tag{3-42}$$

$$l_{1,k_{init}}^{(A)} = c_1 \quad m \tag{3-43}$$

$$l_{2,k_{init}}^{(A)} = c_2 \quad m \tag{3-44}$$

  Subsequently, the corresponding initial shape extent matrix $\mathbf{X}_{k_{init}}^{(A)}$ is calculated by use of Equations 3-32 - 3-34.
- Initial shape uncertainty parameter $\alpha_{k_{init}}^{(A)}$ is set to a small value (close to 2, with $\alpha_{k_{init}}^{(A)} > 2$), which denotes large uncertainty for initial shape extent.

Note that index $^{(A)}$ is used to denote the first state initialization method discussed in this report.

The proposed tracking algorithm for a single pedestrian, discussed in Chapter 5, attempts also to select the initial state by fusing obtained Lidar data and human pose detections. A major difficulty faced on this task is the fact that the mono camera, mounted on the ego-vehicle, does not provide depth information about obtained images. As a result, it is not possible to explicitly transform available human pose detections from 2D image plane coordinates to 3D world (cartesian) coordinates. A method to associate obtained Lidar and camera (e.g.: pose detections) data is presented in Section 4-5. Subsequently, based on this association method, filter state initialization and pedestrian heading measurement creation is discussed in Sections 5-1 and 5-2, respectively.

Chapter 4

Sensor Data Processing

The main topic of interest in terms of this project is the implementation of algorithms achieving extended tracking of VRUs in real automotive applications within the Bayesian framework. In Chapter 2, a brief summary of similar examples found in recent studies is provided, focusing on a specific case study to depict some of the limitations of state-of-the-art approaches. In short, it is mentioned that performance evaluation of estimated shape extend does not take place in previous studies concerning state-of-the-art EOT algorithms for pedestrians in real automotive applications. The main reason for that is the unavailability of ground truth shape extent data for pedestrians, neither in found examples nor in benchmarks like KITTI or MOT. On top of that, widely used performance metrics for shape extent tracking are inappropriate for comparison of arbitrarily shaped objects.

For the aforementioned reasons, using sensor data found in these examples or in benchmarks online is not an appropriate choice. In fact, one of the biggest challenges of this project was the creation of simulation scenaria, so that acquired sensor data resembles to real applications, while also extra ground truth information for objects of interest (e.g.: pedestrian) can be derived. In this project, PreScan software was used to create a simulation close to a real automotive application. PreScan software is widely used for design of automotive scenarios, since objects of interest (vehicles, VRUs), corresponding motion characteristics (object path, velocity), as well as sensor data acquisition are represented quite accurately, in comparison to real automotive applications.

In this Chapter, the examined simulation scenario in PreScan software is described in detail, together with the processing steps of obtained sensor data, including:

- Processing of obtained Lidar data from ego-vehicle to acquire pedestrian position measurements
- Processing of obtained Lidar data from multiple sensors to create ground truth pedestrian shape extent in x-y plane coordinates
- Processing of obtained mono camera images from ego-vehicle using OpenPose library to acquire pedestrian body pose detections (to be used for shape extent

tracking performance evaluation)

- Association of acquired Lidar position measurements and pedestrian body pose detections, which lie in different frame coordinates. Based on the proposed association approach:
  - Initial state for both the baseline and proposed filter is derived
  - An additional heading angle measurement for the pedestrian is derived and used in the nonlinear proposed filter

4-1 PreScan Simulation Description

The objective of this project is to create an algorithm that achieves Extended Object Tracking of a single pedestrian using Lidar and Mono Camera data in automotive applications. In more detail, kinematic state variables of interest are the 2D pedestrian centroid position and the 2D pedestrian velocity (in x-y world coordinates frame). Moreover, orientation and semi-axes lengths of the considered elliptical shape for the pedestrian have to be estimated.

For that reason, the following scenario is considered during simulations taking place in PreScan software: A stationary vehicle (referred to as ego-vehicle in this report) is considered, which has two sensors, a Lidar and a mono camera mounted on it. The Lidar is mounted on the top part of the ego-vehicle and its position is declared as the origin of the 2D world coordinates frame (e.g.: x-y plane). Moreover, the Lidar sensor is considered to have the same specifications as a Velodyne-64 sensor (HDL-64E), namely 0.08° angular resolution in azimuth and 0.4° angular resolution in elevation, respectively. In addition, the mono camera is mounted behind the front mirror of the ego-vehicle, covers a FoV of 60° and returns images with 880×660 pixel resolution.

The object of interest in this scenario is a single pedestrian, moving around the ego-vehicle. In terms of this project, it is assumed that the pedestrian moves by making only forward steps and its motion takes place within the FoV of both sensors mounted on the ego-vehicle (Lidar and mono camera). Based on these assumptions, a path within sensors' FoV is selected, similar to that used in [7] to test the baseline filter explained in Chapter 3. The reason is that we want to test our algorithm in a scenario where the baseline filter is found to perform relatively well in previous studies. In this scenario, pedestrian motion includes straight lines, as well as maneuvers.

Finally, an extra stationary vehicle (referred to as extra-vehicle in this report) is included also in the designed scenario. It is placed opposite to the ego-vehicle, in a symmetrical position with respect to pedestrian path and has a Lidar sensor mounted on it, with similar characteristics to the sensor mounted on the ego-vehicle. The Lidar sensor mounted on the extra-vehicle is only useful for creation of ground truth data for pedestrian shape extent, as explained in Section 4-3. As a result, obtained position measurements from the extra-vehicle are not incorporated in the measurement update step of each filter.

A topview of the considered scenario designed in PreScan software, depicting Lidar sensors position (red circles), the direction of pedestrian motion (black arrow), as well as pedestrian ground truth centroid position (blue cross) for multiple simulation timesteps,

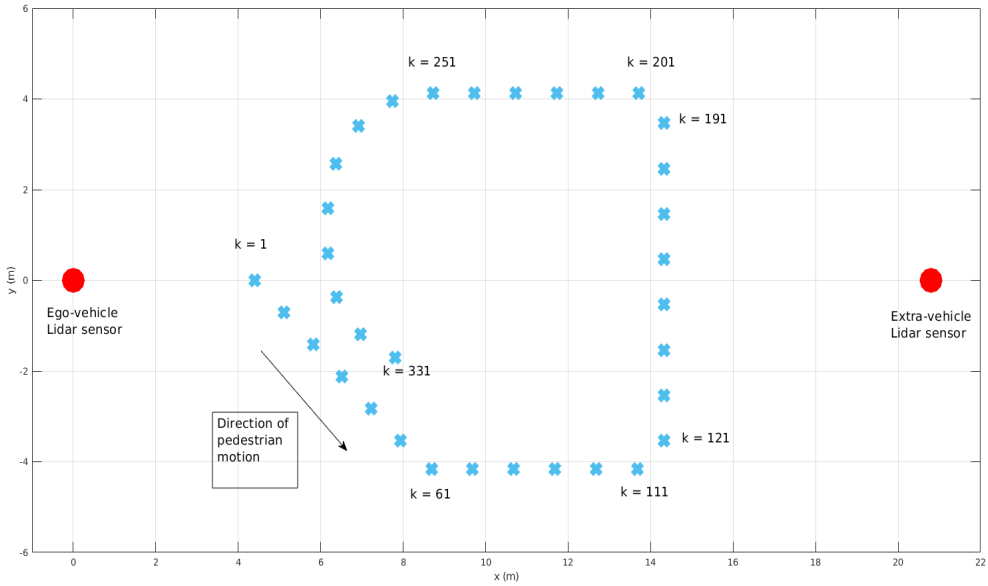


Figure 4-1: Topview of simulation scenario in x-y plane. Shown are the position of ego-/extra-vehicle Lidars (red circles) and selected ground truth position data within pedestrian path.

is shown in Figure 4-1 in 2D world coordinates frame. To begin with, ego-vehicle Lidar sensor is placed at the origin, while extra-vehicle Lidar sensor lies at point $(20.8, 0)$. Pedestrian motion starts at point $(4.4, 0)$ at timestep $k = 1$ and its path direction is declared by the black arrow. After following the depicted straight and maneuver sectors, pedestrian stops at point $(8.68, -1.87)$ at $k = 344$.

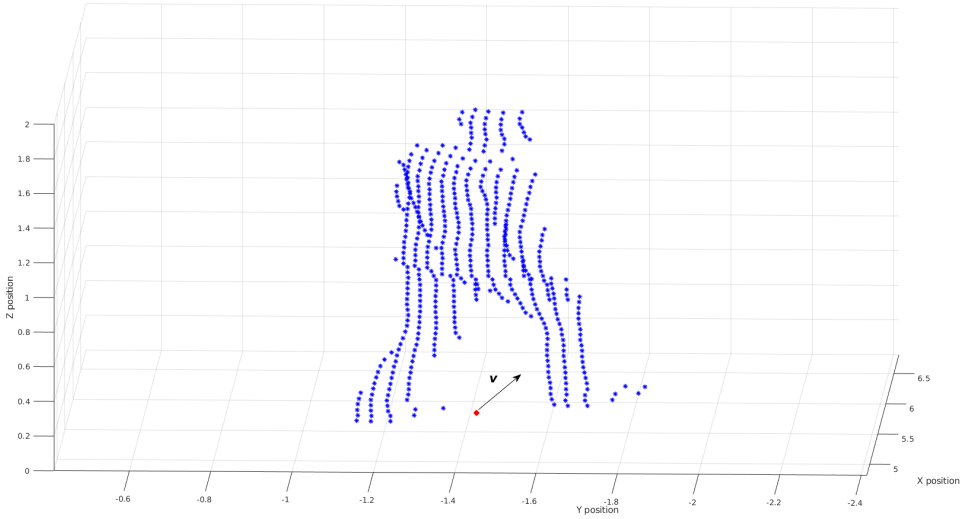
4-2 Lidar Sensor Data Processsing

To begin with, for the sake of simplicity, it is considered that a clustering algorithm already exists, which keeps only Lidar points corresponding to the pedestrian and discards the rest. This consideration is made because the implementation of such a clustering algorithm is out of the scope of this project. In this part of the project, focus is given only on the use of pedestrian sensor data in the tracking process. As a result, PreScan software returns multiple Lidar detections originating from the pedestrian in spherical coordinates (e.g.: range r, azimuth angle ϕ, elevation angle θ).

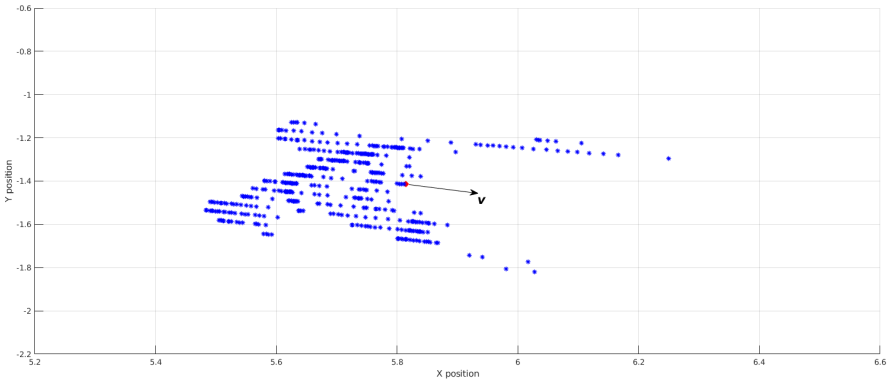
Subsequently, a pre-processing routine is employed which converts obtained Lidar measurements from spherical coordinates to 3D cartesian coordinates (e.g.: position x, position y, position z). This step is necessary, because the baseline filter (presented in Chapter 3), as well as other EOT algorithms (presented in Section 2-2) accept cartesian position measurements obtained from objects of interest as input in the measurement update step. A Lidar-obtained 3D point cloud of the pedestrian for a selected simulation

timestep is shown in Figure 4-2a.

At this point, it should be mentioned that state-of-the-art filters perform object tracking in x-y plane coordinates, meaning that z-coordinate is not accounted for. This is also the case for pedestrian tracking, because of simplicity and effectiveness for automated driving applications. Thus, processed Lidar measurements are projected from 3D cartesian coordinates to the x-y plane. The obtained 2D Lidar position measurements can then be incorporated to the measurement update step of the baseline filter. Zero-mean additive Gaussian noise is also applied to the projected pedestrian Lidar point cloud. For instance, the Lidar-obtained 3D point cloud depicted in Figure 4-2a is projected in x-y world coordinates plane, as shown in Figure 4-2b. Note that in both instances of Figure 4-2, pedestrian velocity vector at corresponding simulation time instance is also demonstrated (black arrow).



(a) Pedestrian 3D point cloud, together with ground truth velocity vector.



(b) Projected pedestrian point cloud in x-y plane together with ground truth velocity vector.

Figure 4-2: Ego-vehicle Lidar oobtained data at simulation timestep $k = 21$.

4-3 Creation of Pedestrian Ground Truth Data for Pedestrian using Lidar Sensor Data

Concerning pedestrian ground truth data, luckily PreScan software stores ground truth kinematic information for all objects that participate in a simulation scenario. As a result, pedestrian 2D centroid position and 2D velocity are automatically generated by PreScan, so they are always available for all simulation timesteps.

A main addition of this M.Sc. thesis project, in comparison to related practical examples discussed in Section 2-2, is that performance evaluation for pedestrian shape tracking takes place. This is achieved by introducing a ground truth shape extent for the pedestrian and applying an appropriate metric to compare it with the estimated one.

Position detections from both ego- and extra-vehicle Lidar sensors are combined to create a ground truth shape for the pedestrian in 2D world coordinates frame (x-y plane). Firstly, Lidar-obtained data from both sensors is processed according to the procedure described in Section 4-2. Subsequently, the random boundary line of the combined Lidar 2D point cloud is calculated. Hence, this arbitrarily shaped boundary is considered as ground truth shape extent of the pedestrian, to be used for performance evaluation of the baseline and proposed tracking algorithms. An example is shown in Figure 4-3, where the combined Lidar point cloud is denoted with blue points and corresponding boundary defined as ground truth pedestrian shape extent is represented by a black closed line for a selected simulation timestep. Moreover, a manually taken topview of pedestrian body in PreScan software for the same simulation timestep is also shown, in order to provide a visual evaluation of the aforementioned method for shape extent ground truth creation.

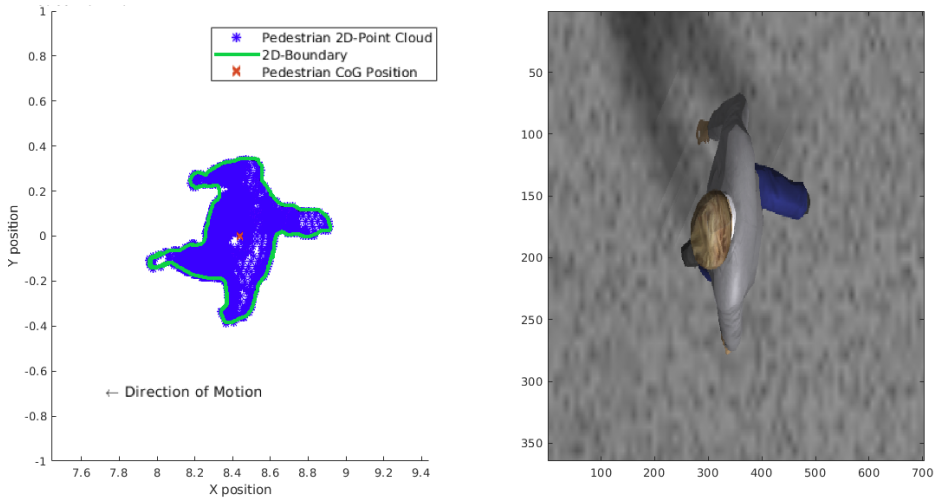


Figure 4-3: Example for pedestrian shape extent creation example. On left, combined Lidar point cloud is shown with blue dots, together with calculated ground truth shape. On right, a manually cropped topview image of pedestrian body, taken from PreScan software visualization of the considered experiment.

A ground truth shape extent for the pedestrian is created in a similar way for all simulation timesteps. A topview of the calculated ground truth pedestrian shape for selected

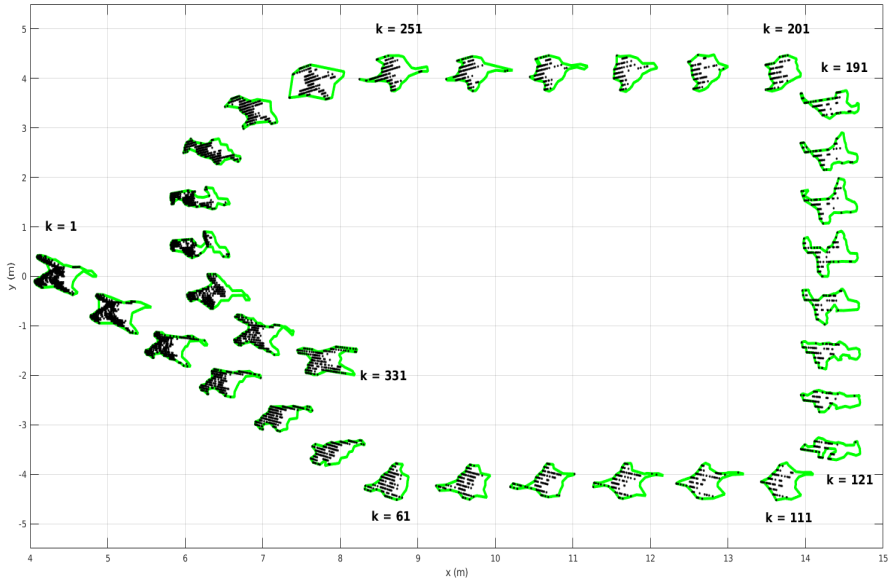


Figure 4-4: A topview of the calculated ground truth pedestrian shape and obtained ego-vehicle Lidar point cloud for selected simulation timesteps in 2D world coordinates frame (x-y plane).

simulation timesteps in 2D world coordinates frame is shown in Figure 4-4. In more detail, calculated shape extent is demonstrated by a thick green boundary, while also Lidar position measurements obtained from the ego-vehicle sensor only are represented with black points.

As already mentioned in Section 2-2-1, widely used EOT performance metrics tailored for comparison of elliptically shaped objects are not suited for this example, where an elliptic estimated shape extent must be compared to an arbitrary ground truth one. Nevertheless, the Modified Hausdorff distance, proposed in [20] for star-convex shapes, is also applied in our simulation scenario for performance evaluation. Modified Hausdorff distance is explained in more detail in Section 5-4.

4-4 Mono Camera Image Data Processing

A mono camera is also mounted on the ego-vehicle, together with the Lidar sensor. More specifically, the mono camera is mounted behind the front mirror of the ego-vehicle, covers a FoV of 60° and returns images with 880×660 pixel resolution. An example of obtained image from the mono camera for a specific simulation timestep is shown Figure 4-5 (left).

In this project, the OpenPose library is employed to detect pedestrian body parts in obtained mono camera images for all simulation timesteps, as depicted in Section 2-3. Since obtained images do not contain depth information, the location of each detected

body part in 2D image pixel coordinates is calculated by the OpenPose library, as described in Equation 2-9. Note that it is out of the scope of this project to create a human pose detection algorithm. Hence, an already existing version of OpenPose is used in this project in order to generate the aforementioned result. An example of mono camera obtained image together with corresponding pedestrian pose detections for a specific simulation timestep is shown in Figure 4-5 (right)

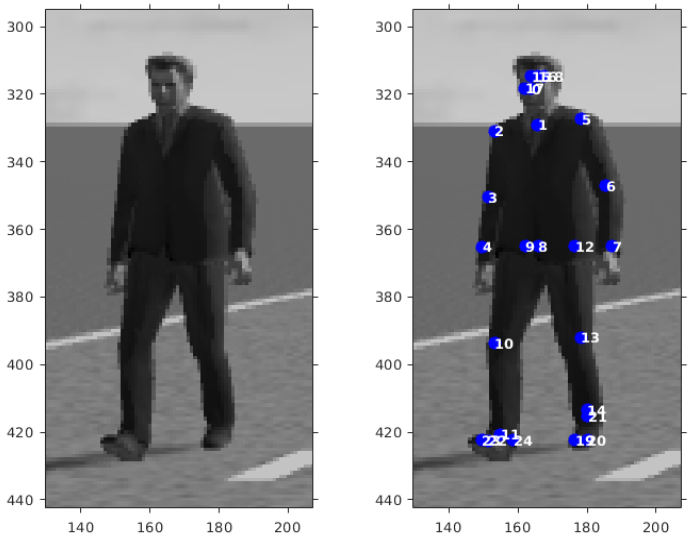


Figure 4-5: Obtained mono camera image (left), together with pedestrian body pose detections (right) at simulation timestep $k = 224$.

4-5 Association of Lidar Position Measurements and Pedestrian Pose Detections

As already mentioned, the vast majority of state-of-the-art EOT algorithms in automotive applications make use of Lidar or Laser-range data obtained from objects of interest (e.g.: vehicles or VRUs). Especially concerning pedestrians, approaches discussed in Section 2-2 enable only Lidar sensors for pedestrian detection. One of the main interests of this project is to investigate whether an additional sensor modality (e.g.: mono camera) can increase accuracy in real automotive extended pedestrian tracking applications. To be more specific, we are interest to investigate the performance of an EOT algorithm, fusing Lidar-obtained position measurements and mono camera-obtained body part pose detections of the pedestrian for filter state initialization and measurement update steps, respectively.

Nevertheless, there is a difference between the coordinate spaces of obtained Lidar measurements, obtained human pose detections and defined position state vector. In fact, Lidar detections refer to 3D world (x-y-z) coordinates, while human pose detections refer

to 2D image pixel coordinates. Moreover, pedestrian position and shape extent tracking takes place in 2D world coordinates (x-y plane). Note that transforming the obtained human body pose detections from 2D image plane coordinates frame to 3D world coordinates frame is not possible, due to the absence of depth information in obtained mono camera images. As a result, fusing Lidar and human pose detections is not a straightforward task.

In this Section, a method to associate obtained Lidar and human pose detections of the pedestrian in each simulation timestep is presented. The aim of this association method is to find the points of obtained ego-vehicle Lidar point cloud that are "close" to selected pedestrian pose detections of interest (e.g.: right/left shoulders/ankles detections, respectively). Then, these Lidar-obtained points can be used in our algorithm to represent obtained pedestrian body parts pose detections in 3D world coordinates frame. To begin with, the following assumptions are considered for the simulation scenario:

- Ego-vehicle Lidar and mono camera sensors are synchrounous
- Pedestrian motion takes place within FoV of both ego-vehicle sensors

Then, the following steps are considered for the proposed association method:

1. Obtained Lidar measurements are projected from 3D (x-y-z) world coordinates to 2D image plane coordinates. After this step, projected Lidar and obtained human pose detections are both available in image plane coordinates. In Figure 4-6, projected Lidar data together with corresponding human pose detections in 2D image plane coordinates are depicted for the same selected simulation timestep.
2. At this point, the sets of projected Lidar data and obtained pedestrian body pose detections, respectively, are available in 2D image plane coordinates. This step

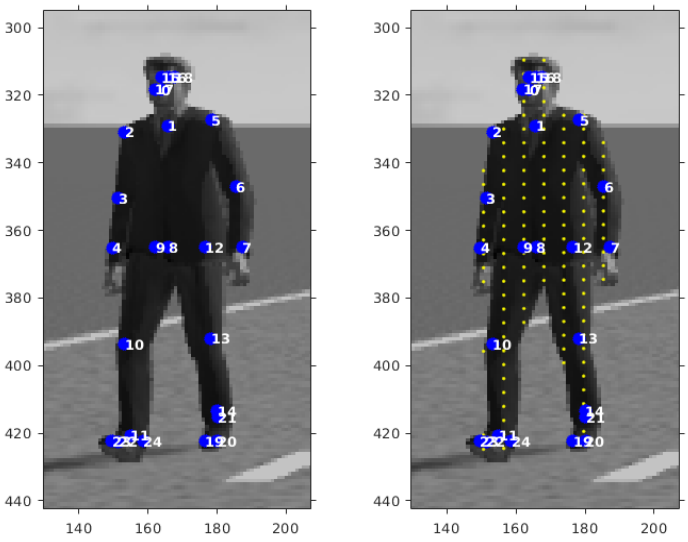


Figure 4-6: Step 1 of proposed association method between Lidar and human pose data at $k = 224$: Project pedestrian Lidar point cloud to 2D image plane coordinates.

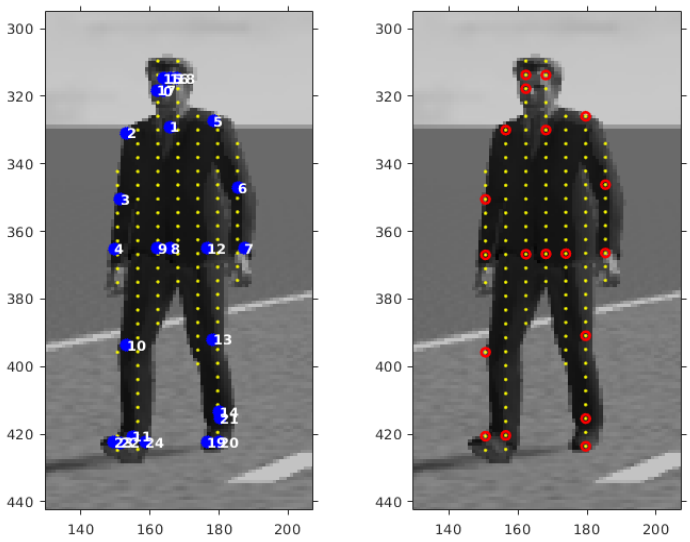


Figure 4-7: Step 2 of proposed association method between Lidar and human pose data at $k = 224$: Find "closest" projected Lidar points to each pedestrian pose detection in 2D image plane.

aims to find the "closest" projected Lidar point to each human pose detection point in 2D image plane coordinates. In more detail, the minimum Euclidean distance between each human pose detection point and neighboring projected Lidar points is selected as distance metric to define the closest Lidar point. In Figure 4-7, selected projected Lidar points for each pedestrian body part (red circles), together with corresponding human pose detections in 2D image plane coordinates are depicted for the same selected simulation timestep.

3. Subsequently, each projected Lidar data, selected as closest to corresponding pedestrian body part detection in 2D image plane coordinates, is associated to its corresponding Lidar measurement in 3D world (x-y-z) plane coordinates. Thus, at this point, 25 Lidar measurements in x-y-z plane have been selected to represent the 25 pedestrian body parts, which are initially obtained by running OpenPose library. An example is shown in Figure 4-8, where the projected Lidar data that are closest to corresponding human pose detections in the 2D image pixel coordinates frame (red circles on left sub-figure) are associated to obtained ego-vehicle Lidar data in the 3D world coordinates frame (red dots on right sub-figure).
4. As already mentioned in this report, tracking of pedestrian position and shape extent takes place in 2D world coordinates frame (e.g.: x-y plane). In Section 4-2, it was denoted that obtained Lidar measurements are projected from 3D cartesian coordinates to the x-y plane. This is also the case for the 25 Lidar measurements in the 3D cartesian coordinates frame that have been associated to pedestrian body pose detections based on the aforementioned steps. As a result, in each simulation timestep, 25 Lidar measurements lying in the x-y plane are derived to represent corresponding obtained pedestrian pose detections. By this way, these detections

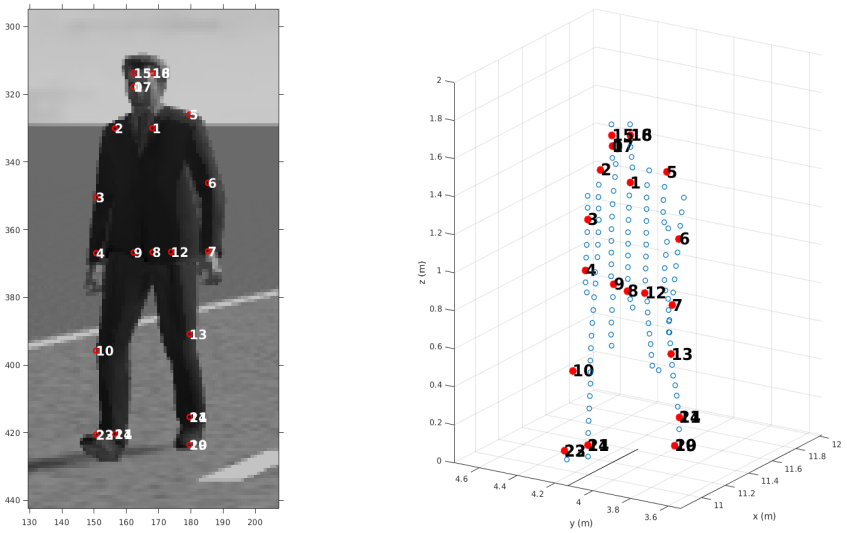


Figure 4-8: Step 3 of proposed association method between Lidar and human pose data at $k = 224$: Associate each "closest" projected Lidar point in 2D image coordinates to its corresponding Lidar measurement in 3D world coordinates.

in 2D can be used to initialize our proposed tracking algorithm, as well as to create an extra measurement (e.g.: pedestrian heading angle measurement) with respect to the baseline filter. An illustration of this step is shown in Figure 4-9 for the same selected simulation timestep.

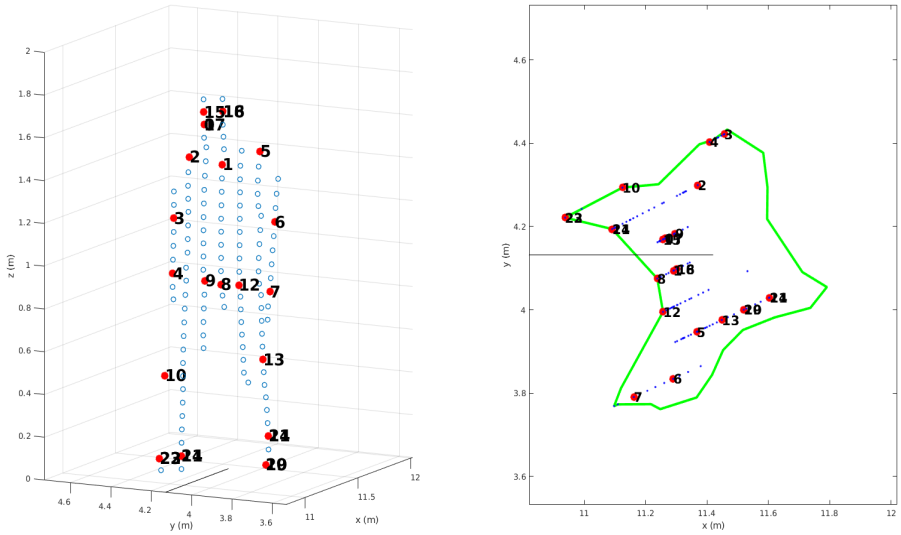


Figure 4-9: Step 4 of proposed association method between Lidar and human pose data at $k = 224$: Discard the z-coordinate in associated points. By this way, pose detections can be represented in 2D world coordinates (x-y plane) by the selected Lidar points.

Chapter 5

Proposed EOT filter for single pedestrian tracking

In Chapter 3, the RMM-based tracking algorithm proposed by Feldmann [7] is presented in detail. As already mentioned, this filter is designed for a single object of interest, while a single sensor (Lidar) is used to obtain multiple 2D-position measurements of the object of interest in each sensor scan. In terms of this project, it is investigated whether an extra sensor modality could improve estimates of the kinematic or shape extent attributes of the filter for a single pedestrian. The extra sensor used is a Mono camera, providing 2D images of the moving pedestrian. Note that no depth information for corresponding images are provided. Subsequently, pose detections of pedestrian body parts (e.g. shoulders, legs, etc) are obtained from each camera image. Processing steps of obtained Lidar and camera data from PreScan software are explained in detail in Chapter 4.

The main contribution of this M.Sc. Thesis project is the incorporation of the human pose detections in a Bayesian algorithm for EOT, such as the RMM-based model. In detail, it is investigated whether an extra sensor modality (Mono camera) could lead to improved performance of the RMM-based filter, especially regarding the limitations presented in the previous sections. The proposed version of EOT algorithm is compared to the baseline filter proposed by Feldmann in [7] with respect to the following aspects:

1. State Initialization of the filter: In our proposed version of the tracking algorithm, human pose detections of shoulders and ankles are used, together with obtained Lidar data, to provide an initial value for the kinematic state and shape vector parameters.
2. Measurement Update step of the filter: In our proposed version of the tracking algorithm, camera-obtained pose detections of pedestrian shoulders are used to create an extra measurement, for pedestrian heading angle. Subsequently, Lidar-obtained point cloud data for pedestrian position is fused with human-pose-obtained pedestrian heading angle in the update step.

In this Chapter, the use of associated Lidar points and human pose detections of the pedestrian (based on the analysis in Section 4-5) is discussed in order to define the initial state variables, as well as a pedestrian heading angle measurement. In addition, the measurement update step of our proposed tracking algorithm is described in detail.

5-1 State Initialization using Pedestrian Pose Detections

5-1-1 Considered Assumptions

Using only Lidar detections to initialize position and shape extent state variables, as well as assigning zero values to initial velocity variables may result in degraded filter performance and slow convergence to true state values. An idea to tackle this issue is to define the corresponding initial state variables by making use of selected 2D pose detections corresponding to human shoulders and ankles, obtained from OpenPose library. Note that these detections are associated to pedestrian Lidar data according to the procedure presented in Section 4-5.

In order to use the obtained 2D pose detections to initialize pedestrian state variables, the following assumptions are made for the extent of pedestrian body and motion in x-y plane (2D world coordinates frame):

- The pedestrian can only move forward during any examined simulation.
- Pose detections for both shoulders and ankles (e.g.: right and left) are available at all time steps of the examined simulation.
- Pedestrian body is approximated by an elliptical shape in x-y plane (even if the considered ground truth shape extent is not elliptical).
- The initial 2D centroid position of pedestrian extent is close to the mean of shoulder detections (mean of Points 2 and 5 in Figure 2-8).
- One of the initial semi-axes lengths of pedestrian extent is considered to be given by the distance between shoulder detections (Points 2 and 5 in Figure 2-8).
- One of the initial semi-axes lengths of pedestrian extent is considered to be given by the distance between ankle detections (Points 11 and 14 in Figure 2-8).
- Initial orientation of the ellipse corresponding to pedestrian extent is considered to be equal to the angle between x-axis and the closest initial semi-axes length in the counter-clockwise direction. As already mentioned in Section 3-5 [7], pedestrian ellipse orientation $\in [0°, 90°]$.

The validity of these assumptions can be shown in Figure 5-1, where selected pose detections are demonstrated in 2D world coordinates with red circles for the same selected simulation timestep. It is shown that the axis between shoulder detections (Points 2 and 5) is almost perpendicular to pedestrian true velocity, while the axis between ankles detections (Points 11 and 14) is close to being parallel to the velocity vector. Hence, using the distances between shoulders and ankles detections in x-y plane, respectively, to denote the initial values of initial ellipse semi-axes lengths is a sensible choice. Concerning initial centroid position, it is shown that the mid-hip detection (Point 8), is not very close to ground truth pedestrian position. Instead, the mean point of shoulders-axis,

seems to be a better choice for pedestrian position initialization. Finally, shoulder detections are also used to derive the initial orientation of the elliptical shape, representing the pedestrian, as well as to create the heading angle measurement to be incorporated in the measurement update step of the proposed filter, as explained in the following Sections of this report.

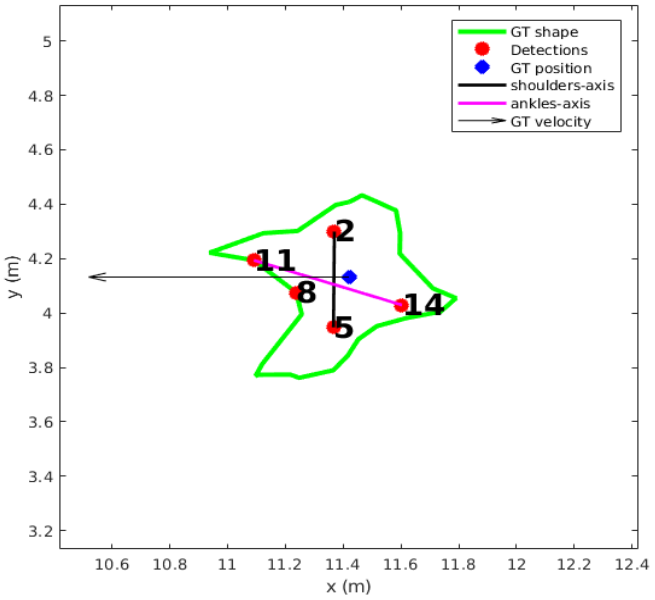


Figure 5-1: Pose detections of shoulders (points 2-5), ankles (points 11-14), mid-hip (point 8) in 2D world coordinates frame, together with corresponding axes at simulation timestep $k = 224$.

5-1-2 Initialization of kinematic state variables

Initial pedestrian 2D position Let us consider that the initial simulation timestep is denoted as k_{init}. In addition, let us assume that shoulder detections for k_{init} are available and represented in image pixel coordinates as:

$$\mathbf{z}^{2(pixel)}_{pose,k_{init}} = \begin{bmatrix} \mathbf{x}^{2}_{pose,k_{init}} \\ \mathbf{y}^{2}_{pose,k_{init}} \\ \mathbf{c}^{2}_{pose,k_{init}} \end{bmatrix}^{(pixel)} \qquad \mathbf{z}^{5(pixel)}_{pose,k_{init}} = \begin{bmatrix} \mathbf{x}^{5}_{pose,k_{init}} \\ \mathbf{y}^{5}_{pose,k_{init}} \\ \mathbf{c}^{5}_{pose,k_{init}} \end{bmatrix}^{(pixel)} \tag{5-1}$$

where $\mathbf{z}^{2(pixel)}_{pose,k_{init}}$ and $\mathbf{z}^{5(pixel)}_{pose,k_{init}}$ correspond to initial detections of right and left shoulder, respectively (see Equation 2-9).

In addition, consider that the transformed 2D points, representing each initial shoulder detection in 2D world coordinates (based on the sensor data processing routine described in Section 4-5) are given as follows:

$$\mathbf{z}_{pose,k_{init}}^{2^{(2D,world)}} = \begin{bmatrix} \mathbf{x}_{pose,k_{init}}^{2} \\ \mathbf{y}_{pose,k_{init}}^{2} \end{bmatrix}^{(world)} \qquad \mathbf{z}_{pose,k_{init}}^{5^{(2D,world)}} = \begin{bmatrix} \mathbf{x}_{pose,k_{init}}^{5} \\ \mathbf{y}_{pose,k_{init}}^{5} \end{bmatrix}^{(world)} \tag{5-2}$$

where $\mathbf{z}_{pose,k_{init}}^{2^{(2D,world)}}$ and $\mathbf{z}_{pose,k_{init}}^{5^{(2D,world)}}$ correspond to initial detections of right and left shoulder, respectively.

Then, based on Equation 5-2, initial centroid position is defined as:

$$\mathbf{r}_{k_{init}}^{(B)} = mean(\mathbf{z}_{pose,k_{init}}^{2^{(2D,world)}}, \mathbf{z}_{pose,k_{init}}^{5^{(2D,world)}}) = \begin{bmatrix} \frac{\mathbf{x}_{pose,k_{init}}^{2} + \mathbf{x}_{pose,k_{init}}^{5}}{2} \\ \frac{\mathbf{y}_{pose,k_{init}}^{2} + \mathbf{y}_{pose,k_{init}}^{5}}{2} \end{bmatrix}^{(world)} = \begin{bmatrix} \mathbf{r}_{k_{init},x} \\ \mathbf{r}_{k_{init},y} \end{bmatrix} \tag{5-3}$$

Note that index $^{(B)}$ is used to denote that this is the second state initialization method discussed in this report.

Initial pedestrian 2D velocity Concerning pedestrian velocity initialization, zero values for x-y velocity are selected in previous studies (see Section 3-6). In this project, a two-point difference of shoulders mean detection for two consecutive simulation timesteps is selected as initial velocity at k_{init}.

In more detail, let us assume that shoulder detections for k_{init} and k_{init-1} are available. The representation of shoulders detections in 2D world coordinates is then given by Equation 5-2 at k_{init}. Similarly, at k_{init-1}, shoulders detections in x-y plane are given as:

$$\mathbf{z}_{pose,k_{init-1}}^{2^{(2D,world)}} = \begin{bmatrix} \mathbf{x}_{pose,k_{init-1}}^{2} \\ \mathbf{y}_{pose,k_{init-1}}^{2} \end{bmatrix}^{(world)} \qquad \mathbf{z}_{pose,k_{init-1}}^{5^{(2D,world)}} = \begin{bmatrix} \mathbf{x}_{pose,k_{init-1}}^{5} \\ \mathbf{y}_{pose,k_{init-1}}^{5} \end{bmatrix}^{(world)} \tag{5-4}$$

where $\mathbf{z}_{pose,k_{init-1}}^{2^{(2D,world)}}$ and $\mathbf{z}_{pose,k_{init-1}}^{5^{(2D,world)}}$ correspond to initial detections of right and left shoulder, respectively, at k_{init-1}.

Then, based on Equation 5-4, initial centroid position at k_{init-1} is defined as:

$$\mathbf{r}_{k_{init-1}} = mean(\mathbf{z}_{pose,k_{init-1}}^{2^{(2D,world)}}, \mathbf{z}_{pose,k_{init-1}}^{5^{(2D,world)}}) = \begin{bmatrix} \frac{\mathbf{x}_{pose,k_{init-1}}^{2} + \mathbf{x}_{pose,k_{init-1}}^{5}}{2} \\ \frac{\mathbf{y}_{pose,k_{init-1}}^{2} + \mathbf{y}_{pose,k_{init-1}}^{5}}{2} \end{bmatrix}^{(world)} = \begin{bmatrix} \mathbf{r}_{k_{init-1},x} \\ \mathbf{r}_{k_{init-1},y} \end{bmatrix} \tag{5-5}$$

As a result, based on Equations 5-3, 5-5, initial centroid velocity is given as:

$$\dot{\mathbf{r}}_{k_{init}}^{(B)} = \begin{bmatrix} \dot{\mathbf{r}}_{k_{init},x} \\ \dot{\mathbf{r}}_{k_{init},y} \end{bmatrix} = \begin{bmatrix} \frac{\mathbf{r}_{k_{init},x} - \mathbf{r}_{k_{init-1},x}}{\Delta t} \\ \frac{\mathbf{r}_{k_{init},y} - \mathbf{r}_{k_{init-1},y}}{\Delta t} \end{bmatrix} \tag{5-6}$$

where Δt is the time interval between two consecutive simulation timesteps. In the considered simulation scenario, data acquisition frequency is equal to $f_{sensor} = 10H$, thus $\Delta t = 0.1s$

5-1-3 Initialization of shape state parameters

Initial ellipse semi-axes lengths After initialization of centroid 2D position and velocity, initial values for orientation and semi-axes lengths should be derived using pedestrian pose detections. Again, let us consider that right/left shoulder detections for the initial simulation timestep k_{init} are available. Their representation in image pixel coordinates and 2D world coordinates is shown in Equation 5-1 and 5-2, respectively. Moreover, ankles detections for the initial simulation timestep k_{init} is represented in image pixel coordinates as:

$$\mathbf{z}_{pose,k_{init}}^{11^{(pixel)}} = \begin{bmatrix} \mathbf{x}_{pose,k_{init}}^{11} \\ \mathbf{y}_{pose,k_{init}}^{11} \\ \mathbf{c}_{pose,k_{init}}^{11} \end{bmatrix}^{(pixel)} \qquad \mathbf{z}_{pose,k_{init}}^{14^{(pixel)}} = \begin{bmatrix} \mathbf{x}_{pose,k_{init}}^{14} \\ \mathbf{y}_{pose,k_{init}}^{14} \\ \mathbf{c}_{pose,k_{init}}^{14} \end{bmatrix}^{(pixel)} \tag{5-7}$$

where $\mathbf{z}_{pose,k_{init}}^{11^{(pixel)}}$ and $\mathbf{z}_{pose,k_{init}}^{14^{(pixel)}}$ correspond to initial detections of right and left ankle, respectively.

Also, consider that transformed 2D points, representing right/left ankles detections in 2D world coordinates (based on the sensor data processing routine described in Section 4-5) are given as:

$$\mathbf{z}_{pose,k_{init}}^{11^{(2D,world)}} = \begin{bmatrix} \mathbf{x}_{pose,k_{init}}^{11} \\ \mathbf{y}_{pose,k_{init}}^{11} \end{bmatrix}^{(world)} \qquad \mathbf{z}_{pose,k_{init}}^{14^{(2D,world)}} = \begin{bmatrix} \mathbf{x}_{pose,k_{init}}^{14} \\ \mathbf{y}_{pose,k_{init}}^{14} \end{bmatrix}^{(world)} \tag{5-8}$$

where $\mathbf{z}_{pose,k_{init}}^{11^{(2D,world)}}$ and $\mathbf{z}_{pose,k_{init}}^{14^{(2D,world)}}$ correspond to initial detections of right and left ankle, respectively, at k_{init}.

As already mentioned, semi-axes lengths of initial extent are derived using the Euclidean distances between shoulders detections and ankles detections, respectively, for the starting timestep of the filter:

$$d_{shoulders} = \|\mathbf{z}_{pose,k_{init}}^{2^{(w)}} - \mathbf{z}_{pose,k_{init}}^{5^{(w)}}\| \tag{5-9}$$

$$d_{ankles} = \|\mathbf{z}_{pose,k_{init}}^{11^{(w)}} - \mathbf{z}_{pose,k_{init}}^{14^{(w)}}\| \tag{5-10}$$

However, it is not immediately clear which Euclidean distance should be applied to the first ($l_{1,k_{init}}$) and second ($l_{2,k_{init}}$) lengths of initial ellipse, according to the vector representation of shape parameters shown in Equation 3-31. At this point, it should be reminded that according to Section 3-5, $l_{1,k}$ denotes the semi-axis length of the ellipse that is closest to x-axis of the 2D world coordinates plane in the counter-clockwise direction. As a result, depending on the relation between x-y coordinates of right and left shoulder detections, as shown in Equation 5-2, and after some geometric calculations, the following cases unveil for initial semi-axes lengths:

1. If $\mathbf{z}^{2^{(w)}}_{pose,k_{init}} \lesseqgtr \mathbf{z}^{5^{(w)}}_{pose,k_{init}}$, then

$$l^{(B)}_{1,k_{init}} = \frac{d_{shoulders}}{2} \tag{5-11}$$

$$l^{(B)}_{2,k_{init}} = \frac{d_{ankles}}{2} \tag{5-12}$$

2. If $\mathbf{x}^{2^{(w)}}_{pose,k_{init}} \lesseqgtr \mathbf{x}^{5^{(w)}}_{pose,k_{init}}$ and $\mathbf{y}^{2^{(w)}}_{pose,k_{init}} \gtreqless \mathbf{y}^{5^{(w)}}_{pose,k_{init}}$, then

$$l^{(B)}_{1,k_{init}} = \frac{d_{ankles}}{2} \tag{5-13}$$

$$l^{(B)}_{2,k_{init}} = \frac{d_{shoulders}}{2} \tag{5-14}$$

Initial ellipse orientation Let us consider again that right/left shoulder detections in 2D world coordinates are available for the initial simulation timestep k_{init} and given by Equation 5-2. Depending on the relation between x-y coordinates of right and left shoulder detections, as shown in Equation 5-2, and after some geometric calculations, the following cases unveil for initial ellipse orientation:

1. If $\mathbf{z}^{2^{(w)}}_{pose,k_{init}} \lesseqgtr \mathbf{z}^{5^{(w)}}_{pose,k_{init}}$, then

$$\phi^{(B)}_{k_{init}} = \arcsin \frac{|\mathbf{y}^{2}_{pose,k_{init}} - \mathbf{y}^{5}_{pose,k_{init}}|}{d_{shoulders}} \tag{5-15}$$

2. If $\mathbf{x}^{2^{(w)}}_{pose,k_{init}} \lesseqgtr \mathbf{x}^{5^{(w)}}_{pose,k_{init}}$ and $\mathbf{y}^{2^{(w)}}_{pose,k_{init}} \gtreqless \mathbf{y}^{5^{(w)}}_{pose,k_{init}}$, then

$$\phi^{(B)}_{k_{init}} = \arccos \frac{|\mathbf{y}^{2}_{pose,k_{init}} - \mathbf{y}^{5}_{pose,k_{init}}|}{d_{shoulders}} \tag{5-16}$$

Note that initial ellipse orientation is bounded on the interval $[0°, 90°]$, as already mentioned in Section 3-5.

Initial shape extent matrix Subsequently, derived initial state variables are used to calculate the initial shape extent matrix $\mathbf{X}^{(B)}_{k_{init}}$ based on Equations 3-32 - 3-34. It is also importnat to mention that there is no point in considering uncertainty measures (e.g. variance) for orientation and semi-axes lengths, since they are not used to calculate the uncertainty of the shape extent estimate! In fact, this is an important limitation of the RMM-based filter. Instead, a small value of shape uncertainty parameter α can be selected (close to 2, with $\alpha^{(B)}_{k_{init}} > 2$), denoting large uncertainty for initial shape extent.

An example of the aforementioned initialization method for a selected simulation timestep ($k = 224$) is depicted in Figure 5-2. It is shown that the calculated initial centroid position for the pedestrian (black rectangle) is closer to its true value (blue star), in comparison to the mean Lidar measurement (green circle). In addition, the calculated initial ellipse (red ellipse) covers the area of the considered ground truth pedestrian shape (green boundary).

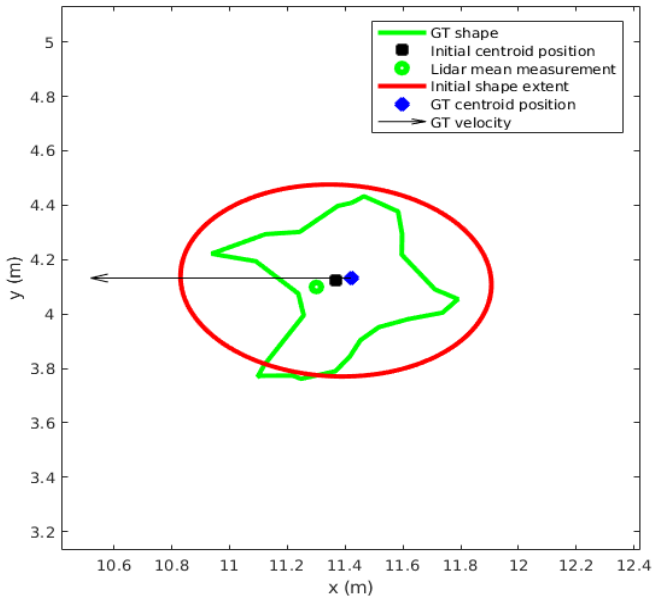


Figure 5-2: Topview of calculated initial state for pedestrian centroid position and shape extent, together with ground truth pedestrian shape extent at simulation timestep $k = 224$.

5-2 Creation of heading angle measurement

The extra measurement created from the association of pedestrian pose detections and Lidar position measurements is the heading angle of the pedestrian, denoted as a_k at each simulation timestep k. Subsequently, this measurement is used together with Lidar position measurements at the proposed measurement update step of the filter. For that purpose, the following assumption is made for pedestrian motion:

- Pedestrian heading direction is always perpendicular to the axis connecting associated shoulders pose detections in 2D world coordinates.

Hence, in a quite similar manner to initialization of ellipse orientation described in Section 5-1, heading angle measurement is created depending on the relation between x-y coordinates of right and left shoulder detections (see Equation 5-2). In more detail, the following cases unveil for each simulation timestep k:

1. If $\mathbf{z}_{pose,k}^{2^{(w)}} < \mathbf{z}_{pose,k}^{5^{(w)}}$, then $a_k \in [-90°, 0°]$ and (based on Equation 5-15) is calculated by:
$$a_k = \arcsin \frac{|\mathbf{y}_{pose,k}^{2} - \mathbf{y}_{pose,k}^{5}|}{d_{shoulders}} - 90° \tag{5-17}$$
2. If $\mathbf{z}_{pose,k}^{2^{(w)}} > \mathbf{z}_{pose,k}^{5^{(w)}}$, then $a_k \in [90°, 180°]$ and (based on Equation 5-15) is calculated by:
$$a_k = \arcsin \frac{|\mathbf{y}_{pose,k}^{2} - \mathbf{y}_{pose,k}^{5}|}{d_{shoulders}} + 90° \tag{5-18}$$

3. If $\mathbf{x}_{pose,k}^{2(w)} > \mathbf{x}_{pose,k}^{5(w)}$ and $\mathbf{y}_{pose,k}^{2(w)} < \mathbf{y}_{pose,k}^{5(w)}$, then $a_k \in [0°, 90°]$ and (based on Equation 5-16) is calculated by:

$$a_k = \arccos \frac{|\mathbf{y}_{pose,k}^{2} - \mathbf{y}_{pose,k}^{5}|}{d_{shoulders}} \tag{5-19}$$

4. If $\mathbf{x}_{pose,k}^{2(w)} < \mathbf{x}_{pose,k}^{5(w)}$ and $\mathbf{y}_{pose,k}^{2(w)} > \mathbf{y}_{pose,k}^{5(w)}$, then $a_k \in [-180°, -90°]$ and(based on Equation 5-16) is calculated by:

$$a_k = \arccos \frac{|\mathbf{y}_{pose,k}^{2} - \mathbf{y}_{pose,k}^{5}|}{d_{shoulders}} - 180° \tag{5-20}$$

5-3 Measurement Update Step using Pedestrian Pose Detections and Lidar Detections

5-3-1 State and Measurement Modeling

Similarly to the baseline filter presented in Chapter 3, the kinematic state of the pedestrian is represented again by the 2D centroid position vector $\mathbf{r}_k$ and the 2D centroid velocity vector $\dot{\mathbf{r}}_k$, as shown in Equations 3-1 - 3-2. In addition, the shape extent matrix $\mathbf{X}_k$ represents pedestrian extent. The relation between $\mathbf{X}_k$ and the ellipse parametric vector $\mathbf{s}_k$ is explained in detail in Section 3-5.

Concerning measurement availability, at each timestep k of the simulation, the Lidar scan returns a random number of n_k measurements, denoted by Equations 3-3 - 3-4. The relation between each position measurement and the kinematic state vector is given by:

$$\mathbf{y_1}_k^j = \mathbf{H_1} \boldsymbol{x}_k + \mathbf{w_1}_k^j \tag{5-21}$$

In detail, $\mathbf{H_1} = [\mathbf{I}_2, \mathbf{0}_2]$ is the matrix mapping kinematic states to position measurements, while $\mathbf{w_1}_k^j$ is additive noise to each position detection.

On top of that, a single heading angle measurement a_k for the pedestrian is available in each simulation timestep, after associating Lidar position and (camera obtained) human pose detections (see Section 5-2). The (non-linear) relation between each heading measurement and the kinematic state vector is given by:

$$\mathbf{y_2}_k = h_2(\mathbf{x}_k) = \tan^{-1}(\frac{\dot{\mathbf{r}}_{y,k}}{\dot{\mathbf{r}}_{x,k}}) \tag{5-22}$$

A schematic representation of the kinematic and elliptical shape extent parameters for the pedestrian, together with corresponding heading angle a is shown in Figure 5-3.

5-3-2 Sequential Measurement Update for Kinematic State

Firstly, let us denote that the prediction step of the proposed filter is exactly similar to that presented in Section 3-4 for the baseline filter. As a result, Assumptions 5-8

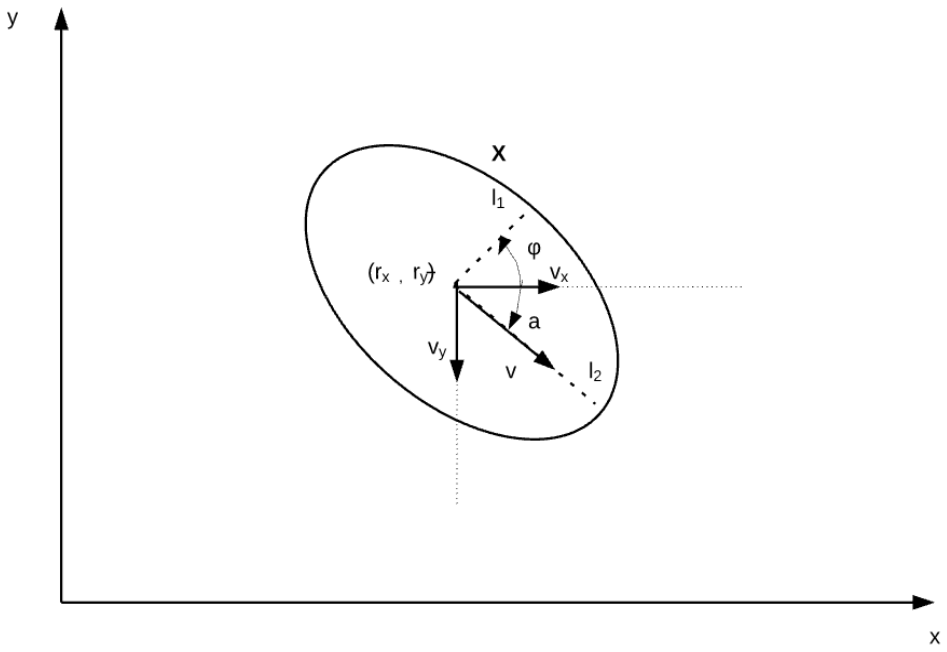


Figure 5-3: Schematic representation of the kinematic and elliptical shape extent parameters for the pedestrian, together with corresponding heading angle a.

are considered to be valid and predicted kinematic and shape extent state is given by Equations 3-27 - 3-30.

Moreover, a sequential measurement update step is proposed, where both Lidar position measurements and heading angle measurements for the pedestrian are incorporated to achieve joint estimation of the kinematic state. In the first step, the standard linear Kalman filtering update equations are employed, together with the mean 2D centroid position measurement. In the second step, an Extended Kalman Filter (EKF) is used to update the kinematic state by incorporating also the heading angle measurement.

To be more specific, let us denote the predicted kinematic state vector $\mathbf{x}_{k|k-1}$ and corresponding covariance matrix $\mathbf{P}_{k|k-1}$. Then, the first update step of the proposed filter is exactly similar to that of the baseline filter, as presented in Section 3-3. In more detail, Assumptions 1-3 are considered to be valid and the updated kinematic state after incorporation of Lidar position detections is given as:

$$\mathbf{x^1}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_{KF}(\overline{\mathbf{y_1}}_k - \mathbf{H_1}\mathbf{x}_{k|k-1}) \tag{5-23}$$

$$\mathbf{P^1}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_{KF}\mathbf{S_1}_{k|k-1}\mathbf{K}_{KF}^T \tag{5-24}$$

where

$$\mathbf{S_1}_{k|k-1} = \mathbf{H_1}\mathbf{P}_{k|k-1}\mathbf{H_1}^T + \frac{\mathbf{Y_1}_{k|k-1}}{n_k} \tag{5-25}$$

$$\mathbf{K}_{KF} = \mathbf{P}_{k|k-1}\mathbf{H_1}^T\mathbf{S_1}^{-1}_{k|k-1} \tag{5-26}$$

$$\mathbf{Y_1}_{k|k-1} = z\mathbf{X}_{k|k-1} + \mathbf{R_1} \tag{5-27}$$

is the approximation of the true innovation covariance, the Kalman gain and the predicted variance of a single measurement, respectively, for the first update step of the proposed filter.

In addition, the second update step of the proposed filter consists of EKF filtering update equations, with use of obtained pedestrian heading angle measurement $\mathbf{y_2}_k = a_k$, as follows:

$$\mathbf{x}_{k|k} = \mathbf{x^1}_{k|k} + \mathbf{K}_{EKF}(\mathbf{y_2}_k - h_2(\mathbf{x^1}_{k|k})) \tag{5-28}$$

$$\mathbf{P}_{k|k} = \mathbf{P^1}_{k|k} - \mathbf{K}_{EKF}\mathbf{S_2}_{k|k}\mathbf{K}_{EKF}^T \tag{5-29}$$

where

$$\mathbf{S_2}_{k|k} = \mathbf{H_2}\mathbf{P^1}_{k|k}{\mathbf{H_2}}^T + \mathbf{R_2} \tag{5-30}$$

$$\mathbf{K}_{EKF} = \mathbf{P^1}_{k|k}{\mathbf{H_2}}^T{\mathbf{S_2}}^{-1}_{k|k} \tag{5-31}$$

$$\mathbf{H}_2 = \left.\frac{\partial h_2}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}^1}} = \begin{bmatrix} 0 & 0 & \frac{-\dot{\mathbf{r}^1}_{y,k}}{\dot{\mathbf{r}^1}^2_{x,k}+\dot{\mathbf{r}^1}^2_{y,k}} & \frac{\dot{\mathbf{r}^1}_{x,k}}{\dot{\mathbf{r}^1}^2_{x,k}+\dot{\mathbf{r}^1}^2_{y,k}} \end{bmatrix} \tag{5-32}$$

is the approximation of the true innovation covariance, the Extended Kalman gain and the linearized matrix mapping kinematic states to heading angle measurement, for the second update step of the proposed filter. As a result, the pair $(\mathbf{x}_{k|k}, \mathbf{P}_{k|k})$ is the output of the proposed measurement update state.

Finally, let us also denote that the update step for the shape extent state is exactly similar to that presented in Section 3-3. As a result, Assumptions 1 and 4 are considered to be valid and updated shape extent parameters are given by Equations 3-22 - 3-26. In other words, the obtained heading angle measurement is only used in the kinematic state update step, while the procedure for shape extent update remains intact.

5-4 Performance Metrics

In EOT, joint estimation of kinematic and shape extent state attributes of objects of interest is performed. This is the case also for both tracking algorithms examined in this report, namely the Lidar-only RMM-based filter proposed by Feldmann [7] (see Chapter 3) and the Lidar and Camera based nonlinear filter presented in Chapter 5. At this point, let us denote that the examined tracking algorithms use an elliptic representation of the pedestrian, while corresponding ground truth shape extent (as described in Section 4-3) is an arbitrary shape. Due to this difference between the representation of estimated and true pedestrian shape, it is not possible to derive a single metric for joint evaluation of kinematic and shape extent tracking. On the contrary, it is possible to evaluate separately the accuracy of estimated kinematic and shape related parameters.

5-4-1 Performance Metrics for kinematic state of pedestrian

Let us denote as $\hat{\mathbf{x}}_k$ the estimated kinematic state of pedestrian at simulation timestep k, consisting of estimated 2D centroid position and velocity, respectively:

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{r}}_k \\ \hat{\dot{\mathbf{r}}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{r}}_{x,k} \\ \hat{\mathbf{r}}_{y,k} \\ \hat{\dot{\mathbf{r}}}_{x,k} \\ \hat{\dot{\mathbf{r}}}_{y,k} \end{bmatrix} \tag{5-33}$$

In addition, let us denote as $\mathbf{x}_k^{GT}$ the ground truth kinematic attributes of the pedestrian at simulation timestep k, consisting of true 2D centroid position and velocity, respectively:

$$\mathbf{x}_k^{gt} = \begin{bmatrix} \mathbf{r}_k^{gt} \\ \dot{\mathbf{r}}_k^{gt} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{x,k}^{gt} \\ \mathbf{r}_{y,k}^{gt} \\ \dot{\mathbf{r}}_{x,k}^{gt} \\ \dot{\mathbf{r}}_{y,k}^{gt} \end{bmatrix} \tag{5-34}$$

RMSE for pedestrian 2D position and velocity The Root Mean Squared Error (RMSE) metric is used to evaluate 2D pedestrian centroid position and 2D pedestrian velocity tracking, respectively. Firstly, the RMSE for 2D pedestrian centroid position is given by:

$$RMSE_{pos} = \sqrt{\frac{1}{2}[(\hat{\mathbf{r}}_{x,k} - \mathbf{r}_{x,k}^{gt})^2 + (\hat{\mathbf{r}}_{y,k} - \mathbf{r}_{y,k}^{gt})^2]} \tag{5-35}$$

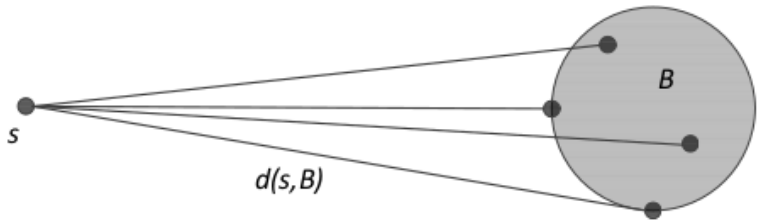
In a similar manner, the RMSE for 2D pedestrian centroid velocity is given by:

$$RMSE_{vel} = \sqrt{\frac{1}{2}[(\hat{\dot{\mathbf{r}}}_{x,k} - \dot{\mathbf{r}}_{x,k}^{gt})^2 + (\hat{\dot{\mathbf{r}}}_{y,k} - \dot{\mathbf{r}}_{y,k}^{gt})^2]} \tag{5-36}$$

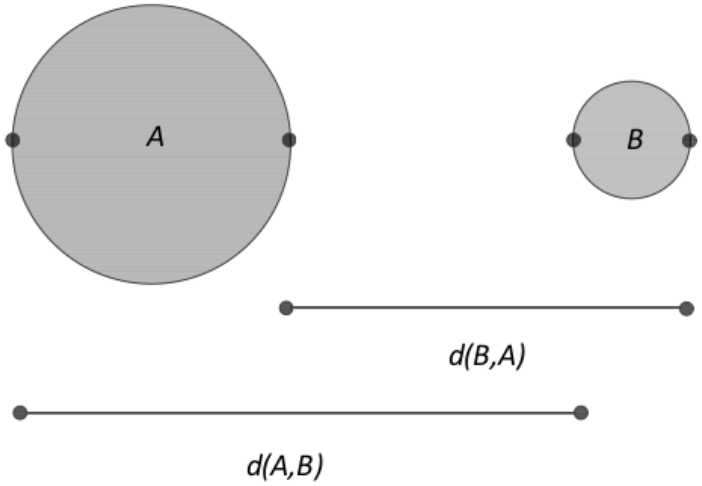
5-4-2 Performance Metrics for shape extent state of pedestrian

Let us denote as $S_{\hat{A}}(\hat{\mathbf{r}}_k, \hat{\mathbf{X}}_k)$ the estimated and as $S_A^{gt}(\mathbf{r}_k^{gt}, \mathbf{X}_k^{gt})$ the ground truth pedestrian shape extent, respectively. According to the aforementioned analysis, $S_{\hat{A}}$ is represented as an ellipse, while S_A^{gt} is an arbitrary shape. The modified Hausdorff distance is a suitable metric for comparison of two shapes with different representation. For that reason, it is proposed as a metric to evaluate estimation of shape extent for a pedestrian in 2D world coordinates (e.g.:x-y plane) in this study.

To begin with, Hausdorff distance metric is widely used to measure the similarity of two point sets. In [20], a modified version of the Hausdorff distance is proposed for shape estimation performance evaluation of two star-convex shapes, represented by the Random Hypersurface Model (RHM - see Section 2-1-4-2 and Figure 2-5). This metric, the so-called modified Hausdorff distance, is also used in this project to compare the estimated elliptic shape $S_{\hat{A}}$ with the ground truth arbitrary shape S_A^{gt}. Note that both $S_{\hat{A}}$ and S_A^{gt} are treated as point sets with boundaries corresponding to their extent.



(a) A representation of $d_E(\hat{a}, S_A^{gt})$ or $d_E(a^{gt}, S_{\hat{A}})$, respectively.



(b) A representation of $d(S_{\hat{A}}, S_A^{gt})$ or $d(S_A^{gt}, S_{\hat{A}})$, respectively.

Figure 5-4: An explanatory example of modified Hausdorff distance for two extended objects. Borrowed from [20].

.

Let us denote as $\hat{a} \in S_{\hat{A}}$ a point belonging to estimated shape extent and as $a^{gt} \in S_A^{gt}$ a point belonging to ground truth shape extent, respectively.

Then, the distance from $S_{\hat{A}}$ to S_A^{gt} is given by:

$$d(S_{\hat{A}}, S_A^{gt}) = \max_{\hat{a} \in S_{\hat{A}}} \{d_E(\hat{a}, S_A^{gt})\} \tag{5-37}$$

where

$$d_E(\hat{a}, S_A^{gt}) = \min_{a^{gt} \in S_A^{gt}} \{d_E(\hat{a}, a^{gt})\} \tag{5-38}$$

Similarly, the distance from S_A^{gt} to $S_{\hat{A}}$ is given by:

$$d(S_A^{gt}, S_{\hat{A}}) = \max_{a^{gt} \in S_A^{gt}} \{d_E(a^{gt}, S_{\hat{A}})\} \tag{5-39}$$

where

$$d_E(a^{gt}, S_{\hat{A}}) = \min_{\hat{a} \in S_{\hat{A}}} \{d_E(a^{gt}, \hat{a})\} \tag{5-40}$$

Finally, based on Equations 5-37,5-39 the modified Hausdorff distance for shape estimation performance evaluation is defined as:

$$d_H(S_{\hat{A}}, S_A^{gt}) = \max\{d(S_{\hat{A}}, S_A^{gt}), d(S_A^{gt}, S_{\hat{A}})\} \tag{5-41}$$

Chapter 6

Evaluation of Results

In the previous Chapters of this report, a detailed analysis of the employed tracking algorithms, as well as processing techniques of sensor data obtained from PreScan software is provided. In detail, the Random Matrix Model (RMM) for extended tracking of a single object of interest is briefly discussed in Section 2-1-5. Subsequently, a tracking algorithm using Lidar position measurements, an RMM representation and a linear measurement update step for a single object of interest is presented in detail in Chapter 3. In more detail, measurement and state modeling approaches and the state initialization procedure followed is included in this Chapter. Note that this filter is used as the baseline tracking algorithm, for evaluation purposes in terms of this study.

Moreover, our proposed tracking algorithm, which makes use of Lidar-obtained position data and human pose detections to estimate the kinematics and shape extent of a single pedestrian is presented in detail in Chapter 5. Sensor data is obtained from a designed scenario in PreScan software and its processing steps are explained in detail in Chapter 4. Based on the association method presented in Section 4-5, it is possible to map obtained pedestrian pose detections from the 2D image plane coordinates frame to 3D world coordinates frame. Associated pedestrian pose detections are then employed to perform an alternative state intialization method (Section 5-1) and create an extra measurement for the pedestrian, regarding its heading angle (Section 5-2). Subsequently, a nonlinear sequential measurement update step for pedestrian kinematic state is presented (Section 5-3), making use of both corresponding Lidar position and heading angle measurements.

In this Chapter, evaluation of obtained results takes place. Firstly, a comparison is made between the two presented initialization methods, with respect to ground truth state variables, in order to get a first indication of whether calculated initial state is close to pedestrian true state. Secondly, the accuracy of created heading angle measurement is investigated with respect to true pedestrian heading, for a single run of the selected PreScan scenario. In fact, it is shown that the association method between Lidar and human pose detections affects crucially the accuracy of both initial state and created heading measurement with respect to their ground truth values.

Moreover, the effect of each state initialization approach to the performance of considered tracking algorithms is investigated. To begin with, performance of the baseline filter is evaluated while using a different state initialization method. Evaluation takes place in two distinct phases:

- In the first evaluation phase, a Monte Carlo simulation with 200 runs takes place for the designed scenario, starting from the same initial simulation timestep k_1. As a result, in each run only one simulation parameter changes, namely the additive Gaussian zero-mean noise in Lidar position measurements.
- In the second evaluation phase, a Monte Carlo simulation with 1000 runs takes place for the designed scenario. In each run, two simulation parameters change, namely the additive Gaussian zero-mean noise in Lidar position measurements and the initial simulation timestep k_1.

In each phase, the performance metrics presented in Section 5-4 are used for evaluation, namely the RMSE for estimated pedestrian centroid position and velocity, as well as the modified Hausdorff distance for estimated pedestrian shape extent.

Finally, the baseline filter, using the conventional state initialization approach, and the proposed filter, using the proposed state initialization method are compared. Again, evaluation takes place in the same two distinct phases mentioned above, while the aforementioned performance metrics are employed for that purpose.

6-1 Evaluation of proposed state initialization approach

In the previous Sections of this report, two alternative approaches to define pedestrian initial state are discussed. To begin with, the most widely used approach in previous studies is presented in Section 3-6. Concerning kinematic state $\mathbf{x}_k$, initial centroid position is set equal to the mean of the first Lidar-obtained position measurement set, while initial centroid velocity is set equal to zero. On top of that, concerning shape extent state, ellipse orientation is set equal to zero, while ellipse axes lengths are set equal to randomly selected constant values. Based on these values, the initial shape extent SPD matrix $\mathbf{X}_k$ is defined, according to Equations 3-31 - 3-34.

On the other hand, an alternative approach is proposed in Section 5-1, where Lidar points corresponding to pedestrian shoulders and ankles detections in x-y plane (2D world coordinates frame) are employed to calculate pedestrian initial state. In short, the mean of shoulders detection is used to initialize centroid position, while a two-point differentiation of the exact same point for the first two simulation timesteps defines initial centroid velocity. Moreover, the distances of shoulders and ankles detections are used to derive values for initial ellipse semi-axes, respectively. Finally, the angle between x-axis and the first considered ellipse axis in the counter-clockwise direction is calculated to derive the initial ellipse orientation value.

6-1-1 Comparison of calculated initial state with ground truth values

Before incorporating the aforementioned state initialization approaches to each tracking algorithm, a first indication of the accuracy of state initialization method based on association of Lidar position measurements and human pose detections (as described in Section 5-1) can be examined by comparing the derived initial values with corresponding ground truth data. In practice, any of the simulation timesteps $k \in [1, 340]$ of the examined scenario (see Section 4-1) can be selected as the initial simulation timestep k_{init}. In other words, pedestrian state initialization could take place at any $k_{init} \in [1, 340]$ and then the filter would start running at timestep k_{init+1}. As a result, initial state is calculated for all simulation timesteps in the following ways:

- The conventional state initialization method, using only Lidar-obtained position measurements, as described in Section 3-6.
- The proposed state initialization method using human body pose detections together with Lidar-obtained position measurements, as described in Section 5-1.

For each of the aforementioned cases, a single run of the simulation scenario takes place and the following quantities are calculated for comparison of initial state derived by each calculation method and ground truth data:

- RMSE for initial pedestrian 2D centroid position derived by the conventional method (Section 3-6) and the human pose based method (Section 5-1), respectively.
- RMSE for initial pedestrian 2D centroid velocity derived by the conventional method (Section 3-6) and the human pose based method (Section 5-1), respectively.
- Absolute error between initial ellipse orientation derived by the human pose based method (Section 5-1) and ground truth ellipse orientation, obtained from PreScan software.

Note that it is not possible to compare initial and true ellipse semi-axes lengths. The reason is that considered pedestrian ground truth shape is not an ellipse, thus no semi-axes lengths are assigned to it.

Evaluation of initial pedestrian 2D centroid position To begin with, the calculated initial centroid position for the pedestrian, based on obtained mean Lidar position measurement is given by Equation 3-40, while the calculated same quantity, based on the mean of associated shoulders detections in 2D world coordinates is given by Equation 5-3. A comparison between RMSE of initial pedestrian 2D position for the two considered state initialization approaches and ground truth centroid position is demonstrated in Figure 6-1.

It is shown that initial 2D position based on Lidar position measurements (conventional initialization approach - blue line) presents an error around $0.05 - 0.15m$. The error exists due to the fact that, in each simulation timestep, some parts of pedestrian body are not detected by the ego-vehicle Lidar sensor. In fact, this is a limitation of the

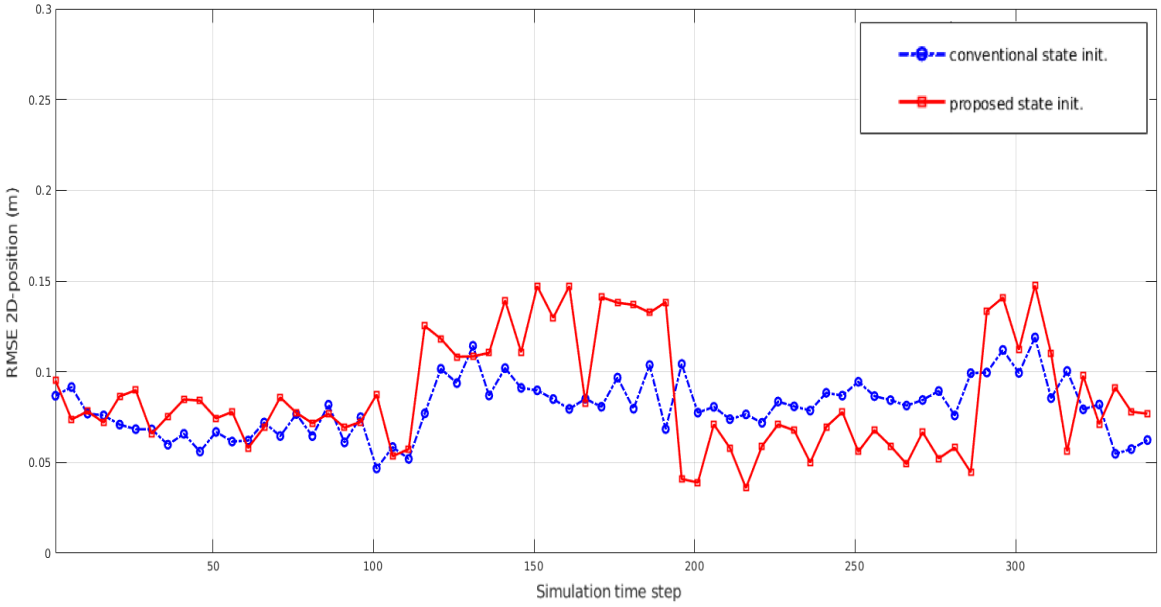


Figure 6-1: RMSE of initial pedestrian 2D position, calculated based on the conventional (blue) and proposed (red) state initialization approaches, respectively, with respect to ground truth pedestrian centroid position, for all simulation timesteps of the selected scenario.

Random Matrix Model representation, which assumes an almost uniform distribution of obtained position measurements over object extent (Equations 3-5 - 3-7 and Figure 3-1c). While assumption might be valid for ideal simulations designed in softwares similar to MATLAB, this in not the case for real automotive applications. For example, at simulation timestep $k = 21$, when the pedestrian is walking away from the ego-vehicle, only the rear part of pedestrian body is detected by the Lidar sensor, while the front body part is not detected, as shown in Figure 4-2. For that reason, there is always an error between the mean Lidar position measurement and the true 2D pedestrian centroid, as also shown in an explanatory topview in Figure 6-2.

Nevertheless, use of associated human shoulder detections in x-y plane (proposed pose-based initialization approach, see Equation 5-3) results in similar or slightly better initial state value for centroid position in most simulation timesteps. The only pedestrian path sectors where proposed pose-based initialization approach performs worse than the conventional one are at $k \in [122, 191]$ and $k \in [290, 310]$, due to sensor to object geometry.

To be more specific, let us consider pedestrian obtained data at simulation timestep $k = 161$. Then, pedestrian true heading is perpendicular to Lidar sensor pointing direction, meaning that only the left part of human body is captured by the Lidar sensor, as shown in Figure 6-3a. Even though OpenPose library manages to detect the right shoulder of the pedestrian in 2D image pixel coordinates, the associated Lidar point (calculated as explained in Section 4-5) lies in the left part of pedestrian body in 3D

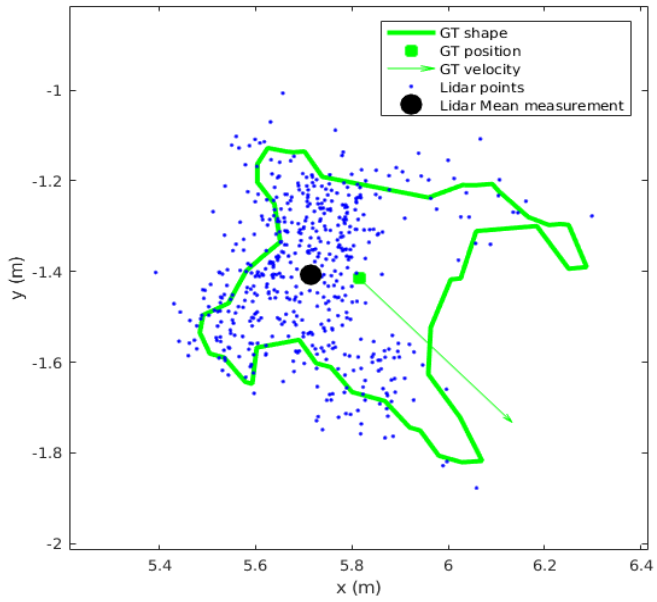
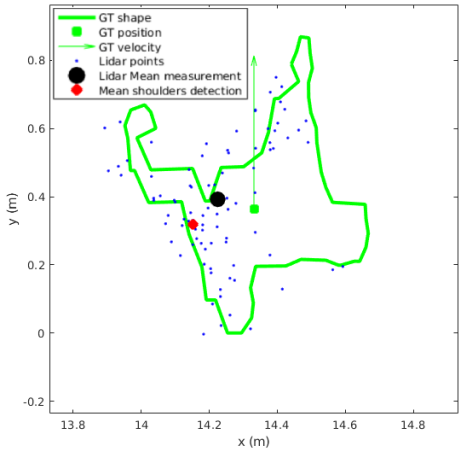


Figure 6-2: Obtained Lidar points, corresponding mean measurement and ground truth information for pedestrian at simulation timestep $k = 21$.

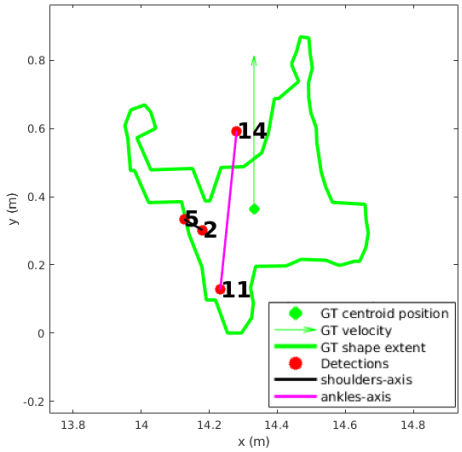
world coordinates frame. As a result, association between right shoulder detection in image pixel coordinates and obtained Lidar point cloud in world coordinates fails, as shown in Figure 6-3b. This leads to an inaccurate position value corresponding to mean shoulders detections in 2D world coordinates during this time interval (see again Figure 6-3a).

Evaluation of initial pedestrian 2D centroid velocity As already mentioned in this report, the conventional state initialization approach considers a zero initial velocity in x-y plane, given by Equation 3-41, while the calculated initial velocity based on a two-point differentiation of the mean of associated shoulder detections for the first two simulation timesteps is defined by Equation 5-6. Note that the aggregated ground truth velocity of the pedestrian is always equal to 1 $\frac{m}{s}$. A comparison between RMSE of initial pedestrian 2D velocity for the two considered state initialization approaches and ground truth centroid position is demonstrated in Figure 6-4.

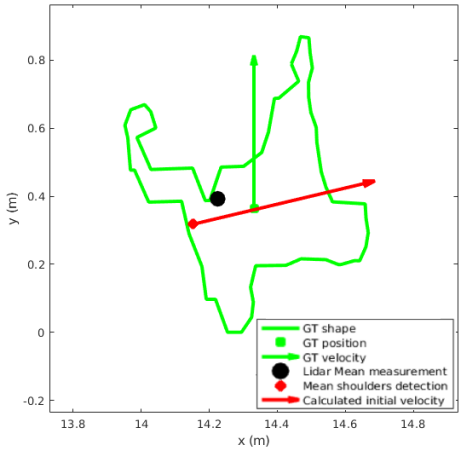
It is shown that initial velocity RMSE calculated by two-point differentiation of mean shoulder detections in consecutive simulation timesteps presents many fluctuations. This is expected, because such calculations, enabling differentiation over a short time interval are generating results prone to noise. On top of that, as already mentioned in the previous paragraph, depending on sensor to pedestrian geometry, associated mean shoulders detection may be far from the actual pedestrian centroid position (see Figure 6-3a). However, this was the only alternative for pedestrian velocity initialization, regarding available sensor data. A visualization of the big difference between ground truth and calculated initial velocity for the pedestrian at simulation timestep $k = 161$ is included



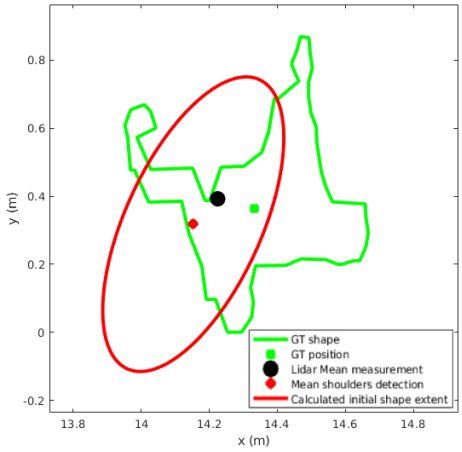
(a) Lidar points and their mean, together with mean shoulders detection.



(b) Shoulders and ankles associated detections, together with corresponding axes.



(c) Calculated and ground truth initial centroid velocity.



(d) Calculated and ground truth initial shape extent.

Figure 6-3: Topview of pedestrian sensor data and calculated initial state at simulation timestep $k = 161$ in 2D world coordinates.

.

in Figure 6-3c. In the following Sections, it is shown that, despite the noisy outcome, the proposed initialization approach results in a decreased error for estimated pedestrian velocity, in comparison to setting a zero initial velocity.

Evaluation of initial ellipse parameters Concerning initial ellipse parameters, the widely used (conventional) approach considers a zero initial ellipse orientation, as well as random constant selected values for initial ellipse semi-axes lengths (see Equations 3-42 - 3-44). On the other hand, based on association of obtained Lidar points and pedes-

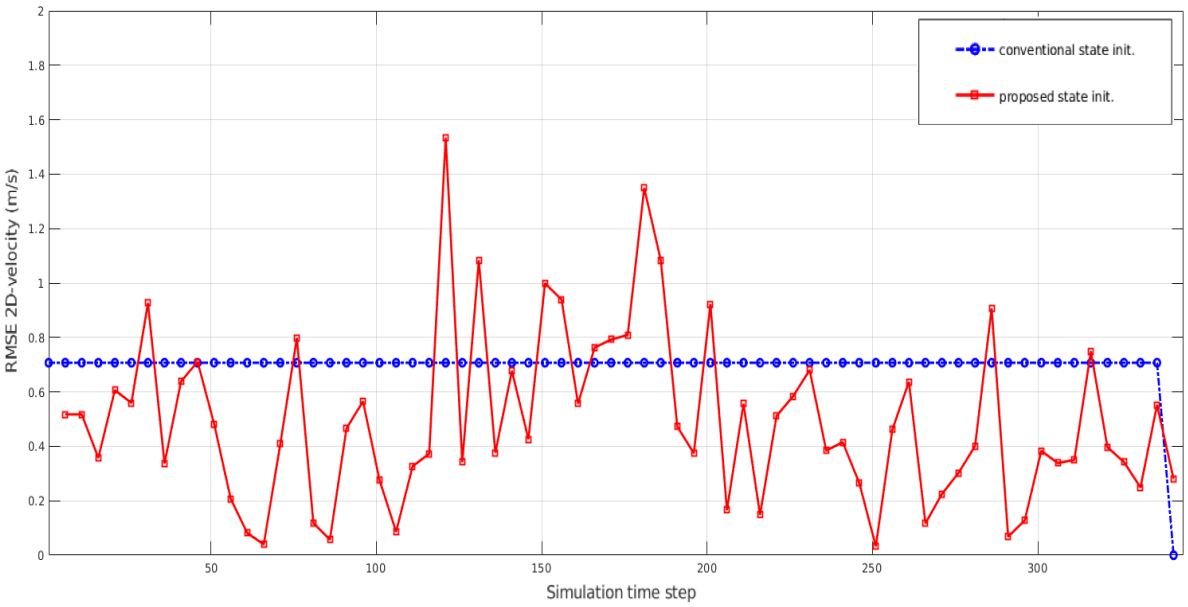


Figure 6-4: RMSE of initial pedestrian 2D velocity, calculated based on the conventional (blue) and proposed (red) state initialization approaches, respectively, with respect to ground truth pedestrian centroid velocity, for all simulation timesteps of the selected scenario.

trian detections of ankles and shoulders, Equations 5-11 - 5-16 are employed to initialize ellipse orientation and lengths in our proposed approach presented in this report.

Since the considered ground truth shape is not an ellipse, it is not possible to compare the calculated initial semi-axes lengths for the pedestrian with any corresponding true values. Nevertheless, it is possible to derive the true ellipse orientation, via the true pedestrian heading angle, which is provided from PreScan software for the designed simulation scenario. The calculated initial pedestrian ellipse orientation versus ground truth pedestrian ellipse orientation is demonstrated in Figure 6-5 for the designed simulation scenario. It is shown that calculated initial ellipse orientation is quite close to the true value at time intervals when the pedestrian is nearby the Lidar sensor, such as $k \in [0, 60]$ and $k \in [251, 281]$. On the contrary, strong fluctuations are shown at $k \in [122, 250]$, especially when true ellipse orientation takes its boundary values, namely 0° and 90°, respectively. In practice, due to the definition of ellipse orientation (see Section 3-5), the boundary values for ellipse orientation (0° and 90°, respectively) refer to the exact same situation for the considered shape extent representation. As a result, even a slight turn of pedestrian body can create such sudden changes on its value when being close to boundary values (e.g.: from values close to 0° to values close to 90° and vice versa).

An example is shown in Figure 6-3d, where the calculated initial ellipse for the pedestrian is presented at $k = 161$. Note that at this simulation timestep true ellipse orientation is equal to 0°. Since calculation of initial ellipse orientation is entirely based on shoulders

pose detections, the aforementioned inaccuracy of the proposed association method followed to represent shoulders detections in 2D world coordinates results in considerable error with respect to true ellipse orientation for some simulation timesteps. For instance, at $k = 161$ calculated initial ellipse orientation is equal to 77°.

Concerning calculated initial ellipse axes lengths, the short distance between shoulder detections in x-y plane at $k = 161$, results in a smaller calculated value for the minor axis in comparison to the expected. For that reason, the calculated initial ellipse shown in Figure 6-3d does not cover a big part of pedestrian shape extent.

In a few words, inaccuracies of the proposed association method for pedestrian shoulder pose detections, presented in Section 4-5, especially in time instances when only one side of pedestrian body is detected by Lidar sensor may result in significant errors for calculated initial state, with respect to true values. Thus, for these simulation timesteps, the conventional initialization approach is more accurate than the proposed one. Nevertheless, the fact that the calculated initial state based on associated pedestrian pose detections is closer to ground truth values for most simulation timesteps is promising, in the sense that it could lead to increased tracking accuracy for filter output in comparison to the conventional initialization case. In order to verify the validity of this statement, the baseline filter, described in Chapter 3, is tested for both initialization approaches in Section 6-1-2.

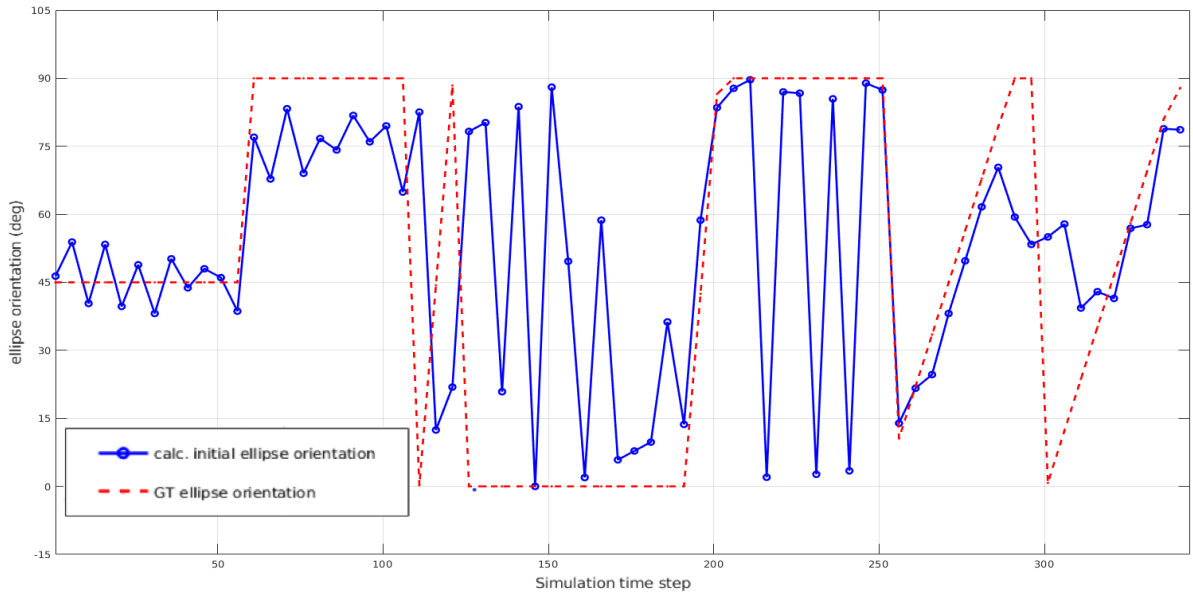
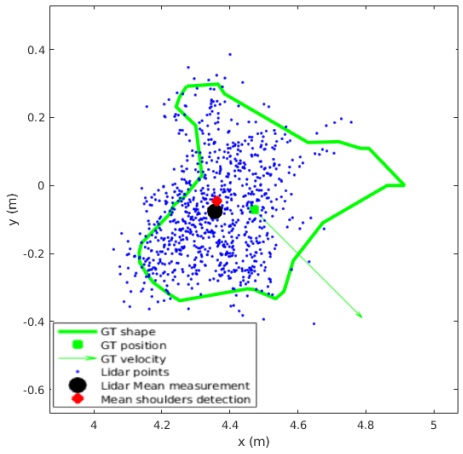


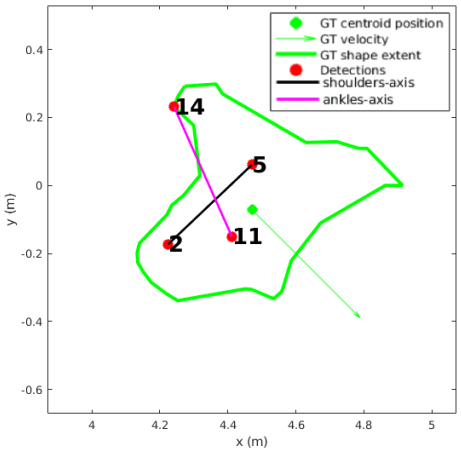
Figure 6-5: Calculated initial pedestrian ellipse orientation (blue) versus ground truth pedestrian ellipse orientation (red), for all simulation timesteps of the selected scenario.

6-1-2 Effect of proposed state initialization approach in baseline tracking algorithm

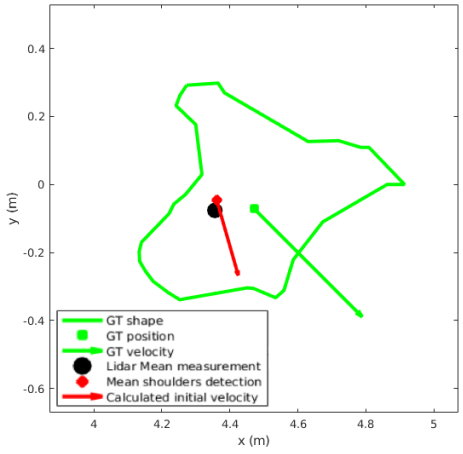
At this point, both the conventional and proposed state initialization approaches are employed for the baseline filter, described in Chapter 3, and a comparison of their performance is made. Employed performance metrics are the RMSE for pedestrian centroid position and velocity, respectively, for kinematics and mean Hausdorff distance for shape extent state. All three metrics are explained in detail in Section 5-4. As already mentioned, evaluation takes place in two distinct phases.



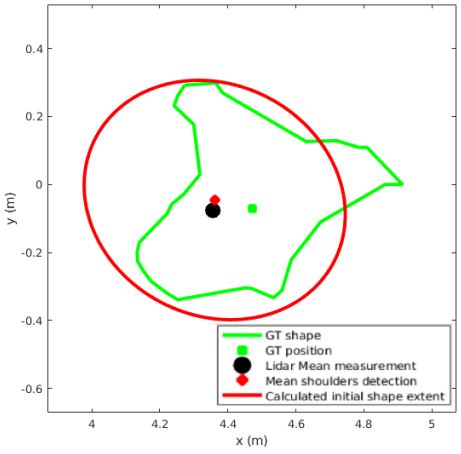
(a) Lidar points and their mean, together with mean shoulders detection.



(b) Shoulders and ankles associated detections, together with corresponding axes.



(c) Calculated and ground truth initial centroid velocity.



(d) Calculated and ground truth initial shape extent.

Figure 6-6: Topview of pedestrian sensor data and calculated initial state at simulation timestep $k = 2$ in 2D world coordinates.

Concerning the first evaluation phase, a Monte Carlo simulation of 200 runs takes place

for the designed scenario. In all runs, the same initial simulation scenario, $k = 2$, is selected. As a result, in each run only the additive Gaussian zero mean white noise in Lidar 2D position measurements changes. In fact, a covariance matrix $C_{meas} = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}$ is defined for measurement noise. A topview of pedestrian sensor data at $k = 2$, together with corresponding pedestrian body pose detections and calculated initial state variables for each initialization approach, is demonstrated in Figure 6-6 for an explanatory run. In more detail, initial state values derived from each approach are summarized in Table 6-1 for the same explanatory run. Note that covariance for initial position is set based on spread of obtained Lidar measurements $\overline{\mathbf{Y}}_k$. Also, kinematic covariance for the proposed initialization approach is smaller than the conventional, meaning that more trust is given for our proposed approach.

	Position	Velocity	Ellipse orientation	Ellipse lengths	Kinematic covariance
Conventional init.	(4.36,-0.08) m	(0,0) $\frac{m}{s}$	0°	(0.3,0.3) m	diag($\frac{1}{4}$ $\overline{\mathbf{Y}}_k$,$\mathbf{I}_2$)
Proposed init.	(4.36,-0.05) m	(0.14,-0.49) $\frac{m}{s}$	52.93°	(0.18,0.3) m	diag($\frac{1}{8}$ $\overline{\mathbf{Y}}_k$,$\frac{1}{2}$ $\mathbf{I}_2$)

Table 6-1: Derived initial state values for the conventional and proposed initialization approach at simulation timestep $k = 2$ of selected scenario.

Evaluation of corresponding performance metrics for the Monte Carlo simulation of 200 runs of the baseline filter is demonstrated in Figures 6-7 - 6-9. In each run, the additive zero-mean Gaussian noise in Lidar position measurements changes. It is shown that the error of the baseline filter in estimated position and shape extent is almost the same for both initialization approaches. There is a very slight difference only for the very first simulation timesteps. This behavior is expected, because selected initial centroid position at $k = 2$ is very close for both cases, as also shown for the considered explanatory run in Table 6-1 and Figure 6-6.

Nevertheless, this is not the case for estimated velocity, since the proposed approach results in a significantly reduced initial error (e.g.: 0.43 $\frac{m}{s}$), in comparison to the conventional initialization approach (e.g.: 0.7 $\frac{m}{s}$), in the first simulation timestep. This is a decreased RMSE of **38.5**% for the initial simulation timestep, while convergence to an error close to zero is also faster in the latter case. Let us also denote that the picks in velocity RMSE take place when the pedestrian makes a maneuver. Since a nearly constant velocity motion model is employed in the baseline filter, maneuvers are not accounted for and velocity error increases at corresponding time instances.

However, starting all Monte Carlo runs from the exact same initial simulation timestep might not be a fair choice concerning performance evaluation of different state initialization approaches. For example, it is possible that the selected initial simulation time instance might favor performance evaluation by producing results like those demonstrated in Figures 6-7 - 6-9. For that reason, a second evaluation phase takes place, where a Monte Carlo simulation of 1000 runs is employed for the designed scenario. In each run, two simulation parameters change, namely the additive Gaussian zero-mean noise in Lidar position measurements and the initial simulation timestep k_1. In other words, any simulation timestep $k > 1$ can be selected to start running the filter. In fact, selection of initial simulation timestep is random and represented by a uniform

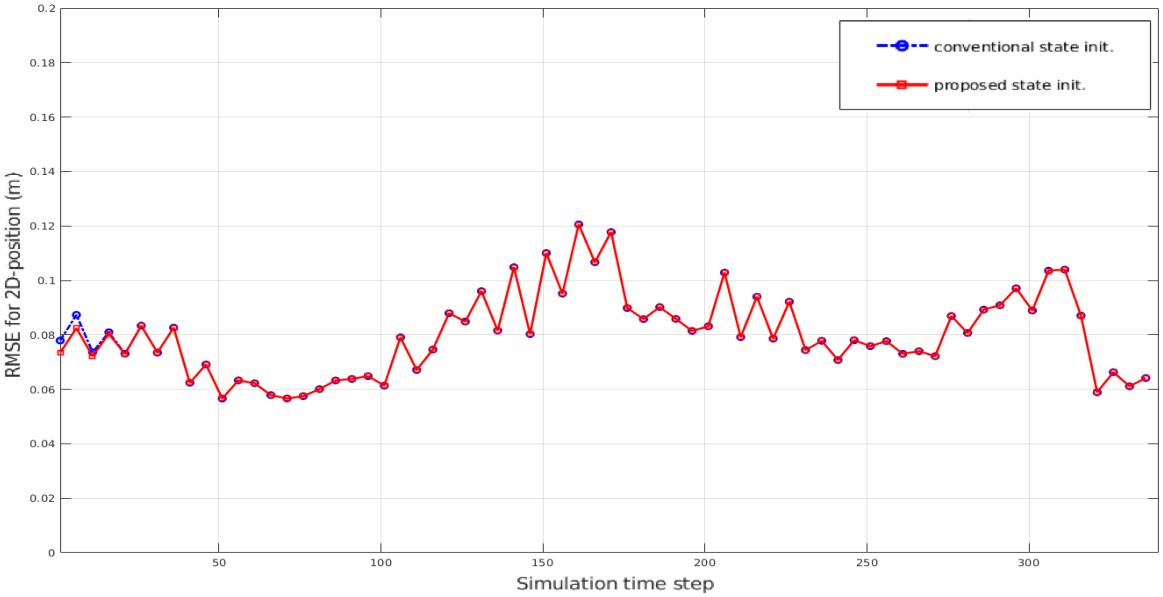


Figure 6-7: RMSE for estimated pedestrian centroid position using the baseline filter and the conventional or proposed initialization approach, respectively, in a Monte Carlo simulation of 200 runs starting at simulation timestep $k = 2$.

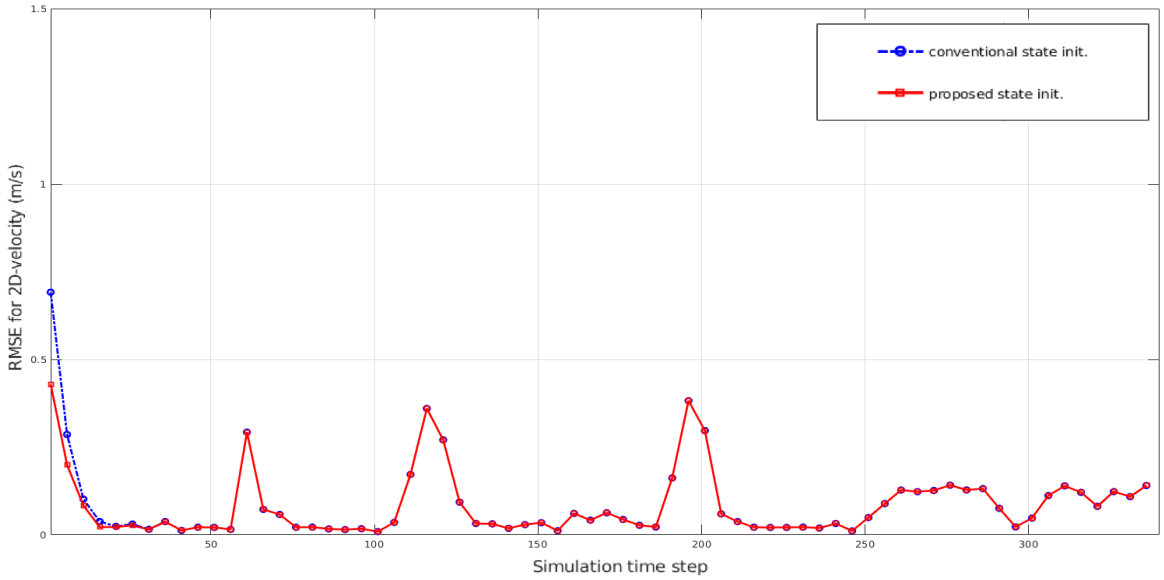


Figure 6-8: RMSE for estimated pedestrian centroid velocity using the baseline filter and the conventional or proposed initialization approach, respectively, in a Monte Carlo simulation of 200 runs starting at simulation timestep $k = 2$.

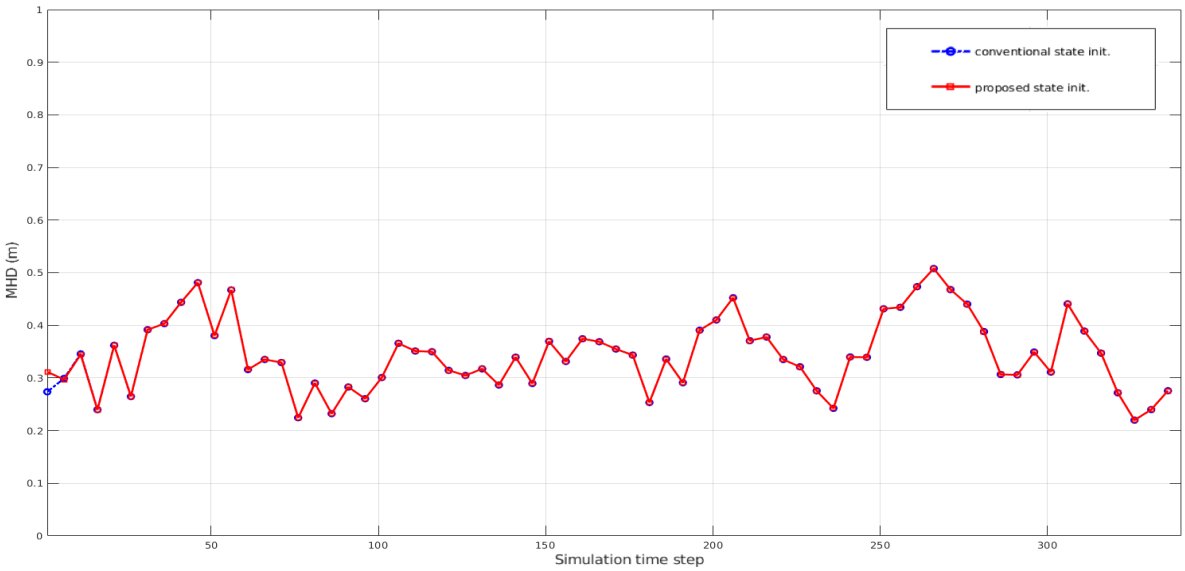


Figure 6-9: Modified Hausdorff distance for estimated pedestrian shape extent using the baseline filter and the conventional or proposed initialization approach, respectively, in a Monte Carlo simulation of 200 runs starting at simulation timestep $k = 2$.

distribution. Moreover, all runs have a predefined duration of 100 simulation timesteps, so that corresponding performance metrics can be calculated in an efficient way. Again, a covariance matrix $C_{meas} = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}$ is defined for measurement noise in each run.

Evaluation of corresponding performance metrics for the Monte Carlo simulation of 1000 runs of the baseline filter with random initial simulation timestep is depicted in Figures 6-10 - 6-12. While position RMSE for both cases is identical and almost constant, velocity RMSE presents again a reduced error (e.g.: 0.45 $\frac{m}{s}$), in comparison to the conventional initialization approach (e.g.: 0.68 $\frac{m}{s}$) for the first simulation timestep. Hence, a decreased velocity RMSE of **33.3**% is achieved.

As a result, it can be concluded that our proposed initialization method does not have a major effect on tracking accuracy of pedestrian centroid position and kinematic shape extent, but results in a significant reduction of velocity error at the first simulation timesteps of the selected scenario, independently of the choice made for the starting simulation timestep.

6-2 Evaluation of proposed tracking algorithm

Except from the two different approaches for pedestrian state initialization, two tracking algorithms with different measurement update step are also discussed in the previous Sections of this report. In more detail, the baseline filter, presented in Chapter 3, makes

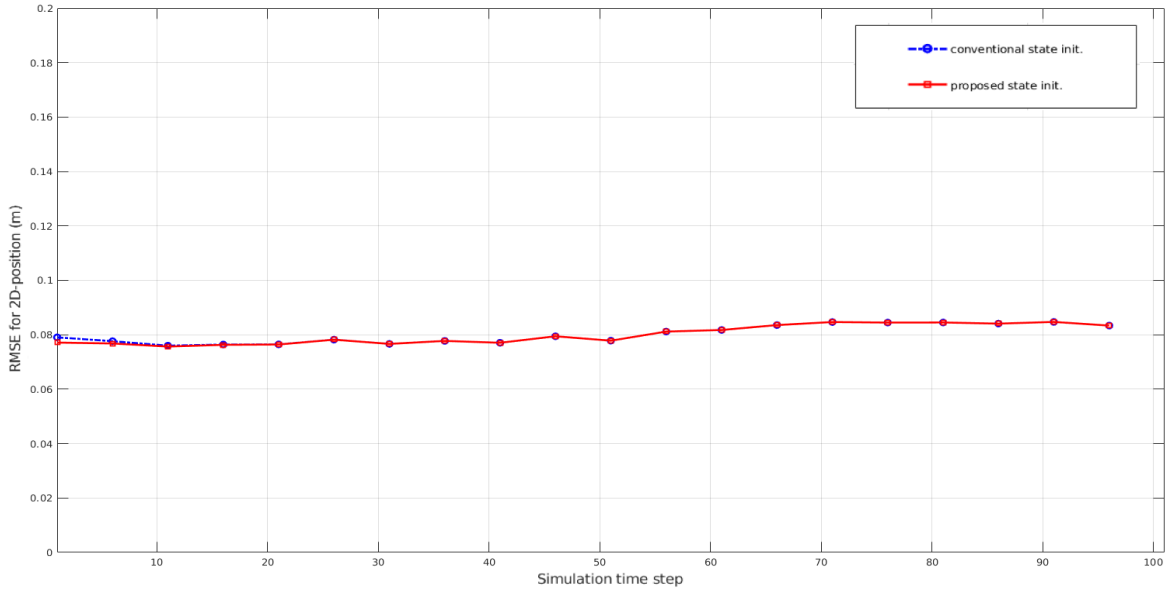


Figure 6-10: RMSE for estimated pedestrian centroid position using the baseline filter and the conventional or proposed initialization approach, respectively, in a Monte Carlo simulation of 1000 runs. Each run starts at a randomly selected simulation timestep.

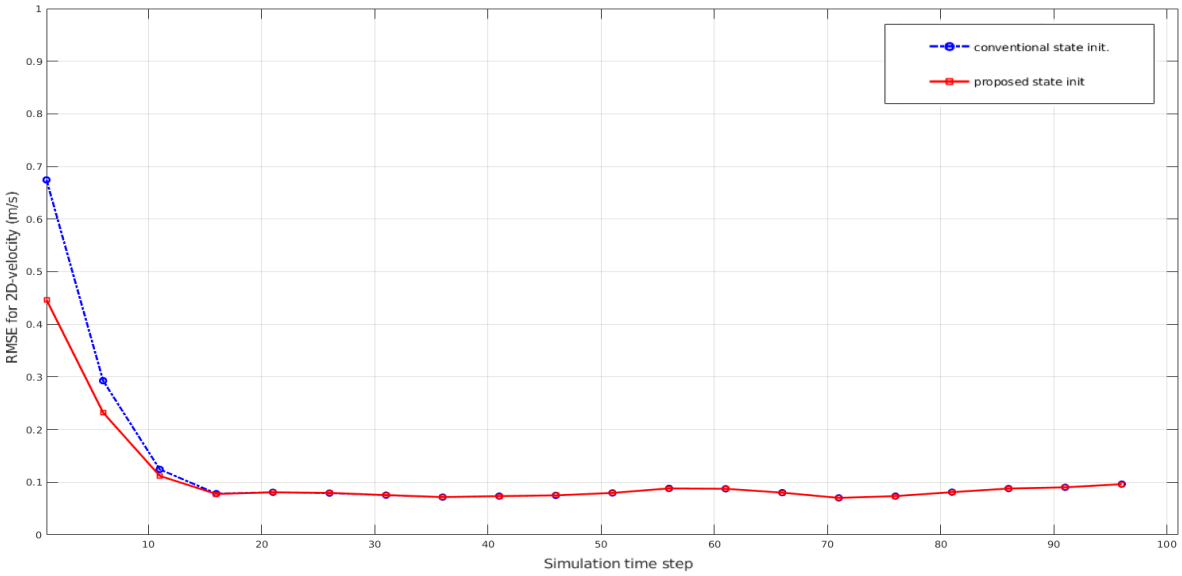


Figure 6-11: RMSE for estimated pedestrian centroid velocity using the baseline filter and the conventional or proposed initialization approach, respectively, in a Monte Carlo simulation of 1000 runs. Each run starts at a randomly selected simulation timestep.

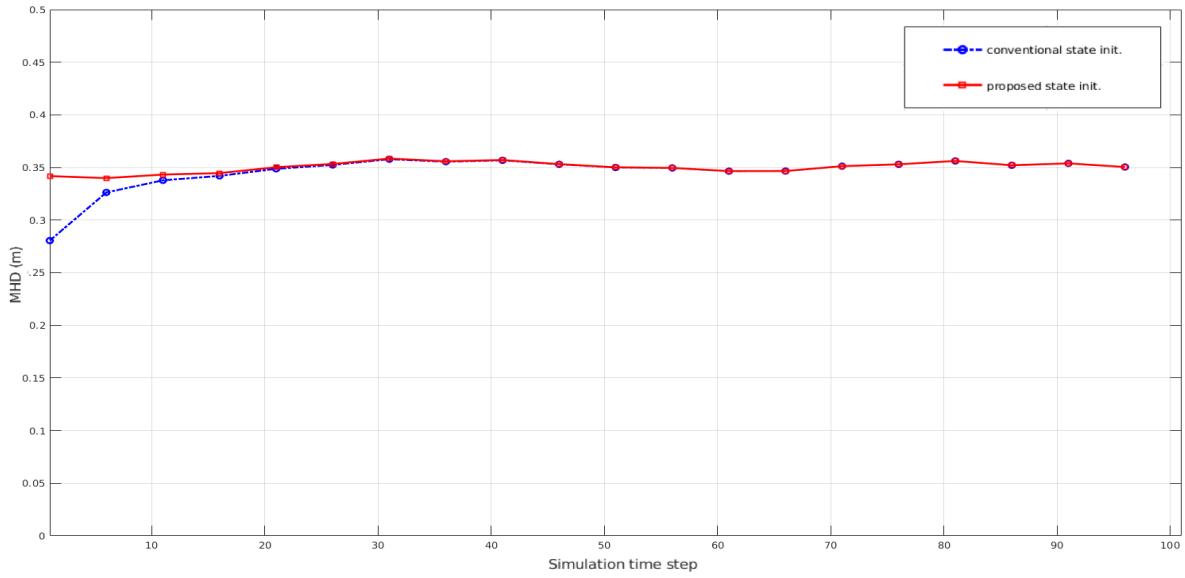


Figure 6-12: Modified Hausdorff distance for estimated pedestrian shape extent using the baseline filter and the conventional or proposed initialization approach, respectively, in a Monte Carlo simulation of 1000 runs. Each run starts at a randomly selected simulation timestep.

use of Lidar position measurements, which are incorporated into a linear measurement update step to jointly estimate kinematic and shape extent state parameters. On the other hand, our proposed tracking algorithm using Lidar position data and calculated pedestrian heading angle (based also on human body pose detections) in a nonlinear sequential measurement update step, is presented in Chapter 5.

In this Section, created pedestrian heading angle measurement (according to Section 5-2) is demonstrated for a single simulation run and its accuracy is discussed. Subsequently, a comparison of the baseline filter, using the conventional state initialization approach, and the proposed filter, using the proposed state initialization approach, takes place. Evaluation of each algorithm is possible via use of the aforementioned performance metrics, presented in Section 5-4.

6-2-1 Evaluation of created heading angle measurement

To begin with, pedestrian heading angle is defined as the angle between x-axis of world coordinates frame and pedestrian velocity vector, as depicted in Figure 5-3, and takes values within $[-180°, 180°]$. In Section 5-2, an approach to create a measurement for pedestrian heading angle is presented, based on its associated shoulders detections in 2D world coordinates. Depending on the exact position of each shoulder detection point in x-y plane, one of Equations 5-17 - 5-20 is selected for that purpose. Subsequently, calculated heading angle measurement is incorporated in the second (nonlinear) step of the sequential measurement update for our proposed filter, presented in Section 5-3-2.

An example of created heading angle measurement of the pedestrian is demonstrated in Figure 6-13, for a single run of the designed simulation scenario. It is shown that the obtained measurement is very noisy and presents an important difference in comparison to ground truth pedestrian heading. To be more specific, in time instances when the pedestrian is close to ego-vehicle sensors, such as $k \in [0, 60]$, $k \in [251, 281]$ and $k \in [320, 340]$, heading measurement is slightly bumpy but still close to its true value. Nevertheless, in cases when only one side of pedestrian body is visible by Lidar sensor, such as $k \in [122, 191]$ and $k \in [280, 320]$, fluctuations in heading measurement are stronger. Finally, it can been observed that very sharp fluctuations exist at $k \in [201, 250]$, where true heading angle takes its boundary value, namely 180°. In practice, the boundary values for pedestrian heading angle ($-180°$ and 180°, respectively) correspond to the exact same situation for pedestrian motion. As a result, even a slight turn of pedestrian body or a slight miscalculation during sensor data processing can create such steep variations on heading angle when it is so close to boundary values (e.g. from values close to 180° to values close to $-180°$ and vice versa).

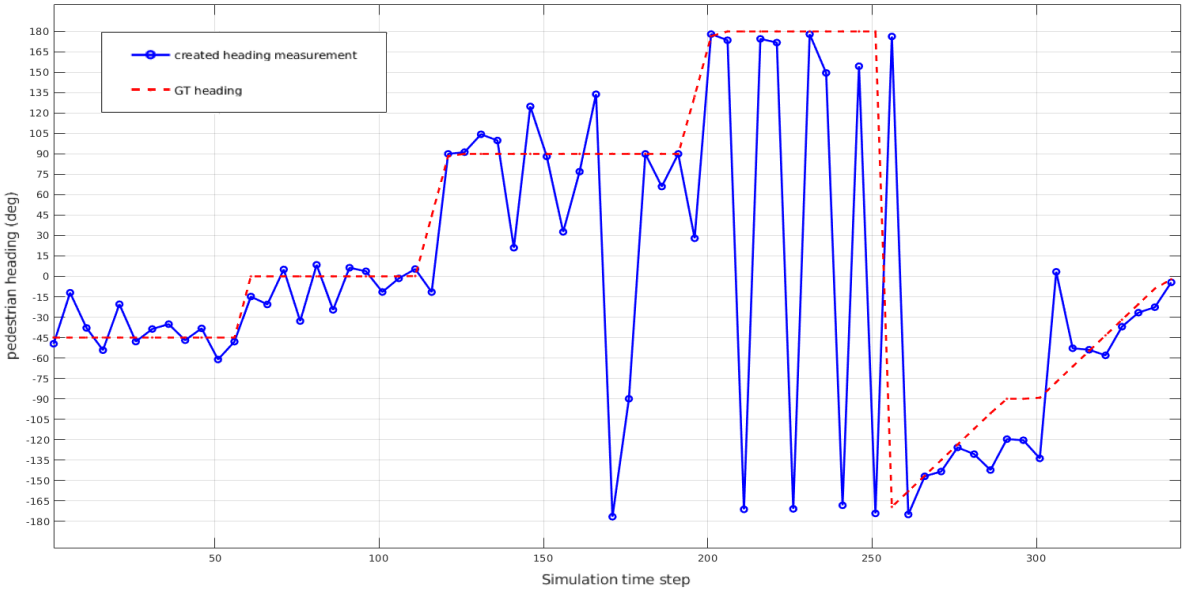
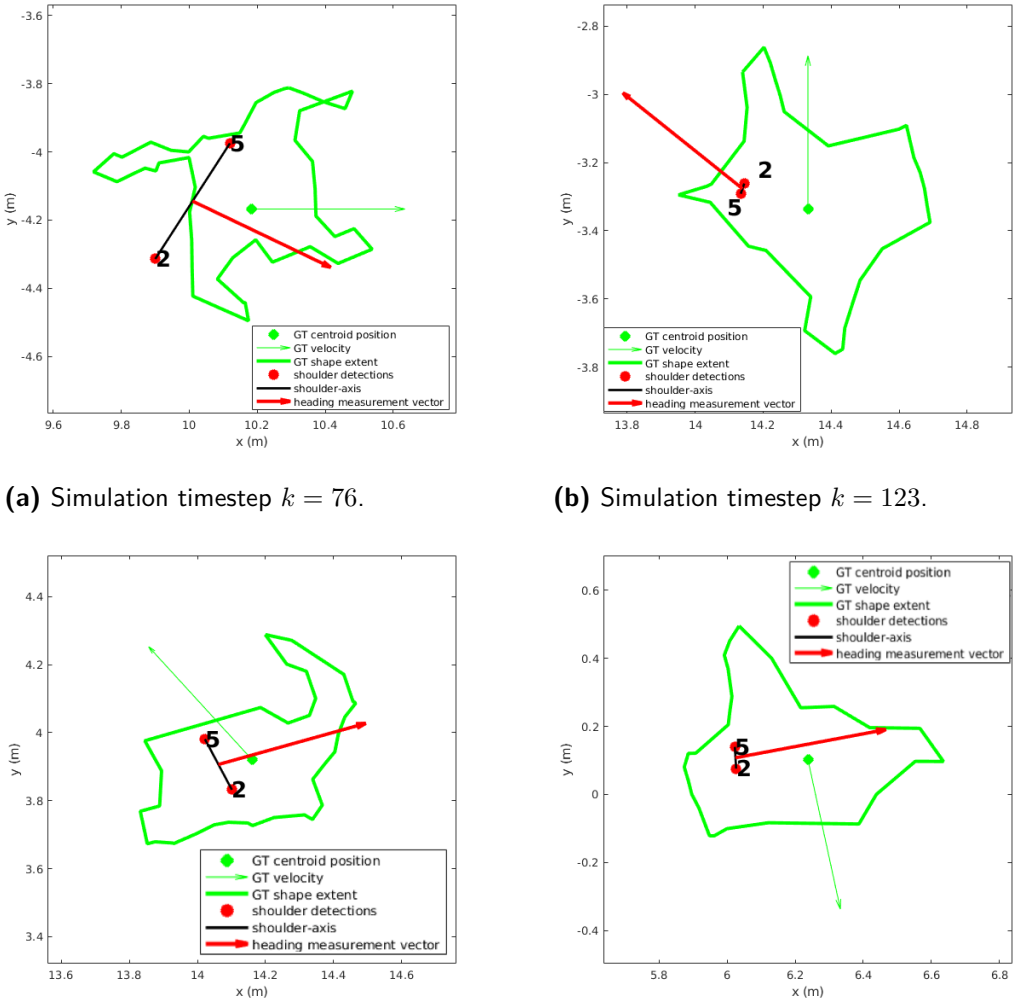


Figure 6-13: Calculated pedestrian heading angle measurement for a single simulation run versus ground truth pedestrian heading angle.

The reason explaining the significant difference between calculated and true pedestrian heading, as well as the observed steep fluctuations of its value, is (again) the inaccuracies of the proposed association method between obtained Lidar position measurements and shoulder pose detections. To be more specific, according to the considered assumption for pedestrian motion mentioned in Section 5-2, calculated heading direction is always considered to be perpendicular to its shoulder detections axis. As a result, firstly the pedestrian shoulder axis is derived in x-y plane and then corresponding heading angle measurement is calculated accordingly for each simulation timestep. Thus, it is clear



(a) Simulation timestep $k = 76$.

(b) Simulation timestep $k = 123$.

(c) Simulation timestep $k = 196$.

(d) Simulation timestep $k = 306$.

Figure 6-14: Topview of pedestrian shoulder detections axis, calculated heading measurement vector and true heading, represented by ground truth velocity vector, for a single simulation run.

that potential errors in the derived position of associated shoulder detections in 2D world coordinates affect directly the calculated value of heading angle measurement. A topview of pedestrian associated shoulder detections in 2D world coordinates, together with calculated heading measurement vector for four selected simulation timesteps, is presented in Figure 6-14. Note that heading angle measurement is defined as the angle between x-axis and demonstrated heading measurement vector for each case. The effect of associated shoulder detections for creation of this measurement for the pedestrian, as well as the accuracy of calculated heading angle with respect to true pedestrian heading (represented by corresponding ground truth velocity vector) is depicted for each selected case. Values of ground truth and calculated heading angle for the same simulation time

instances are included also in Table 6-2.

	$k = 76$	$k = 123$	$k = 196$	$k = 306$
Ground truth heading angle	$0°$	$90°$	$132°$	$-77.76°$
Measurement heading angle	$-32.88°$	$160.9°$	$28°$	$3.33°$

Table 6-2: Values of calculated and ground truth pedestrian heading angle for selected timesteps of a single simulation run.

6-2-2 Comparison of baseline and proposed tracking algorithms

At this point, a comparison takes place between the two tracking algorithms discussed in this study. To begin with, the conventional state initialization approach (see Section 3-6) is employed for the baseline filter, described in detail in Chapter 3. In its linear measurement update step for kinematic state estimation, the mean measurement of obtained Lidar points in 2D world coordinates is used.

On the other hand, the proposed state initialization approach, based on associated pedestrian pose detections in 2D world coordinates (see Section 5-1), is used for the proposed tracking algorithm, which is described in detail in Chapter 5. The sequential measurement update step of the proposed filter for kinematic state estimation consists of two steps: a linear step, using the mean measurement of obtained Lidar measurements (exactly similar to baseline filter) and a nonlinear step, using the created pedestrian heading angle measurement (see Sections 6-2-1 and 5-2).

Note that the update step for shape extent state estimation is exactly similar for both tracking algorithms. As a result, the only difference between the filters is the incorporation of calculated heading angle measurement in the second, nonlinear update step of the proposed tracking algorithm. According to Equations 5-22 and 5-28 - 5-32, the obtained heading measurement is linked directly only to pedestrian velocity state variables. As a result, significant differences between estimated position and shape extent state are not expected for examined filters. In practice, we are interested to investigate whether the available heading measurement together with the nonlinear extension of update step for kinematics estimation can directly improve accuracy of estimated velocity, especially during maneuvers. In case of success, it is possible that position and shape extent tracking accuracy can be improved indirectly.

Again, employed performance metrics are the RMSE for pedestrian centroid position and velocity, respectively, for kinematics and mean Hausdorff distance for shape extent state. All three metrics are explained in detail in Section 5-4. Similarly to state initialization case, evaluation takes place in two distinct phases.

Concerning the first evaluation phase, a Monte Carlo simulation of 200 runs takes place for the designed scenario. In all runs, the same initial simulation scenario, $k = 2$, is selected. As a result, in each run only the additive Gaussian zero mean white noise in Lidar 2D position measurements changes. Again, a covariance matrix $C_{meas} = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}$ is defined for measurement noise. A topview of pedestrian sensor data at $k = 2$, together

with corresponding pedestrian body pose detections and calculated initial state variables for each initialization approach, is demonstrated in Figure 6-6 for an explanatory run. In addition, initial state values derived from each approach are summarized in Table 6-1 for the same explanatory run. Note that covariance for initial position is set based on spread of obtained Lidar measurements $\overline{\mathbf{Y}}_k$. Also, kinematic covariance for the proposed initialization approach is smaller than the conventional, meaning that more trust is given for our proposed state initialization approach, adopted by our proposed filter. Finally, an example of pedestrian heading angle measurement, plus corresponding pedestrian topview instances for different simulation timesteps of an explanatory run are depicted in Figures 6-13 - 6-14, respectively. Of course, this heading measurement presents differences in each Monte Carlo run, because it is affected by Lidar sensor noise in obtained x-y measurements.

Evaluation of corresponding performance metrics for the Monte Carlo simulation of 200 runs of the baseline (blue) and proposed filter (red) is demonstrated in Figures 6-15 - 6-17. It is shown that a big error is present in estimated velocity RMSE (Figure 6-16). This error is caused due to the noisy calculated heading angle measurement incorporated in the nonlinear update step of the proposed filter (Figure 6-13). In more detail, velocity error is small at $k \in [0, 110]$, when heading measurement presents only relatively small fluctuations around its true value. On the other hand, velocity error increases rapidly for $k > 110$, at the time when pedestrian has performed a maneuver and sensor to human body geometry is such that association of shoulder detections and Lidar points fails. This can be observed also in Figure 6-14b, where a significant difference of 70° exists between true and calculated heading. At $k \in [121, 191]$, when only the left part of pedestrian body is visible, velocity error remains high, due to the remaining fluctuations in heading measurement. Subsequently, velocity tracking performance becomes even worse, due to the huge error (around 100°) of calculated heading angle, with respect to true pedestrian heading, at simulation timestep $k = 196$, depicted in Figure 6-14c, and remains in this level at $k \in [201, 251]$, when rapid fluctuations take place between boundary values calculated heading measurement. Finally, velocity RMSE decreases rapidly at $k > 280$, when the pedestrian is approaching the ego-vehicle sensors.

In a few words, our proposed approach for creation of heading angle measurement, similarly to state initialization, depends on the accuracy of the association method employed to map pedestrian shoulder detections from 2D image pixel coordinates frame to 3D world coordinates frame. Since obtained Lidar point cloud is not uniformly spread over the pedestrian in practical automotive applications (similar to the examined scenario) and no depth information is available for obtained pose detections in image plane coordinates, accuracy of proposed association method depends highly on sensor to object geometry. For instance, in situations where a side part of pedestrian body is not detected by ego-vehicle Lidar sensor, the associated point for corresponding shoulder detection in world coordinates is not accurately derived, in comparison to its expected position based on ground truth pedestrian shape. Then , inaccuracies in association method output have a major effect on calculated heading angle, incorporated as input to the measurement update step of the proposed filter, resulting in decreased tracking performance concerning estimated pedestrian velocity.

Nevertheless, initial velocity error at simulation timestep $k = 2$ decreases in the proposed

tracking algorithm, due to corresponding state initialization approach employed. In more detail, velocity RMSE equals 0.69 $\frac{m}{s}$ for the baseline filter and 0.41 $\frac{m}{s}$ for the proposed filter, respectively. Hence, this is a decreased velocity RMSE of $\mathbf{40.5}\%$, however it is not clear whether pedestrian to sensor geometry at the selected initial simulation timestep $k = 2$ favors this result. This is investigated in the reminder of this section, where results of the second evaluation phase are presented.

Concerning estimated centroid position, performance of both tracking algorithms is quite similar, despite the huge error in pedestrian velocity estimation, as shown in Figure 6-15. For both filters, centroid position RMSE takes its higher values for simulation timesteps when a side of pedestrian body does not generate Lidar position measurements, because the associated mean shoulder detection point in 2D world coordinates frame is crucially affected then. Moreover, position error for the proposed filter is slightly higher in comparison to the baseline algorithm for simulation time instances when velocity error reaches its highest values. The reason is that the effect of obtained mean Lidar measurement of the pedestrian is stronger for estimated centroid position, in comparison to the effect of heading angle measurement, according to Equations 5-22 and 5-28 - 5-32.

Finally, estimated shape extent is not affected by the incorporation of heading angle measurement in the proposed filter. This behavior is expected, since no changes are proposed in this study for estimation of pedestrian shape extent. In fact, the representation of pedestrian shape extent state by a 2x2 matrix is a major limitation of the Random Matrix Model approach. Instead of directly estimating the orientation and ellipse lengths for the object of interest, filters employing the RMM-based representation

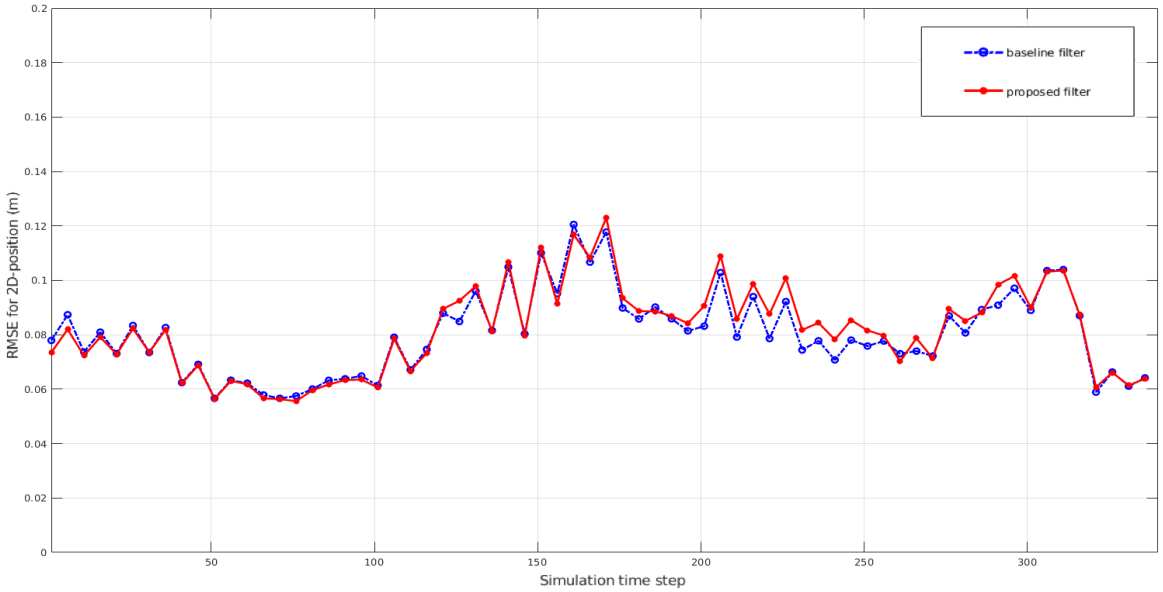


Figure 6-15: RMSE for estimated pedestrian centroid position using the baseline and the proposed tracking algorithm, respectively, in a Monte Carlo simulation of 200 runs starting at simulation timestep $k = 2$.

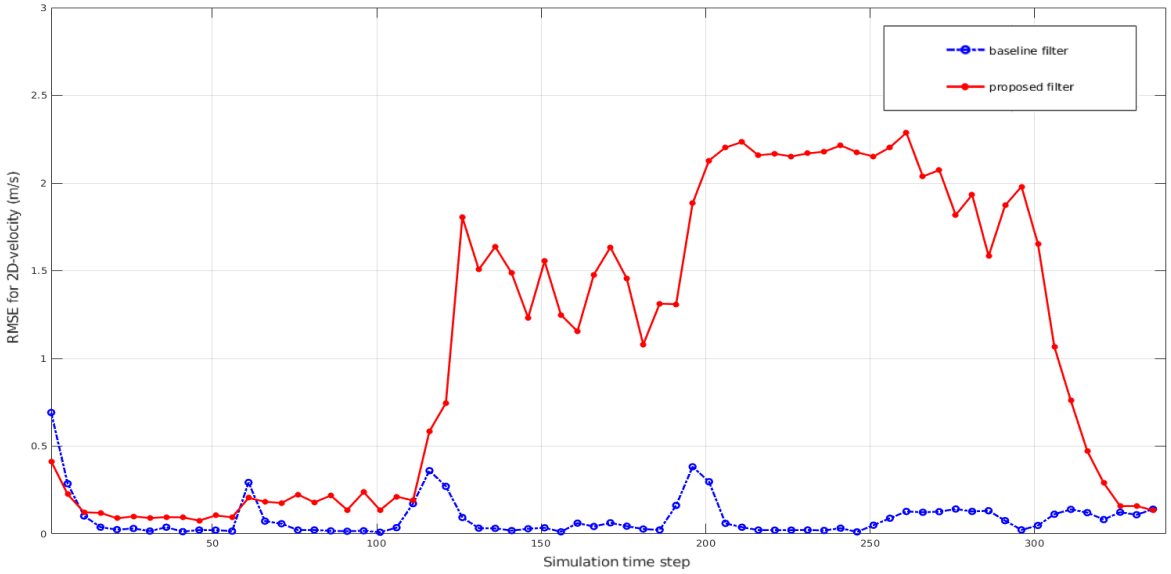


Figure 6-16: RMSE for estimated pedestrian centroid velocity using the baseline and the proposed tracking algorithm, respectively, in a Monte Carlo simulation of 200 runs starting at simulation timestep $k = 2$.

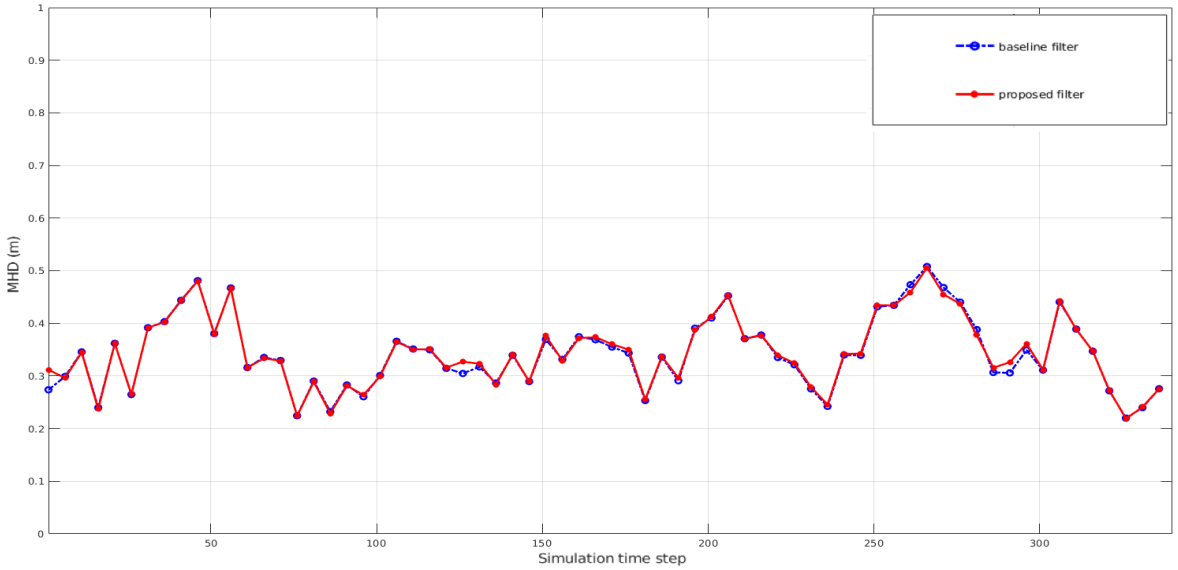


Figure 6-17: Modified Hausdorff distance for estimated pedestrian shape extent using the baseline and the proposed tracking algorithm, respectively, in a Monte Carlo simulation of 200 runs starting at simulation timestep $k = 2$.

provide an estimate for corresponding 2x2 shape extent matrix $\mathbf{X}_k$. Then, parameters of the estimated ellipse are calculated based on Equations 3-35 - 3-39. As a result, it is not possible to incorporate any sensor information concerning pedestrian heading in the update step of corresponding shape extent matrix $\mathbf{X}_k$, as long as this representation is selected for pedestrian state variables.

Finally, a second evaluation phase takes place for the two tracking algorithms, where a Monte Carlo simulation. In each run, two simulation parameters change, namely the additive Gaussian zero-mean noise in Lidar position measurements and the initial simulation timestep k_1. In other words, any simulation timestep $k > 1$ can be selected to start running the filter. In fact, selection of initial simulation timestep is random and represented by a uniform distribution. Moreover, all runs have a predefined duration of 100 simulation timesteps, so that corresponding performance metrics can be calculated in an efficient way. Again, a covariance matrix $C_{meas} = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}$ is defined for measurement noise in each run.

Evaluation of corresponding performance metrics for the Monte Carlo simulation of 1000 runs of the baseline and proposed tracking algorithm, respectively, with random initial simulation timestep is depicted in Figures 6-18 - 6-20. It is shown that inaccurate calculation of pedestrian heading angle measurement results in a significant error for estimated velocity, as expected. Moreover, estimated velocity RMSE at the first simulation timestep is very close for the two filters, even though the state initialization approach using associated shoulder detections in x-y plane is employed for the proposed

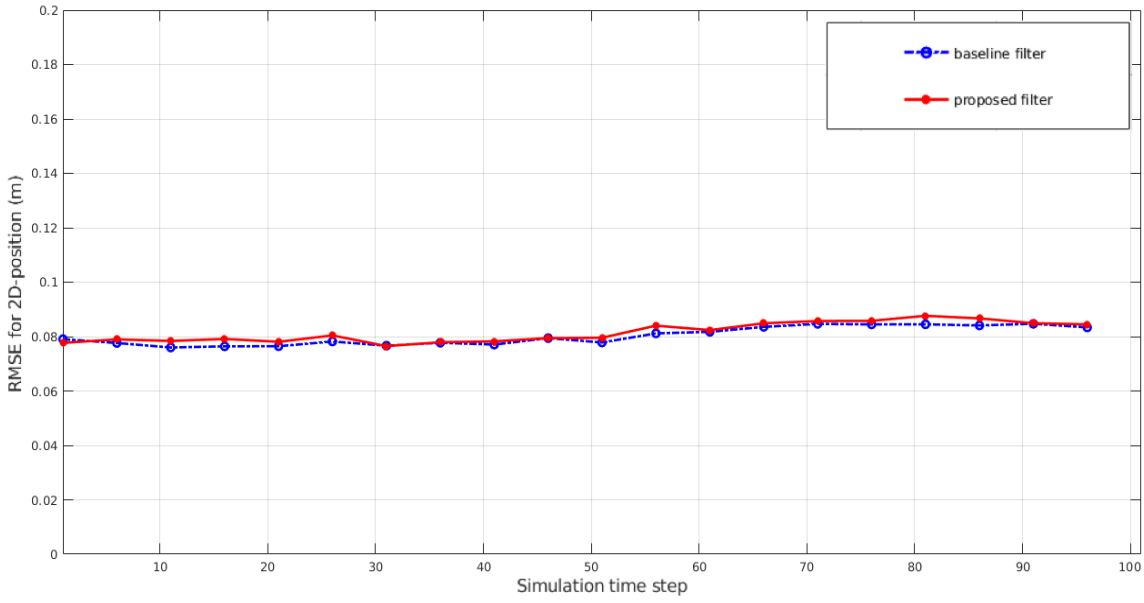


Figure 6-18: RMSE for estimated pedestrian centroid position using the baseline and proposed tracking algorithm, respectively, in a Monte Carlo simulation of 1000 runs. Each run starts at a randomly selected simulation timestep.

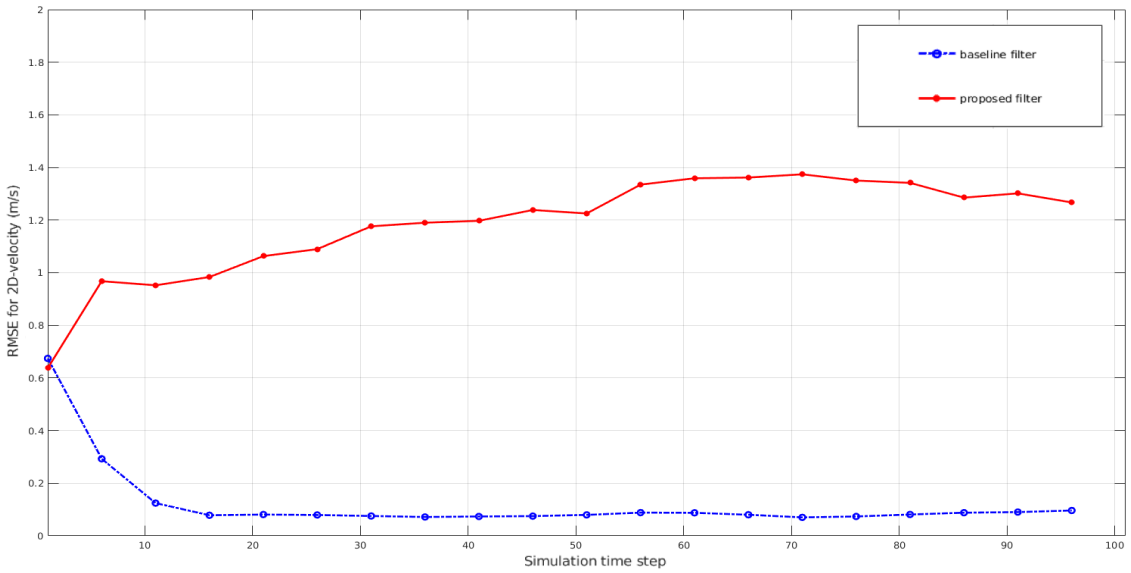


Figure 6-19: RMSE for estimated pedestrian centroid velocity using the baseline and proposed tracking algorithm, respectively, in a Monte Carlo simulation of 1000 runs. Each run starts at a randomly selected simulation timestep.

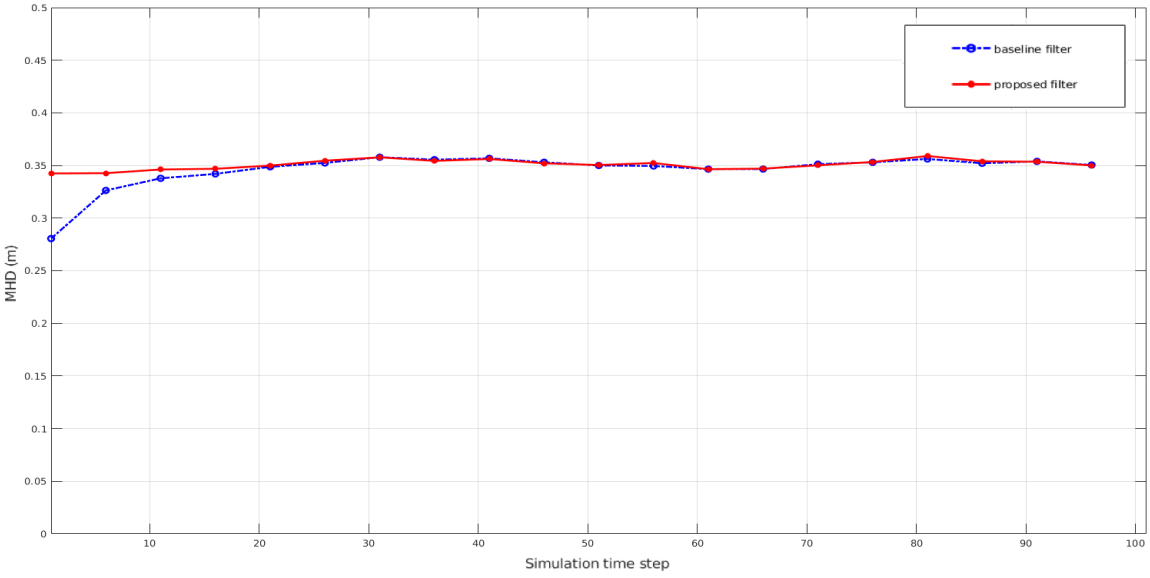


Figure 6-20: Modified Hausdorff distance for estimated pedestrian shape extent using the baseline and proposed tracking algorithm, respectively, in a Monte Carlo simulation of 1000 runs. Each run starts at a randomly selected simulation timestep.

filter. In fact, velocity error is only decreased by **6**% in the latter case. As a result, it can be concluded that the significant decrease of velocity RMSE (**40.5**%) shown at the first timesteps of results presented in Figure 6-17 was favored by the selection of $k = 2$ as initial simulation timestep for the examined scenario. On top of that RMSE for estimated pedestrian centroid position, as well as modified Hausdorff distance for estimated shape extent are almost identical and constant for the two examined tracking algorithms.

Chapter 7

Conclusions and Recommendations

7-1 Conclusions

In Chapter 1 of this report, it is mentioned that the problem addressed in this M.Sc. thesis is extended tracking of a single pedestrian during a real time scenario. In more detail, the goal of this study is to implement an EOT algorithm, within the Bayesian tracking framework, using data from different sensor modalities, in order to achieve accurate estimation of kinematic and shape extent attributes of a single pedestrian.

Created ground truth pedestrian shape extent and corresponding performance metric for EOT algorithms To begin with, an important limitation of previous studies concerning extended VRUs tracking in real automotive applications is the lack of performance evaluation for estimated pedestrian shape extent. Since ground truth data for pedestrian shape is not available in previous studies, as well as in benchmarks found online (e.g.: KITTI, MOT), performance metrics tailored for EOT algorithms cannot be applied for evaluation. In this study, an attempt is made to create an arbitrary ground truth shape for the pedestrian, by combining obtained position detections from multiple Lidar sensors, employed in a simulation scenario designed in PreScan software. As a result, the considered ground truth pedestrian shape extent can be compared to corresponding estimated shape extents in each simulation timestep. The modified Hausdorff distance is a performance metric tailored for comparison of two arbitrary shapes and thus is used for evaluation of pedestrian shape extent tracking in this study.

Advantages and disadvantages of proposed association method for Lidar and human pose detections data Moreover, in terms of this project, two different types of sensor data are available for the pedestrian in each simulation timestep. In short, multiple Lidar obtained position detections are available in 3D world coordinates frame, while mono camera obtained pedestrian body parts pose detections in 2D image pixel coordinates

frame are also provided in each simulation timestep. Due to the unavailability of depth information for obtained pedestrian pose detections, a method is employed in this study to map pose detections of shoulders and ankles from 2D image pixel coordinates to 3D world coordinates, by associating them to obtained Lidar measurements. Nevertheless, the proposed association method has limited accuracy in some simulation timesteps, depending on sensor to object geometry. To be more specific, RMM representation assumes a uniform distribution of Lidar obtained position measurements over object extent for the pedestrian. The same assumption is also considered for the aforementioned association method in our simulation scenario. However, this assumption is not valid in real automotive applications, since there are always parts of pedestrian body that do not generate Lidar detections, depending on sensor to pedestrian geometry. As a result, an attempt to associate pedestrian pose detections corresponding to human body parts that are not detected from the Lidar sensor will always fail. In other words, while the proposed association method aims to tackle the unavailability of depth information in obtained mono camera images, at the same time it might fail to tackle the limitation presented by the considered assumption for obtained Lidar measurement spread in real automotive applications, since Lidar sensor data is not uniformly distributed over object extent.

Effect of proposed association method to proposed state initialization approach In fact, the importance of aforementioned association method between obtained Lidar points and human pose detections is twofold. Firstly, a state initialization approach for an EOT algorithm is proposed in this report, where associated shoulder and ankles pose detections in 2D world coordinates frame are employed to calculate initial centroid position and velocity, as well as initial orientation and semi-axes lengths for the selected as first simulation timestep. Subsequently, this proposed approach is compared to the conventional state initialization procedure, both applied to the baseline tracking algorithm. Even though the association method is prone to failure for some simulation time instances, it is shown that a significantly decreased RMSE for estimated pedestrian velocity is achieved for the selected first simulation timestep when the proposed state initialization is incorporated to the baseline filter. A Monte Carlo simulation of 1000 runs proves also that this improved performance for pedestrian velocity is achieved independently of the selected initial simulation timestep. On the other hand, the proposed state initialization approach has a minor effect on estimated centroid position RMSE and modified Hausdorff distance for estimated shape extent.

Effect of proposed association method to calculated heading angle measurement for the pedestrian Secondly, the heading angle measurement, considered for the pedestrian in terms of the sequential measurement update step of the proposed tracking algorithm, is created based on the aforementioned association method. In fact, it is shown that inaccuracies in association method affect strongly the accuracy of calculated heading measurement, with respect to true heading angle for the pedestrian, especially in time instances when sensor to object geometry is such that the area close to the shoulder of the pedestrian is not detected by Lidar sensor. According to the analysis presented in this report, incorporation of the strongly fluctuating calculated heading angle measurement

model to the nonlinear kinematic measurement update step of the proposed filter results in a significant error increase for estimated pedestrian velocity.

Limitations of Random Matrix Model representation for pedestrian state Despite the significantly increased error for estimated pedestrian velocity due to inaccuracies in incorporated heading angle measurement, the estimated shape extent by the proposed filter is not affected (see Section 6-2-2). This behavior is expected, due to the limitations of the RMM state representation for the pedestrian, which is adopted for both tracking algorithms examined in this report.

To begin with, according to RMM representation, pedestrian state is represented by a kinematic vector $\mathbf{x}_k$, consisting of 2D position and velocity variables, and a 2x2 semi-positive definite matrix $\mathbf{X}_k$, representing an ellipse in x-y world coordinates frame. Moreover, both the baseline and proposed filter assume independence in the measurement update step of kinematic and shape extent state (see Section 3-3 - Assumption 1). Hence, two distinguished steps are defined for measurement update/prediction step of the kinematic and shape extent state of the pedestrian, respectively. For instance, estimated shape extent depends on the spread of obtained Lidar detections, the mean Lidar measurement and the predicted shape extent, respectively, according to Equations 3-22 - 3-26. Since an exactly identical update step for pedestrian shape extent is considered also for the proposed filter, it is obvious that pedestrian heading angle does not affect directly the estimated shape extent.

In addition, due to the representation of shape extent state by corresponding 2x2 matrix $\mathbf{X}_k$, it is not possible to directly estimate shape extent variables such as ellipse orientation and semi-axes lengths or define corresponding uncertainty measures for each estimated ellipse parameter. Instead, both tracking algorithms provide an estimate for the shape extent matrix. Subsequently, corresponding ellipsoid parameters are calculated based on Equations 3-35 - 3-39. In a few words, RMM is the simplest representation approach for an elliptic object and requires a linear Kalman-filter-like tracking algorithm, but it is impossible to explicitly estimate corresponding ellipse state variables in that filter.

7-2 Recommendations

Based on conclusions of this study, some research recommendations for further investigation on this subject are given. This chapter regards the features that are not examined in detail in the content of this M.Sc. thesis, but their investigation is considered essential. Further research may focus on the topics explained in the following paragraphs:

Alternative approach for association of Lidar and human pose detections data As discussed in the previous Section of this report, the association method examined in terms of this project is prone to failure at some simulation timesteps, depending also on the detected Lidar point cloud of the pedestrian. An alternative approach to map effectively pose detections from image plane coordinates to x-y world coordinates would be to assume constant length for all pedestrian body parts (e.g.: torso, arms, legs, etc..).

Then, associating correctly only a single pedestrian pose detection in 2D world coordinates allows as to calculate all remaining pose detections, based on assumed constant lengths of pedestrian body parts.

Alternative approach for pedestrian state representation and tracking In this work, the RMM representation is selected for the pedestrian, because it is the simplest state-of-the-art representation approach for an elliptic object. Nevertheless, the existance of the 2x2 shape extent matrix and the considered independent tracking assumption of kinematics and shape extent state presents limitations to the tracking result.

An alternative approach would be to substitute the RMM representation and instead use an augmented vector containing all kinematic and shape extent state parameters. As a result, it would be possible to jointly and directly estimate all state parameters of interest in a single step, as well as to define corresponding uncertainty values for each state parameter, by running one of the nonlinear tracking algorithms, presented in Section 2-2-1. For instance, the calculated heading angle measurement for the pedestrian could be incorporated directly to the measurement update step and improve tracking of considered ellipse orientation. The main disadvantage of this approach is the increased computational cost of presented nonlinear filters, in comparison to the RMM-based baseline and proposed tracking algorithms.

Bibliography

[1] K. Granström, C. Lundquist, F. Gustafsson, and U. Orguner, "Random Set Methods: Estimation of Multiple Extended Objects," *IEEE Robotics & Automation Magazine*, vol. 21, pp. 73–82, jun 2014.

[2] M. Baum and U. D. Hanebeck, "Shape tracking of extended objects and group targets with star-convex RHMs," *14th International Conference on Information Fusion*, pp. 1–8, 2011.

[3] M. Baum, *Simultaneous Tracking and Shape Estimation of Extended Objects*. PhD thesis, 2013.

[4] K. Granström, M. Baum, and S. Reuter, "Extended Object Tracking: Introduction, Overview and Applications," *Journal of Advances in Information Fusion*, vol. http://arx, pp. 1–30, mar 2016.

[5] K. Granström and U. Orguner, "A PHD filter for tracking multiple extended targets using random matrices," *IEEE Transactions on Signal Processing*, vol. 60, pp. 5657–5671, nov 2012.

[6] M. Beard, S. Reuter, K. Granström, B.-T. Vo, B.-N. Vo, and A. Scheel, "Multiple Extended Target Tracking With Labeled Random Finite Sets," *IEEE Transactions on Signal Processing*, vol. 64, pp. 1638–1653, apr 2016.

[7] M. Feldmann, D. Franken, and W. Koch, "Tracking of Extended Objects and Group Targets Using Random Matrices," *IEEE Transactions on Signal Processing*, vol. 59, pp. 1409–1420, apr 2011.

[8] J. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, pp. 1042–1059, jul 2008.

[9] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of object tracking*. 2011.

[10] D. L. Hall and J. Llinas, *Handbook of Multisensor Data Fusion*. 2001.

[11] H. Durrant-Whyte, "Introduction to estimation and the Kalman filter," *Australian Centre for Field Robotics*, 2001.

[12] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. 1995.

[13] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking," *Digital Signal Processing: A Review Journal*, vol. 25, no. 1, pp. 1–16, 2014.

[14] Y. Bar_Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. 2001.

[15] K. Gilholm, S. J. Godsill, S. Maskell, and D. Salmond, "Poisson models for extended target and group tracking," *Proceedings of the SPIE 5913, Signal and Data Processing of Small Targets 2005*, vol. 5913, pp. 1–12, 2005.

[16] K. Granström and C. Lundquist, "On the Use of Multiple Measurement Models for Extended Target Tracking," *16th International Conference on Information Fusion (FUSION)*, pp. 1534–1541, 2013.

[17] S. Yang and M. Baum, "Second-Order Extended Kalman Filter for Extended Object and Group Tracking," *19th International Conference on Information Fusion (FUSION)*, vol. 3, no. 2, 2016.

[18] S. Yang and M. Baum, "Extended Kalman filter for extended object tracking," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP))*, no. 1, pp. 4386–4390, 2017.

[19] S. Yang, M. Baum, and K. Granstrom, "Metrics for performance evaluation of elliptic extended object tracking methods," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 523–528, 2017.

[20] L. Sun, S. Zhang, B. Ji, and J. Pu, "Performance Evaluation for Shape Estimation of Extended Objects Using A Modified Hausdorff Distance âĹŮ," no. August, pp. 780–784, 2016.

[21] K. Granström, C. Lundquist, and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *14th International Conference on Information Fusion*, pp. 1–8, 2011.

[22] A. Scheel, K. Granström, D. Meissner, S. Reuter, and K. Dietmayer, "Tracking and Data Segmentation Using a GGIW Filter with Mixture Clustering," *Proceedings of the 17th International Conference on Information Fusion (2014)*, 2014.

[23] K. Granström, S. Reuter, M. Fatemi, and L. Svensson, "Pedestrian tracking using Velodyne data - stochastic optimization for extended object tracking," in *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 39–46, 2017.

[24] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[25] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[26] M. Feldmann and W. Koch, "Comments on: "Bayesian Approach to Extended Object and Cluster Tracking using Random Matrices"," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1687–1693, 2012.

[27] M. Feldmann, D. Fränken, and W. Koch, "Tracking of Extended Objects and Group Targets using Random Matrices - A New Approach," *Proceedings of the ISIF International Conference on Information Fusion (FUSION)*, pp. 242–249, 2008.

[28] A. Gupta and D. Nagar, *Matrix Variate Distributions*. Boca Raton, FL: Chapman & Hall/CRC Press, 1999.

[29] K. Granström and U. Orguner, "Properties and approximations of some matrix variate probability density functions," *Technical report from Automatic Control at Linköpings universitet Properties*, 2011.

[30] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Glossary

List of Acronyms

2D	Two Dimensional space
3D	Three Dimensional space
EKF	Extended Kalman Filter
EOT	Extended Object Tracking
FoV	Field of View
GIW	Gaussian Inverse-Wishart
GOT	Group Object Tracking
LMB	Labeled Multi-Bernoulli
LRS	laser Range Scanner
OT	Object Tracking
PDF	Probabilistic Density Function
PPP	Poisson Point Process
RFS	Random Finite Set
RHM	Random Hypersurface Model
RMM	Random Matrix Model
RMSE	Root Mean Squared Error
SOEKF	Second-Order Extended Kalman Filter
SPD	Semi-Positive Definite
VRUs	Vulnerable Road Users