

## Radar Based Road User Detection in Intelligent Vehicles

Palfy, A.

**DOI**

[10.4233/uuid:4778841d-a71a-444e-94ac-fa1fca1c732b](https://doi.org/10.4233/uuid:4778841d-a71a-444e-94ac-fa1fca1c732b)

**Publication date**

2022

**Document Version**

Final published version

**Citation (APA)**

Palfy, A. (2022). *Radar Based Road User Detection in Intelligent Vehicles*. [Dissertation (TU Delft), Delft University of Technology]. <https://doi.org/10.4233/uuid:4778841d-a71a-444e-94ac-fa1fca1c732b>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

# RADAR BASED

— ROAD USER DETECTION IN —

# INTELLIGENT VEHICLES



ANDRAS PALFFY





# **RADAR BASED ROAD USER DETECTION IN INTELLIGENT VEHICLES**



# **RADAR BASED ROAD USER DETECTION IN INTELLIGENT VEHICLES**

## **Dissertation**

for the purpose of obtaining the degree of doctor  
at Delft University of Technology  
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen,  
chair of the Board for Doctorates,  
to be defended publicly on  
Wednesday, 14 December 2022 at 15:00 o'clock

by

**Andras PALFFY**

Master of Science in  
Software Techniques for Digital Signal and Image Processing,  
Cranfield University, United Kingdom,  
born in Budapest, Hungary.

This dissertation has been approved by the promotor.

promotor: Prof. dr. Dariu M. Gavrilă

copromotor: Dr. Julian F. P. Kooij

Composition of the doctoral committee:

Rector Magnificus

Prof. dr. Dariu M. Gavrilă

Dr. Julian F. P. Kooij

Chairperson

Delft University of Technology, promotor

Delft University of Technology, copromotor

*Independent members:*

Prof. dr. Csaba Benedek

Prof. dr. Klaus Dietmayer

Prof. dr. ir. Marcel J.T. Reinders

Prof. dr. Alexander Yarovoy

Dr. Holger Caesar

Pazmany Peter Catholic University, Hungary

Ulm University, Germany

Delft University of Technology

Delft University of Technology

Delft University of Technology



*Keywords:* intelligent vehicles, automotive radars, road user detection

*Printed by:* Gildeprint Drukkerijen

*Style:* TU Delft House Style, with modifications by Moritz Beller  
<https://github.com/Inventitech/phd-thesis-template>

ISBN 978-94-6384-399-7

An electronic version of this dissertation is available at  
<http://repository.tudelft.nl/>.

*If I have seen further, it is by standing on the shoulders of giants.*

Isaac Newton





# CONTENTS

|   |             |
|---|-------------|
| <b>Summary</b>  | <b>xi</b>   |
| <b>Samenvatting</b>   | <b>xiii</b> |
| <b>1 Introduction</b>   | <b>1</b>    |
| 1.1 Scope of this thesis . . . . .  | 4           |
| 1.2 Thesis outline and contributions . . . . .                                | 4           |
| <b>2 Introduction to automotive radars</b>                                    | <b>7</b>    |
| 2.1 Measurement of distance . . . . .   | 11          |
| 2.2 Measurement of velocity . . . . .   | 14          |
| 2.3 Measurement of direction. . . . .   | 17          |
| 2.4 Different types of radar data . . . . .                                   | 18          |
| 2.4.1 2+1D radar point cloud. . . . .   | 18          |
| 2.4.2 Radar cube . . . . .  | 19          |
| 2.4.3 3+1D radar point cloud. . . . .   | 21          |
| 2.5 Advantages of radar sensors . . . . .                                     | 22          |
| 2.6 Disadvantages of radar sensors . . . . .                                  | 23          |
| <b>3 Related work</b>   | <b>27</b>   |
| 3.1 2+1D radars . . . . .   | 28          |
| 3.2 Radar cube . . . . .  | 29          |
| 3.3 3+1D radars . . . . .   | 29          |
| 3.4 The use of Doppler. . . . .   | 30          |
| 3.5 Radar datasets . . . . .  | 30          |
| 3.6 Darting out pedestrians . . . . .   | 31          |
| 3.6.1 Camera based approaches . . . . .                                       | 31          |
| 3.6.2 Radar based approaches . . . . .  | 32          |
| 3.6.3 Fusion based approaches . . . . .                                       | 32          |
| 3.7 Environment modeling . . . . .  | 32          |
| 3.8 Tracking . . . . .  | 33          |
| <b>4 Detection of darting out pedestrians with fusion of camera and radar</b> | <b>35</b>   |
| 4.1 Introduction . . . . .  | 36          |
| 4.2 Proposed approach . . . . .   | 38          |
| 4.2.1 Overview and contributions. . . . .                                     | 38          |
| 4.2.2 State space and notations . . . . .                                     | 39          |
| 4.2.3 Prediction step . . . . .   | 40          |
| 4.2.4 Update step. . . . .  | 41          |

|          |   |           |
|----------|---|-----------|
| 4.3      | Implementation. . . . .   | 43        |
| 4.3.1    | Particle filtering . . . . .  | 43        |
| 4.3.2    | Use of stereo camera data. . . . .                                      | 45        |
| 4.3.3    | Use of radar data . . . . .   | 46        |
| 4.4      | Dataset. . . . .  | 47        |
| 4.5      | Experiments . . . . .   | 49        |
| 4.5.1    | Estimated existence probability in dangerous situations . . . . .       | 50        |
| 4.5.2    | Distinguishing dangerous and non-dangerous scenarios . . . . .          | 53        |
| 4.6      | Discussion . . . . .  | 54        |
| 4.7      | Conclusions and future work . . . . .                                   | 57        |
| <b>5</b> | <b>Multi-class road user detection using the 3D radar cube</b>          | <b>59</b> |
| 5.1      | Introduction . . . . .  | 60        |
| 5.1.1    | Contributions. . . . .  | 63        |
| 5.2      | Proposed method. . . . .  | 63        |
| 5.2.1    | Pre-processing . . . . .  | 64        |
| 5.2.2    | Network . . . . .   | 64        |
| 5.2.3    | Ensemble classifying . . . . .  | 65        |
| 5.2.4    | Object clustering . . . . .   | 65        |
| 5.3      | Dataset. . . . .  | 66        |
| 5.4      | Experiments . . . . .   | 67        |
| 5.4.1    | Baselines. . . . .  | 68        |
| 5.4.2    | Implementation. . . . .   | 68        |
| 5.4.3    | Results . . . . .   | 69        |
| 5.4.4    | Discussion . . . . .  | 72        |
| 5.5      | Conclusion of the chapter. . . . .                                      | 75        |
| <b>6</b> | <b>Multi-class road user detection using the 3+1D radar point cloud</b> | <b>77</b> |
| 6.1      | Introduction . . . . .  | 78        |
| 6.1.1    | Contributions. . . . .  | 80        |
| 6.2      | Methodology. . . . .  | 80        |
| 6.2.1    | 3+1D radar point clouds and Doppler encoding . . . . .                  | 80        |
| 6.2.2    | Accumulation of radar point clouds. . . . .                             | 81        |
| 6.2.3    | Data augmentation . . . . .   | 81        |
| 6.3      | Dataset. . . . .  | 81        |
| 6.3.1    | Measurement setup and provided data . . . . .                           | 82        |
| 6.3.2    | Annotation . . . . .  | 85        |
| 6.4      | Experiments . . . . .   | 85        |
| 6.4.1    | Ablation study: <i>PP-radar</i> . . . . .                               | 86        |
| 6.4.2    | Performance comparison: <i>PP-radar</i> vs. <i>PP-LiDAR</i> . . . . .   | 87        |
| 6.5      | Discussion . . . . .  | 90        |
| 6.6      | Conclusion of the chapter. . . . .                                      | 90        |
| <b>7</b> | <b>Conclusion</b>   | <b>93</b> |
| 7.1      | Summary. . . . .  | 94        |
| 7.2      | Future work . . . . .   | 97        |

---

|                             |            |
|-----------------------------|------------|
| <b>Acknowledgments</b>      | <b>105</b> |
| <b>References</b>           | <b>107</b> |
| <b>Curriculum Vitæ</b>      | <b>121</b> |
| <b>List of Publications</b> | <b>123</b> |



---

# SUMMARY

This thesis addresses the problem of object detection with automotive radar sensors in the field of intelligent vehicles with special attention to vulnerable road users: pedestrians, cyclists, and motorcyclists. It is not the goal of this work to improve the hardware design or signal processing algorithms of the radar sensors themselves, but to take the output of these sensors as “given”, and to propose various ways to use them for road user detection. To facilitate the reading, a brief introduction to the operating principles of radar sensors and their advantages and limitations is first given. Subsequently, the thesis discusses automotive radar based detection of road users based on three different types of radar data.

First, the thesis investigates radar based detection of road users using a commercial automotive radar sensor that outputs the widely used 2+1 dimensional (range, azimuth, and the plus one is the Doppler or velocity dimension) point cloud of radar targets. This data is then used in a sensor fusion approach that combines stereo camera and radar data. The system is used in a specific pedestrian detection scenario: the goal is to detect darting out pedestrians, i.e., pedestrians approaching the road from behind a parked vehicle while being partly or fully occluded. Instead of simply fusing the two sensors, a so-called occlusion aware sensor fusion is proposed to account for the occlusions caused by the parked vehicles. Results show that the inclusion of the radar sensor is beneficial in this task, as darting out pedestrians are detected earlier than when using the camera alone. This is in part due to a unique property of radar signals: they are able to propagate under the parked vehicle and reach the pedestrian even if the pedestrian is completely occluded. In addition, taking the occlusion information into account is shown to also help to better evaluate such scenarios.

The thesis then explores how the previously used point cloud can be better utilized by augmenting it with a lower level of radar data, called the radar cube. The radar cube is a 3D data matrix with three axes: range, azimuth, and Doppler (i.e., relative radial velocity measured by the radar). Unlike the point cloud which provides a single velocity for a point, the radar cube provides the full velocity distribution (i.e., Doppler vector) at each 2D range-azimuth position. Such distribution can capture class-specific modulations of an object’s main velocity caused by its moving parts, e.g., swinging limbs or rotating wheels, rather than a single estimated velocity. A novel neural network is proposed that fuses the 2+1D radar point cloud with this radar cube. Extensive experiments are conducted to show that the incorporation of such low-level data indeed helps in the task of multi-class road user detection.

Finally, the thesis investigates data from next-generation automotive radars. These radars provide elevation information in addition to range, azimuth, and Doppler velocity data, resulting in a 3+1D (three spatial and one Doppler dimension) point cloud. Their point clouds also tend to have higher density compared to the 2+1D radars, so they are becoming more similar to LiDAR point clouds. Therefore, the thesis presents experiments with the application of a 3D object detector to radar point clouds, rather than applying it to LiDAR point clouds, as done previously. Then, rather than comparing it with radar

based benchmarks, the performance of the trained detector is compared with the same detector using a dense high-end 64-layer LiDAR sensor as input. The results show that object detection on high-end 64-layer LiDAR data still outperforms that on 3+1D radar data, but the addition of elevation information and integrating sequential radar scans helps close the gap. To facilitate this experimental study and support further research on this topic, the novel View-of-Delft (VoD) automotive dataset is introduced. It contains 8693 frames of synchronized and calibrated 64-layer LiDAR-, (stereo) camera-, and 3+1D radar- data acquired in complex, urban traffic. It consists of 123106 3D bounding box annotations of both moving and static objects, including 26587 pedestrian, 10800 cyclist and 26949 car labels.

At the same time, the three introduced approaches not only differ in the type of radar data used, but also present three different use cases for automotive radars in a vulnerable road user detection pipeline. The thesis first presents a high-level fusion of stereo camera and radar sensors using the 2+1D point cloud. Then, an approach is presented that semantically segments the 2+1D point cloud into the classes of pedestrians, cyclists, and cars by exploiting the radar cube. Such a segmented point cloud (i.e., a list of points with class information) can be a useful input for mid-level fusion. Finally, it is shown that it is possible to regress 3D bounding boxes (i.e., bounding boxes with position, size, and orientation) for multiple classes using only a single radar sensor which provides 3+1D radar point clouds. Such bounding boxes can serve as input to a high-level fusion aimed at 3D object detection. In addition, the ability to retrieve these bounding boxes without other sensors can increase the redundancy of a vulnerable road user detection pipeline, since the failure of another sensor does not necessarily mean the failure of detection.

The thesis is concluded by discussing its findings and making suggestions for future work both inside and outside of the research field of road user detection.

# SAMENVATTING

Dit proefschrift behandelt het probleem van objectdetectie met auto-radarsensoren op het gebied van intelligente voertuigen met speciale aandacht voor kwetsbare weggebruikers: voetgangers, fietsers en motorrijders. Het is niet het doel van dit werk om het hardwareontwerp of de signaalverwerkingsalgoritmen van de radarsensoren zelf te verbeteren. In plaats daarvan neemt dit proefschrift de output van signaalverwerkingsalgoritmen als gegeven en stelt verschillende manieren voor om ze te gebruiken voor detectie van weggebruikers. Om het lezen voor de lezer te vergemakkelijken, wordt eerst een korte inleiding gegeven in de werkingsprincipes van radarsensoren en hun voordelen en beperkingen. Vervolgens bespreekt het proefschrift autoradargebaseerde detectie van weggebruikers op basis van drie verschillende soorten radargegevens.

Ten eerste onderzoekt het proefschrift op radar gebaseerde detectie van weggebruikers met behulp van een commerciële auto-radarsensor die de veelgebruikte 2+1 dimensionale (bereik, azimut, en de plus één is de Doppler- of snelheidsdimensie) puntenwolk van radardoelen uitvoert. Deze gegevens worden vervolgens gebruikt in een sensorfusie-aanpak die stereocamera- en radargegevens combineert. Het systeem wordt gebruikt in een specifiek voetgangersdetectiescenario: het doel is om wegschietende voetgangers te detecteren, d.w.z. voetgangers die de weg naderen van achter een geparkeerd voertuig terwijl ze gedeeltelijk of volledig zijn afgesloten. In plaats van simpelweg de twee sensoren te fuseren, wordt een zogenaamde occlusiebewuste sensorfusie voorgesteld om rekening te houden met de occlusies veroorzaakt door de geparkeerde voertuigen. De resultaten tonen aan dat het opnemen van de radarsensor gunstig is bij deze taak, omdat wegschietende voetgangers eerder worden gedetecteerd dan wanneer alleen de camera wordt gebruikt. Dit komt mede door een unieke eigenschap van radarsignalen: ze kunnen zich onder het geparkeerde voertuig voortplanten en de voetganger bereiken, zelfs als de voetganger volledig is afgesloten. Bovendien blijkt dat het in aanmerking nemen van de occlusie-informatie ook helpt om dergelijke scenario's beter te evalueren.

Het proefschrift onderzoekt vervolgens hoe de eerder gebruikte puntenwolk beter kan worden benut door deze uit te breiden met een lager niveau van radargegevens, de zogenaamde radarkubus. De radarkubus is een 3D-gegevensmatrix met drie assen: bereik, azimut en Doppler (d.w.z. relatieve radiale snelheid gemeten door de radar). In tegenstelling tot de puntenwolk die een enkele snelheid voor een punt biedt, biedt de radarkubus de volledige snelheidsverdeling (d.w.z. Doppler-vector) op elke 2D-afstand-azimutpositie. Een dergelijke distributie kan klassenspecifieke modulaties van de hoofdsnelheid van een object vastleggen, veroorzaakt door de bewegende delen, bijvoorbeeld zwaaiende ledematen of roterende wielen, in plaats van een enkele geschatte snelheid. Er wordt een nieuw neurale netwerk voorgesteld dat de 2+1D-radarpuntenwolk combineert met deze radarkubus. Er worden uitgebreide experimenten uitgevoerd om aan te tonen dat het incorporeren van dergelijke low-level data inderdaad helpt bij het opsporen van weggebruikers in meerdere klassen.



Ten slotte onderzoekt het proefschrift gegevens van autoradars van de volgende generatie. Deze radars bieden hoogte-informatie naast afstands-, azimut- en Doppler-snelheidsgegevens, wat resulteert in een 3+1D (drie ruimtelijke en één Doppler-dimensie) puntenwolk. Hun puntenwolken hebben meestal ook een hogere dichtheid in vergelijking met de 2+1D-radars, dus ze gaan steeds meer lijken op LiDAR-puntenwolken. Daarom presenteert het proefschrift experimenten met de directe toepassing van een 3D-objectdetector op radarpuntenwolken, wat meestal wordt toegepast op LiDAR-puntenwolken.

In plaats van deze te vergelijken met op radar gebaseerde benchmarks, worden de prestaties van de getrainde detector vergeleken met dezelfde detector met behulp van een hoogwaardige 64-laags LiDAR-sensor als invoer. De resultaten laten zien dat objectdetectie op hoogwaardige 64-laags LiDAR-gegevens nog steeds beter presteert dan op 3+1D-radargegevens, maar de toevoeging van hoogte-informatie en het integreren van sequentiële radarscans helpt de kloof te dichten. Om deze experimentele studie te vergemakkelijken en verder onderzoek over dit onderwerp te ondersteunen, wordt de nieuwe View-of-Delft (VoD) automotive dataset geïntroduceerd. Het bevat 8693 frames van gesynchroniseerde en gekalibreerde 64-laags LiDAR-, (stereo)camera- en 3+1D-radargegevens die zijn verkregen in complex stadsverkeer. Het bestaat uit 123106 3D-begrenzingskader-annotaties van zowel bewegende als statische objecten, waaronder 26587 voetgangers-, 10800-fietsers- en 26949 autolabels.

Tegelijkertijd verschillen de drie geïntroduceerde benaderingen niet alleen in het type radargegevens dat wordt gebruikt, maar presenteren ze ook drie verschillende gebruiksscenario's voor autoradars in een detectiepijplijn voor kwetsbare weggebruikers. Het proefschrift presenteert eerst een fusie op hoog niveau van stereocamera- en radarsensoren met behulp van de 2+1D-puntenwolk.

Vervolgens wordt een benadering gepresenteerd die de 2+1D-puntenwolk semantisch segmenteert in de klassen voetgangers, fietsers en auto's door gebruik te maken van de radarkubus. Zo'n gesegmenteerde puntenwolk (d.w.z. een lijst met punten met klasse-informatie) kan een nuttige input zijn voor fusie op het middenniveau.

Ten slotte wordt aangetoond dat het mogelijk is om 3D-begrenzingsvakken (d.w.z. begrenzingsvakken met positie, grootte en oriëntatie) voor meerdere klassen te regresseren met slechts één enkele radarsensor die 3+1D-radarpuntenwolken levert. Dergelijke begrenzingsvakken kunnen dienen als input voor een fusie op hoog niveau gericht op 3D-objectdetectie. Bovendien kan de mogelijkheid om deze begrenzingsvakken op te halen zonder andere sensoren de redundantie van een detectiepijplijn voor kwetsbare weggebruikers vergroten, aangezien het falen van een andere sensor niet noodzakelijk het falen van de detectie betekent.

Het proefschrift wordt afgesloten met het bespreken van de bevindingen en het doen van suggesties voor toekomstig werk, zowel binnen als buiten het onderzoeksveld van weggebruikersdetectie.

# 1

## INTRODUCTION

*Truly self-driving cars are expected to hit the markets in ten to twenty years  
- and that is true regardless of the current date.*

Dariu M. Gavrilă

THE promise of truly self-driving cars and the benefits expected from them have been widely discussed for over a decade now. Both traditional car manufacturers or suppliers (e.g., Daimler, Ford, Bosch, Continental), and new players in the industry (e.g., Tesla, Lucid Motors, Waymo, Comma.ai, Lyft, Aimotoive) have invested large sums [1] in the development of self-driving capabilities. Meanwhile, researchers from institutes all around the world have also turned their attention to the challenges of self-driving. Robotics and pattern recognition conferences and journals are filled with research papers on scene understanding, object detection and classification, path planning, sensor fusion and control in the domain of intelligent vehicles [2]. In addition, self-driving vehicles have become a topic of discussion for the average citizen [3] - something that rarely occurs with new technologies.

For a discussion with such a broad audience with diverse backgrounds, it was imperative to have a common taxonomy: what exactly do we mean when we talk about self-driving vehicles? In 2018, the Society of Automotive Engineers (SAE) proposed a system with six different levels [4], which has since been widely accepted and used. The levels are as follows. At Level 0 (L0, manual driving), the vehicle lacks any type of driving automation technology. At Level 1 (L1, driver assistance), some automotive systems provide assistance with acceleration, braking, or steering, such as adaptive cruise control. Level 2 (L2, partial driving automation) vehicles not only assist with, but also provide partial automation of the above functions in certain scenarios. At Level 3 (L3, conditional driving automation), the vehicle takes over all driving tasks independently under certain conditions. However, the driver must be available to take control from the automation at any time if necessary. Level 4 vehicles (L4, high driving automation) have continuous and complete control over all driving tasks. They can autonomously carry passengers, who do not need to be ready to take over the control. Nevertheless, Level 4 systems are usually limited to certain conditions (e.g., only good weather or daytime) and operational areas (e.g., designated routes, cordoned yards, or other geographic boundaries). Finally, Level 5 (L5, full driving automation) means that the vehicle can operate independently and universally in any weather condition without geographic boundaries.

The extraordinary interest from various distinct groups in society is likely due to the promised and/or expected benefits of a fully self-driving (L5) vehicle. In general, the most frequently visioned benefits [5][6][7] fall into three main categories: safety, comfort, and financial. Perhaps the most widely accepted argument for the introduction of self-driving cars is the issue of traffic *safety*. The World Health Organization (WHO) estimates that about 1.3 million road traffic fatalities happen every year [8]. Of these, more than half are vulnerable road users (VRUs): pedestrians, cyclists or motorcyclists who can easily be injured and killed in a road environment dominated by cars. In another study, the National Highway Traffic Safety Administration reported that the “critical reason” for 94% of crashes in the United States between 2005 and 2007 was attributed to the driver [9]. If the role of the driver is eliminated or reduced through the deployment of intelligent vehicles, this number could theoretically be lowered, saving thousands of lives each year. On the other hand, this prediction assumes that the self-driving stack itself does not introduce new sources of errors of the same magnitude. Truly self-driving cars are expected to have fewer accidents than human drivers because of two aspects. First, a major advantage of an algorithm performing similarly to a human driver (even if it is not better) is that it never gets tired, distracted, or falls asleep, all three of which are common causes of accidents [9]. Second, a fully equipped

self-driving vehicle could drive better than a human in some aspects. For example, while the average human reaction time in response to visual stimuli is about 250 ms [10], the reaction time of an algorithm can be significantly lower than this with optimized hardware and software implementation. Another advantage of computer-controlled vehicles is the additional source of information that comes from non-visual sensors, such as LiDAR or radar, and from communication with the infrastructure. In other words, such a vehicle, with its LiDAR or radar sensors, might be able to see in the dark or in other poor visibility conditions, and know the exact next maneuvers of other cars by communicating with them - capabilities that are impossible for a human driver.

Another often discussed promise of self-driving cars is the additional *comfort* they could bring into our lives. If truly self-driving, L5 vehicles become widely available, we will no longer have to worry about parking, as our ride could be summoned and discarded on demand, and in the meantime park itself in a remote parking lot. They could also provide the luxury of sleeping, relaxing or working during a commute or a road-trip instead of the exhausting task of driving. A most welcome added convenience would be the increased freedom and mobility of people who for some reason cannot drive themselves, e.g., the blind, disabled, children, or paralyzed citizens. Moreover, such systems would eliminate the need of “designated drivers” after parties and other events.

Last, but not least, there are *financial* motivations for developing a driverless system. A Level 4 or 5 vehicle is a “dream come true” for fleet operators and logistics companies, as it would eliminate or drastically reduce spending on driver salaries. Moreover, such vehicles could also be operated day and night without mandated or health-related breaks, further increasing efficiency and profit of the company. In the meantime, while this would be beneficial to the companies mentioned above, it is also clear that such a transformation of the transportation industry would also mean significant layoffs and a shift in the job market in related fields such as cab, bus, and lorry drivers, raising concerns for our society as a whole.

Of the three main arguments (i.e., safety, comfort, and financial), two only fully emerge if the car is truly and fully self-driving: financial and comfort. Nevertheless, truly self-driving, Level 5 cars do not yet exist, and there is no consensus on when development will reach that point: predictions range from couple years to 75 years [11], while some say such vehicles will never exist [11]. The situation is different when it comes to safety. Safety related benefits can be achieved even without full self-driving capabilities and are already being realized today in L2 and L3 vehicles. That is, by applying algorithms developed for self-driving while the primary driver is still a human, we can detect potential hazards (e.g., possible collision with another car, cyclist, or an emerging pedestrian) and trigger alarms, initiate emergency braking, or even take evasive maneuvers. Doing so can save lives and reduce the number of injuries right now or in the near future, without having to solve every single potential driving situation, or in other words, to reach Level 5, true self-driving.

To achieve this, the first step in a self-driving pipeline is perception: we need to know exactly where and how other road users (parked or moving cars, pedestrians, cyclists, etc.) are in relation to the ego-vehicle. This is crucial to assess whether a situation is dangerous and to plan further steps. Intelligent vehicles have three main types of sensors to perceive the world: cameras, LiDARs and radar sensors. All three have advantages and disadvantages, and no clear winner sensor (or combination of sensors) has yet emerged.

## 1.1 SCOPE OF THIS THESIS

The topic of this thesis is perception in intelligent vehicles with radar sensors. More specifically, my goal is to detect the presence, location, and orientation of other road users using automotive radars, with a particular focus on vulnerable road users. To this end, after introducing the reader to the basics, advantages and disadvantages of radars, I present three different use cases of a radar sensor in a pipeline for detecting vulnerable road users, using three different radar data representations. I conclude the thesis by commenting on the future of road user detection, pointing out remaining questions, and discussing whether radar sensors have a legitimate place in an intelligent vehicle of the future.

As we will see in the following chapters, the basic operating principles of radar are straightforward and were established decades ago [12]. However, I will also show that the exact implementation of a radar based perception system is a complex task in terms of both hardware and software. Both aspects can still be improved and are therefore still widely researched topics today. In this thesis, I do not aim to improve on the low-level sensing aspect of radars, neither by developing new hardware solutions (e.g., different antenna arrays) nor by improving on-board or off-board signal processing algorithms (e.g., new peak-finding or direction-of-arrival regression algorithms). Instead, I take the output of such automotive radars as a “given”, discuss their format, advantages and disadvantages, and make recommendations on how best to use them. In other words, the scope of this work is not to improve the quality of radar sensing, but to suggest approaches to exploit the readily available output of these sensors in the area of vulnerable road user detection in intelligent vehicles.

## 1.2 THESIS OUTLINE AND CONTRIBUTIONS

THIS section provides an overview of the following chapters and then discusses their main contributions individually.

To properly understand both the opportunities and challenges of using such a radar sensor in an intelligent vehicle, we must first discuss the basic operating principles, capabilities, and limitations of automotive radars. To this end, Chapter 2 introduces the reader to the fundamentals of automotive radars and describes the different types of radar data that will be used for perception tasks in the following chapters: 2+1D radar point clouds, the concept of radar cube, and 3+1D point clouds. Subsequently, in Chapter 3 I give a brief overview of the related literature by collecting and categorizing the most important works.

Chapter 4, 5, and 6 differ in what type(s) of radar data of the aforementioned three are used to tackle their addressed perception task. Conventional, commercially available radars output a 2+1D point cloud, which has two spatial dimensions (range and azimuth), and one for Doppler, see Chapter 2. Chapter 4 introduces a method that uses such a point cloud to detect pedestrians in a specific scenario where the pedestrian approaches the road while being partially or fully occluded by a parked vehicle (also called “darting out”). Chapter 5 also uses the exact same radar sensor. However, in addition to the standard 2+1D point cloud, this time it is combined with a lower level of radar data, called the radar cube to detect road users of multiple classes. It is shown that this additional low-level data is beneficial and should be made available to researchers and developers where possible. Chapter 6 also deals with multi-class road user detection, but uses another radar data format that is not yet

widely available: the 3+1D radar point cloud. This means that in addition to the usual range, azimuth, and Doppler dimensions, the radar used in this chapter can also give the elevation angle (i.e., height) of the reflections, see Chapter 2. In other words, instead of the spatially two dimensional, planar point cloud used in the previous chapters, Chapter 6 uses a spatially three dimensional point cloud, somewhat like the point clouds provided by LiDAR sensors.

At the same time, Chapter 4, 5, and 6 not only differ in the type of radar data used, but also present three different use cases for automotive radars in a vulnerable road user detection pipeline. Chapter 4 presents a high-level fusion system for stereo camera and radar sensors. Chapter 5 introduces a novel approach that semantically segments the 2+1D point cloud into the classes of pedestrians, cyclists, and cars. Such a segmented point cloud (i.e., a list of points with class information) can be a useful input for mid-level fusion. As the third use case, in Chapter 6, it is shown that it is possible to regress 3D bounding boxes (i.e., bounding boxes with position, size, and orientation) for multiple classes using only a single radar sensor. Such boxes can serve as input to a high-level fusion aimed at 3D object detection. In addition, the ability to retrieve these boxes without other sensors can increase the redundancy of a vulnerable road user detection pipeline, since the failure of another sensor does not necessarily mean the failure of detection.

Finally, in Chapter 7, the thesis and its results are concluded, unanswered questions are discussed, and suggestions for future work are made, first for the three individual use cases and their possible connections, then for the application of radars with goals other than road user detection, or not in intelligent vehicles. To conclude the chapter, thoughts about the future of point cloud sensors are shared.

The three research related chapters, Chapter 4, 5, and 6, and their contributions are described in more details as follows.

## **DETECTION OF DARTING OUT PEDESTRIANS WITH FUSION OF CAMERA AND RADAR**

In Chapter 4, a high-level fusion approach of stereo camera and 2+1D radar is presented, with the goal of detecting darting out pedestrians as early as possible. Since such scenarios are often complicated by occlusions (e.g., by parked vehicles), the proposed fusion method takes information on what areas are occluded as an input. The contributions of this chapter are as follows. First, I propose a generic (i.e., not unique to radar and camera) occlusion aware multi-sensor Bayesian filter for object detection and tracking. Second, I apply the proposed filter as a radar and stereo camera based system for pedestrian detection and tracking on challenging darting out scenarios. I show that both incorporating occlusion information and the radar sensor into the model helps to detect darting out pedestrians earlier. Third, to facilitate future exploration of these scenarios, I share my dataset with the research community, containing more than 500 relevant scenarios with camera, radar, LiDAR, and odometry data.

## **MULTI-CLASS ROAD USER DETECTION USING THE 3D RADAR CUBE**

Chapter 5 addresses the problem of detecting multiple classes of road users purely by radar. The often mentioned sparseness of the radar point cloud makes this task challenging, since fewer points on an object represent less information. To address this problem, many existing approaches first cluster radar targets to create object proposals. Then, cluster-level features

are extracted to gain more information about the classes of objects, and then these clusters are classified as a whole. However, this “cluster-then-classify” pipeline is prone to clustering errors. In this chapter, I instead introduce a “classify-then-cluster” pipeline and tackle the lack of information by exploiting a lower level of radar data, the radar cube.

My contributions are twofold. First, I propose a radar based, single-frame, multi-class (pedestrian, cyclist, car) moving road user detection method which exploits both target and low-level radar data using a custom-designed CNN. The method provides not only classified radar targets, but also object proposals through a class-specific clustering. Second, using a large-scale, real-world dataset, I show that my method is capable of detecting road users with higher performance than the state of the art, both in target-wise (target classification) and object-wise (object detection) metrics using only a single frame of radar data.

## MULTI-CLASS ROAD USER DETECTION USING THE 3+1D RADAR POINT CLOUDS

The point clouds of the next generation, 3+1D radars have higher density and the additional height dimension compared to the 2+1D radars, making them more similar to LiDAR point clouds (see Figure 6.1 for a visual comparison). Therefore, in Chapter 6, I first experiment with the direct application of a 3D object detector to radar point clouds, which is usually applied to LiDAR point clouds. Then, I compare the performance of the trained detector with the same detector using a dense high-end 64-layer LiDAR sensor as input for a real challenge.

The contribution of this chapter is threefold: First, I examine road user detection with 3+1D radar by applying PointPillars [13], a state-of-the-art multi-class 3D object detector commonly used for LiDAR. I investigate the significance of different features of the 3+1D radar point cloud in an ablation study, including Doppler, RCS (Radar Cross Section), and the elevation information that the traditional 2+1D automotive radars, such as the one used in the previous chapters cannot provide. Second, I compare radar based to LiDAR based detection by training and testing on the same traffic scenes. I show that currently point cloud based detection on dense LiDAR still outperforms detection on radar. However, I also find that the performance gap can be reduced when radar data includes elevation information, and when multiple radar scans are temporally integrated. Additionally, the detection benefits from Doppler measurements, which are unique to radar. Third, I publish the View-of-Delft (VoD) dataset, a novel multi-sensor automotive dataset for multi-class 3D object detection, consisting of calibrated and synchronized LiDAR, camera, and radar data recorded in real-world traffic situations. The View-of-Delft dataset is the largest dataset to date containing 3+1D radar recordings with  $\sim 20$  times as many annotated frames as the Astyx dataset [14], and it is the only publicly available one containing camera, radar, and 64-layer LiDAR data at the same time. Although the experiments in Chapter 6 focus on radar-only methods, the dataset is also suitable for sensor fusion, camera-only, or LiDAR-only methods due to this sensor arrangement, and could be useful for researchers interested in testing their algorithms in cluttered urban traffic.

# 2

2

## INTRODUCTION TO AUTOMOTIVE RADARS

*He's got a radar lock on us!*

Hollywood in Top Gun



THIS chapter is intended to provide a brief introduction to the subject of automotive radars, including their operating principles, capabilities, advantages and disadvantages. Although this topic could make up the content of several university courses, it is not my goal to discuss it in detail here. Instead, I would like to give an overview of the signal processing steps involved and their consequences, sufficient to understand the results of this thesis and/or to help someone conduct similar research.

For more information on the fundamentals of radar sensors, I refer the reader to [12]. For a more detailed discussion on automotive radars specifically, I recommend [15], which provides excellent and detailed material on the subject.

Radar sensors are widely used in aerospace [16][17][18], remote sensing [19] and traffic control [20], but have also been used in the automotive industry for decades [21], long before the introduction of on-board camera or LiDAR sensors. One of the main reasons for this is the fact that most automotive radars can measure an object's distance, direction and relative radial velocity simultaneously, see Figure 2.1. This is a unique feature among sensors, i.e., LiDAR or a camera require multiple frames and some kind of tracking to estimate the speed of an object. Radar, on the other hand, does this instantly with a single measurement. In the following, I will discuss the basics of radars and how such simultaneous measurement is performed.

The term "radar" is an acronym for Radio Detection and Ranging [12]. As the full name implies, a radar is capable of detecting objects and determining their ranges or distances using radio waves. The basic principle of these sensors is quite straightforward. They emit a radio signal that may be reflected by objects they encounter. A fraction of this reflected signal can travel back to the radar sensor's location. The propagation time of the signal between transmission and reception is proportional to the range of the object. Although radars are primarily designed to measure this distance, the received reflection may also contain information about the direction, speed, and reflectivity of the object [12].

One of the most characteristic differences between radars and cameras or LiDARs is that while all sensors use electromagnetic waves, the frequencies of these waves differ significantly. See Figure 2.2 for an overview of the wavelengths and frequencies used by the three sensors in the electromagnetic wave spectrum. While cameras and LiDARs use frequencies near the visible spectrum, radars work with microwaves. From left to right, we see increasing wavelengths and decreasing frequencies. Immediately following the ultraviolet range is what is known as the visible spectrum. This is what humans can see, and normally cameras operate in this range as well, sometimes extending into the infrared range. After the visible spectrum comes the mentioned infrared spectrum. LiDAR sensors operate in this range, just outside the visible spectrum. If one takes a photograph of a LiDAR sensor with a camera, sometimes the laser beams can be seen in the captured image, which are not visible to the naked eye. After the infrared spectrum comes the microwave spectrum, and this is where a typical automotive radar operates (between 20-80 GHz). Note that the frequency is orders of magnitude smaller and the wavelength orders of magnitude larger than those of the other two sensors. As we will see later in this chapter, this drastic difference in operating wavelengths is the reason for some of the key advantages (and some disadvantages) that radar has over camera and LiDAR.

There are two main types of radar sensors [12]. Pulse radars transmit signals in the form of short *pulses* and then wait a short time for reflections from the illuminated targets. The time interval between transmission of the pulse and its reception is then used to determine

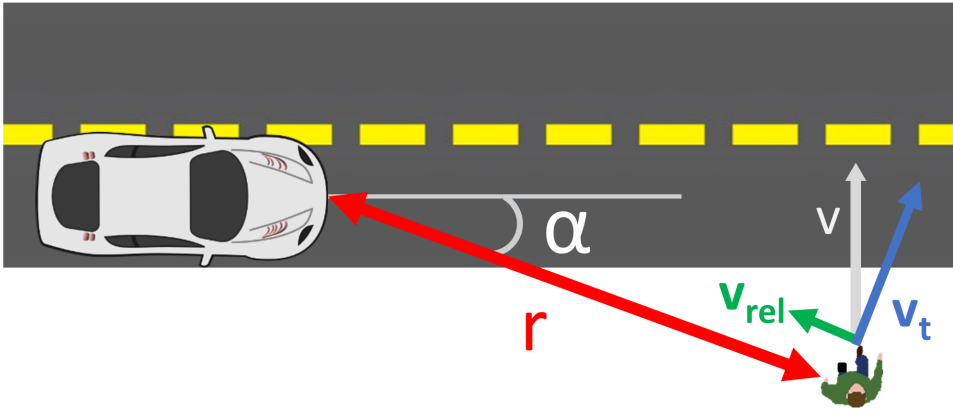


Figure 2.1: Overview of radar based perception. Most automotive radars (in the ego-vehicle, white) can report the other road users' distance  $r$ , azimuth  $\alpha$ , and relative radial velocity  $v_{rel}$ . However, they cannot measure the tangential velocity  $v_t$  and thus the true velocity  $v$ . Some new-generation radars can also provide elevation information, which is neglected in this figure for simplicity.

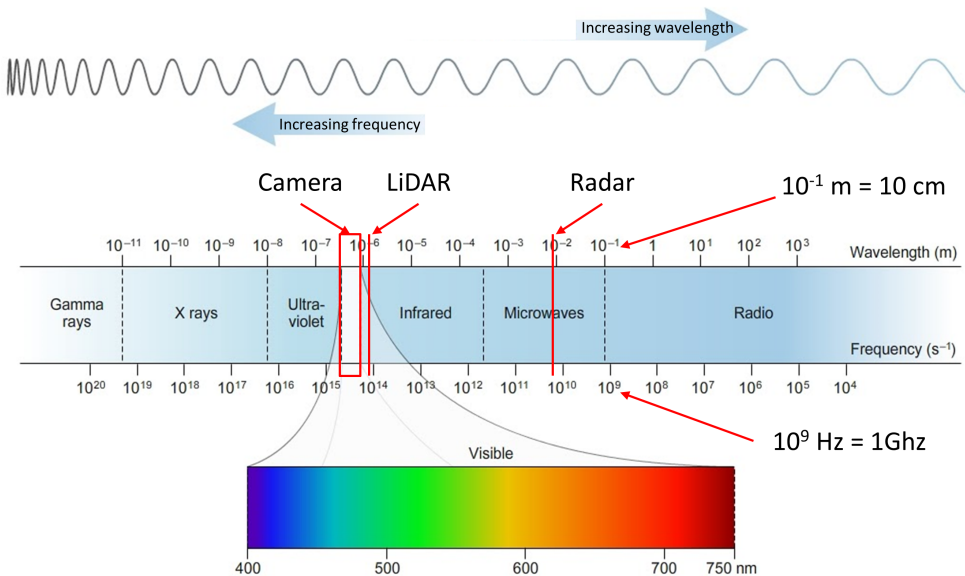


Figure 2.2: Comparison of typical operating wavelengths and frequencies of camera, LiDAR, and radar sensors. Radars use lower frequencies and longer wavelengths than the other two sensors. Original image is from [22].

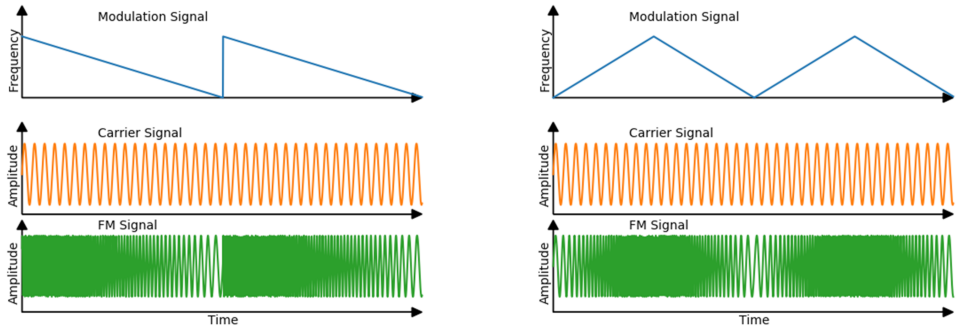


Figure 2.3: Examples of frequency (FM) modulated signals (bottom) generated by a sinusoidal carrier signal (middle) modulated by different modulation signals (top) in frequency modulation.

the target's range. Pulse radars are excellent for measuring the exact distance of objects, especially at longer distances. Also, because they transmit in pulses, they are more difficult to detect and/or “jam”. For this reason, pulse radars are often used in military applications, especially for monitoring airspace. However, they are not able to instantly measure the velocities of the objects. In contrast, continuous wave (CW) radars transmit the signal without interruption, i.e., *continuously*. A good example of continuous wave radars is the “radar gun”, which is used by the police to measure the speed of passing vehicles by pointing the sensor at them. Since its signal is transmitted continuously, a conventional continuous wave radar cannot measure the distance of objects because there is no reference to measure the time delay. However, a special kind of CW radars, called Frequency Modulated Continuous Wave (FMCW) radar, is able to solve this problem.

Both pulse and continuous wave radars are “active”, that is, they emit their own illumination signal. In contrast, “passive” radars use illumination from the environment, such as commercial broadcast and communications signals. In theory, with densely deployed illuminators along roads (in other words, “street lights” for radars), radars in intelligent vehicles could operate in a receiver-only, passive fashion, which could alleviate inference problems [23]. However, this would require considerable investment and modification of infrastructure systems. Thus, passive radars in the automotive sector will remain at the experimental level for the time being, and I will not discuss them further in this thesis. Instead, since the vast majority of automotive radars are active, and in particular, use Frequency Modulated Continuous Waves (FMCW), I will focus on this category in this introduction, and I also use such radars in the following chapters.

FMCW radars are continuous wave radars, meaning they also transmit their radio signal continuously. However, the frequency of the signal is periodically modulated with a modulation function. One iteration of this periodic modulation is called a *chirp*. The simplest type of modulation functions (also called modulation signals) are linear, meaning that the frequency changes at a constant rate. See Figure 2.3 for examples of such signal modulations. In the top row I show the modulation signals that determine how the signal frequency changes. In the middle I show the so-called carrier signal, which is often a sinusoidal waveform with a frequency between 20-80 GHz. In the bottom row are the frequency modulated signals. It

can be seen that the waves of the final, frequency modulated signal become denser when the modulation signal has higher values. In a sense, by modulating the frequency of the signal, a “time stamp” is injected into the signal. We will see later that this makes it possible to measure the distance of objects with continuous wave transmission.

As mentioned earlier, most automotive radars are FMCW radars that can simultaneously measure the distance, direction, and speed of objects. Interestingly, all of these measurements are derived from the radar’s accurate ranging capability. In the following, I will discuss these measurements individually, starting with distance.

## 2.1 MEASUREMENT OF DISTANCE

FOR simplicity, suppose a single radar and a single object in an otherwise completely empty room, see Figure 2.4, top left. The radar emits a single chirp, which is then reflected by the object. Again, for simplicity, I assume a linearly modulated signal whose frequency increases at a constant rate during the chirp. The transmitted signal (also called TX) and the received signal (also called RX) could be plotted on a time-frequency diagram as shown in Figure 2.4, top right. With time, the frequency of the chirp transmitted increases, and after a while the chirp received also appears. It is the exact copy of the transmitted one, but it is delayed in time as it had to reach the object’s location and then propagate back to the sensor. Note that if the object is located at  $d$  distance from the sensor, the signal’s traveled distance is exactly  $2 \cdot d$ . Let us define  $\tau$  as the time it took the radio signal propagating with the speed of light to cover this distance. Given  $\tau$ , calculating  $d$  would be trivial, see Eq. (2.2). However, it is challenging to accurately measure  $\tau$  in automotive scenarios because the objects are orders of magnitude closer than in aerospace applications (meters vs. kilometers), and thus the time delays to be measured are significantly shorter. Furthermore, as mentioned before, with CW radars there is no clear time reference to be used to measure  $\tau$  as we transmit continuously, not in pulses. To address this, FMCW radars exploit the frequency modulation of the transmitted signal.

For this, I will introduce a signal processing unit, which we call a mixer, see Figure 2.5. The mixer takes two sinusoidal signals as inputs and combines them to produce a third signal. In this case, the frequency of the output signal is the difference of the two input frequencies and its phase is the difference of the two input phases. We call the output signal *intermediate frequency signal* or IF signal for short. Let us apply this mixer to the previously described TX and RX signals, see Figure 2.4, bottom right. Note that the third signal, the IF signal appears in the frequency-time diagram only as a horizontal line. This is because the difference between the frequencies of the two signals is constant, i.e.,  $w_1 - w_2$  is always the same. Even better, this constant frequency contains valuable information. That is, if  $S$  is the slope of the chirp, then this frequency (often referred to as beat frequency or  $f_b$ ) is:

$$f_b = S \cdot \tau. \quad (2.1)$$

In other words, the frequency of the IF signal is proportional to the time it takes the radio wave to reach the object and return. Recall that  $\tau$  is the time it takes the radio wave to travel the distance  $d$  to the object twice at the speed of light  $c$ :

$$\tau = \frac{2 \cdot d}{c}. \quad (2.2)$$

2

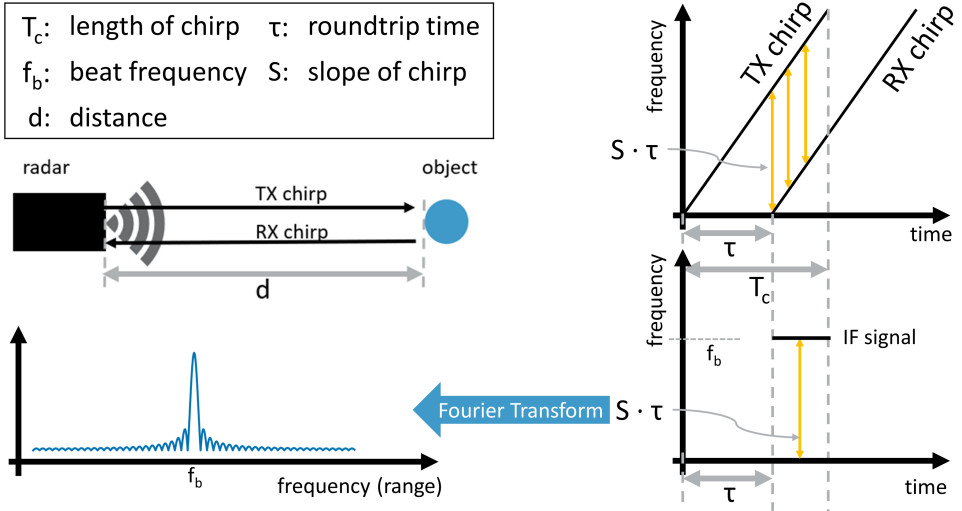


Figure 2.4: Overview of distance measurement of a single object located at distance  $d$  in front of the radar sensor. The radar emits an TX chirp which is reflected by the object. The two signals are then combined to form what is called an IF signal (see also Figure 2.5). The frequency of the IF signal,  $f_b$ , can be determined using the Fourier transform and is proportional to the distance of the object, see Eq. (2.3).

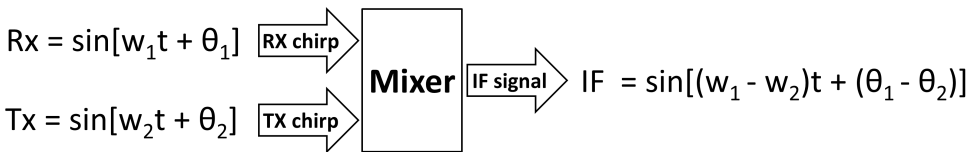


Figure 2.5: Overview of a frequency mixer. Such a mixer takes two sinusoidal input signals (TX and RX) and combines them to produce a third output signal, the IF signal. The IF signal is also sinusoidal and its frequency and phase are the difference between the frequencies and phases of the two input signals.

Then, we can use Eq. (2.1) to derive  $d$  from  $f_b$  and the slope of chirp  $S$ :

$$d = \frac{f_b \cdot c}{2 \cdot S}. \quad (2.3)$$

Since the slope  $S$  is fixed when the radar is parameterized, we only need to measure  $f_b$ . To obtain the frequency of a constant signal, we can use the Fourier transform. The resulting graph shows a high peak at the frequency we are looking for, see Figure 2.4, bottom left.

In summary, to determine the distance of a single object, we transmit a chirp and receive its reflection. If we then combine the two signals (both with changing frequency), we get a third signal called IF signal which has a constant frequency. If we subject this signal to a Fourier transform, we obtain the so-called beat frequency, which is directly proportional to the distance. An overview of this pipeline is given in Figure 2.4. In practice, these signals are discrete, and thus, Fast Fourier Transform (FFT) is used. This step is often referred to as “Range FFT”.

The maximum distance  $d_{max}$  that a radar can cover is an important parameter of the sensor. An object at this distance will yield an IF signal with the highest frequency the radar must be able to successfully sample, i.e., have a higher sampling rate:

$$F_s > f(IF_{d_{max}}) = \frac{2 \cdot S \cdot d_{max}}{c}, \quad (2.4)$$

where  $f(IF_{d_{max}})$  is the frequency of that IF signal, and  $F_s$  is the sampling rate of the radars analog to digital converter (ADC) unit. Thus, the furthest distance at which an object can be detected by the radar system is given as:

$$d_{max} = \frac{F_s \cdot c}{2 \cdot S}. \quad (2.5)$$

Note that so far we have only talked about the distance of a single object, not its direction. The object can be in any direction within the field of view of the sensor at the measured range. I will discuss direction estimation in Section 2.3.

Of course, having only one object in front of the radar sensor is not a realistic scenario. Therefore, we need to consider what happens when there are multiple objects in the scene. Multiple objects will reflect the TX chirp multiple times with different time delays, assuming they are at different distances from the sensor. This results in several different constant frequencies (or “tones”) in the IF signal. If one subjects such an IF signal to a Fourier transform, it yields a frequency spectrum that has several peaks (i.e., different  $f_b$ -s), each corresponding to the range of a perceived object. Radars’ capability to distinguish these objects in the frequency spectrum is called range resolution. It is important to note that the term resolution does not refer to the accuracy of the range measurement, but how close objects can be separated, or in other words, “resolved”. If two objects are too close to each other in terms of distance, the radar system will not be able to separate their  $f_b$  peaks from each other, and therefore will not be able to recognize them as separate objects. Due to the principle of Fourier transform, two signals with constant frequency can be resolved only if their frequency difference is greater than the reciprocal of the observation time. In our case this means:

$$\delta f > \frac{1}{T_c} \quad (2.6)$$

This means the resolution improves with longer observation times, i.e., with longer chirp length  $T_c$ . Using Eq. (2.1) and Eq. (2.2) we can express the minimal distance needed to get  $\delta d$  as:

$$\delta d > d_{res} = \frac{c}{2 \cdot S \cdot T_c} = \frac{c}{2 \cdot B} \quad (2.7)$$

where  $B = S \cdot T_c$  is the frequency range swept by the chirp.

Note that both  $d_{max}$  and  $d_{res}$  are inversely proportional to the slope  $S$  of the chirp (in the case of linear modulation), see Eq. (2.7) and Eq. (2.5). This is unfortunate because while we would like to maximize the maximum detection range  $d_{max}$ , we would like to minimize the resolution  $d_{res}$ . This trade-off also exists in automotive radars and is the reason that short-, medium-, and long-range radars are often used side by side, giving increasing range but degrading resolution respectively. In some cases, a single automotive sensor can be configured in multiple ways to suit the current use case. In this thesis, I use short-range radars and settings, since this is what our main scenario, the detection of road users in urban settings, requires.

## 2.2 MEASUREMENT OF VELOCITY

**N**OW we discuss how FMCW radars can measure the velocity of an object simultaneously with its distance. First, we need to emphasize that radars can only measure the radial velocity of objects (i.e., motion along the vector from the sensor to the object) and not the tangential velocity component (i.e., motion perpendicular to that vector) as shown in Figure 2.1. Therefore, in this section I discuss only the measurement of radial velocity and also explain why the direct measurement of tangential velocity is impossible with radar.

The (radial) velocity is basically the change in distance over time as the object comes closer or moves farther away to/from the sensor. In the radar-related literature, it is often referred to as the range rate because it is the rate at which the range of the object changes. Such change in the distance also changes the roundtrip travel time of the radio signal by a small amount of  $\delta\tau$ . Thus, if we could measure the moved distance or  $\delta\tau$ , the velocity could be calculated. However, radar measures range so frequently that the change in the range (i.e., displacement of the object) during that time period is not significant, and thus  $f_b$  would be identical.

Therefore, we need to find another way to derive velocity from this small change in distance. Recall that the mixer we presented subtracts the two frequencies of the two input signals, but it also subtracts their phases.

For a static, standing object (i.e., zero radial velocity) the IF signal contains a constant frequency. Let us consider what happens if the object moves along the radial direction. In this case the return time  $\tau$  will change a bit, i.e., RX arrives later if the object moved further, and arrives earlier if the object moved closer to the sensor.

In Figure 2.6, I plot the RX and IF signals of a static object in gray and the RX and IF signals of a moving object in blue. Note that the two IF signals have the same frequency, i.e. the objects are so close to each other in range that it is impossible to distinguish them. However, the phases of the two IF signals are completely different. Therefore, the phase is sensitive to small changes in distance, and thus, to velocity.

To fully understand the principles of velocity measurement, we need to recall some properties of the Fourier transform. Remember that the Fourier transform converts a signal

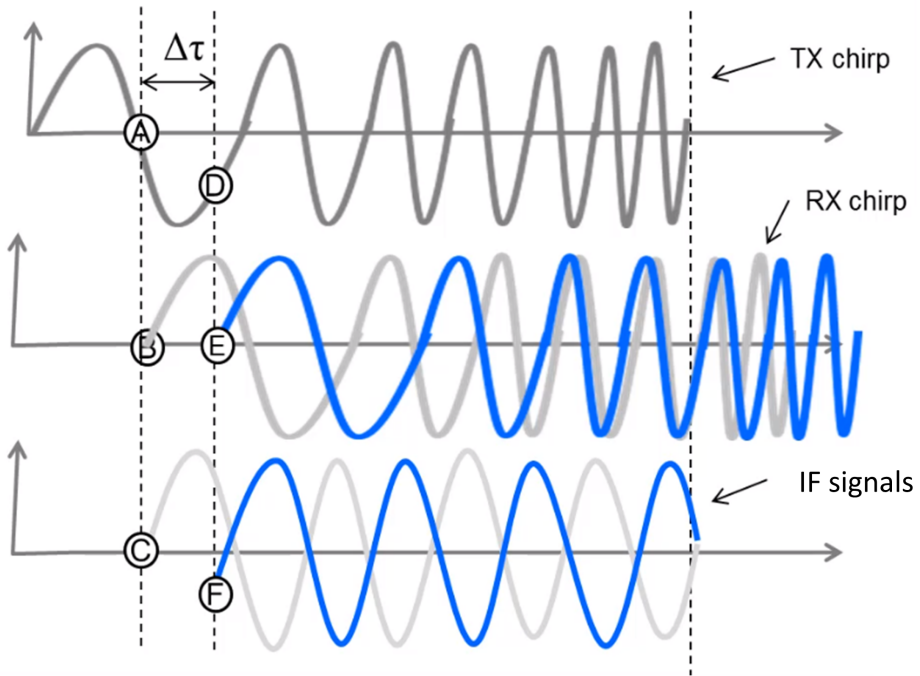


Figure 2.6: TX , RX, and IF signals of an object in its original position (grey), and after its distance changed slightly because of its movement (blue). While the frequency of the IF signal of the moved object is identical to that of the original, its phase differs significantly. This shows that the phase of the IF signal is sensitive to small changes in distance, i.e. motion. Source of image: [15].



from the time domain to the frequency domain, for example, a sinusoidal signal is converted to a single peak in the frequency domain. However, the output of the transform is actually a complex number. Each value is a so-called phasor with an amplitude and a phase value. The phase of the peak corresponds to the initial phase of the sinusoid.

Now that we know that phase is sensitive to small motions and that we can obtain the phase of the IF signal via Fourier transform, let us conclude how the radial velocity is measured. We transmit two chirp signals sequentially that are reflected from the moving object at slightly different distances. The frequencies of the IF signals, and thus the perceived distance of the objects will be identical. However, the phases of the signals will be different. This difference is proportional to the change in distance during a chirp and thus allows the speed of the object to be derived. Observing the phase over a period of time will again result in a periodic signal. Applying the Fourier Transform on this signal (often called the “Doppler FFT”) will provide the rotation frequency of the phasor (or phasors in case of multiple objects), which can be used to derive the object’s radial velocity. Without deriving the exact formula, the velocity is given by:

$$v = \frac{\lambda \cdot \omega}{4 \cdot \pi \cdot T_c}, \quad (2.8)$$

where  $\lambda$  is the wavelength of the carrier signal, and  $\omega$  is the phase difference measured between consecutive chirps. For detailed calculations, please refer to [15]. It is important to note that since velocity measurement is performed by phase difference, there will be ambiguity in the process. For unambiguous measurements, the phase difference  $\omega$  has to be smaller than  $180^\circ$  or  $\pi$ , otherwise the rotation direction of the phasor cannot be determined, in other words, the system cannot decide if an object is incoming or moving away. This puts a limit on the maximum velocity  $v_{max}$  that a radar system with a given configuration can measure:

$$v_{max} = \frac{\lambda \cdot \pi}{4 \cdot \pi \cdot T_c} = \frac{\lambda}{4 \cdot T_c}. \quad (2.9)$$

Velocities with greater absolute value than  $v_{max}$  “overflow” the measurement interval and reappear at the other (negative or positive) end of the velocity range. In other words, objects approaching/moving away too fast relative to the radar sensor may be detected as moving away/approaching respectively, resulting in significant measurement and tracking errors. Therefore, selecting an appropriate configuration for the radar that matches the use case is also crucial for the velocity measurements.

Remember that radars can only measure the radial velocity of object, not the tangential velocity. This is caused by the fact that the velocity is measured by the change in distance from the sensor. A tangentially moving object does not change its distance from the radar, and therefore does not induce changes in the perceived IF signal. Also, it is important to know that radars measure the relative velocity  $v_{rel}$  of objects, not their absolute velocity. With a static sensor (e.g., aerospace, speed control, and traffic monitoring), these values are the same. However, if the radar sensor itself is moving, which is often the case in automotive applications, the sensor’s own motion (often called ego-motion) must be taken into account. More precisely, by accounting for the motion of the sensor that comes from both the translational and rotational movement of the ego-vehicle, we can get the *compensated* or *absolute radial velocity*, a signed scalar value denoted by  $v_r$ . This processing step is called ego-motion compensation, and often performed for automotive related tasks.

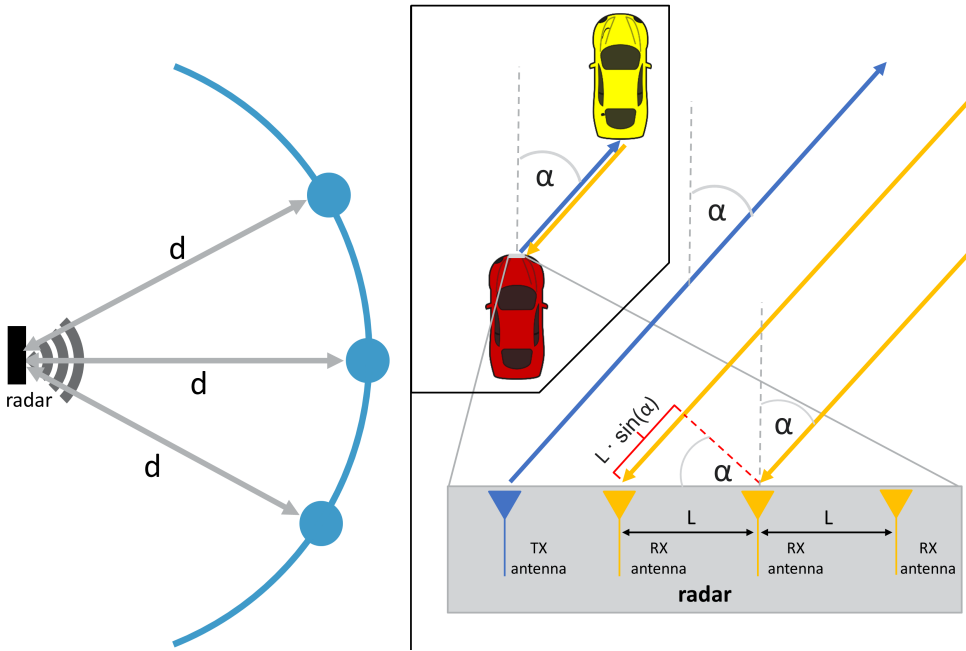


Figure 2.7: Angle estimation with automotive radars. A single antenna is not sufficient to determine the direction of an object because it can be anywhere on an arc at distance  $d$  from the sensor (left). With multiple receiver antennas (RX antennas), the object is detected at slightly different distances (right). These differences can be used to determine the direction. Note that a small change in the azimuth angle has a large effect on the distance difference around  $0^\circ$  and a significantly smaller effect around  $90^\circ$  because of the  $\sin(\alpha)$  dependency. Therefore, angle estimation in front of the sensor ( $|\alpha| \sim 0^\circ$ ) is more accurate than at the sides of the field of view ( $|\alpha| \sim 90^\circ$ ).

## 2.3 MEASUREMENT OF DIRECTION

**A**FTER distance and velocity, we now discuss how the direction of objects relative to the sensor, i.e., their azimuth or elevation angle is measured with radars. As mentioned before, the measurement of direction is also derived from the ranging capabilities of radars. Direction measurement cannot be made with a single antenna because the object could be in any direction on a circle (or sphere in three dimensions) with the measured distance radius, see Figure 2.7, left. In aerospace or marine applications, this challenge is often solved with a rotating directional antenna [18], since the delay caused by rotation is not problematic due to the relatively large distances involved. However, reaction time is more crucial in automotive applications, as objects are significantly closer. Therefore, most automotive radars do not use rotating or moving parts. Instead, a single sensor usually contains multiple antennas arranged in a grid. Each of these antennas performs the same task as described earlier: measuring the distance of nearby objects. However, depending on the direction of the objects, some antennas report a greater measured range than others, which allows us to determine the actual location of the object with triangulation, see Figure 2.7. Note that while the figure presents the regression of a single direction or angle in the horizontal plane, called azimuth, the same principle applies to the angle of elevation when one sets up the antennas

in a two dimensional grid instead of a single line.

The distance between antennas in an automotive radar is limited by the size of the sensor itself. In other words, the antennas are usually close together. This means that the difference in distance between the object and the individual observing antennas is small. Therefore, similar to velocity measurement, it is often estimated by measuring the phase difference of the signals. This means, however, that angle measurement, like velocity measurement before it, is also ambiguous outside a certain angular interval. Objects leaving this angular interval, in which an unambiguous measurement is possible on one side, are detected as returning on the other side. This can also lead to considerable measurement and decision errors and must be taken into account.

In Figure 2.7 (right), one can also see that the difference in the measured distances between the object and the antennas depends on  $\sin(\alpha)$ . Because of the derivative of the sine wave, this means that a small change in the azimuth angle has a large effect on the distance difference around  $0^\circ$  and a significantly smaller effect around  $90^\circ$ . Therefore, a consequence of this measurement approach is the fact that the angle estimate in front of the sensor ( $|\alpha| \sim 0^\circ$ ) is most accurate and deteriorates at the sides of the field of view ( $|\alpha| \sim 90^\circ$ ). This heterogeneous angular accuracy must also be anticipated in any object detection and tracking system using radars.

In practice, the direction estimation is a rather complex signal processing pipeline, often involving “virtual antenna arrays” and the use of super resolution algorithms to further improve the performance [24][25].

We must mention that because radar signals have larger wavelength than other sensors (see Figure 2.2), it can more easily “bounce” or reflect on objects, e.g. walls, or even the side of other cars, which further complicates direction estimation. In other words, the observed direction of an object could be incorrect if the radar signal was reflected on the way to the object or on the way back to the sensor. This phenomenon and its consequences will be discussed in more details later.

## 2.4 DIFFERENT TYPES OF RADAR DATA

THE previous sections gave a brief introduction to the topic of automotive or FMCW radars. It was shown that such sensors are capable of measuring the distance, direction, and relative radial velocity of nearby objects. Now we will discuss the various data formats in which radars can provide this information to users.

### 2.4.1 2+1D RADAR POINT CLOUD

Conventional automotive radars output a sparse point cloud of reflections. These reflections or points are also called *radar targets*. Each point has two spatial dimensions, range  $r$  and azimuth  $\alpha$ , and a third dimension referred to as Doppler, which is the radial velocity  $v_{rel}$  of the target relative to the ego-vehicle [26]. I will refer to this type of sensors as *2+1D radars* in this thesis, as they provide two spatial dimensions, plus one dimension is for Doppler. Each point is defined in this 2+1D space and is given a reflectivity *RCS* value. Since the points in the point cloud are often called radar targets or targets, the features associated with these points (i.e., range  $r$ , azimuth  $\alpha$ , Doppler  $v_{rel}$ , and reflectivity *RCS*) are often referred to as *target-level features*.

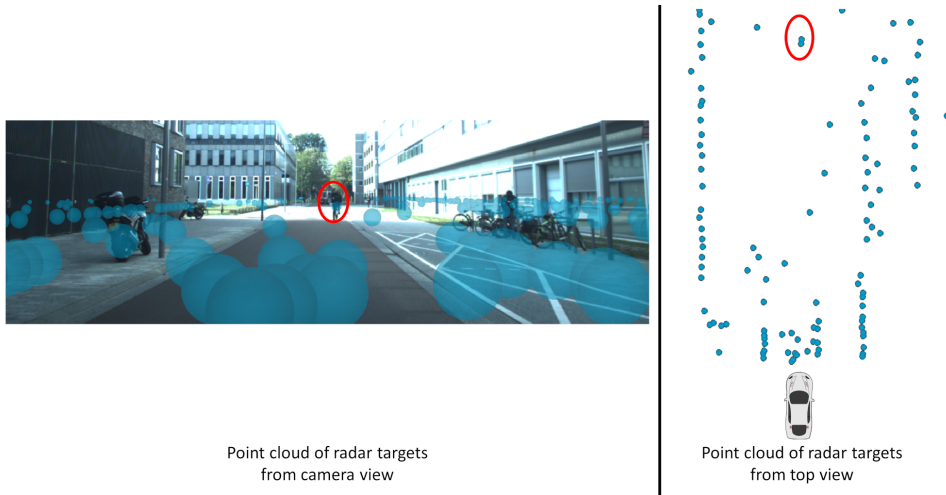


Figure 2.8: Demonstration of the output of a conventional 2+1D radar sensor. The same point cloud is projected onto the camera image (left) and shown in bird’s-eye view (right). Note how the linear structure of the walls and sidewalk (i.e., the curbside) is visible. Such sharp changes in elevation are often visible for the radar. The cyclist was manually highlighted with a red ellipse in both views. The road user has only two radar targets on him. This sparsity of 2+1D automotive radars is one of the major bottlenecks in radar based road user detection.

See Figure 2.8 for an example of the output of such a radar. The image on the right shows the points or targets from the top-down view. The image on the left shows the same targets projected to the camera image. The linear structure of the street, walls and sidewalk (i.e., the curbside) is clearly visible in the top view. Such sharp changes in elevation are often visible for the radar, as they reflect its emitted waves easier (see Section 2.6 for a detailed explanation). Note that for each point only one direction angle is given, namely the azimuth. Hence, these points are distributed in a horizontal plane in front of the sensor. In other words, each point in the camera view is projected at the same height above the ground, since we do not know their elevation. This, however, does not mean that they originate from the same height. In fact, the radar target can be located along a vertical arc (i.e., different elevation angles) at that range and azimuth angle. The density of these point clouds is rather sparse compared to a LiDAR sensor: usually the number of targets per frame varies between 100 and 200. For example, in Figure 2.8 I have marked a cyclist with red circles in both views. Note that there are only two points on the road user. As we will see later in this thesis, the low number of points on nearby objects is often considered one of the major bottlenecks in radar based road user detection.

### 2.4.2 RADAR CUBE

As described earlier in the discussion of the principles of FMCW radars, the extraction of points from the analog signals involves several peak finding steps, e.g., in the output of the range-, Doppler-, or Angle-FFT. Peak finding algorithms often include thresholds that must be properly configured. Such steps also filter out additional information in the form of nearby, smaller peaks. Many works [27][28][29][30] instead focus on “lower level”

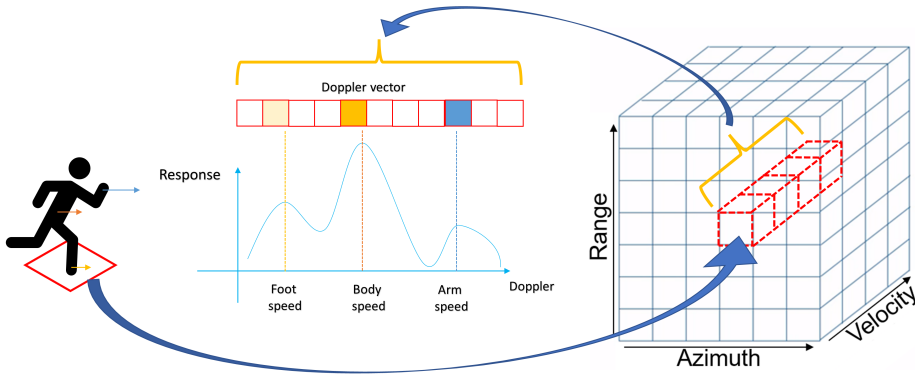


Figure 2.9: Overview of the radar cube. The radar cube is a three dimensional matrix with range, azimuth, and velocity axes. The value of a cell represents the reflectivity in that range/azimuth/Doppler bin. A road user may have multiple moving parts at the same spatial position (i.e., range and azimuth coordinates), e.g., swinging limbs. The column at its range/azimuth coordinate is called the Doppler vector and contains the signature of this complex motion.

radar data to eliminate as much of the thresholding/information loss as possible. Additional information can reveal patterns that are helpful in classifying objects.

The *radar cube* is a representation of such low-level radar data. It is a three dimensional data matrix with axes corresponding to range, azimuth, and velocity (also called Doppler). The value of a cell (i.e., an element in the matrix) represents the measured radar reflectivity in that range/azimuth/Doppler bin. Unlike target-level data, which contains a single velocity value (i.e.,  $v_{rel}$ ) for each radar target location, the radar cube provides the full velocity distribution (often called the *Doppler vector*) at multiple 2D range-azimuth locations. Such distributions can capture modulations of an object's main velocity caused by its moving parts, e.g., swinging limbs or rotating wheels, and have been shown to be a valuable feature for object classification [31][32]. A visual explanation of the radar cube and an example of a Doppler vector of a pedestrian is given in Figure 2.9.

Typically, radar cube features are used by first creating 2D range azimuth or range Doppler projections, or by aggregating the projected Doppler axis over time into a Doppler-time image [29][30]. I will refer to features derived from the radar cube or its projections as *low-level*. Low-level data is also commonly used for classification tasks (i.e., without determining the location of the object), especially using Doppler-Time images [27][28][29]. A disadvantage of such low-level radar data is the lower range and azimuth resolution than those provided by radar targets and the fact that radar phase ambiguity is not yet accounted for, since advanced range interpolation and direction-of-arrival estimation have not yet been performed. In addition, the use of the full radar cube requires the transmission and on-board processing of a significantly larger amount of data than a point cloud representation, which

is difficult to accomplish in a production vehicle. In my experiments, for example, the average radar cube message occupied  $\sim 21$  times more memory than the radar point cloud representation of the same scene.

### 2.4.3 3+1D RADAR POINT CLOUD

Thanks to recent improvements in radar technology, next generation automotive radars also determine the elevation of an object in order to correctly locate the object in three dimensional space. That is, unlike conventional automotive radars, 3+1D radars have three spatial dimensions: range  $r$ , azimuth  $\alpha$ , and elevation  $\theta$ , while still providing Doppler  $v_{rel}$  as the fourth dimension. I will refer to these type of sensors as *3+1D radars* in this thesis, as they provide three spatial dimensions plus one for Doppler.

While it is not necessarily a result of the additional elevation dimension provided, these next generation radars also tend to provide a denser point cloud than before [14]. With the additional elevation information and increased density, 3+1D radar point clouds are somewhat reminiscent of LiDAR point clouds. See Figure 2.10 for a comparison of 2+1D and 3+1D radar point clouds. The 3+1D radar point cloud is more dense, and includes height for each point. Therefore, the shape of objects can be more precisely determined.

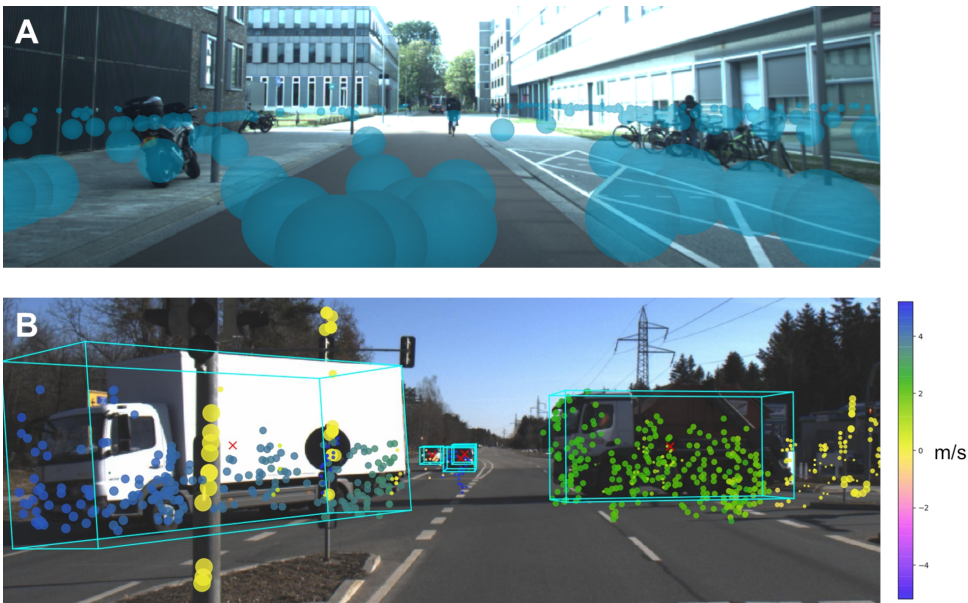


Figure 2.10: Example frame from a 2+1D radar (A), and from a 3+1D radar (B, from the Astyx Dataset [14]). Circles are radar reflection points projected into camera view, colored by their measured relative radial speed. Ground Truth bounding boxes are also plotted.

In this thesis, I have chosen to use N+1D terminology to indicate what type of radar sensor is being discussed at the moment, where N is the number of spatial dimensions provided by the radar, and the +1 stands for the Doppler dimension (if given). This was to avoid adding to the confusion already present in the literature. For example, 3+1D radars are

also often referred to as 4D radars [33] [34], because they have a total of four dimensions. However, there are papers that refer to these sensors as 3D radars, and to their output as 3D point clouds, e.g. [14][35]. On the other hand, calling conventional 2+1D radars “2D radars” can also be confusing, because some radars (often called scanning radars) return neither elevation nor Doppler values [36], effectively providing an actual two dimensional output. With the N+1D terminology used in this thesis, I aim to unambiguously clarify this issue and what type of sensors I am referring to.

## 2.5 ADVANTAGES OF RADAR SENSORS

**I**N the field of perception in intelligent vehicles, radar sensors have several advantages over camera and LiDAR sensors. One of the most important, as mentioned earlier, is the fact that radars can simultaneously measure the position and velocity of objects - something no other sensor can do. This can be exploited in various ways, such as predicting future positions, estimating current orientation, or even classifying an object in the scene based on its velocity profile, as we will see in Chapter 5 and 6.

Another advantage is that in contrast to camera, radar is an active sensor. That is, it relies on its own radiation source to illuminate the scene. This means that, unlike camera, radar is not affected by lighting conditions of the scene. In other words, radars can see in the dark, just like LiDARs.

As shown in Figure 2.2, the radar operates in a different, lower frequency range than the other two sensors'. This also means that its operating wavelength is longer. This has several consequences, some of which are beneficial. For example, a radio wave with a longer wavelength propagates more easily in unclean air, such as rain, snow, smoke, or fog. In other words, it is more robust to adverse weather conditions [37][38].

This longer wavelength has other consequences as well. For example, such waves are more easily reflected from common flat objects on the roads, such as walls, road surfaces, or sides of vehicles. This phenomenon is often referred to as multipath propagation, and it makes it possible to indirectly observe areas of the scene (e.g., behind a corner, behind a parked car) that are not in the line of sight and therefore cannot be observed by the camera, LiDAR, or even the naked eye [39][40].

Finally I should mention two benefits that are key in a production vehicle both for the manufacturer and for a potential customer: price, and the placement (i.e., location, visibility, and design) of the sensor. A decent radar sensor development kit is available to research groups for a few hundred USD. Mass production of such sensors reduces the cost to a fraction of that [39]. In contrast, LiDAR sensors cost at least an order of magnitude more than that [41]. Such a significantly lower price is an attractive factor in the automotive industry, where manufacturing costs are critical.

The possible locations and design constraints of a sensor are also important as they influence the aesthetics and aerodynamics of the car. Since most automotive radars have no moving parts, they occupy limited volume. In contrast, most LiDAR sensors are so-called scanning LiDARs, which have to rotate to cover their field of view, and thus require extra space. This often results in the placement of the sensors on the roof, hampering the design of the vehicle. Further, moving parts can affect durability and increase the maintenance cost of a sensor. In addition to their price and aesthetics, this is another reason why there is currently no production vehicle on the market with a rotating LiDAR sensor. There are also



Figure 2.11: Demonstration of the possible placement of sensors in an intelligent vehicle. The scanning LiDAR sensor requires space to rotate to cover its field of view. Therefore, they are often placed on the roof of research vehicles or robot taxis. Radar sensors, on the other hand, can penetrate the plastic material of the bumper and can therefore be mounted (and hidden) almost anywhere on the vehicle.

LiDAR sensors with few or no moving parts, often referred to as solid state LiDARs [42]. While these have limited resolution and field of view compared to their scanning pairs, they take up no more space than an automotive radar. However, radars have another advantage that facilitates their integration. Because of the longer wavelength mentioned earlier, radar waves can penetrate plastic materials that are often used in vehicle bumpers. In other words, radars, unlike LiDARs, do not require a window to function and can be hidden inside the chassis. Such a window is an obvious design limitation, but also something that must be kept clean for proper operation. See Figure 2.11 for a comparison of the placement of a high-end LiDAR sensor and a radar sensor in our research vehicle. While the rotating LiDAR must be placed on top and kept clear of blocking objects such that it can rotate freely, the radar behind the bumper is not visible at all to the observer.

In fairness, the LiDAR sensor used in our research vehicle and shown in the Figure 2.11, a Velodyne HDL-64, is relatively old. Since its debut, smaller rotating LiDARs have been introduced to the market, typically with a more practical, shorter cylinder shape, from both Velodyne and other manufacturers, such as Ouster. However, the claims of this section are still valid: these sensors have moving parts, and are larger, heavier, less cost-efficient, and more limited in possible placement than a radar sensor.

## 2.6 DISADVANTAGES OF RADAR SENSORS

Of course, radar sensors also have some disadvantages that must be anticipated when they are used in an intelligent vehicle.

First of all, most radar sensors provide point clouds of the environment, similar to LiDAR



or a stereo camera setup. The more valid points such a point cloud has, the more complete the picture it gives of the scene, including other road users. Unfortunately, this is not a strong side of radars: a conventional 2+1D radar outputs 100-200 points in a scan with 10-20 Hz, and even a next-generation 3+1D radar provides only a few times more than that. In contrast, as shown in Chapter 6, a high-end LiDAR sensor with 64 layers provides two orders of magnitude more points for the same area with similar refresh rates. The sparsity of radar point clouds is a serious bottleneck that will be addressed in multiple ways in this thesis.

Another limitation of radar comes from its longer operating wavelength. These signals are specularly reflected by the surfaces of objects, unlike visible and laser light beams that scatter in all directions. This means that only a fraction of the transmitted waves returns to the radar receiver [35]. Therefore, radar based perception is more sensitive to the shape of the objects than LiDARs. For example, a corner of a car may reflect part of the illuminating radar signal back to the sensor, while the flat side of the car reflects the signal but away from the sensor. Radar devices are also sensitive to the material of the illuminated object. That is, depending on the material, the radar signal may be reflected or partially penetrate through the medium reached. For example, objects made of metal, water (or water-filled), and rocks, in descending order, are good reflectors, while wood or plastic are poorly detected. While this property can sometimes be useful as a classification feature, it generally requires special attention and complicates development. For the reasons mentioned above, the points returned in each scan are not only sparse but also irregularly distributed, i.e., they do not follow a predefined scan pattern like LiDAR point clouds, resulting in a less consistent 3D image of the environment.

Next, the aforementioned bouncing property of radar signals, which is caused by the specular reflection discussed above and could make indirect observations possible, can be a disadvantage as well. That is, if a radar target is not known to have been reflected on a surface, its reported position will be incorrect. Such reflected and therefore incorrectly located radar targets are often referred to as “ghost targets” and create clutter in the point cloud. For example, consider an ego-vehicle following another car on a straight road, see Figure 2.12. The radar of the ego-vehicle can detect the car ahead directly or via a reflection on the guardrail. In the latter case, the detected car is erroneously located on the other side of the road. A real life example of such ghost targets is given in Figure 2.13. The side of the parked vehicle in front of the ego-vehicle acts as a “mirror” for the radar sensor. In this specific case, the reflected targets actually originate from the ego-vehicle itself. Note that there could be multiple of these indirect reflections between the actual target and the sensor, which further complicates the situation. Removing ghost targets is a growing research topic, see e.g. [43]. An interesting result of this phenomenon is the fact that 3+1D radars providing elevation information often report points below the road surface. This is caused by reflections that originate above the ground but are reflected somewhere on the road before returning to the radar sensor. See Figure 2.10 for an example. The cars in the distance have radar targets on them, but also below the ground.

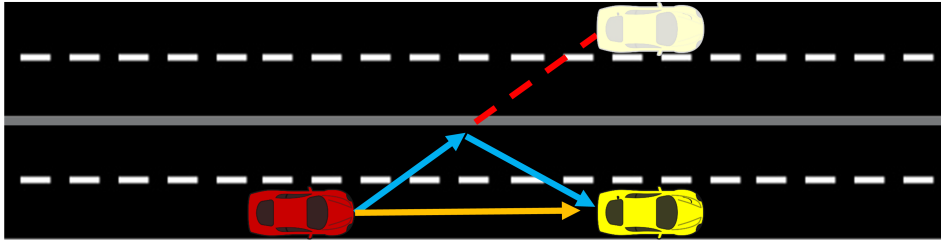


Figure 2.12: Illustration of ghost detections. The ego-vehicle (red) is equipped with a radar sensor. The radar can detect the car in front (yellow) directly (orange arrow) but also via a reflection on the guardrail (blue arrows). In the latter case, the radar will erroneously report the detected car to be also on the other side of the road (dimmed yellow). Such reflected and therefore incorrectly located radar targets are often called ghost targets.



Figure 2.13: Real world examples of ghost targets. On the left, targets are projected into the camera view. On the right is a top view of the scene with the stereo camera point cloud. Notice that the targets are farther away from the ego-vehicle (bottom of top view) than the parked car. This is because the radar signal is reflected off the large flat slide of the parked car, and the radar sensor is detecting the ego-vehicle itself. The colors of the targets indicate the outcome of the classification algorithm presented in Chapter 5. Namely, blue targets have the *car* class assigned to them, red targets have the *cyclist* class assigned to them.



# 3

3

## RELATED WORK

*The man who does not read good books is no better than the man who can't.*

Mark Twain

ROAD user detection is a critical step for any level of vehicle automation from Level 2 and up. Intelligent vehicles have multiple sensors to cope with this task: cameras [44][45][46], LiDARs [13][47], and radars [48][49][50] are used in the literature. Fusing different sensors, e.g. radar with camera [51] or LiDAR with camera [52][53], can further increase reliability and bring redundancy to such systems. In this chapter, I focus on radar based machine perception and review the most important related work, which I categorize based on the type of radar data used. I then discuss the various possibilities and use cases for which Doppler has been used in the literature. This is followed by an overview of publicly available radar datasets. Detection of occluded pedestrians about to cross the road (i.e., “darting”) is a special edge-case in the scope of general road user detection in intelligent vehicles. Nevertheless, this is one of the most difficult cases, causing far too many fatalities. Since the detection of such pedestrians is also the topic of one of the following chapters, I devote a separate section to methods dealing with occluded or darting pedestrians. Finally, this chapter concludes with related work on two topics that are not necessarily part of road user detection, but are nonetheless critical to successful implementation: tracking and environment modeling.

### 3.1 2+1D RADARS

CONVENTIONAL 2+1D automotive radars have been used for multi-class road user detection in various ways, such as using clustering algorithms [54][55], convolutional neural networks (CNNs) [39][49][56], or point cloud processing neural networks [50][57]. The sparsity of the point cloud provided by 2+1D radars is one of the largest bottlenecks of the radar perception domain. Researchers attempted to overcome this challenge and obtain more information in various ways, e.g.: by merging multiple frames over time [39][50], using multiple radars [58], or fusing radar with other sensor modalities [59][60][61][62].

Many road user detection methods start by clustering the radar targets into a set of object proposals based both on their spatial ( $r, \alpha$ ) and radial velocity ( $v_r$ ) features. Then, cluster-level features (i.e. statistical features describing the whole cluster) are extracted from each clusters. Finally, each object proposal (and their radar targets) is classified based on these features by a classification algorithm. In [55], radar targets are first clustered into objects by DBSCAN [63]. Then, several cluster-wise features are extracted, e.g. the variance/mean of  $v_r$  and  $r$ . The performance of various classifiers (Random Forest, Support Vector Machine (SVM), 1-layer Neural Network, etc.) were compared in a single-class (pedestrian) detection task. [54] also uses clusters calculated by DBSCAN as the base of a multi-class (*car, pedestrian, group of pedestrians, cyclist, truck*) detection, but extracts different features, e.g., deviation and spread of  $\alpha$ . Afterwards, Long Short-Term Memory (LSTM) and Random Forest classifiers were compared for the classification step. Falsely merged clusters were corrected manually to focus on the classification task itself. The same authors showed a method [64] to incorporate a priori knowledge about the data into the clustering. [65] also aims to improve the clustering with a multi-stage approach. [66] follows the work of [54] for clustering and classification, but tests and ranks further cluster-wise features in a backward elimination study.

While clustering based methods are widely used, it is often noted [50] [64] that the clustering step is error-prone. Objects can be mistakenly merged or split apart. Finding suitable parameters (e.g. radius and minimum number of points for DBSCAN) is challenging

as the same parameters must be used for all classes, although they have significantly different spatial extension and velocity profiles. For example, a larger radius is beneficial for cars, but could falsely merge pedestrians and cyclists. Another challenge of clustering based methods is that small objects may not have enough reflections to extract meaningful statistical features, e.g. variance. For example, both [54] and [55] set DBSCAN's minimum number of points to form a cluster larger than one, which means that single standing points are thrown away.

To address these challenges, there is a trend to classify each target individually instead of in clusters. Encouraged by the results achieved with semantic segmentation networks on point-clouds from LiDAR or stereo camera setups, e.g. Pointnet++ [67], researchers have tried to apply the same techniques to radar data. However, the output of a single 2+1D radar sweep is often too sparse. To overcome this, they used multiple frames [50] or addressed large road users (cars) only [57].

In general, most existing methods using 2+1D radars do not regress 2D or 3D bounding boxes for the detected objects. Instead, they perform semantic or instance segmentation of the sparse point cloud, i.e. they assign a class label (and potentially an object id) to each radar target individually [49][50][55][68][69][70].

## 3.2 RADAR CUBE

LOW-LEVEL radar data has been used for road user classification, especially for pedestrians. For example, a walking pedestrian's Doppler-time image contains a characteristic walking gait pattern [31][32]. This is beneficial to exploit if the radar sensor is stationary, e.g. in surveillance applications [27][28][29]. Doppler-time features were also used in automotive setups. [30] applies a CNN-LSTM network on Range-Doppler and Doppler-Time spectrograms of 0.5-2 seconds to classify *pedestrian*, *group of pedestrians*, *car*, and *cyclist* classes. [71] pointed out that a long multi-frame observation period is not viable for urban driving, and proposed a single-frame usage of low-level data. Their method still generates object proposals with DBSCAN similar to [54][55], but extracts for each cluster the corresponding area in a 2D Range-Doppler image, which is then classified using conventional computer vision steps. In [56], the full radar cube is used as a multi-channel image input to a CNN network to classify *cars*, *pedestrians*, and *cyclists*. The study only addresses a single-object classification task, i.e. location is not fetched.

## 3.3 3+1D RADARS

GIVEN the limited availability of 3+1D radar sensors, only a handful of studies in the literature have used them. [72] introduced a deep learning-based method to generate the 3+1D radar point cloud from lower-level radar data. Similarly, [26] also described a signal processing pipeline to obtain 3+1D radar point clouds and showed real-life qualitative examples of such a point cloud in highway scenarios. [73] introduced a continuous-time calibration tool for 3+1D radar and monocular camera. [43] used a modified Pointnet network to classify ghost targets in a 3+1D radar point cloud using the LiDAR point cloud as supervision. In [33] a 3+1D radar classification dataset was introduced along with a new neural network designed for this classification task.

Even fewer works used 3+1D radars for object detection tasks. In [74] the authors applied such a sensor to build a static 3D occupancy map of highway and parking lot scenes

after filtering out dynamic targets. Afterward, the map is semantically segmented by image segmentation networks into the street, curbstone, fence, barrier, or parked car classes. Before the work in Chapter 6, the only publicly available automotive detection dataset that contains 3+1D radar data was the Astyx dataset [14]. Despite the small size of the dataset (~500 frames), the authors have successfully used it to perform 3D car detection by fusing radar and camera with the AVOD fusion network [75]. Furthermore, they also compared this radar-camera fusion with LiDAR-camera fusion, although the LiDAR sensor had only 16 layers. Finally, [35] used the combination of two spatially separated low-resolution 3+1D radars to detect vehicles by a novel neural network called RP-net, containing several Pointnet layers. To the best of my knowledge, 3+1D radars have neither been used for multi-class road user detection before, nor have they been compared to high-end LiDAR sensors.

### 3.4 THE USE OF DOPPLER

As introduced in Chapter 2, radars, unlike cameras and LiDARs, are capable to directly measure relative radial velocity (often called the *Doppler*) of objects by the Doppler effect. It is interesting to look at the literature from the point of view of how this unique property of radars was used. The most trivial use of Doppler is to distinguish static and dynamic objects in the environment after ego-motion compensation. E.g., while some research only keeps static radar targets [74][76][77][78], others use the Doppler information to keep only moving reflections to detect dynamic objects [30][39][40][49]. After first clustering the radar point cloud to generate object proposals, basic statistical properties (mean, deviation, etc.) of the velocity spectrum can be used for classification [54][55]. [50] presented in an ablation study that adding Doppler as an input channel to a Pointnet++ network significantly improves semantic segmentation. [49] showed that the (relative) velocity distribution contains valuable class information which can be exploited for multi-class road user detection. With multiple radar targets originating from the same object, it is also possible to regress the 2D velocity vector (and thus, orientation) of the object using the targets' measured radial velocities as samples at different azimuth angles, as [79] showed for cars and [80] for bikes. Thus, it has been shown that the Doppler dimension can be beneficial in 3D object detection in two ways: 1) classification, as classes may have distinct velocity patterns [49][50], and 2) in orientation estimation, as the general velocity (moving direction) of an object is highly correlated with its orientation [79][80]. Despite its advantages, in the few works that used a 3+1D radar sensor, Doppler was either ignored [75], used to filter static radar targets [74], or used as an additional input channel in a point cloud processing network without ego-motion compensation [35]. Although Doppler has been shown to be beneficial for multi-class road user detection using conventional 2+1D automotive radars, 3+1D radars have only been used for single-class (vehicle) detection in the literature [35].

### 3.5 RADAR DATASETS

RECENTLY, several automotive datasets containing radar data have been published for various tasks, such as localization [81][82], object classification [33], or scene understanding with a stationary radar sensor [83]. In this section, I focus on detection datasets that contain realistic recordings from a moving ego-vehicle. Both the RadarScenes [84] and CRUW [85] datasets contain 2+1D radar and camera data and have a large number

of annotations for all three main classes: cars, pedestrians, and cyclists. CRUW contains low-level radar data in the form of range-azimuth images and provides only the 2D BEV position and class of annotated objects. RadarScenes, on the other hand, provides radar data in the form of processed 2+1D radar point clouds. Similar to CRUW, RadarScenes does not have 2D BEV or 3D bounding box annotations. Instead, the authors chose to label each radar target individually. Targets originating from static road users (e.g., parked cars) were not assigned to a class, but were labeled as static. Unfortunately, neither RadarScenes nor CRUW provides LiDAR data. The RADIATE dataset [36] contains radar, camera, and LiDAR data along with 2D BEV bounding box annotations for all three classes. It was collected using a mechanically rotating 2D radar, which provides a 360° dense image of the environment, but does not output Doppler or elevation information. The Zendar dataset [86] provides Synthetic Aperture Radar (SAR) data using a 2+1D radar. While the SAR technique is a well-known method to acquire additional information and resolution about static objects, it is less applicable in dynamic object detection. Unfortunately, the dataset only has annotations for the car class. The nuScenes dataset [87] contains data from all three sensor modalities, and they provide a large number of 3D bounding box annotations. However, the output of the equipped 2+1D radar sensors is considered too sparse for radar-only detection methods by some in the research community [26][84], and the used LiDAR sensor has only 32 layers. Finally, the Astyx dataset [14] is the only one to use a 3+1D radar, and it also contains data from a camera and a 16-layer LiDAR. Unfortunately, its limited size (~500 frames) and highly imbalanced classes (e.g., only 39/11 pedestrians/cyclists annotations) make it ill-suited for multi-class object detection research.

## 3.6 DARTING OUT PEDESTRIANS

**I**N this section, I discuss camera-, radar-, and fusion based methods for pedestrian detection, with a focus on darting out scenarios and occluded pedestrians.

### 3.6.1 CAMERA BASED APPROACHES

Cameras are often used for pedestrian detection as they provide rich information while being relatively inexpensive. In recent years, convolutional neural networks (CNNs) and deep learning methods [45][46] dominate in this field. The problem of occlusion is widely recognized, e.g., many benchmarks define separate metrics for different levels of occlusion [46][88]. For an overview of camera based methods that consider occlusions, see [89]. Several approaches aimed to explicitly account for occlusions by learning a set of component detectors and fusing their results to detect partially occluded pedestrians [90][91][92][93]. More recently, researchers proposed special loss functions [94][95] or top-down approaches [96] to jointly estimate the state of close-by pedestrians occluding each other, especially in challenging, crowded scenes. Other methods introduced hard negative mining to increase the occlusion tolerance of networks [97], or proposed to explicitly collect more training data of partially occluded pedestrians [46] to address the problem. However, none of these methods used a global environment model to describe the occlusions that may affect the number and attributes of detections, e.g., result in no/fewer detections behind cars. See Section 3.7 for more details about environment modeling.



### 3.6.2 RADAR BASED APPROACHES

[98] presented a tracking method using track-before-detection and particle filtering. The system was also tested in scenes of the pedestrian entering and exiting an occluded region behind a car. The radar was able to provide measurements even in the occlusion. However, the occlusion itself was not considered, and although they compared the performance to a camera based detection system, no fusion occurred. In [99], a binary classification system of pedestrians and static objects was presented that uses low-level radar data as input and extracts hand-crafted features. The system was evaluated using darting out scenarios, but no sensor fusion was used, nor was occlusion investigated as a possible source of information. [39] exploited that radar signals often “bounce” off large flat surfaces. They showed that it is possible to detect moving road users outside the direct line-of-sight with reflected radar measurements by using building facades or parked vehicles as relay walls. In [100], the authors explicitly addressed the detection of fully occluded, darting out pedestrians with radar. They designed an experimental setup with a static radar sensor in an indoor area (the pedestrian was behind a corner) and an outdoor area (the pedestrian was behind a van). Movement of the occluded pedestrian is then classified by clustering into different behavior types, such as walking towards, walking out of it, and walking inside the occluded region. None of these methods considered occlusion as a source of information, and none of them compared or fused camera and radar sensors to detect darting out pedestrians in realistic environments, i.e., from a moving ego-vehicle.

### 3.6.3 FUSION BASED APPROACHES

SENSOR fusion was extensively researched to provide more robust perception solutions either via model-based (mathematical, e.g., Kalman or particle filters, evidence modeling) [51][101][102], or data-driven approaches (e.g., with neural networks) [103][104]. In this subsection, I focus on fusion systems that use radar, with particular attention to whether and how these systems address occluded pedestrians. In [29], LiDAR and radar were fused to detect pedestrians in a static experimental setup. First, a binary occlusion map of the scene was created by detecting occluding objects with LiDAR. This map was then used to select which sensors to use for detection: both sensors for unoccluded regions, and purely radar for occluded regions, exploiting its multipath property. In [102], all three sensors were combined in a multi-class system for detecting moving objects, including pedestrians, in an intelligent vehicle setup using an occupancy grid representation. The LiDAR was used as the main sensor to detect moving objects, while camera and radar were mainly used for classification. The influence of occlusions was not considered. None of the fused systems found were developed for use in intelligent vehicles to address darting out scenarios, or considered occlusion as a source of information beyond helping sensor selection.

## 3.7 ENVIRONMENT MODELING

MODELING occluded areas in the environment is often done in bird’s-eye view (BEV). A common approach is to aggregate range measurements from radar or LiDAR sensors into a 2D occupancy grid and then project “shadows” behind the extracted objects [105][106]. For automated driving applications, creating an environment model with information from on-board camera sensor(s) can lead to a faster process (i.e., it does not need to be accumulated)

and provides more information about the nature of the occluding object (e.g., whether it is a car) due to the rich texture information. In [40], the goal was to explicitly model only the occlusions caused by (parked) vehicles. To this end, 2D detections in the image plane were fetched from the *car*, *bus*, *truck*, and *van* classes from the Single Shot Multibox Detector (SSD) [107]. Depth (i.e., distance from the ego-vehicle) was estimated by projecting the bounding boxes into the stereo point cloud and taking the median distance of the points from the camera inside the box, resulting in “2.5D” detections (i.e., they are lines in bird’s-eye view). Areas behind these detections were considered occluded, creating a binary map. While this solution resulted in fast processing time and contributed to earlier detection of darting out pedestrians in the experiments, it also had some drawbacks. By assigning a single distance to the entire occluding vehicle, parts of the vehicle closer/farther than that distance are incorrectly considered “regular” unoccluded/occluded (but still walkable) regions. However, a pedestrian cannot be physically present in either of these halves. Modeling occlusion with a bounding box also has limitations in width and height, e.g., a pedestrian may be more visible behind the shorter parts of a car than behind its tallest point, but these two cases are treated identically.

An alternative camera based approach to creating a more accurate occlusion model that is still computationally efficient may be to use stixels [108]. Stixels are rectangular column-wise group of pixels based on disparity information with the goal of reducing the complexity of the stereo point cloud. Since the original publication [108], researchers have integrated class information [109] and later instance information into stixels [110]. The latter are referred to as Instance Stixels and could be a well suited input for an occlusion model because they follow the shape of an occluding car (both in depth and width/height) and are still computationally efficient to compute and process. In addition, the same Instance Stixels representation can also serve as input to a road user detection and tracking system by providing the location and height of the object, e.g., a pedestrian or cyclist.

### 3.8 TRACKING

**R**OAD users are often tracked with Kalman Filters both in camera based [111] and radar based [30] detection systems. Kalman filters can only model linear motion. Situations with possibly non-linear motion dynamics, e.g., a pedestrian who may or may not stop at the road side, can be handled by using an “extended” Kalman Filter, or by switching between multiple linear motion models with a switching dynamic system [111]. Another commonly used method for pedestrian tracking is the particle filter [98][112][40][113] which estimates the posterior distribution over the state space using a set of weighted particles. Analogous to Kalman Filters for single-object tracking, probability hypothesis density (PHD) filters can be used to jointly estimate the non-constant number of targets and their states [114].

Unlike Kalman Filters, a particle filter can handle non-linear motion dynamics, and can represent arbitrary, potentially multi-modal distributions. To satisfy our use case (detecting and tracking a road user), a filter should not only track an object of interest (e.g., a pedestrian), but also report a probability that it is present in the scene. [112][115] give solutions to incorporate this existence probability into particle filters.



## 4



# DETECTION OF DARTING OUT PEDESTRIANS WITH FUSION OF CAMERA AND RADAR

4

*There are relatively few things that kill people that  
are young other than car accidents and suicide.*

Kay Redfield Jamison

---

This chapter is based on  A. Palfy, J. F. P. Kooij, and D. M. Gavrilă, “Occlusion aware sensor fusion for early crossing pedestrian detection,” *IEEE Intelligent Vehicles Symposium*, pp. 1768–1774, 2019 [40], and  A. Palfy, J. F. P. Kooij, and D. M. Gavrilă, “Detecting darting out pedestrians with occlusion aware sensor fusion of radar and stereo camera,” *Accepted to IEEE Transactions on Intelligent Vehicles*, 2022..

## 4.1 INTRODUCTION

ABOUT 23% of the 1.35 million traffic fatalities world-wide involve pedestrians [8]. Automated driving has the potential to significantly reduce these traffic deaths, yet the sensor-based detection and tracking of pedestrians from a moving vehicle remains challenging. Pedestrians have a wide variation in appearance, can quickly alter their course, and can step onto the road at pretty much any location.

Intelligent vehicles can use multiple sensors to cope with this task: cameras [44][45][46], radars [48][49][50] and LiDARs [47][13]. Fusing different sensors, e.g., camera with radar [51] or camera with LiDAR [52], can increase the reliability and redundancy of such systems. In this chapter, I consider the fusion of a (stereo) camera with a radar. These are low-cost sensors with complementary strengths that are well established in driver assistance context on the market. Cameras provide color/texture information at a fine horizontal and vertical resolution. Radar sensors provide accurate depth information, can directly measure the radial velocities and are more robust to adverse weather and lighting conditions.

Pedestrian sensing is often complicated in urban scenarios by occlusions, such as by parked vehicles. A substantial 26% of the accidents with crossing pedestrian analyzed in [116] involved some form of visual occlusion. In fact, this case is so important that the consumer advocacy group Euro NCAP designates a special test scenario for it, titled “Running Child from Nearside from Obstruction” [117]. This case of a pedestrian *darting out* [118] is illustrated in Figure 4.1. It is particularly dangerous because neither a human driver nor the pedestrian have initially a clear, direct view of the other. Similarly, in an automated driving setting, a parked vehicle would block direct line-of-sight from the sensors of the ego-vehicle to the pedestrian. However, the extent of this blockage depends on the sensor’s type and on the size and shape of the occlusion.

A camera may see the upper body of a pedestrian behind a passenger car, while a person behind a larger vehicle, such as a truck or a van may be invisible to the sensor. On the other hand, commercially available 2+1D radars, which provide two spatial dimensions (range and azimuth) and one dimension for Doppler (radial velocity), are often able to detect the reflections of a pedestrian even in complete occlusion due to multipath propagation [99][39]. That is, the reflected radar signal may “bounce” off other parked cars or the ground beneath the occluding vehicle and reach the ego-vehicle’s sensor even if there is no direct line-of-sight. Such indirect reflections are weaker and occur less frequently than direct ones[99], but they could still provide valuable information about a potentially darting out pedestrian.

Since both camera and radar sensors are affected by occlusions, their fusion preferably requires an occlusion model that describes how many detections to expect from each sensor in the differently occluded areas of the scene (e.g., to expect fewer detections behind cars). In addition, the occlusion model could also provide information about the expected properties of such detections, e.g., that the visible part of a partially occluded pedestrian may be smaller than an unoccluded one. The stereo camera is a suitable sensor to create this occlusion model because it provides rich and dense textural and depth information that can help accurately detect and model the occluding vehicle itself.

In this chapter, I present a Bayesian occlusion aware sensor fusion system designed to detect darting out pedestrians. I show that incorporating an occlusion model into such a sensor fusion system helps to detect darting out pedestrians earlier; thus precious time is

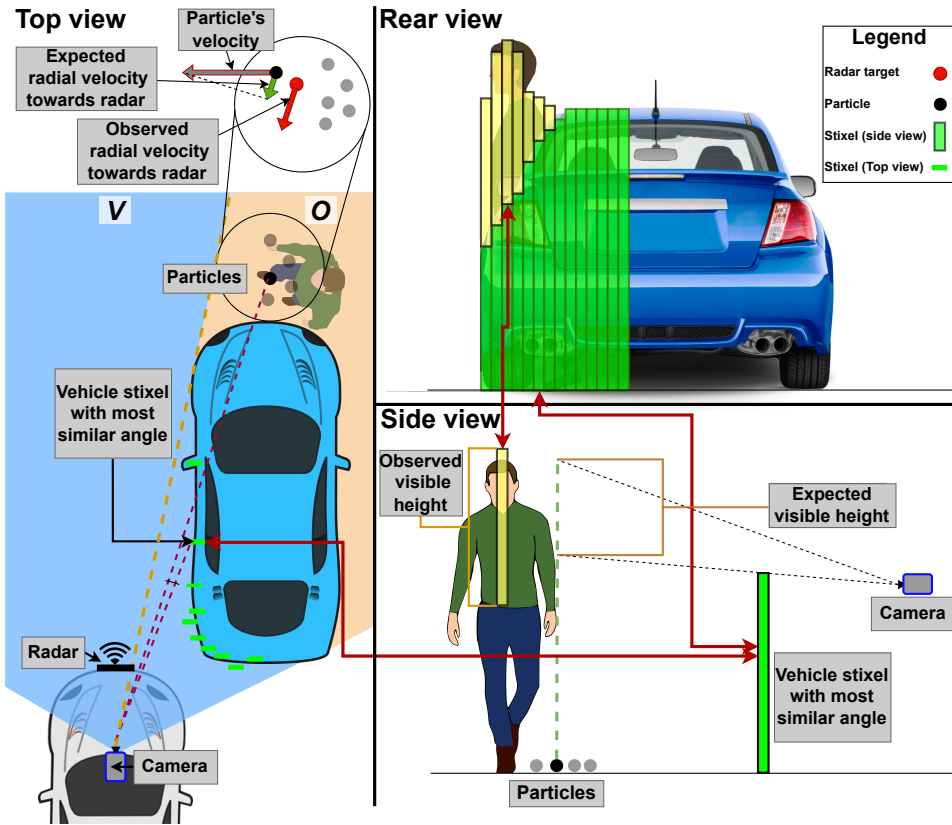


Figure 4.1: Darting out scenario: a pedestrian steps out from behind a parked car (blue) which blocks the line-of-sight of the ego-vehicle (white). I propose to detect such pedestrians with the fusion of stereo camera and radar in an occlusion aware way, i.e., first building an occlusion model of the environment and then expecting fewer and different detections (e.g. shorter visible parts of the pedestrian) from the occluded regions ( $O$ ) than from the visible, unoccluded ones ( $V$ ).

gained to initiate emergency braking or steering, if needed. While I consider the fusion of (stereo) camera and radar, the framework is suitable to integrate other sensors, e.g., LiDAR.

The chapter is structured as follows. In Section 4.2, I present my generic occlusion aware Bayesian multi-sensor fusion filter. Details of how this filter was implemented with radar and stereo camera sensors, and applied to darting out scenarios are discussed in Section 4.3. Section 4.4 describes the dataset that was created and used for this work. In Section 4.5, I present my experiments and results, which are discussed in-depth in Section 4.6. Finally, Section 4.7 concludes the chapter. For an overview of relevant works, please refer to Section 3.6.

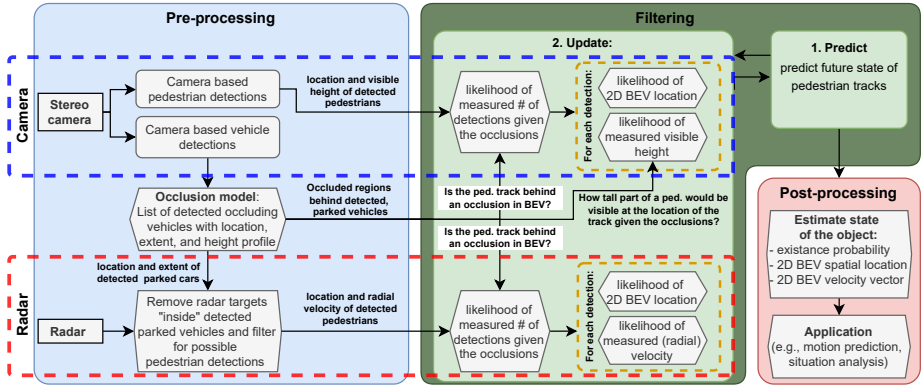


Figure 4.2: Overview of my pipeline. Sensor data is processed to get the pedestrian detections and the occlusion model (**Pre-processing**). Future states of the object in the filter are predicted, and then their likelihood are updated with the detections considering the occlusion model (**Filtering**). The estimated existence probability and state of the object are calculated, which information can be used in subsequent applications (**Post-processing**). Blue/red dashed boxes mark camera/radar specific steps that are described in Section 4.3.

## 4.2 PROPOSED APPROACH

### 4.2.1 OVERVIEW AND CONTRIBUTIONS

The goal of this chapter is to fuse radar and stereo camera (Figure 4.2, blue and red dashed rectangles) by incorporating occlusion information to detect darting out pedestrians. To this end, I propose a generic Bayesian filter to fuse these sensors in an occlusion aware manner. This estimates not only the 2D position and velocity of an object's center on the ground plane (i.e. BEV), but also the probability that the object of interest (i.e., a pedestrian) is present in the scene. The used state space will be discussed in details in Subsection 4.2.2.

First, in a prediction step, I define the prior distribution of the filter given previous measurements (Figure 4.2, Predict step). The distribution of predicted positions and velocities are defined for three cases: a new object entering the scene, an object leaving the scene, and finally a tracked object remaining in the scene. Please refer to Subsection 4.2.3 for details.

After the prediction step, the new detections are fetched from each sensor (Figure 4.2, Pre-processing) and incorporated in the update step (Figure 4.2, Update step), which is discussed in details in Subsection 4.2.4. I assume conditional independence of the sensors given the true state of the object, thus I can perform the update with their set of detections individually whenever they arrive, even if the sensors operate asynchronously at different frame rates. I describe here the update in a generic way and define sensor specific details, e.g. measurement models, later in Section 4.3.

When updating with any of the sensors, its  $K$  detections (as determined by its measurement model) are fed into the filter. This updates the likelihood of the hypotheses in two ways. First, the likelihoods of measuring  $K$  detections with this sensor are calculated. Since the number of detections depends on the position of the object, I can incorporate information from an occlusion model here. That is, my system adjusts the expected number of detections to the visibility of a position and expects more/less detections at unoccluded/occluded locations, see Figure 4.1. Second, I also consider the unique capabilities of the sensors. That is, I estimate the likelihood of the attribute of the detection based on the estimated state

of the object. Here I could use, for example, the velocity measurement of a radar or the classification confidence of a camera. I can also evaluate the size of the visible part of a pedestrian given its assumed occlusion condition.

The occlusion model can be retrieved from a single or from a combination of sensors, or from an independent source. In this chapter, it will be provided by the stereo camera, see Section 4.3.

Finally, after the filtering, the object's probability of existence and state (i.e., 2D BEV center location and velocity) can be estimated and used in subsequent processing steps, e.g., in predicting future positions or evaluating the dangerousness of the scene (Figure 4.2, Post-processing).

My contributions are as follows.

1. I propose a generic occlusion aware multi-sensor Bayesian filter for object detection and tracking.
2. I apply the proposed filter as a radar and stereo camera based pedestrian detection and tracking system on challenging darting out scenarios. I show that incorporating occlusion information and the radar sensor into my model helps to detect darting out pedestrians earlier while keeping the number of false alarms low when the pedestrian stays behind the car.
3. I share my dataset<sup>1</sup> containing more than 500 relevant scenarios with camera, radar, LiDAR, and odometry data.

This work builds upon our previous conference publication [40], where we initially proposed an occlusion aware Bayesian filter for darting out pedestrians based on stereo camera and radar. This work features an improved sensor measurement model (incorporation of additional attributes besides location, see Subsections 4.3.2 and 4.3.3). Among others, the occlusion extent is now more accurately represented by a height profile derived from instance segmentation rather than by a bounding box derived from an object detector (see Section 3.7). In terms of validation, this work features a significantly enlarged dataset and added experimentation.

### 4.2.2 STATE SPACE AND NOTATIONS

Now I discuss the mathematical formulation of my proposed generic occlusion aware, multi-sensor Bayesian filter without sensor related specifics. Let the space  $\mathcal{T}$  consist of a 2D (lateral and longitudinal) position and velocity, and a binary flag marking if the tracked object (e.g., a pedestrian) exists. Let  $\mathbf{h}$  be a state vector in  $\mathcal{T}$  (vectors are written in boldface):

$$\mathcal{T} : \mathcal{R} \times \mathcal{R} \times \mathcal{R} \times \mathcal{R} \times \{0, 1\}, \quad (4.1)$$

$$\mathbf{h} \in \mathcal{T}, \mathbf{h} = (\mathbf{x}, \mathbf{v}, \mathcal{E}), \quad (4.2)$$

where  $\mathbf{x} = (x, y)$  and  $\mathbf{v} = (v_x, v_y)$  are the object's 2D BEV position and velocity vectors on the ground plane, and  $\mathcal{E}$  represents the existence probability. I.e.,  $\mathcal{E} = 1$  means there is a pedestrian in the scene and  $\mathcal{E} = 0$  represents its absence.

<sup>1</sup>The dataset is freely available at <https://intelligent-vehicles.org/datasets/> to academic and non-profit organizations for non-commercial, scientific use.



I define a Bayesian filter for detection and tracking which estimates the posterior state distribution  $P(\mathbf{h}_t|\mathcal{Z}_{1:t})$  given all measurements  $\mathcal{Z}_{1:t}$ . The filter operates on-line, integrating new measurements into a posterior using Bayes' theorem:

$$P(\mathbf{h}_t|\mathcal{Z}_{1:t}) \propto P(\mathcal{Z}_t|\mathbf{h}_t) \cdot P(\mathbf{h}_t|\mathcal{Z}_{1:t-1}), \quad (4.3)$$

where  $\mathcal{Z}_t$  is the set of all sensor detections at current time  $t$ . Here the prior distribution  $P(\mathbf{h}_t|\mathcal{Z}_{1:t-1})$  for time  $t$  is obtained by applying a state transition probability on the previous posterior, and integrating over the previous state  $\mathbf{h}_{t-1}$  following the Chapman-Kolmogorov equation:

$$P(\mathbf{h}_t|\mathcal{Z}_{1:t-1}) = \int P(\mathbf{h}_t|\mathbf{h}_{t-1}) \cdot P(\mathbf{h}_{t-1}|\mathcal{Z}_{1:t-1})d\mathbf{h}_{t-1}. \quad (4.4)$$

I am thus required to define the state transition distribution  $P(\mathbf{h}_t|\mathbf{h}_{t-1})$  for the filter's prediction step, and measurement likelihood function  $P(\mathcal{Z}_t|\mathbf{h}_t)$  for the update step, which I will derive in the following subsections. Note that the posterior contains the expected existence probability of a pedestrian in the scene:

$$P(\mathcal{E}_t|\mathcal{Z}_{1:t}) = \iint P(\mathbf{h}_t|\mathcal{Z}_{1:t})d\mathbf{x}_td\mathbf{v}_t. \quad (4.5)$$

### 4.2.3 PREDICTION STEP

The state transition distribution is factorized into two terms:

$$P(\mathbf{h}_t|\mathbf{h}_{t-1}) = P(\mathcal{E}_t|\mathbf{h}_{t-1}) \cdot P(\mathbf{x}_t, \mathbf{v}_t|\mathcal{E}_t, \mathbf{h}_{t-1}). \quad (4.6)$$

The first term estimates the object presence flag  $\mathcal{E}$ . A new object can appear with a probability of  $p_n$ . Unlike  $p_n$ ,  $p_s(\mathbf{h}_{t-1})$ , the probability that an object stays in the scene depends on the previous state  $\mathbf{h}_{t-1}$ , because the position of the object affects the probability that it will suddenly leave the region of interest. Using these, I can determine the probability of  $\mathcal{E}$  given the previous state  $\mathbf{h}_{t-1}$  for entering (new), not present and not entering, staying, and leaving objects respectively:

$$P(\mathcal{E}_t = 1|\mathcal{E}_{t-1} = 0, \mathbf{h}_{t-1}) = p_n, \quad (4.7)$$

$$P(\mathcal{E}_t = 0|\mathcal{E}_{t-1} = 0, \mathbf{h}_{t-1}) = 1 - p_n, \quad (4.8)$$

$$P(\mathcal{E}_t = 1|\mathcal{E}_{t-1} = 1, \mathbf{h}_{t-1}) = p_s(\mathbf{h}_{t-1}), \quad (4.9)$$

$$P(\mathcal{E}_t = 0|\mathcal{E}_{t-1} = 1, \mathbf{h}_{t-1}) = 1 - p_s(\mathbf{h}_{t-1}). \quad (4.10)$$

In case an object is present ( $\mathcal{E}_t = 1$ ), the values of  $\mathbf{x}$  and  $\mathbf{v}$  are distributed as follows for entering and staying objects respectively:

$$P(\mathbf{x}_t, \mathbf{v}_t|\mathcal{E}_t = 1, \mathcal{E}_{t-1} = 0, \mathbf{h}_{t-1}) = p_e(\mathbf{x}_t, \mathbf{v}_t), \quad (4.11)$$

$$P(\mathbf{x}_t, \mathbf{v}_t|\mathcal{E}_t = 1, \mathcal{E}_{t-1} = 1, \mathbf{h}_{t-1}) = P(\mathbf{x}_t, \mathbf{v}_t|\mathbf{x}_{t-1}, \mathbf{v}_{t-1}).$$

For this last term, I use a constant velocity dynamic model similar to [111], with a normally distributed acceleration noise  $\mathbf{a} \sim N(0, \Sigma_a)$ :

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \mathbf{a}\Delta t, \quad (4.12)$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}_{t-1}\Delta t + \frac{1}{2}\mathbf{a}\Delta t^2. \quad (4.13)$$

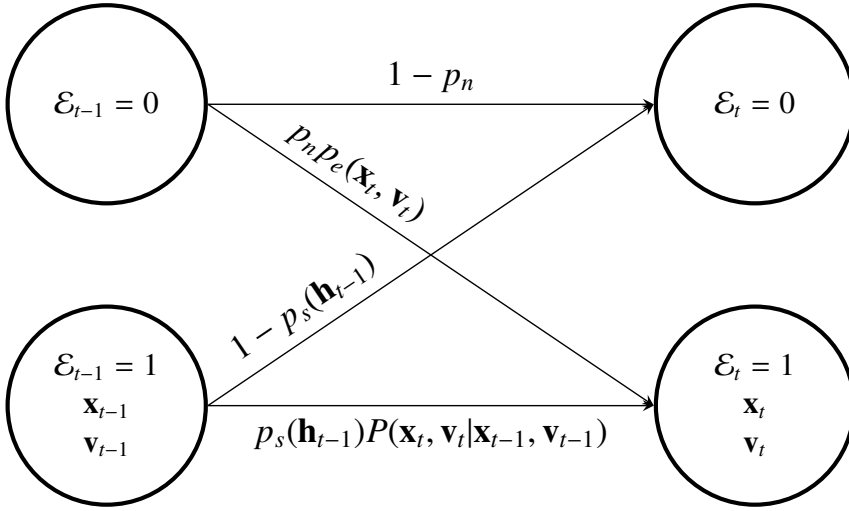


Figure 4.3: Transition of states.  $\mathcal{E}_t = 0$  denotes the lack of object, and  $\mathcal{E}_t = 1$  denotes the presence of an object with the configuration of  $\mathbf{x}_t, \mathbf{v}_t$  at timestamp  $t$ .

Through the introduction of the binary flag  $\mathcal{E}$ , the full state transition can be regarded as a state machine, see Figure 4.3.

#### 4.2.4 UPDATE STEP

Now I describe the likelihood  $P(\mathcal{Z}_t | \mathbf{h}_t)$ . I follow the common assumption of conditional independence for the sensors, thus the single-sensor update step described here can be applied independently to each. The sensor  $s$  returns  $K$  detections at once:  $\mathcal{Z} = \{z^1, \dots, z^K\}$ . Each detection  $z^k$  contains a 2D BEV location and some additional attributes:  $z^k = [z_{pos}, z_{attr}]$ . To include occlusion awareness, my measurement model introduces several auxiliary variables, with conditional dependencies as shown in the graphical model of Figure 4.4. These variables and their distributions will be introduced in the next paragraphs, where I first distinguish between the expected number of detections, which differentiates my occlusion aware from the naive approach, and then the likelihood term for a single measurement  $z^k$ .

**Detection rates** The total number of detections ( $K$ ) is the sum of foreground ( $K^F$ ) and background ( $K^B$ ) detections:  $K = K^F + K^B$ . If I consider detections as conditionally independent events occurring during a fixed interval, it is natural to model the number of foreground (true positive) and background (false positive) detections with two Poisson distributions. Let us denote the corresponding detection rates with  $\lambda^F(\mathbf{x}, \mathcal{E})$  and  $\lambda^B$  for the foreground and background detections respectively. The values of  $K^B, K^F$  follow Poisson distributions,  $K^B \sim Pois(\lambda^B)$  and  $K^F \sim Pois(\lambda^F)$ , with scalar parameters  $\lambda^B$  and  $\lambda^F$ . The total number of detections  $K$  is then also Poisson distributed:

$$P(K | \lambda^B, \lambda^F) = Pois(\lambda^B + \lambda^F). \quad (4.14)$$

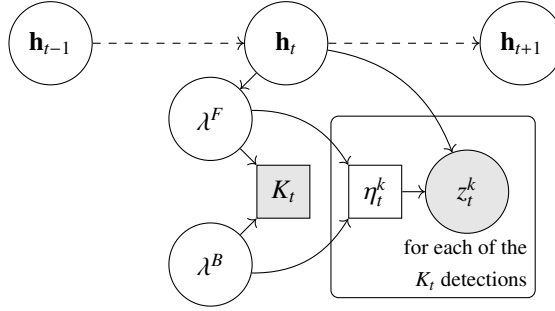


Figure 4.4: Graphical model of the probabilistic dependencies in a single time slice  $t$  with  $K$  detections  $\mathcal{Z}_t = \{z_t^1, \dots, z_t^K\}$ .  $\mathbf{h}_t$  is the state vector and  $\lambda^B, \lambda^F$  are the expected detection rates. The binary flag  $\eta_t^k$  denotes if the  $k^{\text{th}}$  detection  $z_t^k$  comes from foreground or background. Discrete/real variables are shown with square/circle nodes. Observed variables are shaded.

4

I distinguish between my novel occlusion aware filter by the way it determines the value of the foreground detection rate, as opposed to the naive approach. In the proposed *Occlusion Aware Filter (OAF)* approach, the number of foreground detections depends both on the object's presence and location. A benefit of Poisson distributions is that I can incorporate the occlusion information here with a spatially dependent rate parameter, i.e., more true detections are expected if the pedestrian is unoccluded (i.e. visible) than if the pedestrian is occluded:

$$\lambda^F = \begin{cases} \lambda_{unocc}^F & \text{if } \mathbf{x} \in V, \\ \lambda_{occ}^F & \text{if } \mathbf{x} \in O, \end{cases} \quad (\text{OAF}) \quad (4.15)$$

where  $\lambda_{unocc}^F, \lambda_{occ}^F$  indicate the expected detection rates in unoccluded ( $V$ ), occluded ( $O$ ) areas respectively, see Figure 4.1. The extent of these areas will be determined by the implementation-specific environment occlusion model.

In contrast, a naive (i.e., not occlusion aware) filter assumes that  $\lambda^F$  is constant, targeting the more typical unoccluded case:

$$\lambda^F = \lambda_{unocc}^F. \quad (\text{naive approach}) \quad (4.16)$$

The occlusion aware filter behaves the same as a naive one in unoccluded cases, but in occluded positions it adapts its expected rate  $\lambda^F$ .

**Measurement likelihood** Derived from the properties of Poisson distributions, the number of false and true positive detections given  $K$  are distributed as Binomial distributions parametrized by the ratio of  $\lambda^B$  and  $\lambda^F$ . Thus, the probability of a detection  $z^k$  being foreground/background is (given  $K$  number of detections):

$$P(\eta^k = 1 | \lambda^B, \lambda^F) = \frac{\lambda^F}{\lambda^F + \lambda^B}, \quad (4.17)$$

$$P(\eta^k = 0 | \lambda^B, \lambda^F) = \frac{\lambda^B}{\lambda^F + \lambda^B}, \quad (4.18)$$

where the binary flag  $\eta^k$  denotes if the  $k^{\text{th}}$  detection  $z^k$  comes from the tracked object, i.e., is a true positive detection. Since every detection is conditioned on  $\mathcal{E}$  and  $\eta^k$  latent variables, I have to define the likelihood function  $P(z^k|\mathcal{E}, \mathbf{x}, \mathbf{v}, \eta^k)$  for the following cases:  $(\mathcal{E} = 1, \eta^k = 1)$ ,  $(\mathcal{E} = 1, \eta^k = 0)$ ,  $(\mathcal{E} = 0, \eta^k = 0)$ , which stand for the true positive and for the false positive cases, with and without a present pedestrian, respectively.

Unlike [40], in which we only considered location of detections, here I compose this likelihood function with two parts: a spatial component (i.e., likelihood of detection's location) and an attribute component (likelihood of such a detection at that location). I assume that true positive (foreground) detections are spatially distributed around the object's position  $\mathbf{x}$  described by some distribution  $L_F(z_{pos}|\mathbf{x})$ , and that false (background) detections are distributed as described by some distribution  $L_B(z^k)$ . Similarly, I define the attribute likelihood functions,  $A_F(z_{attr}|\mathbf{x}, \mathbf{v})$  and  $A_B(z_{attr})$  for true and false detections, but also conditioned on velocity. Then:

$$\begin{aligned} P(z^k|\mathcal{E} = 1, \mathbf{x}, \mathbf{v}, \eta^k = 1) &= L_F(z_{pos}|\mathbf{x}) \cdot A_F(z_{attr}|\mathbf{x}, \mathbf{v}), \\ P(z^k|\eta^k = 0) &= L_B(z_{pos}) \cdot A_B(z_{attr}). \end{aligned} \quad (4.19)$$

Finally, the complete likelihood of a single measurement is:

$$\begin{aligned} P(z^k|\mathcal{E}, \mathbf{x}, \mathbf{v}, \lambda^B, \lambda^F) &= P(z^k|\mathcal{E}, \mathbf{x}, \mathbf{v}, \eta^k = 1) \cdot P(\eta^k = 1|\lambda^B, \lambda^F) \\ &+ P(z^k|\mathcal{E}, \mathbf{x}, \mathbf{v}, \eta^k = 0) \cdot P(\eta^k = 0|\lambda^B, \lambda^F). \end{aligned} \quad (4.20)$$

Since all  $K$  detections are conditionally independent given  $\mathbf{x}$  and  $\mathcal{E}$ , and  $\lambda^B$  and  $\lambda^F$  are determined through position  $\mathbf{x}$  and areas  $V$  plus  $O$ , the full measurement likelihood becomes:

$$P(\mathcal{Z}_t|\mathbf{h}_t) = P(K|\lambda^B, \lambda^F) \cdot \prod_{k=1}^K P(z^k|\mathbf{h}_t, \lambda^B, \lambda^F). \quad (4.21)$$

## 4.3 IMPLEMENTATION

**F**IRST, I describe how the Bayesian filter was implemented with a particle filter. Then, I discuss how the attribute likelihood function was implemented for the two sensors. A summary of the model parameters is given in Table 4.1.

### 4.3.1 PARTICLE FILTERING

For inference, I use a particle filter to represent the posterior distribution in my model by a set of samples (i.e., particles). Unlike, say, a multiple-model Kalman Filter, it is straightforward to include information about occlusion in the particle filter, i.e. particles in occluded areas are treated differently than those in unoccluded areas, and to represent uniform initial uncertainty over the bounded occlusion region. Furthermore, such a system is easy to scale for the available hardware resources by changing the number of particles.

To include the existence probability in the filter, I follow [112]. Of  $N$  particles, the first one (index 0) will represent all hypotheses with non-present pedestrian, called the negative particle. The remaining  $N - 1 = N_s$  particles (called the positive ones) represent the cases of a present pedestrian:

$$\mathcal{E}_t = 0 \rightarrow w_t^{(0)}, \quad (4.22)$$

$$\mathcal{E}_t = 1 \rightarrow (\mathbf{h}_t^{(i)}, w_t^{(i)}) \text{ for } i = 1 \dots N_s. \quad (4.23)$$

| Parameter                              | Short description                               | In my experiments           |
|--|---|-----------------------------|
| $p_s(\mathbf{h}_t^{(i)})$              | Probability for a ped. to stay                  | 0.95 in ROI                 |
| $p_n$                                  | Probability of an entering ped.                 | 0.2                         |
| $p_e(\mathbf{h}_t)$                    | Distrib. of $\mathbf{h}_t$ for an entering ped. | Uniform in ROI              |
| $\lambda_{unocc}^F$                    | Exp. # of detections (unoccluded)               | 1/1.5 for cam./rad.         |
| $\lambda_{occ}^F$                      | Exp. # of detections (occluded)                 | 0.1/0.3 for cam./rad.       |
| $\lambda^B$                            | Exp. # of background detections                 | 0.05/0.1 for cam./rad.      |
| $L_B(z_{pos})$                         | Spatial likelihood (background)                 | Uniform in ROI              |
| $L_F(z_{pos} \mathbf{x})$              | Spatial likelihood (foreground)                 | Eq. (4.32), Eq. (4.34)      |
| $A_B(z_{attr})$                        | Attribute likelihood (background)               | Eq. (4.33), Eq. (4.36)      |
| $A_F(z_{attr} \mathbf{x}, \mathbf{v})$ | Attribute likelihood (foreground)               | Eq. (4.33), Eq. (4.36)      |
| $\Sigma_{cx}, \Sigma_{rx}$             | std. dev. used in $L_F$                         | 0.2 m, 0.3 m                |
| $\Sigma_{ch}^F, \Sigma_{rv}^F$         | std. dev. used in $A_F$                         | 0.7 m, 0.8 m/s              |
| $\Sigma_{ch}^B, \Sigma_{rv}^B$         | std. dev. used in $A_B$                         | 1.5 m, 3 m/s                |
| $W_{speed}$                            | Exp. distrib. of particle speeds                | $N(p, \Sigma_w)$            |
| $W_{dir}$                              | Exp. distrib. of particle orientation           | Uniform in $\pm 22.5^\circ$ |

Table 4.1: List of model parameters and their experimental value settings.

where  $\mathbf{h}_t^{(i)} = [\mathbf{x}_t^{(i)}, \mathbf{v}_t^{(i)}, \mathcal{E}_t^{(i)} = 1]$  is the state of the  $i^{th}$  particle,  $w_t^{(i)}$  is the weight assigned to it, and  $\mathcal{E}_t^{(i)} = 1$  marks that these  $N_s$  particles represent hypotheses of a present pedestrian. Thus, the estimated probability of a non-present/existing pedestrian given all detections is the normalized weight of the first particle/summed weights of all the others, see Eq. (4.5):

$$P(\mathcal{E}_t = 0 | \mathcal{Z}_{1:t}) = w_t^{(0)}, \quad P(\mathcal{E}_t = 1 | \mathcal{Z}_{1:t}) = \sum_{i=1}^{N_s} w_t^{(i)}. \quad (4.24)$$

To obtain the estimated state of the pedestrian, I use the weighted average of the particles along the hypothesis space:

$$\tilde{\mathbf{h}}_t = [\tilde{\mathbf{x}}_t, \tilde{\mathbf{v}}_t, \tilde{\mathcal{E}}] = \sum_{i=1}^{N_s} w_t^{(i)} \cdot \mathbf{h}_t^{(i)}, \quad (4.25)$$

where  $\tilde{\mathbf{x}}_t = (\tilde{x}_t, \tilde{y}_t)$  is the estimated position,  $\tilde{\mathbf{v}}_t = (\tilde{v}_{x,t}, \tilde{v}_{y,t})$  is the estimated velocity vector of the pedestrian, and  $\tilde{\mathcal{E}}$  is the estimate of the pedestrian being present, see Eq. (4.24).

### INITIALIZATION

Particles' positions are initialized uniformly across the Region of Interest (ROI). Their velocity is drawn from normal distribution  $W_{speed} \sim N(p, \Sigma_w)$  around slow walking pace  $p = 1$  m/s and their orientation is drawn from a uniform distribution  $W_{dir}$  between  $\pm 22.5^\circ$ , where  $0^\circ$  is the orientation perpendicular to the movement of the ego-vehicle, pointing towards the road.

### PREDICTION STEP

The input of the prediction step are  $N_s$  uniformly weighted particles representing the present pedestrian, and one particle representing the  $\mathcal{E}_t = 0$  hypothesis.

Predicted variables are marked with  $\hat{\cdot}$  sign. First, I estimate the next weight of the negative particle as follows:

$$P(\mathcal{E}_t = 0 | \mathcal{Z}_{1:t-1}) = \hat{w}_t^{(0)} = \frac{w_{np}}{w_{np} + w_p}, \quad (4.26)$$

where  $w_p$ ,  $w_{np}$  are the cumulative weights of present, and not present predicted states using Eq. (4.7) - Eq. (4.10):

$$w_p = p_n \cdot w_{t-1}^{(0)} + \sum_{i=1}^{N_s} (p_s(\mathbf{h}_t^{(i)})) w_{t-1}^{(i)}, \quad (4.27)$$

$$w_{np} = (1 - p_n) \cdot w_{t-1}^{(0)} + \sum_{i=1}^{N_s} (1 - p_s(\mathbf{h}_t^{(i)})) w_{t-1}^{(i)}. \quad (4.28)$$

Afterwards, I sample  $N_s$  new positive particles, which are either a mutation of an existing particle moved by the dynamic model, or a completely new (entering) one, see Eq. (4.11). An existing particle stays in the scene with probability  $p_s(\mathbf{h}_t^{(i)})$ , or is replaced by a new one with probability of  $1 - p_s(\mathbf{h}_t^{(i)})$ :

$$\mathbf{h}_{t-1}^{(i)} \rightarrow \begin{cases} \hat{\mathbf{h}}_t^{(i)} \sim P(\mathbf{h}_t | \mathbf{h}_{t-1}^{(i)}) & \text{if moved particle,} \\ \hat{\mathbf{h}}_t^{(i)} \sim p_e(\mathbf{h}_t) & \text{if new particle.} \end{cases} \quad (4.29)$$

All weights of the predicted positive particles are then set uniformly:

$$\hat{w}_t^{(i)} = \frac{1 - \hat{w}_t^{(0)}}{N_s} \quad \forall i = 1 \dots N_s. \quad (4.30)$$

### UPDATE STEP

Particles are updated by new detections using the measurement likelihood Eq. (4.21):

$$w_t^{(i)} \propto \hat{w}_t^{(i)} \cdot P(\mathcal{Z}_t | \hat{\mathbf{h}}_t^{(i)}). \quad (4.31)$$

Details of the attribute likelihood calculations are discussed later in Subsection 4.3.2 and 4.3.3. After the update, all weights are renormalized. To avoid sample degeneracy, I resample the positive particles if the Effective Sample Size (ESS) drops below a threshold [119].

### 4.3.2 USE OF STEREO CAMERA DATA

The camera sensor data is used for two purposes: 1) to update the filter with camera based pedestrian detections and 2) to update the occlusion model, see Figure 4.2, top. For both tasks, I use the Instance Stixel representation [110]. Stixels [108] are rectangular upright sticks in the 3D space, perpendicular to the estimated ground plane. With the extension of [110], each stixel has the following parameters: a 3D position of their bottom, a height, a

class label (among others: *car*, *bus*, *truck*, *person*, *sky*) and an instance id. In this way, objects of interest (pedestrians and occluding vehicles) are represented by a loose set of stixels connected by their class and instance information. Unlike the bounding box representation used in [40], these stixels better describe the shape and extent of objects in both bird's-eye and camera perspectives (e.g., varying visible height of cars) while keeping the processing load low. For a visual introduction to stixels I refer the reader to Figure 4.1. Qualitative examples of Instance Stixels can be seen in Figure 4.7. First I filter the stixels to keep only those from the relevant classes: *pedestrian* stixels as input for the particle filter and vehicle stixels (i.e. from *car*, *truck*, and *bus* classes) to update the occlusion model.

#### UPDATE OF THE OCCLUSION MODEL

The stixels of vehicles that are close enough (i.e., at least one of their stixels is in ROI) are fitted with a bird's-eye view 2D rectangle to model the position and extent of the parked vehicles. The fitting is done with plausible minimum widths and lengths to avoid unrealistically small car assumptions. I consider the projected region behind the farther end of these car models as *occluded* as shown in Figure 4.1 and 4.7. I also store the set of stixels for each car to calculate the height of the occlusion for later use, see below.

#### UPDATE OF THE FILTER

The pedestrian stixels are grouped by their instance id. Then, the average 2D BEV position of the stixels and their largest height range in meters (i.e., the difference between the lowest and highest stixels ends) are computed to create a pedestrian detection for the filter:  $z = [z_{pos}, z_{attr} = z_{height}]$ . The position  $z_{pos}$  is then used in the spatial component, which is modeled with a normal distribution, with standard deviation  $\Sigma_{cx}$ :

$$L_F(z_{pos}|\mathbf{x}_t^{(i)}) = N(z_{pos}|\mathbf{x}_t^{(i)}, \Sigma_{cx}). \quad (4.32)$$

The height  $z_{height}$  is used to calculate the attribute likelihood  $A_F(z_{attr}|\mathbf{x}_t^{(i)}, \mathbf{v}_t^{(i)})$ . I consider the likelihood of observing a pedestrian with visible  $z_{height}$  at the location  $\mathbf{x}_t^{(i)}$  of each particle, given the current occlusion model. First, I compute the expected observable height  $\tilde{h}_t^{(i)}$  for each occluded particle by looking up the car stixel with the most similar angle to it, see Figure 4.1. Then, the height of this stixel is scaled by the distance of the particle to get how tall objects would be occluded by the stixel/parked car at the particle's location. Afterwards, the expected observable height  $\tilde{h}_t^{(i)}$  is the difference between the occluded height and the expected height of a pedestrian  $m_{height}$ . For example, behind a tall van I expect to see no part of a pedestrian ( $\tilde{h}_t^{(i)} = 0$ ), while at an unoccluded location the full height of the pedestrian should be visible ( $\tilde{h}_t^{(i)} = m_{height}$ ). Finally, I model both  $A_F(z_{attr}|\mathbf{x}_t^{(i)}, \mathbf{v}_t^{(i)})$  and  $A_B(z_{attr})$  as zero mean normal distributions with standard deviations  $\Sigma_{ch}^F$  and  $\Sigma_{ch}^B$ :

$$\begin{aligned} d_{height} &= z_{height} - \tilde{h}_t^{(i)}, \\ A_F(z_{attr}|\mathbf{x}_t^{(i)}, \mathbf{v}_t^{(i)}) &= N(d_{height}|0, \Sigma_{ch}^F), \\ A_B(z_{attr}) &= N(d_{height}|0, \Sigma_{ch}^B). \end{aligned} \quad (4.33)$$

### 4.3.3 USE OF RADAR DATA

Radar data is solely used as an input to my pedestrian detection filter. For an overview of radar specific steps, see Figure 4.2, bottom. The equipped radar outputs a sparse point

cloud of reflections called *radar targets*. Each point has two spatial dimensions, range  $r$  and azimuth  $\alpha$ , and a third dimension referred to as Doppler, which is the radial velocity  $v_{rel}$  of the target relative to the ego-vehicle. First, I perform ego-motion compensation for  $v_{rel}$ . That is, by eliminating the motion of the sensor that comes from both the translational and rotational movement of the ego-vehicle I get the *compensated radial velocity*, a signed scalar value denoted by  $v_r$ , describing the ego-motion compensated (i.e., absolute) radial velocity of the point. In a next step, I filter the reflections based on their RCS and  $v_r$ , i.e. I remove targets with very weak reflections or too low velocities to only keep ones that could potentially originate from a darting pedestrian. I also eliminate radar targets that are located in the rectangles fitted to the parked cars since a pedestrian cannot be present there, but the high reflectivity of these cars could yield a moving radar target in case of a faulty ego-motion compensation. The remaining reflections are considered as detections for the filter:  $z = [z_{pos}, z_{attr} = z_{vel} = v_r]$ .

The position  $z_{pos}$  is then used in the spatial component and modeled with a normal distribution analogous to the camera, with standard deviation  $\Sigma_{rx}$ :

$$L_F(z_{pos}|\mathbf{x}_t^{(i)}) = N(z_{pos}|\mathbf{x}_t^{(i)}, \Sigma_{rx}). \quad (4.34)$$

In addition to the spatial distribution, the radar also has an attribute likelihood component  $A_F(z_{attr}|\mathbf{x}_t^{(i)}, \mathbf{v}_t^{(i)})$ . I consider the likelihood of observing a radar reflection with the measured radial velocity  $z_{vel}$  given the location and velocity of each particles. Let us define  $\mathbf{los} = \mathbf{x}_t^{(i)} - \mathbf{x}_{radar}$  as the line-of-sight vector pointing from the radar to the particle. I calculate the expected radial velocity  $\tilde{v}_{r,t}^{(i)}$  as the particle's velocity  $\mathbf{v}_t^{(i)}$ 's projection to this vector (i.e., its radial component):

$$\tilde{v}_{r,t}^{(i)} = \frac{\mathbf{los} \cdot \mathbf{v}_t^{(i)}}{\|\mathbf{los}\|}. \quad (4.35)$$

Finally, I model both  $A_F(z_{attr}|\mathbf{x}_t^{(i)}, \mathbf{v}_t^{(i)})$  and  $A_B(z_{attr})$  as zero mean normal distributions with standard deviations  $\Sigma_{rv}^F$  and  $\Sigma_{rv}^B$ :

$$\begin{aligned} d_{vel} &= z_{vel} - \tilde{v}_{r,t}^{(i)}, \\ A_F(z_{attr}|\mathbf{x}_t^{(i)}, \mathbf{v}_t^{(i)}) &= N(d_{vel}|0, \Sigma_{rv}^F), \\ A_B(z_{attr}) &= N(d_{vel}|0, \Sigma_{rv}^B). \end{aligned} \quad (4.36)$$

## 4.4 DATASET

THE dataset was captured by our prototype vehicle [120] in Delft, the Netherlands. I recorded the output of a Continental 400 radar mounted behind the front bumper (2+1D: range, azimuth, Doppler, ~13 Hz, ~100 m range, ~120° field of view), an IDS stereo camera (1936 × 1216 px, ~10 Hz, 35 cm baseline) mounted on the windshield, a Velodyne HDL-64 LiDAR (64 layers, ~10 Hz) scanner on the roof, and the ego-vehicle's odometry (Spatial Dual GNSS/INS/AHRS sensor and wheel odometry fused via an Unscented Kalman Filter, ~30 Hz). All sensors were jointly calibrated following [121]. While the LiDAR data is not used in this chapter, it will be made available for future work.

The dataset contains 501 recordings, each with a length between 8-20 seconds. In each recording, the ego-vehicle approaches or passes (at least) one parked vehicle with a pedestrian





Figure 4.5: Examples of darting out pedestrians from the dataset.

behind it. All recordings were performed in a real environment, with driving speeds suitable for the environment (mean: 4.0 m/s, std.: 0.57 m/s). The pedestrian either steps out from behind the parked vehicle (“*darting*” or “*walking*” sequences) or remains there (“*staying*” sequences). Participants were instructed which action to perform next, but were free to choose their walking speed during *darting*, or their activity (imitating e.g. phone call, bagging groceries, slight movement) during *staying* recordings. See Figure 4.5 for examples of darting out pedestrians. Fifteen subjects with different heights participated in the experiment (mean: 178 cm, standard deviation: 8.5 cm). In total, more than 100 different parked vehicles were used as occlusion, ranging from passenger cars (partial occlusion) to vans (full occlusion).

The resulting dataset contains 249 walking and staying 252 sequences. For each sequence, I manually annotated its type (darting or staying), the pedestrian’s height, the occluding vehicle’s type (car or van) and some environment conditions (e.g., harsh lighting, leaves on the ground, etc.). I have also marked the first timestamps where a) the head, b) the body center, c) one of the feet, and d) the entire body of the pedestrian is visible, see Figure 4.6. This allows a temporal alignment of the sequences and a better understanding of the visual occlusion in the case of different occluding vehicles.



Figure 4.6: Example of annotated frames on a walking out sequence. I marked the first frames where (a) the pedestrian’s head, (b) the body center, (c) one of the feet, and (d), full body is visible.

## 4.5 EXPERIMENTS

IN my experiments, I investigate how the fusion of stereo camera and radar sensors, and the incorporation of occlusion information help to detect darting out pedestrians. For this purpose, I compare the following methods: *naive camera*, *naive fusion*, *OAF camera*, and *OAF fusion*, where “naive”/“OAF” stands for naive/occlusion aware filtering. The *naive camera* and *naive fusion* are methods that use only the camera/both sensors to update the filter in a naive way, see Eq. (4.16). Similarly, *OAF camera* and *OAF fusion* use only camera/both sensors to update, but in an occlusion aware way, i.e., they are “occlusion aware filters”, see Eq. (4.15). All four methods above use Instance Stixels (IS) as camera based pedestrian detections to update the filter, while *OAF camera* and *OAF fusion* also use Instance Stixels to model the occlusions. To study the benefits of the improvements introduced in this chapter, I compare the methods above with the fusion method from our previous publication [40]: *OAF<sup>SSD</sup> fusion*. This is also an occlusion aware filter fusing both sensors, similar to *OAF fusion*, but it uses the output of the Single Shot Detector (SSD) instead of Instance Stixels (IS) as camera based method. Further, in contrast to the other methods, *OAF<sup>SSD</sup> fusion* does not use the attribute likelihood components introduced in Subsection 4.3.2 and 4.3.3, only the spatial component. Note that unlike IS, SSD provides detections as bounding boxes, not involving the height profile of the cars. Hence, the height related attribute likelihood component would not be possible to calculate with SSD. An overview of the compared methods is given in Table 4.2. Both the “Filtering” and the “Post-processing” module from Figure 4.2 (including the presented application example) run at a processing speed of over 500 Hz for all methods

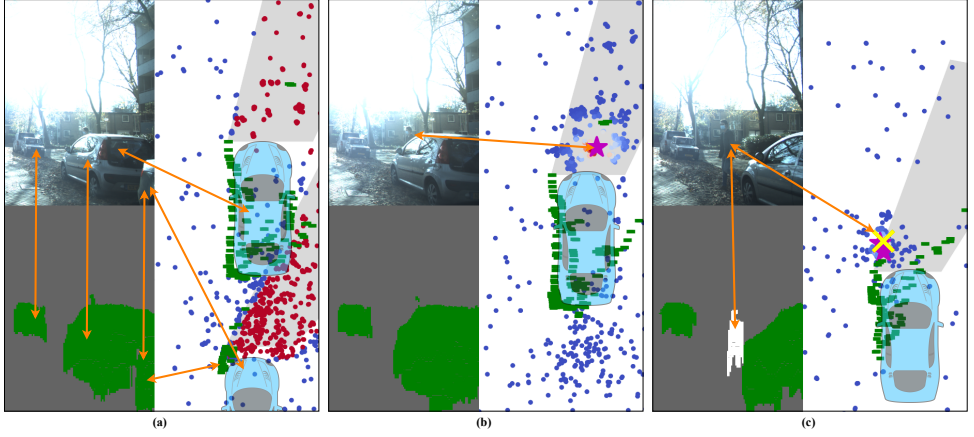


Figure 4.7: Camera views (top left), stixel images (bottom left) and top views (right) of the scene at consecutive timestamps using *OAF fusion*. Vehicle/pedestrian stixels are shown with green/white colors (bottom left). Vehicle stixels are also shown as short green lines on the top view, representing the outlines of the detected parked cars. Occlusion (greyish areas) is calculated as the “shadow” of these cars. Initially, the particles (blue to red, for small to high relative weights) have higher density and weights in occluded regions (a) and converge on the pedestrian’s position after being detected first by the radar (magenta star, (b)) and then by the camera sensor (yellow ‘x’, (c)). All views are cropped for easier visibility. Orange arrows connect corresponding objects in different views.

with 1000 particles in an optimized Python based implementation using the Robot Operating System (ROS) on a high-end system PC (64 GB RAM, TITAN X (Pascal) GPU, Intel Xeon CPU E5-1560 CPU). This brings a negligible overhead compared to the camera based detection modules (off-the-shelf implementation of SSD and IS, including the occlusion model) running around 14 Hz, and the radar related pre-processing steps running at over 200 Hz.

The framework has a set of parameters and distributions that should be tuned to the characteristics (type, accuracy, noise, etc.) of the user’s sensors. A brief overview of these can be found in Table 4.1. In this research, the parameters were empirically tuned on the distinct dataset used in [40] and during in-vehicle experiments, and visually validated on the first few sequences of the new dataset. ROI was defined as a 4.5 m wide, 14 m long rectangle in front of the ego-vehicle. For the camera, I use  $\lambda_{unocc}^F = 1$  because detection is reliable in this range in unoccluded regions.  $\lambda_{occ}^F$  is set to 0.1 for occluded locations. Few false positives occur in the ROI, so  $\lambda^B$  is set to 0.05. For radar, I set  $\lambda_{unocc}^F = 1.5$  for unoccluded positions, since multiple reflections are often received from the same pedestrian. In occlusions, I set  $\lambda_{unocc}^F = 0.3$  as I still expect some reflections due to the multipath propagation. An average rate of  $\lambda^B = 0.1$  is expected for radar, as false positives occur more often than with camera due to e.g. incorrect ego-motion compensation.

#### 4.5.1 ESTIMATED EXISTENCE PROBABILITY IN DANGEROUS SITUATIONS

In my first experiment, I ran the methods on all walking sequences and recorded the reported existence probabilities as in Eq. (4.5). The sequences were temporally aligned by marking the first moment when the pedestrian’s body center was visible as  $t = 0$ , see Figure 4.6.

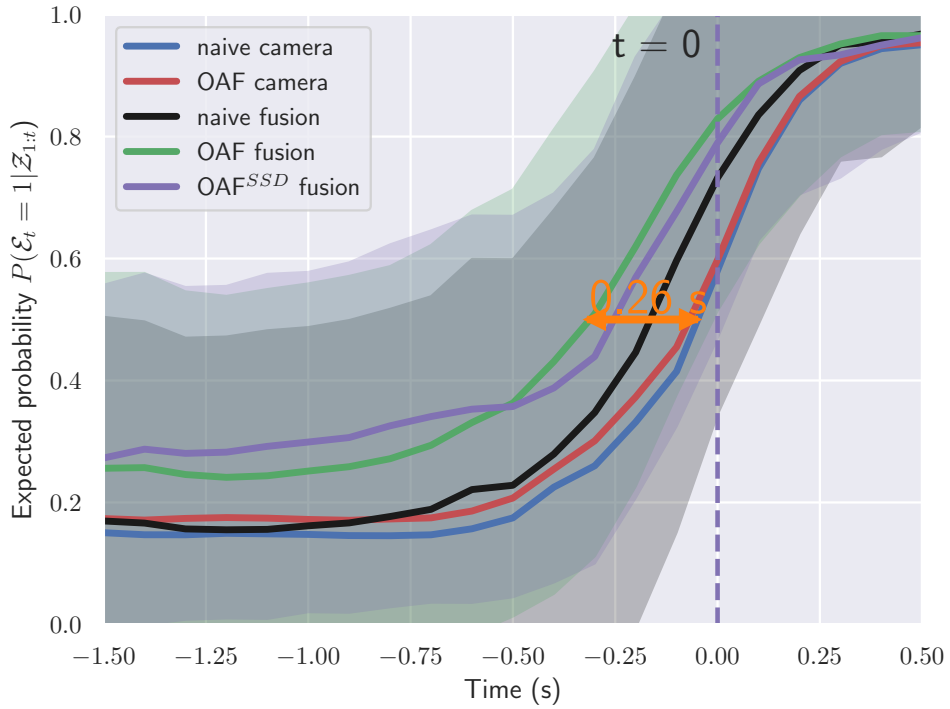


Figure 4.8: Estimated probabilities of a pedestrian being present, averaged over all walking sequences, with standard deviation around the mean for fusion methods.  $t = 0$  is the first moment when the pedestrian's body center was visible. The addition of radar results in earlier detection than using the camera alone.

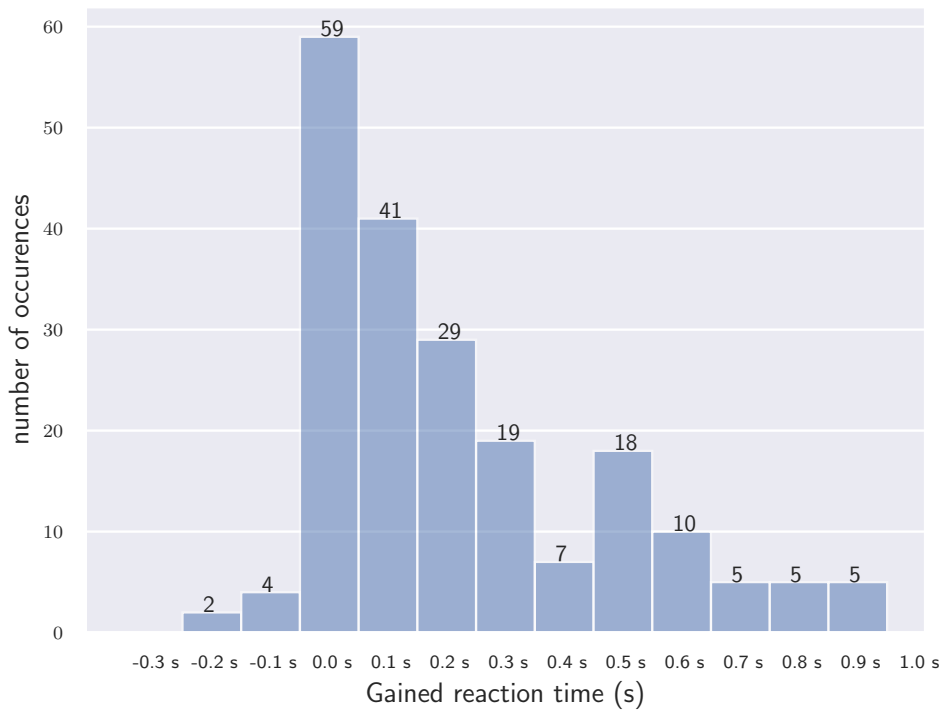


Figure 4.9: Histogram of the gained reaction times. Time difference is calculated between the moments *naive camera* and *OAF fusion* reaches the threshold  $P(\mathcal{E}_t = 1 | \mathcal{Z}_{1:t}) = 0.5$ . For clarity, here I only show the sequences where both methods reach the threshold within the time window of  $[-1 \text{ s}, 0.5 \text{ s}]$ .

| Method                               | Radar | Camera based method | Occlusion awareness | Attribute likelihood component |
|--------------------------------------|-------|---------------------|---------------------|--------------------------------|
| <i>naive camera</i>                  | ✗     | IS                  | ✗                   | ✓                              |
| <i>OAF camera</i>                    | ✗     | IS                  | ✓                   | ✓                              |
| <i>naive fusion</i>                  | ✓     | IS                  | ✗                   | ✓                              |
| <i>OAF<sup>SSD</sup> fusion</i> [40] | ✓     | SSD                 | ✓                   | ✗                              |
| <b><i>OAF fusion</i></b> (proposed)  | ✓     | IS                  | ✓                   | ✓                              |

Table 4.2: Overview of the compared methods with their sensor whether they use radar, type of camera based method (IS: Instance Stixels, SSD: Single Shot Detector), whether they are occlusion aware, and whether they implement the attribute likelihood components.

Then, for each timestamp, and for each method, I calculate the mean estimated probability by averaging over all walking sequences as in [40], see Figure 4.8. In general, the inclusion of radar helps to detect the pedestrian earlier. I.e., any chosen threshold of probability is reached earlier by the three fusion methods (*naive fusion* / *OAF fusion* / *OAF<sup>SSD</sup> fusion*) using both sensors, than by the methods using only the camera. For example, on average, the threshold  $P(\mathcal{E}_t = 1 | \mathcal{Z}_{1:t}) = 0.5$  is reached 0.26 seconds earlier by *OAF fusion* than by *naive camera*. When examining only smaller occluding vehicles (i.e., cars), this time gain increases to 0.30 s. In contrast, for sequences with a van as an occlusion, the measured gain is only 0.12 s.

The previously discussed threshold of  $P(\mathcal{E}_t = 1 | \mathcal{Z}_{1:t}) = 0.5$  is reached 0.15 seconds earlier by the proposed occlusion aware fusion *OAF fusion* than by the naive method *naive fusion*. *OAF fusion* also reports higher probabilities at all times when the pedestrian is occluded ( $t < 0$ ). *OAF<sup>SSD</sup> fusion* reaches the same threshold later than *OAF fusion* by 0.06 seconds, but still earlier than *naive fusion*.

I also examine the sequences individually, and calculate the time difference between the reported probabilities of *naive camera* and *OAF fusion* to be over 0.5. A histogram of the gained reaction times can be found in Figure 4.9. In the large majority ( $\sim 68\%$ ) of dangerous scenarios *OAF fusion* gains some additional reaction time over *naive camera*.

In Figure 4.7, I show an example of a walking scene to demonstrate how *OAF fusion* behaves when there has been no prior detection, and then when first the radar and then the camera has detected the pedestrian.

#### 4.5.2 DISTINGUISHING DANGEROUS AND NON-DANGEROUS SCENARIOS

Similar to [44] I classify the scene into two classes:  $c = \textit{darting}$  (there is a darting pedestrian, dangerous scenario) or  $c = \textit{non-darting}$  (there is no pedestrian, or he/she is not darting). To do this, I estimate the probability  $P(\mathcal{E}_t = 1 | \mathcal{Z}_{1:t})$  of a present pedestrian of any kind (staying or darting) by Eq. (4.24). I also estimate whether the assumed-to-be present pedestrian darts out creating a dangerous scenario  $P(c = \textit{darting} | \mathcal{Z}_{1:t}, \mathcal{E} = 1)$  based on the estimated state of the pedestrian  $\hat{\mathbf{h}}_t$ , see Eq. (4.25). The pedestrian is assumed to be darting if he/she is already on the road in front of the ego-vehicle:  $\tilde{x}_t > \textit{dangerousPos}$  (axis is perpendicular to the movement of the ego-vehicle, increases towards the road), or he/she

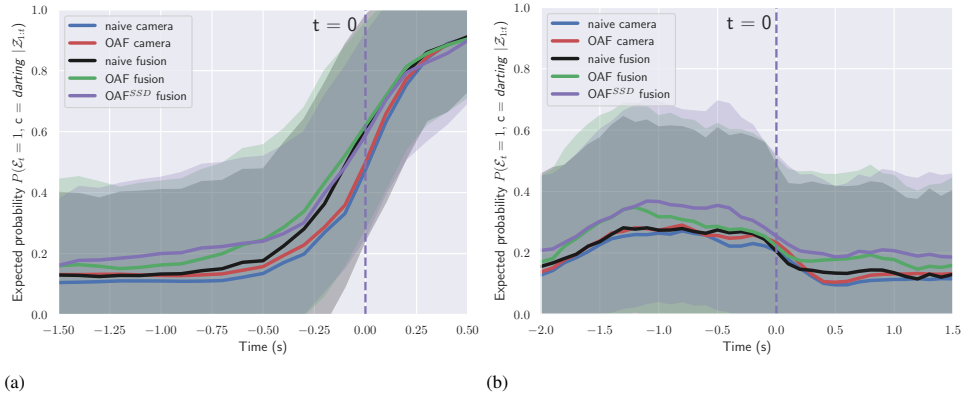


Figure 4.10: Estimated probabilities of a darting pedestrian, averaged over all walking (a) and staying (b) sequences, with standard deviation around the mean for fusion methods.  $t = 0$  is the first moment when the pedestrian’s body center was visible for walking scenes, and the last moment when the occluding vehicle was visible for staying scenes. For walking scenes (a), the addition of radar results in earlier detection than using the camera alone. For staying scenes (b), all methods reports slightly increased, but still small probabilities of danger (i.e. darting) before the passing.

has a lateral velocity component large enough to assume he/she will be on the road later:  $\tilde{v}_{x,t} > \text{dangerousSpeed}$ . Similarly, I assume that the pedestrian will not dart out if he/she is far enough from the road:  $\tilde{x}_t < \text{safePos}$ , or their lateral velocity is close to zero/pointing away from the road:  $\tilde{v}_{x,t} < \text{safeSpeed}$ . Probabilities for values between these limits ( $\text{dangerousPos} > \tilde{x}_t > \text{safePos}$  and  $\text{dangerousSpeed} > \tilde{v}_{x,t} > \text{safeSpeed}$ ) are linearly interpolated. I evaluate the probability of darting based on these two conditions (spatial and velocity) independently, and then take the maximum of the two values for safety. Finally, the probability of a present, darting pedestrian is calculated by multiplying the two probabilities:  $P(\mathcal{E}_t = 1, c = \text{darting} | \mathcal{Z}_{1:t}) = P(\mathcal{E}_t = 1 | \mathcal{Z}_{1:t}) \cdot P(c = \text{darting} | \mathcal{Z}_{1:t}, \mathcal{E} = 1)$ . In the staying scenarios, the pedestrian’s body center was not always visible during the recording as the pedestrian may have remained hidden completely. Hence, unlike for walking scenes, I marked the last moment the occluding vehicle was still visible as  $t = 0$ , to represent the moment when the ego-vehicle passes the occluding, parked vehicle. For each timestamp, I average the probability of a darting pedestrian  $P(\mathcal{E}_t = 1, c = \text{darting} | \mathcal{Z}_{1:t})$  for walking and staying scenes separately, see Figure 4.10. For the walking cases, the fusion methods (*naive fusion*, *OAF fusion* and *OAF<sup>SSD</sup> fusion*) report higher probabilities of a darting pedestrian earlier, see Subfigure 4.10a. When evaluating the non-dangerous staying scenarios, all methods report a small, but moderately increased probability of a darting pedestrian in the moments before the occluding vehicle and the staying pedestrian are passed, and significantly decreased probabilities after the drive-by, see Subfigure 4.10b.

## 4.6 DISCUSSION

THE benefits of including radar in darting out pedestrian detection has been shown in Subsection 4.5.1, where all fusion methods reacted earlier than the methods using only the camera. Such an earlier detection may mean additional reaction time in case of a dangerous

situation. One cause is that radars can often detect pedestrians behind parked vehicles, as their reflected radar signal may be able to propagate under the occluding vehicle and reach the sensor. Some of the gains could also be the result of cases when the camera was not able to detect the already visible pedestrian (e.g. caused by harsh lighting), but the radar was. The radar is mounted on the front of the ego-vehicle, as is common in the industry, see Figure 4.1. This could provide a slightly better viewing angle and also contribute to the earlier detections.

Vans tend to be taller than passenger cars and therefore occlude/hide the darting out pedestrian longer. Hence, it would be intuitive that in sequences with vans as occluding vehicles, radar would assist detection more. However, the opposite is the case: reaction time gained was significantly greater for smaller occluding vehicles such as cars than for vans (0.30 s vs 0.12 s for threshold  $P(\mathcal{E}_t = 1 | \mathcal{Z}_{1:t}) = 0.5$ ). My hypothesis is that this non-intuitive result may have been caused by the length of these vehicles. I suspect that radar signals have to bounce multiple times or in different angles under longer vehicles (i.e. vans), which may reduce or eliminate the reflections. If this assumption is true, it may be beneficial to also estimate the length of the parked vehicle and explicitly include it in the fusion pipeline (i.e., expect fewer reflections behind longer vehicles).

The benefits of occlusion awareness become clear when we compare the naive methods (*naive camera*, *naive fusion*) with their occlusion aware pairs (*OAF camera*, *OAF fusion*). For example, *OAF fusion* reports a higher probability of a pedestrian being present than *naive fusion* at all times when the pedestrian is occluded ( $t < 0$ ). The reason for this is twofold. First, *OAF fusion* is occlusion aware, and thus it “acknowledges” that parts of the scenes are occluded and cannot be properly observed, leading to uncertainty. That is, the absence or low number of detections from these areas is not considered hard evidence for the absence of a pedestrian, unlike in naive methods, e.g. *naive fusion*. Instead, particles behind occlusions are weighted higher compared to the unoccluded particles to represent this uncertainty, resulting in higher a priori awareness to these locations even before any detections occur, see Figure 4.7, left. Similarly, this elevated a priori awareness of an occlusion aware method is also observable between *OAF camera* and *naive camera* for  $t < 0$  moments. Such “caution” resembles the behavior of a human driver approaching highly occluded regions where pedestrians might be. Second, detections originating from these occluded regions are valued more than in the naive methods, because the number of detections received better fits the expectations in Eq. (4.17). As a result, the likelihoods are higher for the same detections than when processed by a naive method, e.g. *naive fusion*, see Eq. (4.21).

The occlusion aware fusion approach presented in this chapter, *OAF fusion*, responded earlier to darting out pedestrians than its older version *OAF<sup>SSD</sup> fusion* from [40]. The reason for this, I believe, is twofold. First, as described in Section 3.6, the occlusion model used by *OAF<sup>SSD</sup> fusion* was often inaccurate. A more accurate model of the occlusions (see Subsection 4.3.2) helped to better evaluate measurements in this study. Although this occlusion model was created using stixels, this improvement could also be achieved using other methods to obtain more accurate occlusion information. Second, this work introduced the concept of attribute likelihood components. More specifically, for camera based detections, even small patches of detections were accepted as reasonable, valid measurements if they matched the occlusion model. This was also supported by the decision to use instance segmentation as input instead of standard object detection, since the former tends to provide more partial detections, which suits my use-case. In the case of radar, the attribute component meant



comparing expected and observed radial velocities. This filters out unrealistic radar targets, which could originate from other road users or simply from noise. On the other hand, radar detections that matched the prior expectations of the object's motion were highly valued and increased the probability earlier. It is noteworthy, however, that *OAF<sup>SSD</sup> fusion* still responded earlier than *naive fusion*, suggesting that even its simpler, SSD based occlusion model benefited more than the attribute likelihood components and the use of stixels as camera based detection. This means that, depending on the application and available resources, using a simpler occlusion model (i.e., SSD instead of IS) could be satisfactory with the benefit of reduced computational load.

In the second experiment, I presented an example application of my methods to distinguish dangerous and non-dangerous situations. For scenes where the pedestrian remains behind the car (i.e. not in danger), the estimated probabilities somewhat increase during the drive-by, but remain small. This observed increase can be explained by the way the particles are initialized with a walking speed and an movement orientation pointing to the road, which intentionally introduces a bias towards the darting hypothesis. Such increase in uncertainty about whether the sighted pedestrian will dart out is similar to the reaction of a human driver, who, having noticed a pedestrian in a similar situation, would also slow down/be more cautious for safety reasons. The occlusion aware fusion methods (*OAF fusion* and *OAF<sup>SSD</sup> fusion*) show further increased caution due to perceived occlusions in the scene, which increase a priori uncertainty by design. *OAF fusion*, however, shows lower estimated probabilities for all  $t < 0$  timestamps than *OAF<sup>SSD</sup> fusion*, again suggesting that the new occlusion model based on Instance Stixels is superior to the one based on SSD, and follows the shape of the occlusions more closely.

All filters depend heavily on the quality of the inputs, especially from the camera, where I expect "high-end" detections (e.g. pedestrian instance stixels) from an off-the-shelf module, even under occlusion. The quality of camera based detections also affects the reliability of the filter over the occlusion model. Common errors arise from radar targets that are incorrectly reported as moving by the radar due to poor ego-motion estimation, and from camera based detections that mistake vertically shaped objects (e.g., trees) for pedestrians.

The proposed system can be further improved in several ways. For example, an additional use of the occlusion model would be to adjust the expected background noise for the radar. That is, instead of uniform distribution, it might be beneficial to increase the expected noise near parked vehicles with highly reflective metallic chassis, and decrease it in uncluttered regions.

Integrating additional sensors, (e.g., LiDAR) into the framework is straightforward. In particular, replacing or supporting the 2+1D radar used in this chapter with a 3+1D radar similar to that used in [122] could be interesting for three reasons. First, the elevation information and increased density of the radar point cloud could be used in a more advanced pedestrian classification step, as shown in [122]. Second, the elevation information could be further used in this particular use case by filtering the radar targets based on their elevation angle, leaving only those that are received from below the parked, occluding vehicle - as these targets could be the result of multi-path propagation. This step would help filter out false positive radar reflections that originate from the chassis of parked cars and not from occluded pedestrians. Third, in [122] the 3+1D radar has been shown to be capable of detecting both moving and parked vehicles. As such, it could contribute directly to the

occlusion model and reduce or even eliminate the need for the camera sensor.

To generalize the filter for other road users, one has to adjust the prior velocity and *RCS* values, e.g., faster and more reflective targets should be expected from a cyclist. For the camera based detectors (IS, SSD), the expected class of object has to be changed. Multiple road users can also be tracked with the filter by modifying the state estimation step in Eq. (4.25) to expect more than one peak in the particle distribution. Consideration of objects other than vehicles as occlusions, e.g., walls, is also possible, and the observed visible height should be treated as in this study. However, the type of occlusion must be considered for radar, since multipath propagation is not possible if the occlusion has no space under it, such as walls.

Finally, I did state estimation in this research and showed quantitative benefits of both fusion and occlusion awareness. However, extending the scope to trajectory prediction, the gained reaction times detecting/predicting dangerous situations could be even greater.

## 4.7 CONCLUSIONS AND FUTURE WORK

**I**N this chapter I proposed a generic occlusion aware multi-sensor Bayesian filter to detect occluded crossing pedestrians. To facilitate my and future research of these scenarios, I published a dataset of more than 500 relevant scenarios with stereo camera, radar, LiDAR, and odometry data. I applied the proposed filter to camera and radar data using this dataset, and provided techniques to account for the unique characteristics of these sensors. My results show that both the inclusion of radar sensor and occlusion information is beneficial for this use case, as pedestrians are detected earlier in dangerous walking scenarios. For example, the threshold of 0.5 for the estimated existence probability of a pedestrian in the scene is reached on average 0.26 seconds earlier by my occlusion aware fusion than by a naive camera only detector, and 0.15 seconds earlier than by the method that fuses the two sensors in a naive way.

I also showed in an application example of my filter that it can distinguish between dangerous and non-dangerous situations, which is necessary to avoid false alarms. In this task, too, the inclusion of the radar proved to be beneficial.

Future work may include a more precise expected distribution of background noise, improved scene classification by extending the scope for trajectory prediction, and the inclusion of further sensors, more particularly a 3+1D radar as discussed Section 4.6.



# 5

## MULTI-CLASS ROAD USER DETECTION USING THE 3D RADAR CUBE

5

*Information is a source of learning. But unless it is organized, processed, and available to the right people in a format for decision making, it is a burden, not a benefit.*

William Grosvenor Pollard

## 5.1 INTRODUCTION

As discussed in Subsection 2.4.1, most commercially available radars output a 2+1D point cloud of reflections called *radar targets* in every frame (also called scan, or sweep). Each radar target has the following features: range  $r$  and azimuth  $\alpha$ , Radar Cross Section  $RCS$  (i.e. reflectivity), and radial speed. The latter can be provided relative to the ego-vehicle, i.e. as relative radial velocity  $v_{rel}$ , or after ego-motion compensation, i.e. as absolute radial velocity  $v_r$  (see Section 2.2 for details). These features are often called *target-level*. Since a single reflection does not convey enough information to segment and classify an entire object, many radar based road user detection methods [54][55][66] first cluster radar targets by their target-level features, see Section 3.1. Clusters are then classified as a whole based on derived statistical features (e.g., mean, variance of  $r$ ,  $v_r$ ,  $RCS$  of contained radar targets), and the same class label is assigned to all radar targets in the cluster. See Figure 5.1 for an overview of a traditional radar based detection pipeline.

A disadvantage of these pipelines is that object segmentation and classification performance depend on the success of the initial clustering step, which is often noted [50][64] that to be error-prone, see Figure 5.2. For example, objects can be mistakenly merged (Figure 5.2, A) or split apart (Figure 5.2, B). Finding suitable parameters (e.g., radius and minimum number of points for DBSCAN [63] clustering algorithm) is challenging as the same parameters must be used for all classes, although they have significantly different spatial extension and velocity profiles. For example, a larger radius is beneficial for cars, but could falsely merge pedestrians and cyclists. Another challenge of clustering based methods is that small objects may not have enough reflections to extract meaningful statistical features, e.g., variance. For example, both [54] and [55] set DBSCAN's minimum number of points to form a cluster larger than one, which means that single standing points are thrown away (Figure 5.2, C).

Various methods [30][31][32] instead explore using the *low-level radar cube* extracted from an earlier signal processing stage of the radar. Further examples were discussed in Section 3.2. The radar cube is a 3D data matrix with axes corresponding to range, azimuth, and velocity (also called Doppler), and a cell's value represents the measured radar reflectivity in that range/azimuth/Doppler bin, see Subsection 2.4.2. In contrast to the target-level data,

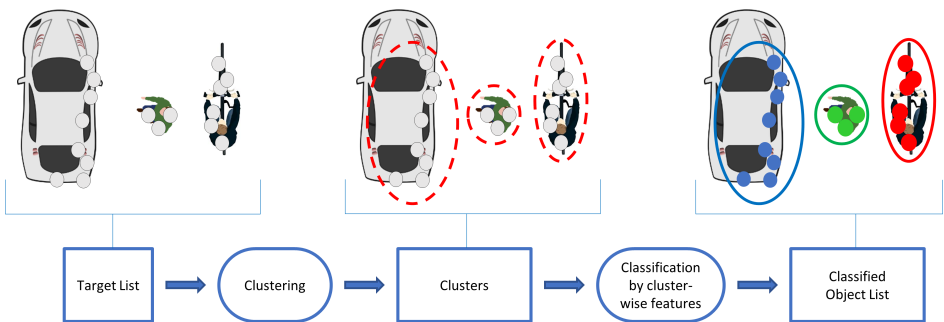


Figure 5.1: Overview of the traditional radar based road user detection pipeline. The target list is first clustered into clusters, which are considered as object proposals. Then, cluster-wise features are extracted and used in classification, assigning a single class label to the whole cluster. In other words, the cluster is classified as a whole.

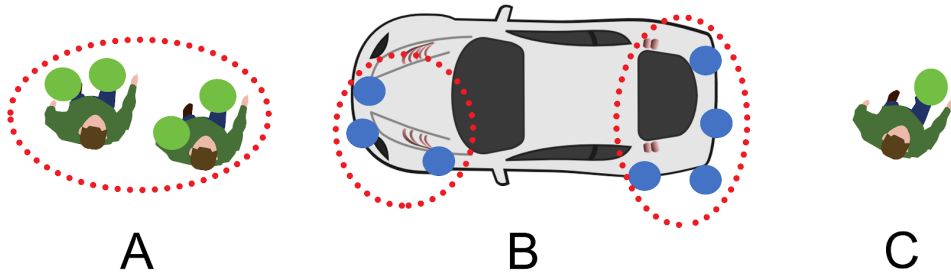


Figure 5.2: Challenging cases for cluster-wise classification methods. A: Objects may be clustered together (red circle). B: Large objects may be split up into several clusters. C: Object with only one reflection. Radar targets are shown as dots, colored green/blue for pedestrian/car ground truth class.

the radar cube provides the complete speed distribution (i.e., Doppler vector) at multiple 2D range-azimuth locations. Such distributions can capture modulations of an object’s main velocity caused by its moving parts, e.g., swinging limbs or rotating wheels, and were shown to be a valuable feature for object classification [31][32]. Commonly radar cube features are computed by first generating 2D range-azimuth or range-Doppler projections, or by aggregating the projected Doppler axis over time into a Doppler-time image [29][30]. I will call features derived from the 3D cube or its projections *low-level*. A downside of such low-level radar data is the lower range and azimuth resolution than the radar targets, and that radar phase ambiguity is not yet addressed, since no advanced range interpolation and direction-of-arrival estimation has taken place.

| Method                   | Basis          | Features    | Classes      | Time window            |
|--------------------------|----------------|-------------|--------------|------------------------|
| <i>Prophet</i> [55] †    | clusters       | target      | single       | 1 frame (50 ms)        |
| <i>Schumann</i> [54] †   | clusters       | target      | multi        | 2 frames (150 ms)      |
| <i>Prophet</i> [71]      | clusters       | both        | single       | 1 frame                |
| <i>Schumann</i> [50]     | targets        | target      | multi        | 0.5 s                  |
| <i>Angelov</i> [30]      | targets        | low         | multi        | 0.5-2 s                |
| <i>RTCnet (proposed)</i> | <i>targets</i> | <i>both</i> | <i>multi</i> | <i>1 frame (75 ms)</i> |

Table 5.1: Overview of the most closely-related methods. †: marks methods selected as baselines.

A detailed overview of radar based methods for object detection using 2+1D radar point clouds was given in Section 3.1, while in Section 3.2 the methods using low-level data were discussed in detail. Here I provide only an overview of the most relevant methods in Table 5.1 with their basis of classification (cluster-wise or target-wise), level of features (target or low), number of classified classes, and the required time window to collect suitable amount of data. None of the found methods avoids error-prone clustering for classification and operates with a low latency for urban driving (i.e., one or two radar sweeps (75 – 150 ms)) at the same time.

In this chapter I propose a radar based, multi-class moving road user detection method, which exploits *both* expert knowledge at the target-level (accurate 2D location, addressed

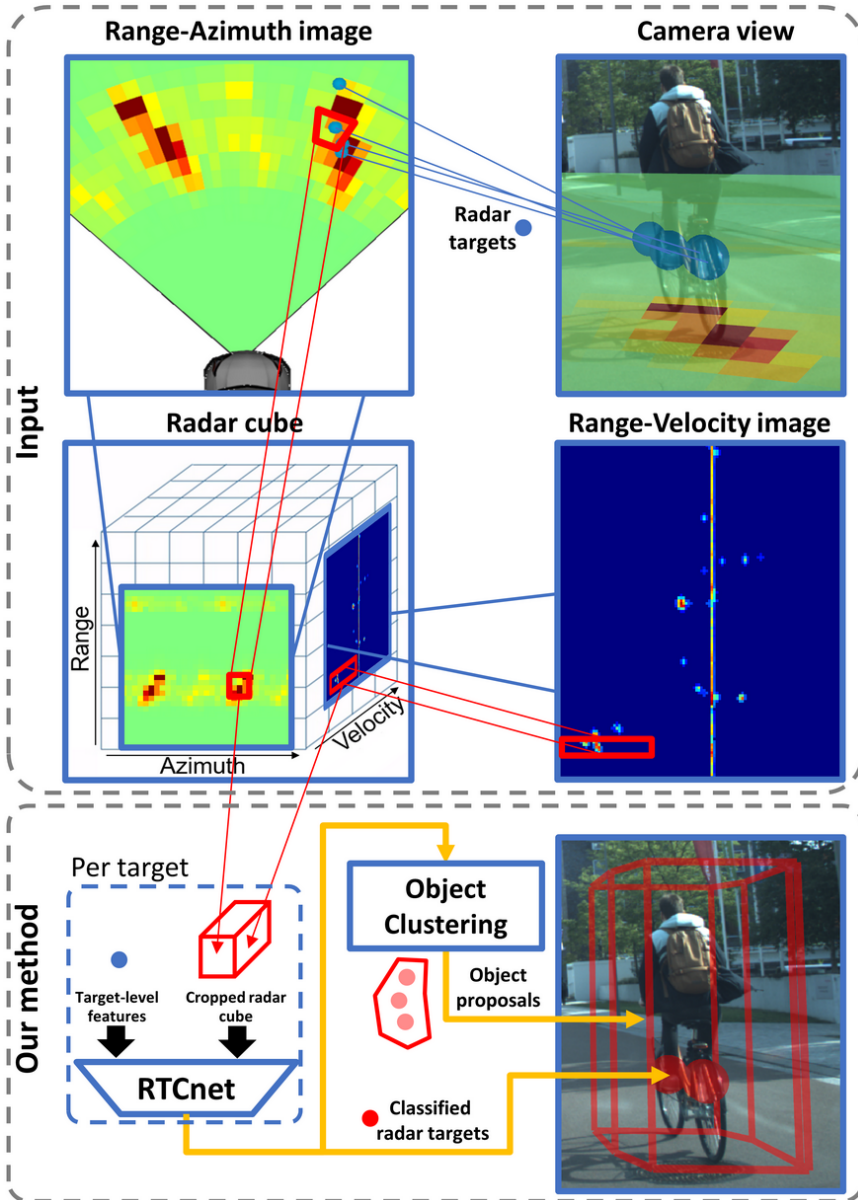


Figure 5.3: Inputs (radar cube and radar targets, top), main processing blocks (RTCnet and object clustering, bottom left), and outputs (classified radar targets and object proposals, bottom right) of my proposed method. Classified radar targets are shown as colored spheres at the sensor’s height. Object proposals are visualized by a convex hull around the clustered targets on the ground plane and at 2 *m*.

phase ambiguity), and low-level information from the *full* 3D radar cube rather than a 2D projection. Importantly, the inclusion of low-level data enables classification of individual radar targets before any object clustering; the latter step can benefit from the obtained class scores. At the core of the method is a Convolutional Neural Network (CNN) called Radar Target Classification Network, or *RTCnet* for short. See Figure 5.3 for an overview of the method's inputs (radar targets and cube) and outputs (classified targets and object proposals).

The proposed method can provide class information on both radar target-level and object-level. Target-level class labels are valuable for sensor fusion operating on intermediate-level, i.e., handling multiple measurements per object [40][123]. The proposed target-level classification is more robust than cluster-wise classification where the initial clustering step must manage to separate radar targets from different objects, and keep those coming from the same object together, see Figure 5.2. The object-level class information output provides instances that are both segmented and classified (object detection), which is valuable for high-level (i.e., late) sensor fusion. While traditional methods must perform clustering with a single set of parameters for all classes, my approach enables use of class-specific clustering parameters (e.g. larger object radius for cars).

### 5.1.1 CONTRIBUTIONS

The main contributions of the chapter are as follows.

1. I propose a radar based, single-frame, multi-class (*pedestrian, cyclist, car*) moving road user detection method, which exploits both target-level and low-level radar data by a specially designed CNN. The method provides both classified radar targets and object proposals by a class-specific clustering.
2. I show on a large-scale, real-world dataset that the method is able to detect road users with higher than state-of-the-art performance both in target-wise (target classification) and object-wise (object detection) metrics using only a single frame of radar data.

## 5.2 PROPOSED METHOD

**I**N this research, I combine the advantages of target-level (accurate range and azimuth estimation) and low-level data (more information in speed domain) by mapping the radar targets into the radar cube and cropping a smaller block around it in all three dimensions (Subsection 5.2.1). *RTCnet* classifies each target individually based on the fused low-level and target-level data. The network consists of three parts (Subsection 5.2.2). The first encodes the data in spatial domains (range, azimuth) and grasps the surroundings' Doppler distribution. The second is applied on this output to extract class information from the distribution of speed. Finally, the third part provides classifications scores by two fully connected layers (FC). The output is either multi-class (one score for each class) or binary. In the latter case, an ensemble voting (Subsection 5.2.3) step combines the result of several binary classifiers similarly to [124]. A class-specific clustering step (i.e. the radar targets' predicted class information is used) generates an object list output (Subsection 5.2.4). See Figure 5.4 for an overview of the method. The software of my pipeline is available on our website<sup>1</sup>.

<sup>1</sup><https://github.com/tudelft-iv/RTCnet>



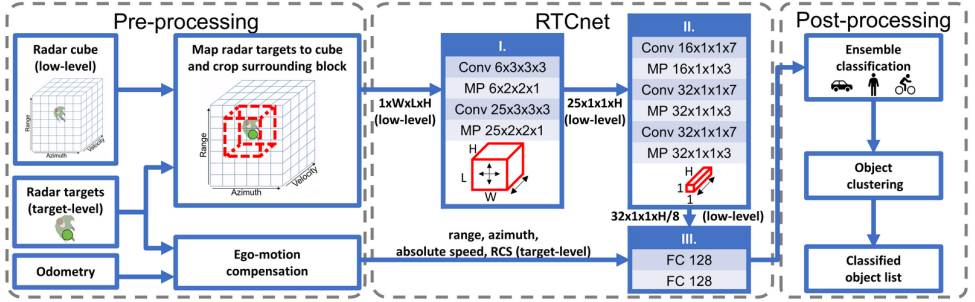


Figure 5.4: Overview of the pipeline. A block around each radar target is cropped from radar cube. *RTCnet* has three parts. I. encodes range and azimuth dimensions. II. extracts class information from the speed distribution. III. provides scores based on II. and target-level features. Ensembling assigns a class label to each radar target. The class-specific clustering provides object proposals.

### 5.2.1 PRE-PROCESSING

First, a single frame of radar targets and a single frame of the radar cube (low-level data) is fetched. Each radar target's speed is compensated for ego-motion similarly to [54]. As I only address moving road users, radar targets with low compensated (absolute) velocity are considered as static and are filtered out. Then, corresponding target-level and low-level radar data are connected. That is, I look up each remaining dynamic radar target's corresponding range/azimuth/Doppler bins, i.e. a grid cell in the radar cube based on their reported range, azimuth and (relative) velocity  $(r, \alpha, v_{rel})$ . Afterwards, a 3D block of the radar cube is cropped around each radar target's grid cell with radius in range/azimuth/Doppler dimensions  $(L, W, H)$ . This block is a region descriptor, containing the Micro-Doppler information of the neighboring cells. Since the radar cube is not compensated for ego-motion, it is crucial to use the relative radial velocity  $v_{rel}$  for connecting target- and low-level features. Note that not the full depth in velocity of the cube is cropped, just a part around the main velocity of the target. After this step, target-level radial velocities are compensated for ego-motion, i.e., I use  $v_r$ , not  $v_{rel}$ . See "Pre-processing" part in Figure 5.4.

### 5.2.2 NETWORK

*RTCnet* consists of three main modules as seen in Figure 5.4, marked as I, II, and III. The first two are convolution steps to process the cropped low-level data, while the final fully connected layers also consider the high-level features to calculate the classification scores.

#### DOWN-SAMPLE RANGE AND AZIMUTH DIMENSIONS

The first part's aim is to encode the radar target's spatial neighborhood's Doppler distribution into a tensor without extension in range or azimuth. In other words, it transforms the  $1 \times W \times L \times H$  sized data to a  $C \times 1 \times 1 \times H$  sized tensor (sizes are given as *Channel*  $\times$  *Azimuth*  $\times$  *Range*  $\times$  *Doppler*), where  $C$  was chosen as 25. To do this, it contains two 3D convolutions (Conv) with the kernel sizes of  $6 \times 3 \times 3 \times 3$  and  $25 \times 3 \times 3 \times 3$  (padding is 1). Both convolutional layers are followed by a maxpool (MP) layer with the kernel sizes of  $6 \times 2 \times 2 \times 1$  and  $25 \times 2 \times 2 \times 1$  with 0 padding to down-sample in the spatial dimensions. Technically, the first module of the

network can be skipped if both  $L$  and  $W = 1$ , i.e., I could use only the Doppler vector of the exact spatial bin containing the target. However, I found using an extended radius beneficial.

### PROCESSING THE DOPPLER DIMENSION

The second part of the network operates on the output of the first which is a  $25 \times 1 \times 1 \times H$  sized tensor. The aim of this module is to extract class information from the speed distribution around the target. To do this, I use three 1D convolutions along the Doppler dimension with the kernel size of 7 and output channel sizes of 16, 32, 32. Each convolution is followed by a maxpool layer with the kernel size of 3 and stride of 2, which halves the length of the input. The output of this module is a  $32 \times 1 \times 1 \times H/8$  block.

### SCORE CALCULATION

The output of the second module is flattened and concatenated to the target-level features ( $r$ ,  $\alpha$ ,  $v_r$ ,  $RCS$ ), and fed into the third one. I use two fully connected layers with 128 nodes each to provide scores. The output layer has either four nodes (one for each class) for multi-class classification or two for each of the binary tasks. In the latter case, ensemble voting is applied, see next subsection.

### 5.2.3 ENSEMBLE CLASSIFYING

With four output nodes, it is possible to train the third module to perform multi-class classification directly. I also implemented an ensemble voting system of binary classifiers (networks with two output nodes). That is, aside training a single, multi-class network, I followed [124] and trained One-vs-All (OvA) and One-vs-One (OvO) binary classifiers for each class (e.g. car-vs-all) and pair of classes (e.g. car-vs-cyclist), 10 in total. The final prediction scores depend on the voting of all the binary models. OvO scores are weighted by the summation of the corresponding OvA scores to achieve a more balanced result. Although I experimented with ensembling multi-class classifiers trained on bootstrapped training data as well, it yielded worse results.

### 5.2.4 OBJECT CLUSTERING

The output of the network (or voting) is a predicted class label for each target individually. To obtain proposals for object detection, I cluster the classified radar targets with DBSCAN incorporating the predicted class information, i.e. radar targets with *bike/pedestrian/car* predicted labels are clustered in separate steps. As metric, I used a spatial threshold  $\gamma_{xy}$  on the Euclidean distance in the  $x, y$  space (2D Cartesian spatial position), and a separate speed threshold  $\gamma_v$  in velocity dimension as in *Prophet* [55] or [65]. The advantage of clustering each class separately is that no universal parameter set is needed for DBSCAN. Instead, I can use different parameters for each class, e.g. larger radius for cars and small ones for pedestrians (Figure 5.2, A and B). Furthermore, swapping the clustering and classification step makes it possible to consider objects with a single reflection, e.g. setting *MinPoints* to one for pedestrian labeled radar targets (Figure 5.2, C). A possible drawback is that if a subset of an object's reflections are misclassified (e.g. a car with multiple targets, most labeled *car* and some as *cyclist*), the falsely classified targets (i.e. the *cyclist* ones) will be mistakenly clustered into a separate object. To address this, I perform a filtering on the produced object proposals, calculating their spatial, (radial) velocity, and class score distribution distances

(scores are handled as 4D vector, and I take their Euclidean distance after normalization). If two clusters have different classes and are close enough in all dimensions (see parameters in Subsection 5.4.2), I merge the smaller class to the larger (i.e. pedestrians to cyclists and cars, cyclists to cars) given that the cluster from the larger class has more radar targets.

### 5.3 DATASET

THE real-world dataset contains  $\sim 1$  hour of driving in urban environment with our demonstrator vehicle [120]. I recorded both the target-level and low-level output of our radar, a Continental 400 series mounted behind the front bumper. I also recorded the output of a stereo camera ( $1936 \times 1216$  px) mounted on the wind-shield, and the ego-vehicle’s odometry (filtered location and ego-speed).

Annotation was fetched automatically from the camera sensor using the Single Shot Multibox Detector (SSD) [107] trained on the EuroCity Persons dataset [46]. Distance is estimated by projecting each bounding box into the stereo point cloud computed by the Semi-Global Matching algorithm (SGM) [125], and taking the median distance of the points inside each. In a second iteration, mislabeled ground truth were manually corrected, e.g. cyclist annotated as pedestrian. See Figure 5.5 for an overview of the annotation process.

The training set contains more than  $30/15/9 \times 10^3$  pedestrian/cyclist/car instances respectively (one object may appear on several frames), see Table 5.2. Figure 5.9 shows the distribution of radar targets in the training set distance-wise. To further extend the training dataset, I augmented the data by mirroring the radar frames and adding a zero-mean, 0.05 std Gaussian noise to the normalized  $r$  and  $v_r$  features. Training and testing sets are from two independent driving (33 and 31 minutes long) which took place on different days and routes. Validation set is a 10% split of training dataset after shuffling.

|   | Pedestrians | Bikers | Cars  |
|---|-------------|--------|-------|
| Number of instances                       | 31300       | 15290  | 9362  |
| Number of radar targets                   | 63814       | 45804  | 30906 |
| Avg. number of radar targets per instance | 2.04        | 3.00   | 3.30  |
| Instances with only one radar target      | 12990       | 3526   | 2878  |
| Ratio of instances with one radar target  | 41.5%       | 18.8%  | 37.6% |

Table 5.2: Number of instances from each class in my training set. Many road users have only one radar reflection, which is not enough to extract meaningful statistical features via clustering.

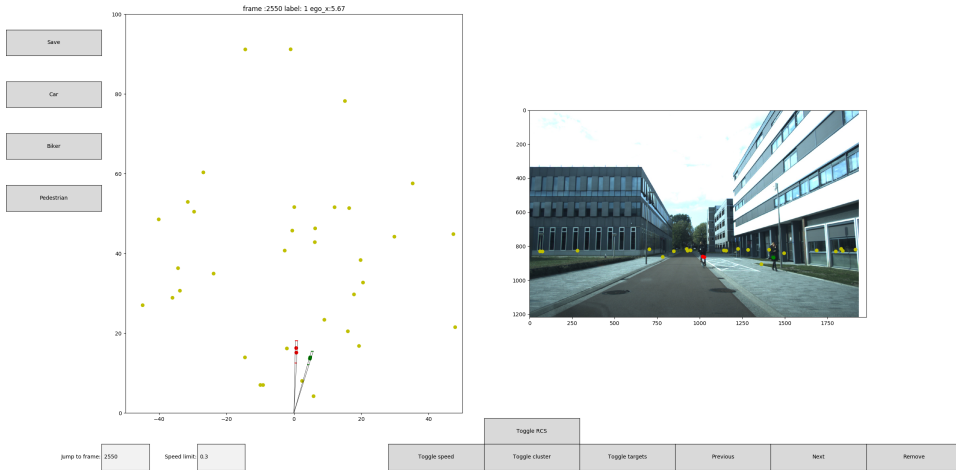


Figure 5.5: Demonstration of the annotation pipeline. Annotations were automatically fetched from the camera sensor by projecting the detections of the Single Shot Multibox Detector (SSD) [107] to the stereo point cloud. On the left, top view of the radar point cloud is shown with two road users (red: cyclist, green: pedestrian) automatically detected and annotated. On the right, the same scene is projected into the camera view for visualization. In a second iteration, the automatically labeled targets were manually reviewed using this annotator tool.

## 5.4 EXPERIMENTS

I compared the proposed method, *RTCnet* with binary bagging (from now on, referred to as *RTCnet*) to two baselines in two experiments to examine their radar target classification and object detection capabilities.

In the first experiment, I examined their performance in classification task, using a target-wise metric, i.e. a true positive is a correctly classified target [50]. For cluster-wise methods (the baselines) the predicted label of a cluster is assigned to each radar target inside following [50]. Furthermore, I also performed an ablation study to see how different features benefit the method in this classification (adaptation in brackets). *RTCnet (no ensemble)* is a single, multi-class network to see if ensembling is beneficial. *RTCnet (no RCS)* is identical to *RTCnet*, but the RCS target-level feature is removed to examine its importance. Similarly, in *RTCnet (no speed)* the absolute speed of the targets is unknown to the networks, only the relative speed distribution (in the low-level data) is given. Finally, *RTCnet (no low-level)* is a significantly modified version as it only uses target-level features. That is, the first and second convolutional parts are skipped, and the radar targets are fed to the third fully connected part directly. Note that in contrast to *RTCnet (no speed)*, *RTCnet (no low-level)* has access to the absolute speed of the target, but lacks the relative speed distribution. Object clustering is skipped in the first experiment.

In the second experiment, I compare the methods in object detection task, examining the whole pipeline, including the object clustering step. Predictions and annotations are compared by their intersection and union calculated in number of targets, see Figure 5.6. A true positive is a prediction that has an Intersection Over Union (IoU) bigger than or equal

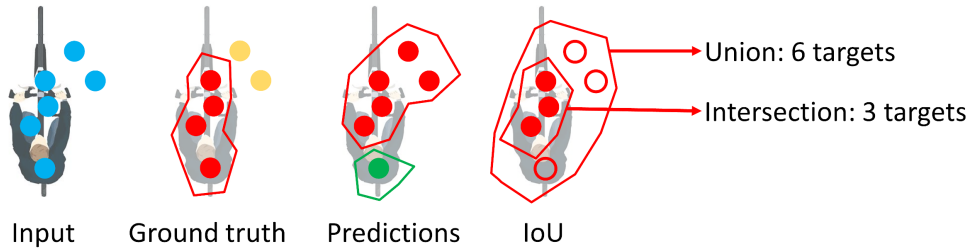


Figure 5.6: Object-level metric. Intersection and Union are defined by number of radar targets.  $\frac{\text{Intersection}}{\text{Union}} \geq 0.5$  counts as a true positive. In this example, there is a true positive cyclist and a false positive pedestrian detection.

to 0.5 with an annotated object. Further detections of the same ground truth object count as false positives. All presented results were measured on moving radar targets to focus on moving road users.

## 5

### 5.4.1 BASELINES

I selected *Schumann* [54] as a baseline because it is the only multi-object, multi-class detection method found with small latency, see Table 5.1. As no other research handled multiple classes, I selected *Prophet* [55] as my second baseline, which is a single-class pedestrian detector, but the negative training and testing set contained cars, dogs, and cyclists. I re-implemented their full pipeline (DBSCAN clustering and cluster-wise classification) and trained their algorithms with my training set. Optimal DBSCAN parameters are sensor specific (depending on density, resolution, etc.), thus I optimized the threshold in spatial dimensions  $\gamma_{xy}$  (0.5 m – 1.5 m, step size 0.1 m) and the threshold in velocity  $\gamma_v$  (0.5 – 1.5 m/s, step size 0.1 m/s) on the validation set for both baselines independently. I used the same metric as in the object clustering. Both baselines have features describing the number of static radar targets in the cluster. I also searched for an optimal speed threshold  $v_{min}$  (0–0.5 m/s, step size 0.1 m/s) for both to define these static radar targets. All reported results for baselines were reached by using their optimal settings, see Table 5.3. *MinPoints* was set to two as in *Prophet* [55] (increasing it further would exclude almost all pedestrians, see Table 5.2). In *Schumann* [54] the authors used manually corrected clusters (i.e. separating objects falsely merged by DBSCAN) to focus on the classification. I did not correct them to examine real-life application possibilities. I implemented a Random Forest classifier with 50 trees for both baselines, as *Prophet* [55] reported it to be the best for their features. *Schumann* [54] also tested LSTM, but used several frames aggregated as input.

### 5.4.2 IMPLEMENTATION

I set  $L = W = 5$ ,  $H = 32$  as the size of the cropped block. Speed threshold to filter out static objects is a sensor specific parameter and was set to 0.3 m/s based on empirical evidence. Table 5.3 shows the DBSCAN parameters for both baselines and for the class-specific clustering step. The thresholds to merge clusters during object clustering were set to 1 m spatially, and 0.6 for scores. The velocity threshold is 2 m/s for pedestrian to cyclist, and 1.2 m/s for pedestrian/cyclist to car merges.

I normalized the data to be zero-mean and have a standard deviation of 1 feature-wise for

| Method                    | $\gamma_{xy}$ | $\gamma_v$ | <i>MinPoints</i> | $v_{min}$ |
|---------------------------|---------------|------------|------------------|-----------|
| <i>Prophet</i> [55]       | 1.2 m         | 1.3 m/s    | 2                | 0.4 m/s   |
| <i>Schumann</i> [54]      | 1.3 m         | 1.4 m/s    | 2                | 0.4 m/s   |
| Class-specific (peds.)    | 0.5 m         | 2.0 m/s    | 1                | –         |
| Class-specific (cyclists) | 1.6 m         | 1.5 m/s    | 2                | –         |
| Class-specific (cars)     | 4.0 m         | 1.0 m/s    | 3                | –         |

Table 5.3: Optimized DBSCAN parameters for the two baselines, and for the class-specific clustering for each class.

| Method                       | Pedestrian  | Cyclist     | Car         | Other       | Avg.        |
|------------------------------|-------------|-------------|-------------|-------------|-------------|
| <i>Prophet</i> [55]          | 0.61        | 0.58        | 0.34        | 0.91        | 0.61        |
| <i>Schumann</i> [54]         | 0.67        | <b>0.68</b> | 0.46        | <b>0.92</b> | 0.68        |
| <i>RTCnet</i> (no low-level) | 0.56        | 0.63        | 0.33        | 0.90        | 0.61        |
| <i>RTCnet</i> (no speed)     | 0.66        | 0.63        | 0.36        | 0.91        | 0.64        |
| <i>RTCnet</i> (no RCS)       | <b>0.71</b> | 0.66        | 0.48        | 0.91        | 0.69        |
| <i>RTCnet</i> (no ensemble)  | 0.67        | 0.65        | 0.47        | 0.89        | 0.67        |
| <i>RTCnet</i>                | <b>0.71</b> | 0.67        | <b>0.50</b> | <b>0.92</b> | <b>0.70</b> |

Table 5.4: Target-wise F1 scores per class (best in bold). *RTCnet* outperforms the baselines on average. The ablation study shows benefits of ensembling and using low-level data.

$r$ ,  $\alpha$ ,  $v_r$ , *RCS*, and for the whole radar cube. At inference time, variance and mean values calculated from training data are used. I used PyTorch [126] for training with a cross-entropy loss (after softmax) in 10 training epochs. Inference time is  $\sim 0.04$  s on a high-end PC (Nvidia TITAN V GPU, Intel Xeon E5-1650 CPU, 64 GB RAM), including all moving radar targets, the 10 binary classifiers and the ensembling.

### 5.4.3 RESULTS

#### TARGET CLASSIFICATION

I present the results of the target classification experiment in Table 5.4. Target-wise F1 scores for all classes and their macro-average are given for each method. *RTCnet* outperformed the two cluster-wise baselines reaching an average F1 score of 0.70. *Schumann* [54] has slightly better results on *cyclists* than *RTCnet* (0.68 vs 0.67), but performed significantly worse on *pedestrians* (0.67 vs 0.71) and *cars* (0.46. vs 0.50). The ablation study showed that removing each feature yields worse results than the complete pipeline, but the one without reflectivity information (*RTCnet* (no *RCS*)) comes close with an average of 0.69. Removing the low-level features (*RTCnet* (no low-level)) decreased the performance significantly to an average of 0.61. The multi-class (single) network *RTCnet* (no ensemble) outperforms the baselines on the *car* class, but performs worse on *cyclists*. Ensemble voting brings significant improvement on all classes. Example of correct and incorrect target classifications are shown in Figure 5.7 and 5.8 for all road user classes. In Figure 5.9 I show how the classification performance (target-wise F1 score) changes over distance (with 5 m bins) for each class, along with the number of radar targets in the training set. Although most annotation fall



Figure 5.7: Examples of correctly classified radar targets by *RTCnet*, projected to image plane. Radar targets with pedestrian/cyclist/car labels are marked by green/red/blue. Static objects and the class *other* are not shown.

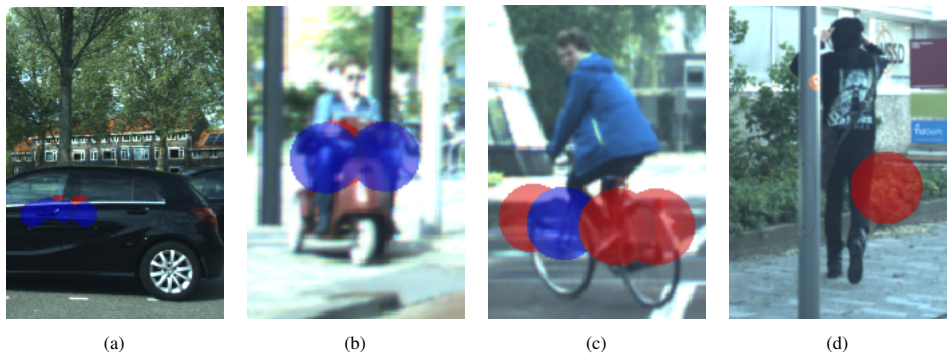


Figure 5.8: Examples of radar targets misclassified by *RTCnet*, caused by: flat surfaces acting as mirrors and creating ghost targets (a), unusual vehicles (b), partial misclassification of an objects' reflections (c), and strong reflections nearby (d).

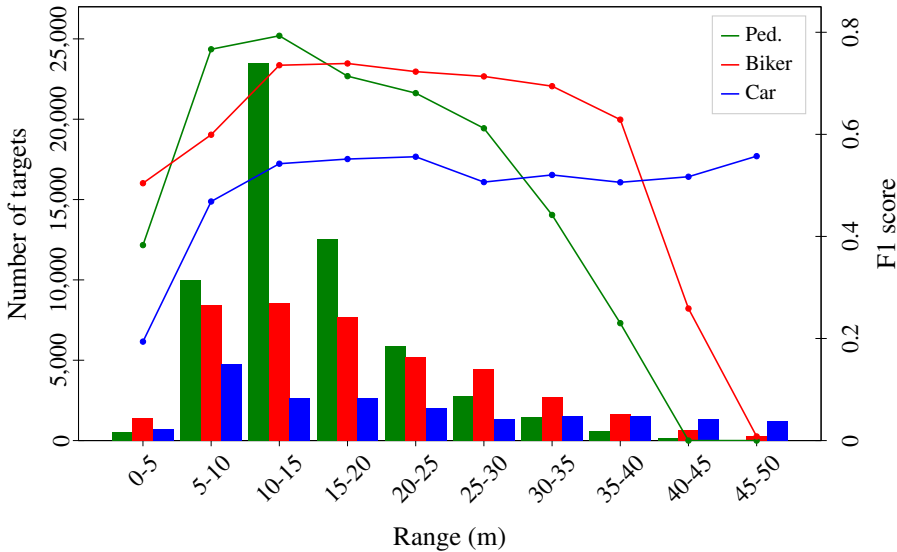


Figure 5.9: Target-wise F1 scores (lines) and number of targets in training set (bars) as a function of distance from ego-vehicle.

|                          | Pedestrian  | Cyclist     | Car         | Avg.        |
|--------------------------|-------------|-------------|-------------|-------------|
| <i>Prophet</i> [55]      | 0.48        | 0.50        | 0.23        | 0.40        |
| <i>Schumann</i> [54]     | 0.54        | <b>0.60</b> | 0.31        | 0.48        |
| <i>RTCnet</i> (proposed) | <b>0.61</b> | 0.59        | <b>0.47</b> | <b>0.56</b> |

Table 5.5: F1 scores object-wise (best score in bold). *RTCnet* outperforms the baselines on average.

into the 5 – 20 m range, the network performs reasonably beyond that distance, especially for the larger objects (*cyclist*, *car*). I trained One-vs-All classifiers both for *RTCnet* and *Schumann* [54] for each road user class, and plotted their performance on receiver operating characteristic (ROC) curves in Figure 5.10. The varied threshold is cluster-wise for *Schumann* [54] and target-wise for *RTCnet*. My method has a larger area under the curve of all classes.

## OBJECT DETECTION

The results of the second experiment are shown in Table 5.5. *RTCnet* reached slightly worse results on *cyclists* than *Schumann* [54] (0.59 vs 0.60), but significantly outperformed it on *pedestrians* (0.61 vs 0.54), *cars* (0.47 vs 0.31), and on average (0.56 vs 0.48). Figure 5.11 shows how *Schumann* [54] and *RTCnet* handled two real-life cases from Figure 5.2. Examples of both correct and incorrect object detections by *RTCnet* are shown in Figure 5.13. A link to a video of the results can be found on our website<sup>2</sup>.

<sup>2</sup><http://intelligent-vehicles.org/publications/>



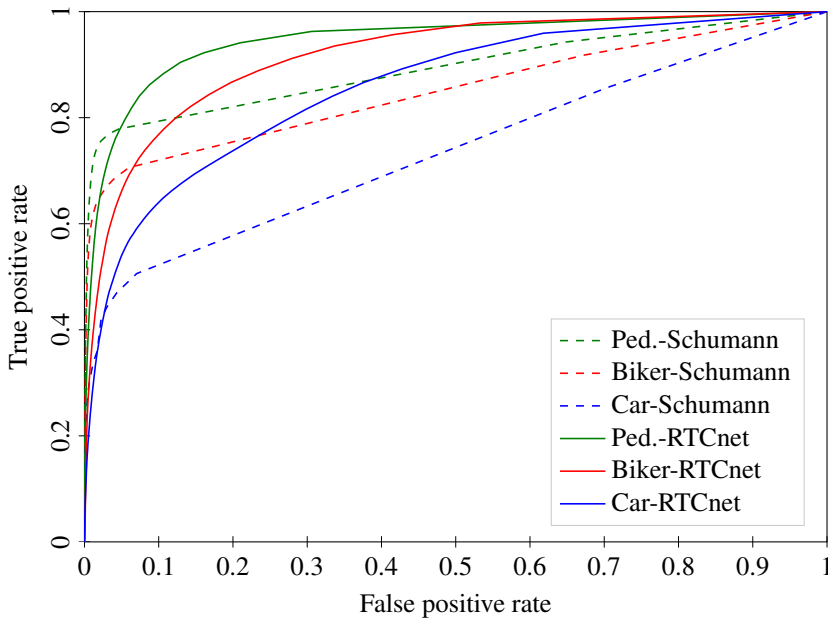


Figure 5.10: ROC curves of road user classes by my method and *Schumann* [54]. Each curve is calculated by changing the decision threshold of a One-vs-All binary classifier.

#### 5.4.4 DISCUSSION

My method outperformed the baselines in target classification mainly due to two reasons. First, the classification does not depend on a clustering step. This decreases the impact of cases shown in Figure 5.2 and allows to handle objects that contain a single radar target (a common occurrence, especially for pedestrians, see Table 5.2). Second, I included low-level radar data, which brings information on the speed distribution around the radar target. To demonstrate that this inclusion is beneficial, I showed that only using target-level data and only the third module of the network (*RTCnet (no low-level)*) caused a significant drop in performance from 0.70 to 0.61 average F1 score. I examined the effect of removing absolute speed from the data too with *RTCnet (no speed)*. While the performance dropped, my network was still able to classify the radar targets by the relative speed distribution around them. The results of *RTCnet (no low-level)* and *RTCnet (no speed)* prove that the relative velocity distribution (i.e. the low-level radar data) indeed contains valuable class information. Interestingly, excluding RCS value did not have a significant impact on the performance. Based on my experiments, an ensemble of binary classifiers results in fewer inter-class miss-classifications than using a single multi-class network. However, inference is of course significantly faster using a single network instead of 10.

Note that even VRUs in occlusion (see Figure 5.7a, 5.7b, 5.7g) are often classified correctly caused by the multipath propagation of radar [40]. This, and its uniform performance in darkness/shadows/bright environments makes radar a useful complementary sensor for camera. Typical errors are shown in Figure 5.8. Radar is easily reflected by flat surfaces (e.g.

side of cars) acting like mirrors, creating *ghost targets*. E.g. in Figure 5.8a our ego-vehicle was reflected creating several false positives. Figure 5.8b is an example of hard to categorize road users. Many errors come from the confusion of *car* and *cyclist* caused by the similarity of their Doppler signature and reflectivity, see Figure 5.8c. Figure 5.8d shows that a strong reflection nearby can mislead the classifier. Since the proposed method does not throw away single targets in a clustering step, it has to deal with more noise reflections than a cluster-wise method. However, the results in *other* class suggest that it learned to ignore them.

The combination of my network and the clustering step outperformed the baseline methods in the object detection task. This is mainly because by swapping the clustering and classifying steps, classes can be clustered with different parameters. That is a significant advantage of my pipeline, as instead of finding a single set of clustering parameters to handle each class, I can tune them separately to fit each, see Table 5.3. This is especially useful in *pedestrian* and *car* classes, which are smaller/larger than the optimal spatial radius  $\gamma_{xy} = 1.2\text{--}1.3\text{ m}$  found for the baselines. However, this radius fits bicycles well, which results in good performance on the *cyclists* class for *Schumann* [54] both on target-level and object-level. Figure 5.11 shows two examples. DBSCAN falsely separated the car and the bus into several clusters, but merged the pedestrians into a single one using the optimized parameters, which caused *Schumann* [54] to fail. My method managed to classify each radar target individually and cluster them correctly (i.e. keep the vehicles in a single cluster, but separate the pedestrians) using the class-specific clustering parameters. Although I used DBSCAN in this work, we can expect this advantage to stand using different types of clustering. Figure 5.12 shows an other example where multiple objects from different classes were successfully detected and clustered. In Figure 5.13a I show a single mis-classified radar target, probably reflected by the speed bump. The resulting false positive pedestrian detection is trade-off of setting *MinPoints* to one for pedestrians. As mentioned, cyclists and cars are often confused. This is especially true if several cyclists ride side-by-side, see 5.13a, since their radar characteristics (extension, speed, reflectivity) are car-like. Both errors usually occur for a single frame only, and can be alleviated by a temporal filtering and tracking system.

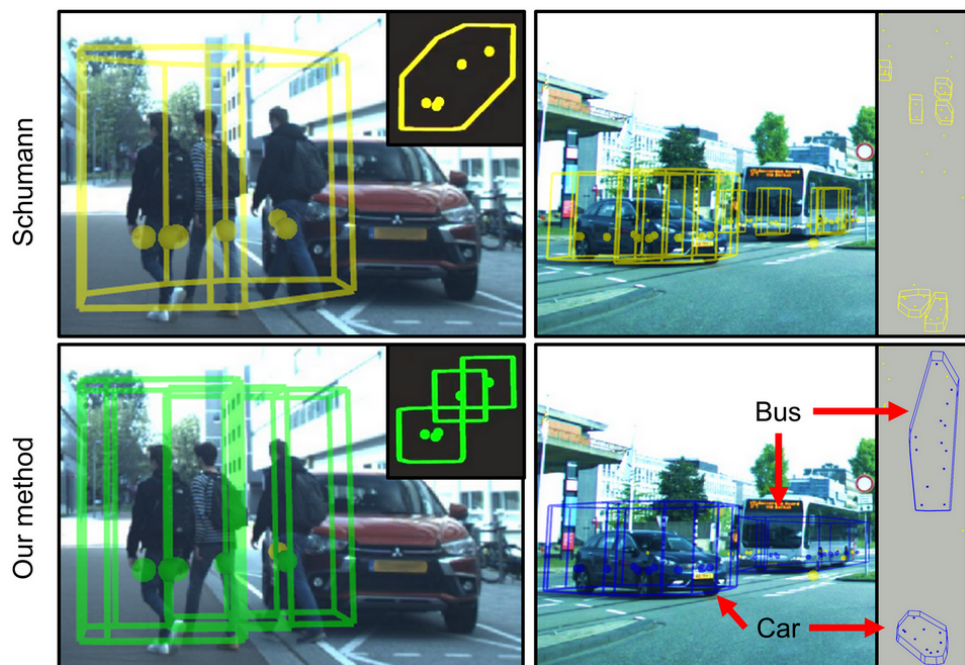


Figure 5.11: Challenging cases for clustering, camera and top view. DBSCAN falsely split the car and the bus but merged the pedestrians into a single cluster, making *Schumann* [54] (top) fail. My method (bottom) managed to classify the radar targets and cluster them correctly using class-specific parameters. Yellow marks *other* class.

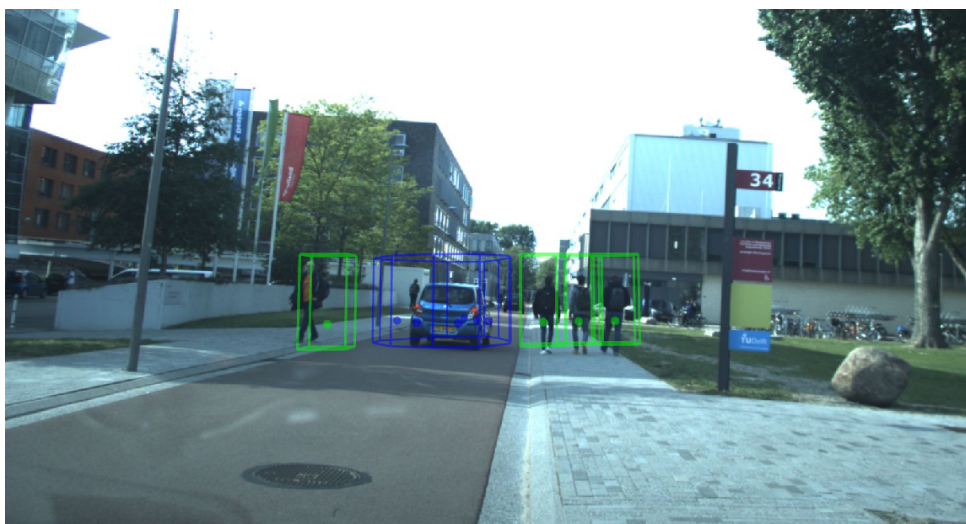


Figure 5.12: Correctly classified and clustered radar targets from multiple classes (green for pedestrian, blue for car). The height of the bounding boxes is predefined and is aimed only for visualization.

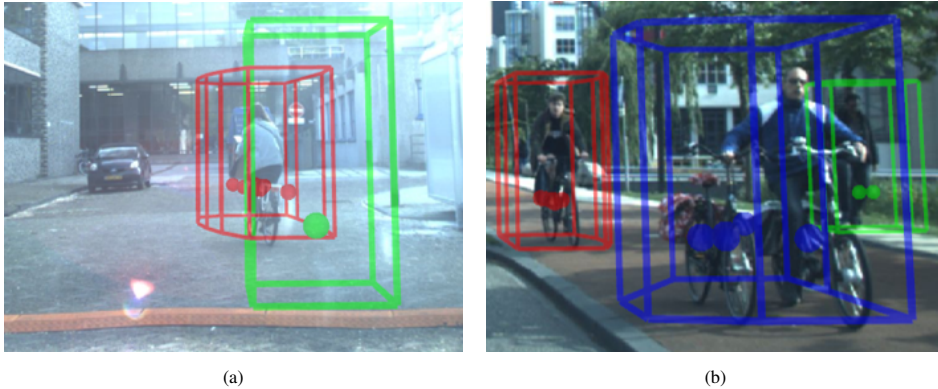


Figure 5.13: Examples of correct and incorrect object detections of my method. A mis-classified radar target triggered a false positive pedestrian detection on (a). Bicycles moving side-by-side at the same speed are detected as a car on (b) because they have radar characteristics (speed, extension, reflectivity) similar to a car.

## 5.5 CONCLUSION OF THE CHAPTER

**I**N this chapter, a radar based, single-frame, multi-class road user detection method was proposed. It exploits class information in low-level radar data by applying a specially designed neural network to a cropped block of the radar cube around each radar target and the target-level features. A clustering step was introduced to create object proposals.

In extensive experiments on a real-life dataset I showed that the proposed method improves upon the baselines in target-wise classification by reaching an average F1 score of 0.70 (vs. 0.68 *Schumann* [54]). Furthermore, I demonstrated the importance of low-level features and ensembling in an ablation study. I showed that the proposed method outperforms the baselines overall in object-wise classification by yielding an average F1 score of 0.56 (vs. 0.48 *Schumann* [54]).

Future work may include a more advanced object clustering procedure, e.g., by training a separate head of the network to encode a distance metric for DBSCAN. Temporal integration and/or tracking of objects could further improve the method's performance and usability. Finally, extending the proposed framework to incorporate data from additional sensor modalities (e.g. camera, LiDAR) is worthwhile.



# 6

## MULTI-CLASS ROAD USER DETECTION USING THE 3+1D RADAR POINT CLOUD

*Roads? Where we're going, we don't need roads.*

Dr. Emmett Brown in Back To The Future

6

## 6.1 INTRODUCTION

**I**N recent years, developments in both radar technology and proposed algorithms have enabled the use of 2+1D radars for road user detection [49][50][54][56][57]. The methods presented in Chapter 4 and 5 are also good examples of such applications.

Despite these improvements, the sparsity of point clouds provided by traditional automotive radars is still a bottleneck in object detection research. Due to their small number of points, it is challenging to regress accurate 2D bird's-eye view (BEV) bounding boxes, especially for smaller objects such as pedestrians. Furthermore, the lack of elevation information (i.e., the height of the points) makes it nearly impossible to infer the height and vertical offset of objects, i.e., to regress 3D bounding boxes. Hence, unlike LiDAR based detectors, most 2+1D radar based object detection methods do not regress bounding boxes either in 2D (BEV) or in 3D, but instead perform semantic or instance segmentation of the 2+1D radar point clouds [49][50][55][68][69][70]. Bounding box regression on sparse radar point clouds remains challenging since the objects usually only have a few points on them, providing little spatial information on the exact location and extent of the true bounding box.

The latest improvement in automotive radar technology, *3+1D radars* may help to overcome these limitations. As discussed in Subsection 2.4.3, unlike traditional automotive radars, 3+1D radars have three spatial dimensions: range, azimuth, and elevation, while still providing Doppler as a fourth dimension. They also tend to provide a denser point cloud [14]. With the additional elevation information and increased density, 3+1D radar point clouds are somewhat reminiscent of LiDAR point clouds. Therefore, these radars may be better suited for multi-class 3D bounding box regression, and it is intuitive to apply object detection networks developed for LiDAR data to them. Nonetheless, 3+1D radars have only been used for the single-class car detection task [35][75], not for pedestrian, cyclist, or multi-class detection tasks.

I see two possible reasons for this. First, the object detection networks regularly used for LiDAR input were not designed with the Doppler dimension in mind, and it is unclear how best to incorporate this additional information. Furthermore, the measured Doppler values depend on the direction in which the object is located, so many data augmentation techniques often applied to LiDAR point clouds are not suitable for radar ones.

Second, while many datasets contain several thousand 3D bounding box annotations for multiple classes on LiDAR data [87][88][127], there is a clear lack of datasets with radar data. In Section 3.5 a detailed overview of publicly available datasets with radar data was given. To be suitable for multi-class road user detection tasks with radar (either pure radar or sensor fusion), I argue that an automotive dataset should meet the following requirements:

1. use a next-generation 3+1D radar to provide both elevation and Doppler information,
2. equip high-end sensors from the other modalities as well, i.e., a high definition camera and a 64-layer LiDAR,
3. provide annotations for the objects that include their extent and orientation (2D BEV or 3D bounding boxes), and
4. should have reasonable number of annotations for the most important urban road users: pedestrians, cars, and cyclists.

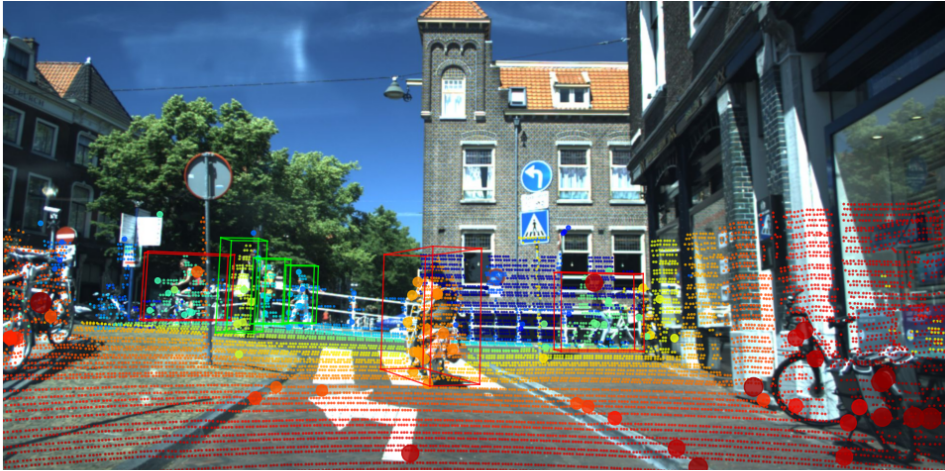


Figure 6.1: Example scene from the View-of-Delft (VoD) dataset. The recordings contain camera images, LiDAR point clouds (shown here as lines of small dots), and 3+1D radar data (shown as large dots), along with accurate localization information and 3D bounding box annotations (cyclist/pedestrian class labels are colored red/green). LiDAR and radar points are colored based on their distance from the ego-vehicle.

| Name                        | Radar data           | Camera | LiDAR            | Size        | Vehicles    | Pedestrians | Cyclists  | Annotation       |
|-----------------------------|----------------------|--------|------------------|-------------|-------------|-------------|-----------|------------------|
| RadarScenes (2021) [84]     | 4×2+1D, front/side   | mono   | ✗                | 832k frames | 326636/3889 | 128197/1529 | 61051/268 | point-wise       |
| CRUW (2021) [85]            | 2×2+1D, front/side   | stereo | ✗                | 396k frames | 23330/-     | 31980/-     | 13347/-   | 2D position      |
| RADIATE (2020) [36]         | 1×2D, surround       | stereo | 32 layers        | 3 hours     | 185810/-    | 10970/-     | 499/-     | 2D bboxes        |
| Zendar (2020) [86]          | 1×2+1D, front        | mono   | 16 layers        | 4780 frames | 11300/-     | 0/-         | 0/-       | 2D bboxes        |
| nuScenes (2019) [87]        | 5×2+1D, surround     | 6×mono | 32 layers        | 400k frames | 598849/-    | 217913/-    | 7331/-    | 3D bboxes        |
| Astyx (2019) [14]           | 1×3+1D, front        | mono   | 16 layers        | 546 frames  | 3087/-      | 39/-        | 11/-      | 3D bboxes        |
| <b>View-of-Delft (2021)</b> | <b>1×3+1D, front</b> | stereo | <b>64 layers</b> | 8693 frames | 27273/429   | 26587/380   | 10800/183 | <b>3D bboxes</b> |

Table 6.1: Comparison of publicly available radar detection datasets with sensors used, type of annotation, and the number of vehicle (sum of *car*, *truck*, and *bus*), *pedestrian*, and *cyclist* annotations (individual annotations/unique instances, where unique object id is available). Top/bottom sections are datasets with radars providing 2D/3D spatial coordinates.

Table 6.1 summarizes the most relevant currently available radar detection datasets according to these requirements, listing their number of cyclist, pedestrian, and vehicle annotations, the type of those annotations, the sensor modalities used, and whether they have Doppler data. In conclusion, no existing publicly available dataset satisfies all the requirements. For example, the only publicly available detection dataset [14] with 3+1D radar data has only ~500 frames, with fewer than 40 annotations for pedestrians or cyclists, and thus, it is not suitable for multi-class object detection.

In this chapter, I apply a state-of-the-art object detector (PointPillars [13]), commonly used for LiDAR 3D data, to such 3+1D radar data. I incorporate the Doppler information, and explore how it influences the detection performance. Furthermore, I investigate how elevation information and the use of past radar scans (i.e. temporal information) increase road user detection performance. I also discuss what kind of data augmentation methods are applicable to 3+1D radar data. Finally, I compare my best radar based object detection method with a PointPillars network operating with LiDAR data, and examine the two sensors'



performance and capabilities as a function of class and distance.

To facilitate the experimental study, I introduce the View-of-Delft<sup>1</sup> (VoD) dataset, a multi-sensor automotive dataset for multi-class 3D object detection, see Figure 6.1.

### 6.1.1 CONTRIBUTIONS

The main contributions of the chapter are as follows:

1. I examine road user detection with 3+1D radar using PointPillars [13], a state-of-the-art multi-class 3D object detector commonly used for LiDAR. I investigate the importance of different features of the radar point cloud in an ablation study, including Doppler, RCS, and the elevation information that traditional 2+1D automotive radars cannot provide.
2. I compare radar based to LiDAR based detection by training and testing on the same traffic scenes. I show that currently point cloud based detection on dense LiDAR still outperforms detection on radar. However, I also find that the performance gap can be reduced when radar data includes elevation information, and when multiple radar scans are temporally integrated. Additionally, the detection benefits from Doppler measurements, which are unique to radar.
3. I publish the View-of-Delft (VoD) dataset, a novel multi-sensor automotive dataset for multi-class 3D object detection, consisting of calibrated and synchronized LiDAR, camera, and radar data recorded in real-world traffic situations with annotations for both static and moving road users. The View-of-Delft dataset is the largest dataset containing 3+1D radar recordings with ~20 times as many annotated frames as the Astyx dataset [14], and it is the only public dataset containing camera, (any kind of) radar, and 64-layer LiDAR data at the same time. Although this work focuses on radar-only methods, the dataset is also suitable for sensor fusion, camera-only, or LiDAR-only methods due to this sensor arrangement, and could be useful for researchers interested in cluttered urban traffic.

## 6

## 6.2 METHODOLOGY

**T**HIS work uses PointPillars [13] as the baseline state-of-the-art multi-class object detector. While PointPillars is typically trained on LiDAR data, I instead train it on 3+1D radar point clouds. In this section I detail the available features of the radar input, and describe how to encode Doppler. I also discuss data augmentation techniques and describe temporal merging of multiple radar scans.

### 6.2.1 3+1D RADAR POINT CLOUDS AND DOPPLER ENCODING

The 3+1D radar outputs a point cloud with spatial, Doppler and reflectivity channels for each scan, giving a total of five features for each point:  $r$  range,  $\alpha$  azimuth,  $\theta$  elevation,  $v_{rel}$  relative radial velocity, and RCS reflectivity. Since most point cloud based object detectors use Cartesian coordinates, I also transform the radar point cloud:  $p = [x, y, z, v_{rel}, RCS]$ , where  $p$  denotes a point, and  $x, y, z$  are the three spatial coordinates with  $x$  and  $y$  axes pointing

<sup>1</sup>Named after the famous painting by Johannes Vermeer (pun intended).

forward and left respectively w.r.t. the vehicle, see Figure 6.3. *Compensated radial velocity* is a signed scalar value denoted by  $v_r$ , describing the ego-motion compensated (i.e. absolute) radial velocity of the point. To obtain it, I perform ego-motion compensation for  $v_{rel}$  by eliminating the motion of the sensor that comes from both the translational and rotational movement of the ego-vehicle. Examples of such encoding of Doppler for multi-class object detection include [49] and [50].  $v_r$  was used as additional decoration for the radar points and it was normalized feature-wise to have zero-mean and unit standard deviation.

### 6.2.2 ACCUMULATION OF RADAR POINT CLOUDS

I experiment with incorporating multiple radar scans in the object detector similar to what [87] has been done for LiDAR and [50] for 2+1D radar data. Aside from the advantage of richer point clouds, merging also provides temporal information, which may help object detectors not only in localization but in classification as well. Accumulation is implemented by transforming point clouds from previous scans to the coordinate system of the last scan and appending a scalar time id denoted by  $t$  to each point indicating which scan it originates from. E.g., a point from the current scan has a  $t = 0$ , while a point from the third most recent scan has a  $t = -2$ . The encoder includes this time id as an extra decoration for the radar points. Note that a “scan” is not the same as a “frame” defined in Section 4.4. While radar point clouds in the frames are synchronized with the LiDAR sensor, here I merge the last scans received from the radar independently of other sensors.

### 6.2.3 DATA AUGMENTATION

Not every data augmentation method used in LiDAR research is directly applicable to radar point clouds since the  $v_r$  measured by the radar should remain correlated with the angle at which the object is observed. The same object with the same kinematics (speed and direction) would be detected with different velocity measurements at a different azimuth or elevation angle, i.e., after being translated during augmentation. Similarly, it is not possible to rotate the ground truth bounding boxes and the points within them locally (around their vertical axis), as this changes the radial component of the object velocity in an unknown way. Finally, rotating the radar point cloud around the sensor (e.g., around its vertical axis) does not affect the measured relative radial velocities. However, this is not true for the ego-motion compensated radial velocities, since the compensation uses the angles between the motion vector of the radar and the direction of the objects. Therefore, commonly used augmentation methods such as translation and rotation of the point cloud or rotation of the ground truth boxes can even be detrimental in the case of radar point clouds. However, mirroring the point cloud to the longitudinal axis and scaling are applicable, as the (absolute) observation angles of radar points do not change. Note that augmentation by scaling is only valid if the origin is the radar sensor itself.

## 6.3 DATASET

**I**N this section, I present the View-of-Delft dataset, including the sensor setup used and the annotations provided<sup>2</sup>. The dataset was recorded while driving with our demonstrator

<sup>2</sup>The VoD dataset, including its annotations for the training and validation sets, is freely available at [intelligent-vehicles.org/datasets/view-of-delft/](https://intelligent-vehicles.org/datasets/view-of-delft/) to academic and non-profit organizations for non-commercial, scientific use.

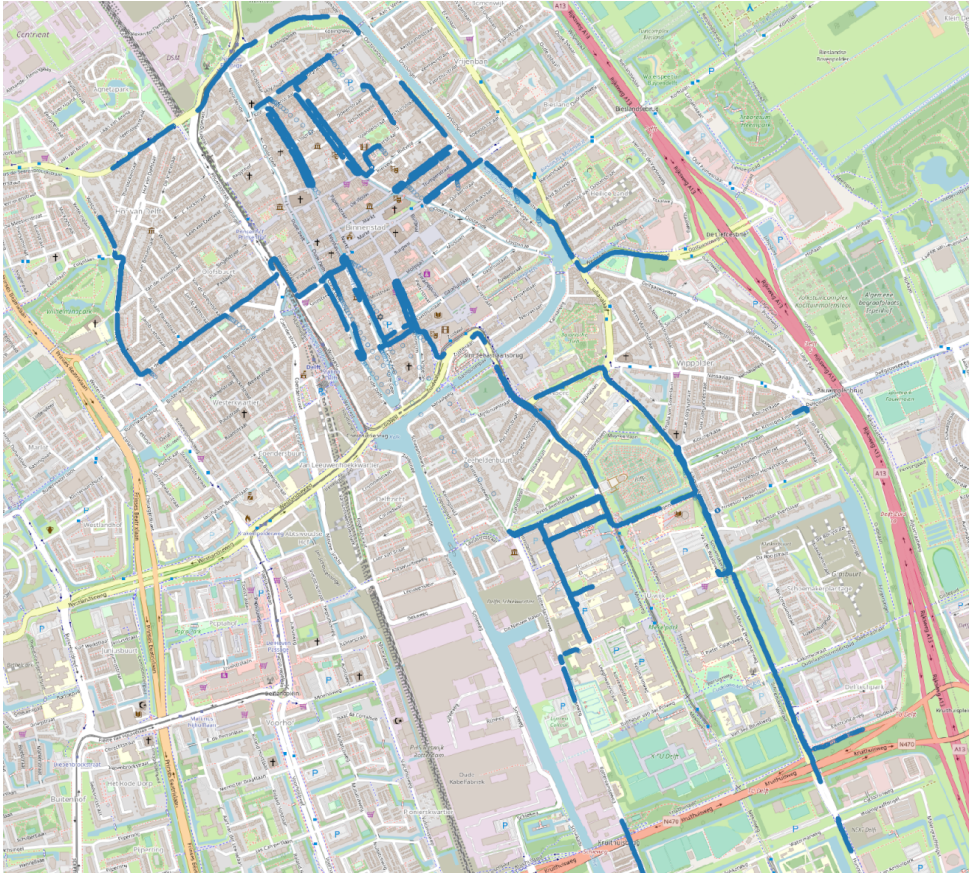


Figure 6.2: Recording Zone. This figure shows the GPS traces (blue) of the recordings in the urban region of Delft, the Netherlands.

vehicle [120] through *campus*, *suburb* and *old-town* locations in the city of Delft (The Netherlands). Recordings were selected with a preference for scenarios containing vulnerable road users (VRUs), i.e., pedestrians and cyclists. See Figure 6.2 for an overview of the recording locations.

### 6.3.1 MEASUREMENT SETUP AND PROVIDED DATA

I recorded the output of the following sensors: a ZF FRGen21 3+1D radar (see Table 6.2 for specifications,  $\sim 13$  Hz) mounted behind the front bumper, a stereo camera ( $1936 \times 1216$  px,  $\sim 30$  Hz) mounted on the windshield, a Velodyne HDL-64 S3 LiDAR ( $\sim 10$  Hz) scanner on the roof, and the ego vehicle's odometry (filtered combination of RTK GPS, IMU, and wheel odometry,  $\sim 100$  Hz). All sensors were jointly calibrated following [121], with a specially designed calibration board that can be reliably detected by LiDAR, stereo camera and radar sensors alike. See Figure 6.3 for a general overview of the sensor setup.

I provide the dataset in synchronized “frames” similar to [88], consisting of a LiDAR



Figure 6.3: Recording platform. Our Toyota Prius 2013 platform is equipped with a stereo camera setup, a rotating 3D LiDAR sensor, a ZF FRGen 21 3+1D radar, and a combined GPS/IMU inertial navigation system.

|            | range         | velocity | azimuth      | elevation   |
|------------|---------------|----------|--------------|-------------|
| Accuracy   | $\leq 0.02$ m | 0.01 m/s | $0.15^\circ$ | $0.3^\circ$ |
| Resolution | $\leq 0.2$ m  | 0.1 m/s  | $1.5^\circ$  | $1.5^\circ$ |

Table 6.2: Native accuracy and resolution along the four dimensions of our radar sensor configuration. On-board signal processing provides further resolution gains.

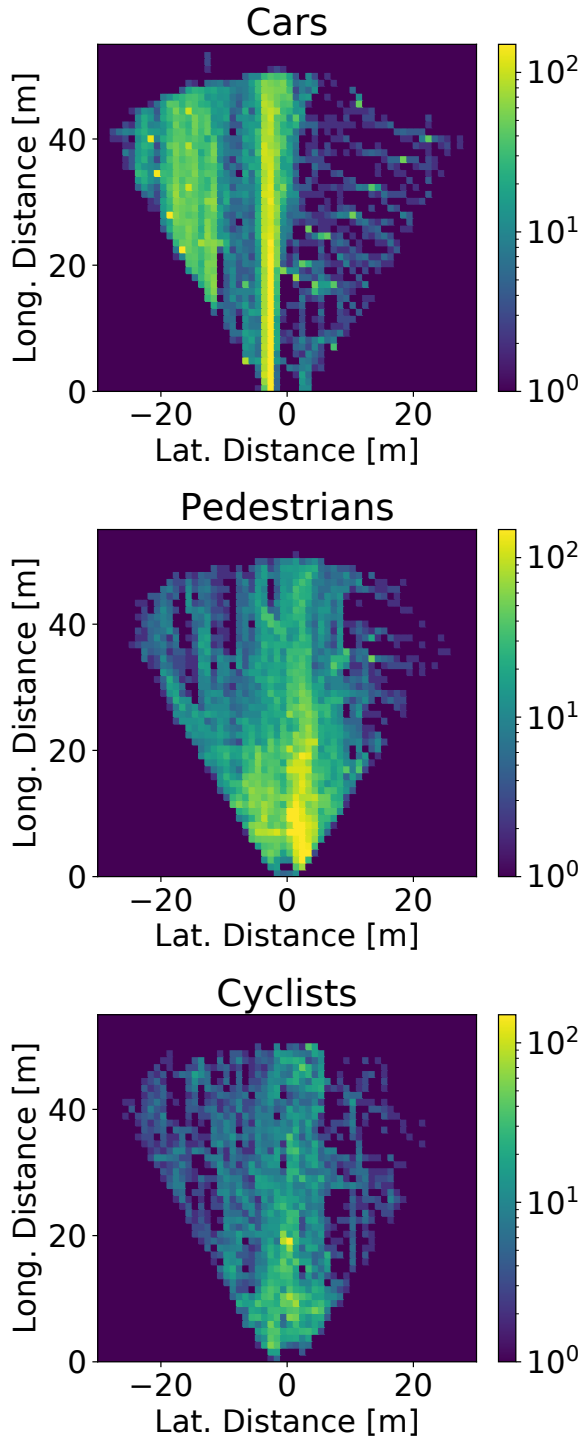


Figure 6.4: Overall spatial distribution of cars, pedestrians, and cyclists in the dataset as a log plot. The ego-vehicle is positioned at (0, 0), looking upwards. Each pixel corresponds to one square meter area. Darkest blue means zero annotation.

|              | car           | pedestrian    | cyclist      | rider         | unused bicycle | bicycle rack | human depiction | moped or scooter | motor      | other       | $\Sigma$ |
|--------------|---------------|---------------|--------------|---------------|----------------|--------------|-----------------|------------------|------------|-------------|----------|
| # objects    | 26949 (21.9%) | 26587 (21.6%) | 10800 (8.8%) | 12809 (10.4%) | 24933 (20.3%)  | 12025 (9.8%) | 370 (0.3%)      | 5403 (4.4%)      | 629 (0.5%) | 2601 (2.1%) | 123106   |
| # unique obj | 423 (22.6%)   | 380 (20.3%)   | 183 (9.8%)   | 222 (11.8%)   | 372 (19.8%)    | 156 (8.3%)   | 10 (0.5%)       | 80 (4.3%)        | 13 (0.7%)  | 36 (1.9%)   | 1875     |
| % moving     | 7.2%          | 73.2%         | 96.1%        | 95.5%         | 0.7%           | 0.0%         | 0.0%            | 10.7%            | 59.9%      | 34.5%       | 37.4%    |

Table 6.3: Dataset statistics: number of annotated objects (top), number of unique objects (middle), and percentage of moving objects (bottom), per class. The ratios compared to the whole dataset are given in brackets. The “other” column combines the classes *ride\_other*, *vehicle\_other*, *truck*, and *ride\_uncertain*.

point cloud, a rectified mono-camera image, a radar point cloud, and a transformation describing the odometry. Timestamps of the LiDAR sensor were chosen as “lead”, and I chose the closest camera, radar and odometry information available (maximum tolerated time difference is set to 0.05 seconds). The frames are sequential in time with 10 Hz (after synchronization) and they are organized into clips with an average length of  $\sim 40$  seconds. The LiDAR and radar point clouds are ego-motion compensated, both for ego-motion between the capture of LiDAR/radar and camera data, and for ego-motion during the scan (i.e., one full rotation of the LiDAR sensor). The dataset follows the popular KITTI dataset [88] both in the defined coordinate systems (see Figure 6.3) and in the file structure. The main advantage of this choice is that several open-source toolkits and detection methods are directly applicable to the dataset. In addition to this synchronized version of the dataset, I also make the “raw” asynchronous recorded data available, including all radar scans at 13 Hz, and rectified camera images at 30 Hz from both the left and right cameras. This can benefit researchers seeking richer temporal data for detection, tracking, prediction, or other tasks.

6

### 6.3.2 ANNOTATION

Any object of interest (static or moving) within 50 meters of the LiDAR sensor and partially or fully within the camera’s field of view (horizontal FoV:  $\pm 32^\circ$ , vertical FoV:  $\pm 22^\circ$ ) was annotated with a 3D bounding box with nine degrees of freedom (9 DoF): its 3D position, extent, and orientation. Annotation was done by [understand.ai](https://understand.ai) (a subsidiary of DSpace) using the camera images and LiDAR data. To help the annotators work, sequential LiDAR point clouds were spatially aligned using odometry information, and object annotations were semi-automatically tracked/propagated to subsequent frames. Note however, that final labels on all frames were manually tuned and validated. 13 object classes were annotated, see Table 6.3 for their object count. For each object, we also annotated the level of occlusion for two types of occlusions (“spatial” and “lighting”) and an activity attribute (“stopped”, “moving”, “parked”, “pushed”, “sitting”). Furthermore, some physical objects were assigned unique object ids over frames to make the dataset suitable for tracking and prediction tasks. Annotation instructions with detailed descriptions of the classes and attributes will be shared along with the dataset.

## 6.4 EXPERIMENTS

**I**N the experiments I consider object detection performance on three object classes: *car*, *pedestrian* and *cyclist*. The spatial distributions of these classes are shown in Figure 6.4. Unlike [30][49][50][84], I considered both static and moving objects in my experiments. I split the dataset into a training, validation, and testing set in a ratio of 59%/15%/26% such that frames from the same clip will only be present in one split. The clips are assigned to

| Method                         | Features               | Entire annotated area |             |             |             |             | In Driving Corridor |             |             |             |             |
|--------------------------------|------------------------|-----------------------|-------------|-------------|-------------|-------------|---------------------|-------------|-------------|-------------|-------------|
|                                |                        | Car                   | Pedestrian  | Cyclist     | mAP         | mAOS        | Car                 | Pedestrian  | Cyclist     | mAP         | mAOS        |
| <i>PP-radar (no elevation)</i> | $x, y, RCS, v_r$       | 32.4                  | 28.8        | 34.6        | 31.9        | 25.1        | 68.2                | 44.2        | 63.6        | 58.6        | 50.1        |
| <i>PP-radar (no Doppler)</i>   | $x, y, z, RCS,$        | 35.6                  | 21.3        | 30.4        | 29.1        | 22.1        | 67.3                | 31.0        | 58.7        | 52.3        | 41.0        |
| <i>PP-radar (no RCS)</i>       | $x, y, z, v_r$         | 33.9                  | 33.1        | 42.7        | 36.6        | 30.3        | 66.8                | 45.3        | <b>67.2</b> | 59.8        | 55.6        |
| <i>PP-radar</i>                | $x, y, z, RCS, v_r$    | <b>35.9</b>           | <b>34.9</b> | <b>43.1</b> | <b>38.0</b> | <b>30.5</b> | <b>74.1</b>         | <b>47.8</b> | 67.1        | <b>63.0</b> | <b>56.8</b> |
| <i>PP-radar (3 scans)</i>      | $x, y, z, RCS, v_r, t$ | 44.4                  | 40.4        | <b>54.2</b> | 46.3        | 39.1        | 78.4                | 56.9        | <b>76.6</b> | 70.6        | 67.1        |
| <i>PP-radar (5 scans)</i>      | $x, y, z, RCS, v_r, t$ | <b>44.8</b>           | <b>42.1</b> | 54.0        | <b>47.0</b> | <b>39.6</b> | <b>78.8</b>         | <b>59.2</b> | 76.1        | <b>71.4</b> | <b>68.2</b> |
| <i>PP-LiDAR (LiDAR)</i>        | $x, y, z, intensity$   | 75.6                  | 55.1        | 55.4        | 62.1        | 49.4        | 90.8                | 71.4        | 82.5        | 81.6        | 70.3        |

Table 6.4: Results for all tested methods on the entire annotated area and within the “Driving Corridor” only. Top: Ablation study of radar features. Middle: study of temporal information. Bottom: LiDAR based detector. Bold face highlights best radar results per section. All class-specific columns involve AP calculated with a 3D IoU (0.5 for car, 0.25 for pedestrian/cyclist).

splits such that the number of annotations (both static and moving) of the three main classes (cars, pedestrians, and cyclists) are proportionally distributed among the splits.

I use two performance measures following the KITTI benchmark [88]: Average Precision (AP) and Average Orientation Similarity (AOS). For AP, I calculate the intersection over union (IoU) of the predicted and ground truth bounding boxes in 3D, and require a 50% overlap for *car*, and 25% overlap for *pedestrian* and *cyclist* classes as in [88]. Mean AP (mAP) and mean AOS (mAOS) are calculated by averaging class-wise results. I report results for two regions: 1) the entire annotated region (camera FoV up to 50 meters) and 2) a more safety-relevant region called “Driving Corridor”, defined as a rectangle on the ground plane in front of the ego-vehicle as  $[-4\text{ m} < x < +4\text{ m}, z < 25\text{ m}]$  in camera coordinates.

In the experiments, I will refer to several sensor data and feature combinations: *PP-LiDAR* is PointPillars trained on LiDAR data, with the 4 typically used input features: spatial coordinates and intensity. This method will serve as a baseline for my radar-LiDAR comparison experiment. *PP-radar* is also a PointPillars network, but trained on 3+1D radar data with all 5 features, using spatial coordinates, reflectivity, and Doppler. In contrast, *PP-radar (no X)* has the feature  $X$  removed and is trained only with 4 features. Finally, *PP-radar (N scans)* is a *PP-radar* using  $N$  accumulated radar scans as described in Subsection 6.2.2. The implementation is built on OpenPCDet [128]. All networks are trained in a multi-class fashion.

#### 6.4.1 ABLATION STUDY: *PP-radar*

See Table 6.4 for the performances of the various PointPillars networks in the ablation study, for the entire coverage area and within the “Driving Corridor” region. The results show that removing the Doppler information (*PP-radar (no Doppler)*) significantly degrades performance for the two VRU classes (pedestrian: 34.9 vs. 21.3, cyclist 43.1 vs. 30.4 for the entire annotated area). Furthermore, it hampers the orientation estimation overall (mAOS: 30.5 vs. 22.1). The results also show that removing either elevation information or *RCS* (i.e. *PP-radar (no elevation)* or *PP-radar (no RCS)*) hurts the performance (mAP: 38.0 vs. 31.9 vs. 36.6 for the entire annotated area). Finally, I examined whether including radar targets from previous scans to provide temporal information makes a significant difference. I trained and evaluated two additional networks using points from the last three and five scans, respectively, to create *PP-radar (3 scans)* and *PP-radar (5 scans)*. Adding further scans increased the overall performance (mAP: 38.0 vs. 47.0 for single/five scans) and improved

orientation estimation (mAOS: 30.5 vs. 39.6 for single/five scans).

Examples of correct and incorrect detections by *PP-radar* are shown in Figure 6.7 and 6.8 for all road user classes.

### 6.4.2 PERFORMANCE COMPARISON: *PP-radar* VS. *PP-LiDAR*

I subsequently compare the object detection performance of *PP-radar* and *PP-LiDAR*, see Table 6.4. *PP-LiDAR* outperformed *PP-radar* in all three classes by a clear margin (mAP: 62.1 vs. 38.0). The relative performance gap decreases when I consider only the “Driving Corridor” region (mAP: 81.6 vs. 63.0). Figure 6.5 provides performance as a function of distance. See next section for an interpretation of these results. Figure 6.6 shows performance as a function of required IoU overlap. An interesting trend that can be seen is that the performance of radar drops off earlier than LiDAR at higher IoU thresholds. This suggests that radar correctly detects and classifies many objects but has difficulty determining their exact 3D position, which hampers overall performance.

On average, *PP-radar* inference took 40% less time than *PP-LiDAR* inference (7.8 ms vs. 12.9 ms on average measuring only the feed-forward step).



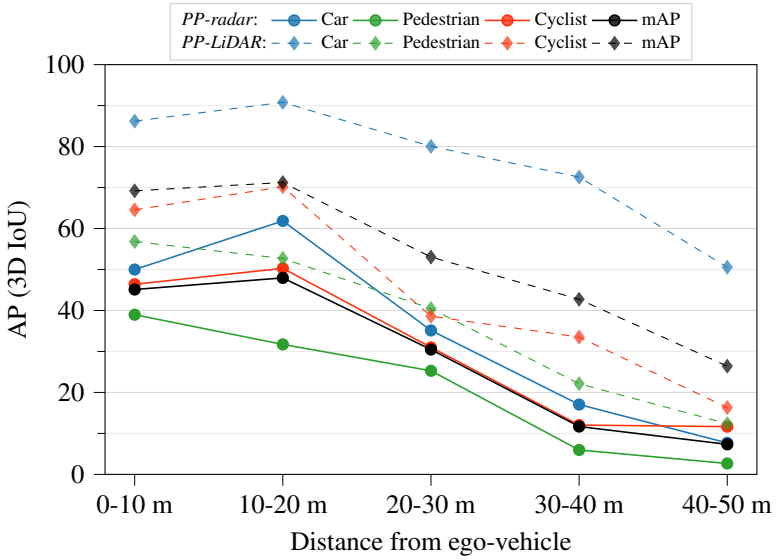


Figure 6.5: Performance of *PP-LiDAR* (dashed, diamond) and *PP-radar* (solid, circles) over distance for each class (3D IoU=0.5 for car, IoU=0.25 for pedestrian/cyclist).

6

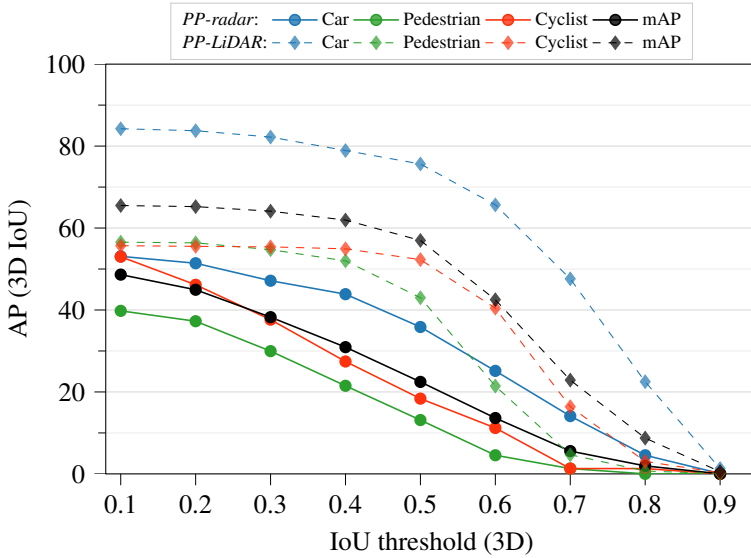


Figure 6.6: Performance of *PP-LiDAR* (dashed, diamond) and *PP-radar* (solid, circles) with different 3D IoU thresholds.



Figure 6.7: Examples of correctly detected objects by *PP-radar* projected to the image plane. Car/pedestrian/cyclist detections are shown as blue/green/red bounding boxes. Dots are radar targets colored by distance from the sensor.

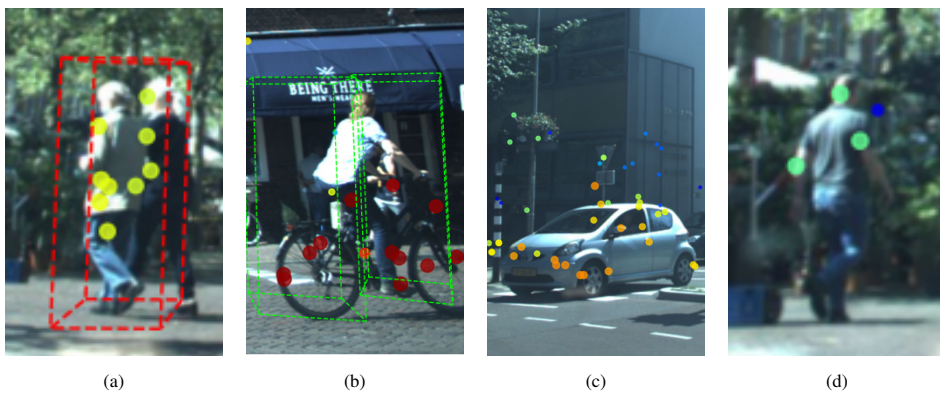


Figure 6.8: Examples of incorrect detections by *PP-radar*: (a) merged smaller objects (two pedestrians are detected as a single cyclist), (b) larger objects split into smaller ones (one cyclist is detected as two pedestrians), (c) strong reflections and clutter nearby (metal poles and high curbs) and (d) distant objects with too few reflections (far away pedestrian).

## 6.5 DISCUSSION

**I**n general, object detection performance will be determined by multiple factors: the number of 3D points lying on a particular object of the target class, their individual positional accuracy, their spatial configuration and additional attributes (e.g. velocity), their saliency vs. objects of the non-target class, and lastly, the size of the training set.

All radar based methods using Doppler performed best on the *cyclist* class. In contrast to pedestrians, and especially cars, the vast majority of cyclists in the dataset are moving, see Table 6.3. The circular motion of the wheels and pedalling, plus the highly reflective metal frame near the center causes a clear and distinctive reflection pattern that radar can more reliably detect. On the *car* class the radar methods performed more poorly relative to the large size of these objects. This can be explained by the few moving cars in the dataset, and by the fact that many are parked on the other side of the road or canal at larger distances (see Figure 6.4), and thus have few reflections. Figure 6.5 confirms that nearby cars are detected better. When focusing on just the safety-critical “Driving Corridor” region in front of the vehicle, radar performs considerably better for all classes, see Table 6.4. This performance is more relevant for driver assistance or automated driving.

The comparison of *PP-LiDAR* and *PP-radar* showed that the former has clearly higher overall performance. This can be attributed to the much higher point density of the specific type of 64-layer LiDAR sensor used (average number of points in the annotated area: LiDAR: 21344, radar: 216). Also the high viewpoint of the LiDAR sensor, on the roof of the car, benefits object detection performance as there is less pronounced occlusion. The radar sensor comes, however, with clear advantages in terms of cost and ease of packaging.

Accumulating multiple radar scans was shown to yield substantial performance improvements. This is because of the increased point density, but presumably also because the past scans provide temporal information, which can help classification (change in Doppler signature over time is class-specific, e.g., swinging limbs). Thus using multiple scans closes the relative performance gap to LiDAR somewhat.

Compromising on object detection performance might be acceptable if, as a result of the much lower point cloud density, embedding on special hardware (with certain memory and processing limitations) becomes possible. Further improvements of radar resolution and target extraction (i.e., peak finding), and/or the availability of low-level data (e.g. radar cube [49]) could further improve object detection.

## 6.6 CONCLUSION OF THE CHAPTER

**I** performed an experimental study on multi-class road user detection (PointPillars) on 64-layer 3D LiDAR data and 3+1D radar data. In ablation studies, I showed that the addition of elevation data (as in a next-generation automotive radar) clearly increases object detection performance (from 31.9 to 38.0 mAP). Doppler information remains essential for radar based object detection as its removal would greatly degrade performance (mAP 38.0 vs. 29.1). *RCS* information helps too (mAP 38.0 vs. 36.6 if removed).

Results indicate that object detection on 64-layer LiDAR data still substantially outperforms that on 3+1D radar data, when using the same PointPillars model (mAP 62.1 vs. 38.0). However, accumulating successive radar scans closes the gap to LiDAR to some degree (mAP 62.1 vs. 47.0 for five radar scans) especially in the “Driving Corridor” (mAP 81.6

vs. 71.4 for five radar scans).

For the experimental study, I introduced the View-of-Delft (VoD) dataset, a multi-sensor dataset for multi-class 3D object detection, consisting of calibrated, synchronized, and annotated LiDAR, camera, and 3+1D radar data. It is the largest dataset containing 3+1D radar recordings, suitable to facilitate future research on radar-only, camera-only, LiDAR-only, or fusion methods for object detection and tracking.



# 7

## CONCLUSION

*There is no real ending. It's just the place where you stop the story.*

Frank Herbert

THE preceding chapters have investigated radar based detection of road users and its potential use cases in intelligent vehicles using three different types of radar data. In this chapter, I summarize the research and my main results, and finally, I conclude the thesis with suggestions for future work based on the new questions that arose from my research.

## 7.1 SUMMARY

FIRST, I present the key methodical and experimental findings of the chapters individually. Then, I discuss broader topics related to sensor fusion and new radar datasets that emerged during my research.

### DETECTION OF DARTING OUT PEDESTRIANS WITH FUSION OF CAMERA AND RADAR

In Chapter 4 a generic occlusion aware multisensor Bayesian filter was proposed to detect occluded crossing pedestrians. The proposed method adapts both the expected rate and properties of detections in different areas according to the visibility of the sensors. I applied the proposed filter to camera and radar data using a new dataset, and provided techniques to account for the unique characteristics of these sensors. The results showed that both the inclusion of radar sensor and occlusion information is beneficial for this use case, as pedestrians were detected earlier in dangerous walking scenarios. For example, the threshold of 0.5 for the estimated existence probability of a pedestrian in the scene was reached on average 0.26 seconds earlier by the proposed occlusion aware fusion than by a naive camera only detector. It was also shown in an application example of the filter that it can distinguish between dangerous and non-dangerous situations, which is necessary to avoid false alarms. In this task, too, the inclusion of the radar was proved to be beneficial.

### MULTI-CLASS ROAD USER DETECTION USING THE 3D RADAR CUBE

In Chapter 5, I went from a scene specific, single-class, fusion based object detection task to generic multi-class (i.e., pedestrians, cyclists, cars) object detection using only the radar sensor. To do this, a novel, radar based, single-frame, multi-class road user detection method was proposed. Besides using the 2+1D radar point cloud, it also exploits class information in low-level radar data by applying a specially designed neural network to a cropped block of the radar cube around each radar target and the target-level features. Its first part contains convolutions in the spatial domains (range, azimuth) to capture the Doppler distribution of the environment in a single column vector. This vector is the input for the second part, where the layers focus on obtaining correspondences in the velocity distribution by convolutions along the Doppler dimension. Finally, two fully connected layers provide scores that are passed to the ensemble voting. A clustering step was introduced to create object proposals by grouping the classified targets. It was shown in extensive experiments on a real-life dataset that the proposed method improves upon the baselines in target-wise classification by reaching an average F1 score of 0.70 (vs. 0.68 by the baseline *Schumann* [54]). I showed that the proposed method also outperforms the baselines overall in object-wise classification by yielding an average F1 score of 0.56 (vs. 0.48 by *Schumann* [54]). In addition, the importance of low-level features and ensembling was demonstrated in an ablation study.

## MULTI-CLASS ROAD USER DETECTION USING THE 3+1D RADAR POINT CLOUD

Chapter 6 addressed the same task as Chapter 5: detecting road users of multiple classes with only a radar sensor. However, instead of using low-level radar data from a conventional 2+1D radar, this time I used a different data source that is not yet widely available: a next-generation 3+1D radar. These sensors provide elevation information (i.e., height) for each point, and they also tend to output a denser point cloud than the 2+1D sensors. Thus, 3+1D radar point clouds are somewhat similar to LiDAR point clouds. Therefore, this chapter presented an experimental study on the task of multi-class road user detection on 64-layer 3D LiDAR data and 3+1D radar data. Unlike in Chapter 5, in this chapter, the sparse radar point cloud was not semantically segmented. Instead, I used the higher density and available elevation information to output the detections in the form of 3D bounding boxes. In ablation studies, it was shown that the addition of elevation data (as the main difference between conventional 2+1D radars and 3+1D next-generation automotive radars) clearly increases object detection performance from 31.9 to 38.0 mAP (mean Average Precision). Consistent with the findings of Chapter 5 using 2+1D radars, Doppler information remains essential for radar based object detection using the 3+1D radar as well, since its omission would greatly degrade performance (mAP 38.0 vs. 29.1). Also, it was shown that the Radar Cross Section (*RCS*) information helps too (mAP 38.0 vs. 36.6 if removed). The results furthermore showed that object detection on a high-end 64-layer LiDAR data still substantially outperforms that on 3+1D radar data, when using the same PointPillars model (mAP 62.1 vs. 38.0). However, accumulating successive radar scans closes the gap to LiDAR to some degree (mAP 62.1 vs. 47.0 for 5 radar scans) especially in the safety critical “Driving Corridor” (mAP 81.6 vs. 71.4 for 5 radar scans). For the experimental study, and for the benefit of the intelligent vehicles research field, Chapter 6 also introduced the View-of-Delft (VoD) dataset, a multi-sensor dataset for multi-class 3D object detection.

## ON SENSOR AND DATA FUSION

Chapter 4 introduced a high-level fusion for the specific task of darting out pedestrian detection. The method uses a stereo camera and a 2+1D radar. However, the fusion itself is generic, meaning that the addition of new sensors (i.e., camera, radar, or even LiDAR) is straightforward. The detections of the new sensors should be fed into the filter in a similar way as the detections of the already integrated sensors. In my opinion, besides the fusion of the two sensors, there is another fusion related step in this chapter: the incorporation of occlusion information. Although the occlusion model in this study was provided by the camera sensor, this was an implementation decision - the model could also be retrieved from other sensors, other vehicles, or even from smart infrastructure.

The high-level or late-fusion approach had clear advantages in this task, as it allowed for greater modularity in the pipeline. This made it possible to use off-the-shelf, pre-trained detectors, e.g., SSD [107] or Instance Stixels [110]. It also allowed the same module to be used for multiple purposes, e.g., Instance Stixels were used both as camera based pedestrian detections and as inputs to the occlusion model. Unlike an early fusion model, this modular high-level fusion model does not require large amounts of training data with all the associated sensors - a problem that currently seriously limits the field of radar research, as discussed in Chapter 6.



In Chapter 5, data from multiple sensors were combined indirectly. The presented method for detecting road users is based solely on the radar sensor. However, the dataset used to train and test the detector was automatically pre-annotated by SSD [107], a stereo camera based detector, which is a fast and inexpensive way to obtain a large amount of annotated data with reasonable quality. It can be argued that this cross-sensor supervised training is also related to the topic of sensor fusion, as the information and capabilities of multiple sensors are used to achieve better overall detection performance. Note that the annotations obtained in this way were later manually corrected to minimize the likelihood of bias being transferred from one sensor to the other. For example, the camera detector often confused bicyclists with pedestrians, especially in hard lighting. Radar, on the other hand, has the advantage of “seeing” the metal frame of the bicycle and the pedaling motion as strong discriminators. This results in enhanced detection of cyclists, see e.g., Chapter 5 and 6, which is a less frequently mentioned benefit of radar sensors and could be a key motivation for fusion systems to incorporate radars.

## ON NEW RADAR DATASETS

As discussed in Chapter 6, while many datasets contain several thousand 3D bounding box annotations for multiple classes on LiDAR and camera data [88][127], there is an obvious lack of datasets containing radar data, especially along with the other sensor modalities. This not only complicates the development of novel algorithms for radars, but also the comparison of the newly proposed methods with the existing state of the art. The lack of publicly available datasets is therefore one of the major bottlenecks in the development of radar based applications today.

A good example of this struggle is the fact that each of the previous three chapters (Chapter 4, 5, and 6) used datasets designed and recorded by our research group, as no publicly available automotive dataset contained radar data that was suitable for the given use cases. Creating such datasets is a demanding task and often a serious engineering challenge, requiring significant human and financial resources from each individual research team. Publicly available datasets would alleviate this often unnecessary burden.

As a result of this thesis, two datasets containing radar data are published. The first one focuses on the specific scenario of darting out pedestrians (see Chapter 4) and contains more than 500 relevant sequences with stereo camera, radar, LiDAR, and odometry data. Furthermore, the type of sequence (darting or staying), the height of the pedestrian, the occluding vehicle’s type (car or van), and some environmental conditions (e.g., harsh lighting, leaves on the ground, etc.) were also annotated to provide some meta information. These data were suitable for developing an early warning system for darting out pedestrians.

The second dataset published as a result of this thesis is the View-of-Delft (VoD) dataset presented in Chapter 6. This is a novel multi-sensor dataset for multi-class 3D object detection in dense urban scenarios, consisting of calibrated, synchronized, and annotated LiDAR, camera, and 3+1D radar data. It is the largest dataset to date containing 3+1D radar data, and it is suitable for future research not only on pure radar, camera, and LiDAR data, but also on sensor fusion methods. For each object of interest, its 3D location and extent with a bounding box with nine degrees of freedom were annotated, along with its activity and occlusion information. Furthermore, objects are assigned unique IDs that are consistent across frames, allowing for object tracking and prediction tasks. In addition to

the data itself, detailed documentation, a development kit, and a benchmark server were also provided. Hopefully this contribution will help promote radar and fusion based methods to improve the safety of vulnerable road users. If the need arises, extension of this dataset with further recordings is time consuming, but straightforward since the documentation, annotation instructions, and data processing pipelines are already available.

Unfortunately, the low-level data recorded and used for the research in Chapter 5 could not be shared due to confidentiality agreements with the manufacturer. This is a common reason for research groups not releasing their datasets, and again an example of how difficult it is to obtain radar data to develop new algorithms or compare them to baseline methods. However, in my view, the chapter contributed to the field of radar datasets in an indirect way by showing that automatic annotation of radar point clouds based on stereo cameras, while not ideal, is feasible and provides a quick and inexpensive way to collect large amounts of training data.

## 7.2 FUTURE WORK

**A**LTHOUGH this thesis has provided a step forward for radar based road user detection, many challenges remain for future research. I first discuss possible future work for radar based approaches using the research presented in the previous chapters as stepping stones. Then I present how I envision the future of automotive radars in sensor fusion based approaches, which is followed by a broader discussion of future work. I conclude the chapter by sharing my views on the future of 3D sensors in intelligent vehicles.

### ON RADAR BASED ROAD USER DETECTION

One way to improve the fusion filter proposed in Chapter 4 is to use its generic property, i.e., integrating additional sensors - e.g., a LiDAR or additional radars - into the proposed modular framework is straightforward. In particular, replacing the 2+1D radar used in this chapter with a 3+1D radar similar to that used in Chapter 6 could be interesting for three reasons. First, the elevation information and increased density of the radar point cloud could be used to improve the pedestrian classification step, as shown in Chapter 6. Second, the elevation information could also be used in this particular use case to filter the radar targets, leaving only those that are received from below the parked, occluding vehicle - as these targets could be the result of multipath propagation. And third, the 3+1D radar has been shown to be capable of detecting both moving and parked vehicles in Chapter 6, and as such it could contribute directly to the occlusion model. In Chapter 5, the low-level data has been shown to be beneficial for pedestrian detection, thus it too could be used in an improved pedestrian classification step for the filter. Furthermore, as discussed in Chapter 2 and 5, this lower level of radar data contains more information than the radar point cloud. That is, there may be some reflections here that are too weak to be reported as radar targets in the point cloud. However, since the reflections originating from the occluded pedestrians are often weak due to multipath propagation, the weak signals in the radar cube can be highly valuable for the use case.

For the research presented in Chapter 5, it would be interesting to see how the resolution both in the spatial and in the Doppler dimensions of this radar cube influences the performance. I expect higher classification accuracy but also higher computational load when

we increase the resolution. On a related note, the resolution and/or signal-to-noise ratio of the radar cube could be further improved by combining two or even more radar cubes from multiple radar sensors. The cubes could be interpolated or averaged as a pre-processing step, or concatenated as channels so that the network learns the merging itself. One of the distinguishing properties of the proposed approach, *RTCnet*, is that it enables users to set different, class-specific clustering parameters for the various road user classes. However, these parameters are still hand selected. Future work may include a more advanced object clustering procedure, e.g., by training a separate head of the network to encode a distance metric for a clustering algorithm, or to regress instance information as well. I expect that this could lead to better clustering and thus, better object proposals. While the approach was developed specifically to work with a single frame of radar data, temporal integration, similar to some of the experiments presented in Chapter 6, could further improve the performance and usability of the method by increasing robustness, and by exploiting class-specific changes in the Doppler pattern over time, such as those caused by arm swing or wheel rotation. Finally, the proposed method provides a semantically segmented radar point cloud, which is an ideal input for a mid- or high-level fusion system, for example, the system presented in Chapter 4. Therefore, it would be interesting to test the proposed method as part of a larger fusion system that includes other modalities, e.g., camera, LiDAR, or other radars, as mentioned above.

In Chapter 6, the LiDAR based detection still outperformed the radar based detection, However, these results need to be analyzed in terms of price and practicality of the sensors. Future work could include further comparisons of the radar with other, cheaper sensors, such as a (stereo) camera or lower resolution LiDAR sensors simulated by artificially down-sampling the 64-layer LiDAR. The chapter has experimented with the application of PointPillars to radar data, a widely used point cloud processing network that is most often applied to LiDAR data. While such an application has been shown to be useful, an interesting future research would be to determine if networks developed for LiDAR data are the best choice for radar data. In my opinion, the answer is yes, but with some adjustments. That is, researchers should pay particular attention to the Doppler channel, the ghost targets caused by multipath propagation, and non-uniform angular resolution - properties unique to radar data, see Chapter 2. The discussion of the limited data augmentation opportunities caused by the Doppler channel in Chapter 6 could be a good example of such radar specific network research for point cloud processing. As mentioned in Chapter 2, radar signals are often reflected from flat surfaces such as walls, sides of vehicles, or the road itself. This can be an advantageous property, as shown in Chapter 4. However, in the case of 3+1D radar point clouds, it also means that some points will be perceived as being below the road surface after the radar signal bounces off it. It would be interesting to see if treating these points separately yields performance gains, e.g., by estimating the road surface with one of the sensors and estimating the original position of the points by reflecting them above the ground. Finally, Chapter 5 has shown that the addition of low-level radar data to the 2+1D radar point cloud significantly improves object detection. Such low-level data is not yet available from 3+1D radars, primarily because of bandwidth issues. However, it would be interesting to see if combining the 3+1D radar point clouds with low-level data also provides such benefits. Note that the radar cube in this case would be four dimensional, not three dimensional like the one used in Chapter 5, since the elevation dimension is added. Thus, an extended version of the proposed *RTCnet* would be required. One possibility would be to use

the *RTCnet* as proposed in Chapter 5: by using the radar targets as “anchors” or proposals and cropping the (now 4D) radar cube around them for classification. Alternatively, one could use *RTCnet* to encode the radar cube in a bird’s-eye view representation and merge it with the encoded pillar-wise features (i.e., the pseudo-image generated by the feature extraction of PointPillars) using channel concatenation.

To conclude the section and extend the proposed future work beyond the research of this thesis, we can observe some reoccurring problems via the examples above. First, we have seen both in the literature and in the three research chapters that the ability of radar to measure velocity instantaneously is of great use in detecting road users. Ignoring Doppler can lead to a simpler processing pipeline and easier application of networks developed for LiDAR data on radar data, as done in e.g. [75], and it also makes data augmentation more straightforward, as discussed in Chapter 6. Nevertheless, I strongly recommend that researchers continue to use Doppler information for all radar related research. However, it is clear that more data augmentation techniques are needed that are compatible with the concept of radial velocity. Such augmentations should be an important topic for future work.

Another property that is unique to radar is the possibility of indirect observation due to multipath propagation. While such indirectly received targets can be filtered out [43], they can also be exploited, as shown in [39] or in Chapter 4. In either case, researchers in the future should be aware of this property and actively anticipate and address it in their pipeline.

In Chapter 6, it was clearly shown that the elevation information from next generation radars helps in object detection. Therefore, I expect that 2+1D radars will soon be obsolete. Such sensors should be used in future research only when absolutely necessary, for example, for financial reasons.

Lack of datasets is a common challenge in radar research, which has also strongly influenced the chapters of this thesis, see Section 7.1. In the general area of machine learning, there is a clear trend: supervised learning is increasingly seen as an approach that is not scalable in terms of funding or time. As a result, researchers are looking for ways to use unlabeled data, e.g., with self-supervised camera [129] or LiDAR [130] approaches, or with cross-sensor supervision provided either by camera for LiDAR [131] or vice versa [132]. Given that there are orders of magnitude less publicly available annotated radar data than camera or LiDAR data, such “tricks” should be even more important to radar researchers. There are already some good examples in the literature: for example, a self-supervised 3+1D radar based scene flow estimation was introduced in [133]. Cross-sensor supervision was presented in Chapter 5, where I used the stereo camera sensor to annotate both the training and test sets. The authors of [134] also address scene flow estimation using cross-sensor supervision from the camera. However, even if (pre-)training of future detectors is done in some sort of self-/cross-supervised manner, I think manually annotated, high-quality, multi-modal datasets will always be needed for comparison and evaluation. I encourage researchers to record and publish such datasets to facilitate our field, similar to the one in Chapter 6.

Finally, it has become clear in the three preceding chapters that the processed point clouds of currently available automotive radars do not provide enough information to individually solve the object detection tasks in an intelligent vehicle at a satisfactory level. In my opinion, radar researchers therefore still need to find ways to extend this available information in the near future. One possibility is to use multiple radar sensors, possibly

with different viewpoints [35]. Another possibility is to gain access to a lower level of radar data (i.e., the radar cube) that has been shown in Chapter 5 to contain valuable information for classification tasks. As mentioned earlier, such extended access would be particularly interesting future work for 3+1D radars. Last but not least, it is also possible to extend the amount of information by fusing radar with other sensor modalities, as discussed in the following subsection.

## ON SENSOR FUSION

The complementary properties of radar and camera are often exploited [62][85][135] in the literature. Nevertheless, these radar-camera fusion approaches for 3D object detection still lag behind methods that use high-end LiDAR on the nuScenes benchmark [87], possibly because the radar data in the dataset comes from low-quality 2+1D radars [26][84]. As far as I know, the only published research using 3+1D radars for radar-camera fusion is [75]. Despite the somewhat disappointing results of those using 2+1D radars and the lack of fusion approaches using 3+1D radars, I believe that fusion of cameras and next generation 3+1D radars should be an important topic of future work due to the complementary nature and cost-effectiveness of this sensor setup [41].

In Chapter 4 a late fusion of the two sensors was presented, and it was also shown to have the advantage of modularity, see Section 7.1. However, such a fusion depends heavily on the individual performance of each sensor and its pipeline, which may not yet be satisfactory in the case of radar for general use cases, see Chapter 6. Instead, I suggest that researchers experiment with early or intermediate fusion. Furthermore, because point clouds from 3+1D radars are beginning to resemble LiDAR point clouds (see Chapter 6) literature on LiDAR-camera fusion may provide inspiration. However, researchers should consider the still significant density difference between LiDARs and radars: methods that rely heavily or exclusively on LiDAR to determine the precise 3D position of objects may not be appropriate for radar.

The leading LiDAR-camera fusion approaches can be classified into three groups: “PointPainting”, “Frustum based methods”, and “Pseudo LiDAR based methods”. PointPainting methods [53][136][137] “paint” the 3D points with features extracted from the camera image, e.g., color, semantic class scores or semantic features. The mapping between the 2D features and the 3D points is typically done by projecting the points onto the image. Clearly, such methods rely heavily on LiDAR for 3D geometric/spatial cues, thus, PointPainting may not be the best approach for radar-camera fusion.

Frustum based methods use some sort of 2D image based detection to “extract” the corresponding LiDAR point subset [138][139][140]. The extracted point cloud is then used to regress a 3D bounding box. A disadvantage of this approach is that it relies heavily on camera based detections: if an object is not detected in the camera image, no downstream 3D detection steps are performed. Furthermore, this approach also assumes sufficient availability of 3D points within the extended frustum - a requirement that radars cannot yet reliably fulfill.

Finally, Pseudo LiDAR based methods [141][142] argue that detection performance is degraded for distant objects with few LiDAR points, even if these instances are visible in the camera image. Their solution is to lift image pixels into 3D space, extending the LiDAR point cloud with further points, and as such, extracting both spatial and semantic cues from the camera. The problem addressed by these methods is exactly the same as the challenge caused by the sparsity of 3+1D radar point clouds: there are not enough 3D points (LiDAR

or radar) available on certain objects of interest to detect them. Therefore, I suggest radar researchers to develop similar approaches, i.e., to fuse images with the radar point cloud by reprojecting the pixels into 3D space, thus creating a “Pseudo LiDAR” point cloud.

The potential of such a fusion approach was demonstrated in a recent MSc thesis [143] supervised by me. In a first stage, Pseudo LiDAR representation of the input monocular image is generated by a novel network, which takes as input the radar point cloud, as well as a monocular depth map and a panoptic mask predicted by pre-trained state-of-the-art networks. In a second stage, the generated Pseudo LiDAR point cloud is appended to the 3+1D radar point cloud. The resulting fused point cloud is then used to train an off-the-shelf point cloud based object detection network. The results on the View-of-Delft dataset [122] were promising: the proposed approach significantly outperformed several state-of-the-art radar-camera fusion methods (proposed fusion vs. best baseline: 53.6 mAP vs. 50.8 mAP) and yielded comparable performance to a network using LiDAR input when evaluated in the safety-critical Driving Corridor (80.5 mAP vs. 81.6 mAP).

## FUTURE WORK IN A BROADER SCOPE

It is clear that for a fully self-driving car, it is not enough to just detect road users: they also need prediction of their possible future locations. I believe that such a prediction can benefit from radar data for several reasons. As discussed in Chapter 2, radars provide velocity information (also called Doppler) that is clearly a valuable input to the prediction. Further, we have seen in Chapter 6 that Doppler is also a strong prior for orientation estimation, which in turn is also critical for estimating future positions. Finally, I see a less direct way to improve position prediction by radar: intention recognition. As mentioned in both Chapter 5 and Chapter 6, the moving parts, e.g., the swinging limbs of VRUs, have a strong influence on the Doppler dimension. Regressing the pose and motion of the limbs with radars could open new possibilities to estimate the intention of VRUs (e.g., whether they will cross or stop at the curbside) and thus further help to predict their future locations. In my opinion, these capabilities of radars are currently underutilized, and I hope that my results and the dataset presented in Chapter 6 will help to promote such research.

Although this work has focused on the use of radar for road user detection, radars in intelligent cars can also be used for various other use cases, such as free road estimation, odometry, place recognition, or localization and mapping tasks. The results of this thesis can benefit such applications as well. For example, the ghost targets discussed in Chapter 2, or the multipath propagation property exploited in Chapter 4 can influence any radar-related research. Similarly, regardless of the actual task of a point cloud processing network (e.g., detection or place recognition), the discussion of the limitations of data augmentation in Chapter 6 will be relevant.

As discussed in Chapter 2, radars are commonly used in various applications that are not related to intelligent cars, but are nevertheless often related to transportation, such as traffic monitoring, airspace surveillance, security surveillance, or navigation of small unmanned aerial or ground vehicles (UAVs and UGVs). Applying some of my results to these use cases is quite straightforward. For example, pedestrian detection using the models proposed in Chapter 5 or Chapter 6 is possible directly by radars in UGVs such as delivery robots, or by radars mounted in surveillance stations. In some other cases, with a little extra work, the ideas from this thesis can be used for entirely new applications. For example, it should be possible

to detect and/or distinguish birds and drones near airports based on the characteristic motion of their rotors and wings using the radar cube and a network similar to the ones in Chapter 5.

As someone who has spent five years working on intelligent vehicles, I think fully self-driving, L5 cars are still a long way off - definitely further than the average citizen tends to believe. Along the way, we will face unprecedented situations involving multiple disciplines, not just engineering. One area that I believe will be affected is the study of law. For example, the question of who is responsible, and to what extent, in the event of an accident involving a vehicle that partially (i.e., L2, L3) or fully (i.e., L4 or L5) drives itself is complex, and already causing roadblocks [144]. The owner of the vehicle, the “driver”/passenger, the manufacturer, the programmer, and the regulatory agencies that tested and then approved the vehicle for public road use could all be considered responsible. Another area where I think a lot of research needs to be done is (social) psychology. The acceptance of self-driving vehicles is anything but trivial and depends heavily on different backgrounds. I expect for example professional lorry or taxi drivers to organize large-scale protests - a form of resistance they have historically been good at [145]. Similarly, I expect the behavior of other road users to change once supposedly self-driving vehicles appear on the road. On the one hand, people might trust these vehicles too much and, for example, cross the street without looking. On the other hand, people, especially children, might have less respect for self-driving vehicles and may “harass” them, for example by jumping in front of them or taking advantage of their patience - a phenomenon that has already been observed several times [146]. In any case, I think we have a long way to go before we will really see truly self-driving, Level 5 cars, and the road ahead is full of unanswered questions and possible future research topics for many fields.

## 7

## THE FUTURE OF 3D SENSORS IN INTELLIGENT VEHICLES

In my opinion, currently there is no trivial trend emerging in the industry about which sensor or combination of sensors is best for driver assistance or self-driving purposes. Some traditional car manufacturers such as Toyota, Volvo, or Skoda focus on driver assistance, L2 tasks, and tend to choose the camera with the combination of radar. The radar is mostly used for collision avoidance or automatic cruise control application, instead of being exploited in object classification and detection. Others, such as Daimler or Audi, already sell cars that in theory, are L3 compliant, using all three main sensors, including a low-resolution LiDAR. This step is not without setbacks though, for example, Audi in 2020 decided to abandon its L3 plans [144], claiming that “currently there is no legal framework for Level 3 automated driving”. Interestingly, stereo cameras seem to be less popular than in the last decade - possibly caused by the additional calibration complexity. Many new players in the industry, such as Waymo, Lyft, or Cruise, have chosen to directly address the challenges of self-driving by building Level 4 robotic taxi fleets. To that end, they usually limit the area of operation, and have a detailed and often updated 3D map of that area created by LiDAR to help with navigation. This mapping approach, and the apparent superiority of LiDAR sensors in object detection [87] are key reasons why high-end LiDAR sensors are more widely used by these companies compared to traditional car manufacturers. Since the vehicles are not intended for end users but to become part of a fleet, the increased manufacturing costs and design constraints caused by the choice of LiDAR sensors are less important. Finally, Tesla recently made a controversial decision [147] to completely remove radar from its perception suites and also not use LiDAR in

its vehicles. This means that Tesla plans to achieve Level 4 and 5 using only camera sensors - a decision questioned by many in both the industry [147] and the scientific community [148].

In my opinion, radar still has a legitimate place in any modern intelligent vehicle, despite the advances of other sensors, and will continue to do so in the future. Moreover, it has been shown in the previous chapters that they can do much more than simple collision avoidance. Hence, they should be integrated into the object detection pipeline, improving detection reliability with information not available from other sensors (e.g., velocity measurements, indirect reflections) and bringing increased redundancy to such systems. On the other hand, despite significant improvements in hardware design and signal processing in recent years, it is clear that the amount of information provided by an automotive radar sensor is not yet comparable to the amount of information provided by a high-end LiDAR sensor, see Chapter 6.

Therefore, in my view, there are two paths for the 3D sensor of the future. The first option is that the density of radar point clouds continues to increase while other benefits, such as reasonable cost and robustness against adverse weather are maintained. The other option is that the manufacturing price of LiDAR sensors is significantly reduced without sacrificing too much in the quality and quantity of the data output. I expect that future LiDAR sensors will also need to provide velocity measurements to be competitive, using signal processing techniques from the radar world. In either case, the result will be a sensor that provides a dense, spatially three dimensional point cloud of the environment, and also provides relative radial velocity measurements for each point. This means that most of the findings and ideas from the previous chapters will still be applicable. I am convinced that these sensors of the future will help improve road safety, and I hope that this thesis will contribute to that end.





---

## ACKNOWLEDGMENTS

This thesis, the involved papers, and any related research, result or experiment could not have been done without the support of people (sometimes literally, sometimes figuratively) around me.

First and foremost, I would like to express my appreciation and gratefulness to my advisor and promotor, prof. dr. Dariu M. Gavrila. Dariu, you are the most rigorous reviewer I have ever met - and I will always be glad to have you on our side in rebuttals. I have learned so much from you in so many areas (research, presentation skills, leadership, or skiing) that I cannot list them all here, but I will certainly never forget them.

Next, I wish to thank my copromotor, dr. Julian F.P. Kooij. Julian, you could always open one of my paper drafts or a script and almost immediately point out an error (or errors). I have always envied this and hope that I have learned a fraction of your skills in these five years.

I further want to thank the committee members: Prof. dr. Csaba Benedek, Prof. dr. Klaus Dietmayer, Prof. dr. ir. Marcel J.T. Reinders, Prof. dr. Alexander Yarovoy, and Dr. Holger Caesar for their time and feedback on the last part of this journey.

Next, I would like to mention Ewoud Pool, Thomas Hehn, Joris Domhof, Tugrul Irmak, Yanggu Zheng, Jork Stapel, and Zimin Xia. You have been with me almost the whole time and I think we have formed an excellent team with long lasting friendships. Thank you for all the fun we had together, I will cherish every moment.

Dear Alberto Bertipaglia, Vishrut Jain, Xiaolin He, Wilbert Tabone, Mubariz Zaffar, Hidde Boekema, Jetze Schuurmans, Marko Cvetković, Shiming Wang, and Varun Kotian, we had less time together. Nevertheless, I have several fond memories shared with each of you, and you still found ways to help me in many cases. I am grateful for that and hope to get to know you even better in the future by visiting some more food trucks or pubs together.

Markus Roth, Markus Braun, Christoph Rist, Christian Münch, Sebastian Krebs, and dr. Fabian Flohr, I feel like we did not have the chance to meet enough in real life during these years, but the few occasions, including epic ski trips, were always a pleasure. Also, I knew I could always turn to you for your perception knowledge, and for that I am grateful.

Dear Dr. Laura Ferranti, Oscar de Groot, Bruno Brito, Dr. Barys Shyrokau, Prof. dr. Riender Happee, thank you for the shared trips, lunches, coffees and memories.

Many thanks to Frank Everdij, Ronald Ensing, and Mario Garzon Oviedo, who probably saved me months of engineering work several times, whether it was taking apart the Prius, putting together the ROS stack, or just teaching me some Linux-based magic that I probably still do not understand. Thank you for your help and lessons. I also want to thank Ehsan Hoseini his help with all the electrical engineering problems I faced during the integration of our new radar sensor.

I would like to thank my three MSc students, Jiaao Dong, Srimannarayana Baratam, and Balázs Szekeres for their hard work towards our shared publications. I am proud to be your daily supervisor and friend, and hope to keep in touch in one way or the other.

One of my greatest take away that I learned at TU Delft is the fact that research institutes are actually run by the secretaries. I am grateful to Karin van Tongeren and Noortje Fousert for their help in countless cases. I am particularly fond of Hanneke Hustinx, who became my friend and favorite secretary of all time over the last five years.

I am very fortunate to have several friendships that, instead of disappearing, grew stronger when I moved abroad. There are too many to list them all here, but I am grateful for the many conversations, phone calls, visits, coffees and beers we have shared over these five years. Each of them has helped me move forward a little. I must mention four of them by name: Dóra Babicz, Márton Hunyady, Domonkos Huszár and Lóránt Kovács, because they accompanied me during my university studies, worked with me at my first job and not only supported me mentally to take on this challenge, but also often offered me concrete help with programming and writing tasks.

I would like to thank my parents and my siblings for their unconditional support and for everything they did for me not only during my PhD, but also until then. Mama, Papa, you already guessed that I would become a doctor when I had no idea what a doctor was. In a sense, this is also your PhD.

Last but not least, I would like to thank Ági, who accompanied me almost during the entire ride. You always supported me, regardless of the distance and obstacles, and understood that I had to do this. My marriage proposal to you is not cited as often as the results of my research proposals, but it was undoubtedly my smartest idea. I feel very fortunate to have had you by my side during my doctoral studies, and I look forward to our future together.

## REFERENCES

- [1] Daniel Holland-Letz, Matthias Kässer, Benedikt Kloss, and Thibaut Müller. Mobility's future: An investment reality check. <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mobilitys-future-an-investment-reality-check>, 2021.
- [2] Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art. *Foundations and Trends in Computer Graphics and Vision*, page 1–308, 2020.
- [3] Kareem Othman. Public acceptance and perception of autonomous vehicles: a comprehensive review. *AI and Ethics*, 1(3):355–387, 2021.
- [4] SAE International: On-Road Automated Driving committee. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. [https://doi.org/10.4271/J3016\\_202104](https://doi.org/10.4271/J3016_202104), 2021.
- [5] Rasheed Hussain, JooYoung Lee, and Sherali Zeadally. Autonomous Cars: Social and Economic Implications. *IT Professional*, 20(6):70–77, 2018.
- [6] Il Bae, Jaeyoung Moon, and Jeongseok Seo. Toward a Comfortable Driving Experience for a Self-Driving Shuttle Bus. *Electronics*, 8(9), 2019.
- [7] Alexandre Moreira Nascimento, Lucio Flavio Vismari, Caroline Bianca Santos Tancredi Molina, Paulo Sergio Cugnasca, João Batista Camargo, Jorge Rady de Almeida, Rafia Inam, Elena Fersman, Maria Valeria Marquezini, and Alberto Yukinobu Hata. A Systematic Literature Review About the Impact of Artificial Intelligence on Autonomous Vehicle Safety. *IEEE Transactions on Intelligent Transportation Systems*, 21(12):4928–4946, 2020.
- [8] World Health Organization. Global Status Report on Road Safety. <https://www.who.int/publications/i/item/9789241565684>, 2018.
- [9] National Highway Traffic Safety Administration. Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey. *Traffic Safety Facts Crash-Stats*, 2015.
- [10] Shunichi Kasahara, Jun Nishida, and Pedro Lopes. Preemptive Action: Accelerating Human Reaction Using Electrical Muscle Stimulation Without Compromising Agency. *CHI Conference on Human Factors in Computing Systems*, page 1–15, 2019.

- [11] Wilbert Tabone, Joost de Winter, Claudia Ackermann, Jonas Bärghman, Martin Baumann, Shuchisnigdha Deb, Colleen Emmenegger, Azra Habibovic, Marjan Hagenzieker, P.A. Hancock, Riender Happee, Josef Krems, John D. Lee, Marieke Martens, Natasha Merat, Don Norman, Thomas B. Sheridan, and Neville A. Stanton. Vulnerable road users and the coming wave of automated vehicles: Expert perspectives. *Transportation Research Interdisciplinary Perspectives*, 9:100293, 2021.
- [12] William A. Holm. Continuous Wave Radar. *Principles of Modern Radar*, pages 397–421, 1987.
- [13] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12689–12697, 2019.
- [14] Michael Meyer and Georg Kuschik. Automotive radar dataset for deep learning based 3D object detection. *European Radar Conference*, pages 129–132, 2019.
- [15] Sandeep Rao, Texas Instruments. Introduction to mmwave Sensing: FMCW Radars. <https://training.ti.com/node/1139153?context=1128486-1139153>, 2020.
- [16] Svante Björklund. Target Detection and Classification of Small Drones by Boosting on Radar Micro-Doppler. *European Radar Conference*, pages 182–185, 2018.
- [17] Hongbo Sun, Beom-Seok Oh, Xin Guo, and Zhiping Lin. IEEE Improving the Doppler Resolution of Ground-Based Surveillance Radar for Drone Detection. *Transactions on Aerospace and Electronic Systems*, 55(6):3667–3673, 2019.
- [18] Muhammed Emir çakıcı, Feyza Yıldırım Okay, and Suat Özdemir. Real-time aircraft tracking system: A survey and a deep learning based model. *International Symposium on Networks, Computers and Communications*, pages 1–6, 2021.
- [19] Lars M.H. Ulander, Per-Olov Frörlind, Anders Gustavsson, Rolf Ragnarsson, and Gunnar Stenström. VHF/UHF bistatic and passive SAR ground imaging. *IEEE Radar Conference*, pages 0669–0673, 2015.
- [20] Wogong Zhang, Nannan Li, Han Zha, Qi Wang, Jie Zhang, Xuan Li, Jinzhong Yu, and Erich Kasper. A Software-Adaptive 77GHz Radar Sensor for Traffic Applications. *IEEE MTT-S International Wireless Symposium*, pages 1–3, 2021.
- [21] Holger H. Meinel. Evolving automotive radar — From the very beginnings into the future. *European Conference on Antennas and Propagation*, pages 3107–3114, 2014.
- [22] Macmillan Learning. The electromagnetic spectrum. <https://sites.google.com/site/chempendix/em-spectrum/>, 2008.
- [23] Giovanni Paolo Blasone, Fabiola Colone, and Pierfrancesco Lombardo. Passive radar concept for automotive applications. *IEEE Radar Conference*, pages 1–5, 2022.

- 
- [24] Yuwei Cheng, Jingran Su, Hongyu Chen, and Yimin Liu. A New Automotive Radar 4D Point Clouds Detector by Using Deep Learning. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8398–8402, 2021.
- [25] AmirHosein Oveis, Marco Martorella, Mohammad Ali Sebt, and Ali Noroozi. Enhanced azimuth resolution in synthetic aperture radar using the music algorithm. *European Radar Conference*, pages 140–143, 2021.
- [26] Florian Engels, Philipp Heidenreich, Markus Wintermantel, Lukas Stacker, Muhammed Al Kadi, and Abdelhak M. Zoubir. Automotive Radar Signal Processing: Research Directions and Practical Challenges. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):865–878, 2021.
- [27] Dave Tahmoush and Jerry Silvius. Radar micro-doppler for long range front-view gait recognition. *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6, 2009.
- [28] Shigeaki Okumura, Takuro Sato, Takuya Sakamoto, and Toru Sato. Technique of Tracking Multiple Pedestrians Using Monostatic Ultra-wideband Doppler Radar with Adaptive Doppler Spectrum Estimation. *International Symposium on Antennas and Propagation*, pages 320–321, 2016.
- [29] Jihoon Kwon and Nojun Kwak. Human detection by neural networks using a low-cost short-range Doppler radar sensor. *IEEE Radar Conference*, pages 755–760, 2017.
- [30] Aleksandar Angelov, Andrew Robertson, Roderick Murray-Smith, and Francesco Fioranelli. Practical classification of different moving targets using automotive radar and deep neural networks. *IET Radar, Sonar & Navigation*, 12(10):1082–1089, 2018.
- [31] Eugen Schubert, Martin Kunert, Andreas Frischen, and Wolfgang Menzel. A multi-reflection-point target model for classification of pedestrians by automotive radar. *European Radar Conference*, pages 181–184, 2014.
- [32] Eugen Schubert, Frank Meinl, Martin Kunert, and Wolfgang Menzel. High resolution automotive radar measurements of vulnerable road users - pedestrians & cyclists. *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, 2015.
- [33] Jie Bai, Lianqing Zheng, Sen Li, Bin Tan, Sihan Chen, and Libo Huang. Radar transformer: An object classification network based on 4D MMW imaging radar. *Sensors*, 21(11), 2021.
- [34] Martin Stolz, Maximilian Wolf, Frank Meinl, Martin Kunert, and Wolfgang Menzel. A New Antenna Array and Signal Processing Concept for an Automotive 4D Radar. *European Radar Conference*, pages 63–66, 2018.
- [35] Kshitiz Bansal, Keshav Rungta, Siyuan Zhu, and Dinesh Bharadia. Pointillism: Accurate 3D bounding box estimation with multi-radars. *ACM Conference on Embedded Networked Sensor Systems*, pages 340–353, 2020.

- [36] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. RADIATE: A Radar Dataset for Automotive Perception in Bad Weather. *IEEE International Conference on Robotics and Automation*, pages 1–7, 2021.
- [37] Sinan Hasirlioglu and Andreas Riener. Introduction to rain and fog attenuation on automotive surround sensors. *IEEE International Conference on Intelligent Transportation Systems*, pages 1–7, 2017.
- [38] Sayanan Sivaraman and Mohan Manubhai Trivedi. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1773–1795, 2013.
- [39] Nicolas Scheiner, Florian Kraus, Fangyin Wei, Buu Phan, Fahim Mannan, Nils Appenrodt, Werner Ritter, Jürgen Dickmann, Klaus Dietmayer, Bernhard Sick, and Felix Heide. Seeing around Street Corners: Non-Line-of-Sight Detection and Tracking In-the-Wild Using Doppler Radar. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2065–2074, 2020.
- [40] Andras Palffy, Julian F. P. Kooij, and Dariu M. Gavrila. Occlusion aware sensor fusion for early crossing pedestrian detection. *IEEE Intelligent Vehicles Symposium*, pages 1768–1774, 2019.
- [41] Sean Campbell, Niall O’Mahony, Lenka Krpalcova, Daniel Riordan, Joseph Walsh, Aidan Murphy, and Conor Ryan. Sensor Technology in Autonomous Vehicles : A review. *Irish Signals and Systems Conference*, pages 1–4, 2018.
- [42] Dinh Van Nam and Kim Gon-Woo. Solid-State LiDAR based-SLAM: A Concise Review and Application. *IEEE International Conference on Big Data and Smart Computing*, pages 302–305, 2021.
- [43] Mahdi Chamseddine, Jason Rambach, Didier Stricker, and Oliver Wasenmüller. Ghost Target Detection in 3D Radar Data using Point Cloud based Deep Neural Network. *International Conference on Pattern Recognition*, pages 10398–10403, 2021.
- [44] Christoph G. Keller and Dariu M. Gavrila. Will the pedestrian cross? A study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):494–506, 2014.
- [45] Antonio Brunetti, Domenico Buongiorno, Gianpaolo Francesco Trotta, and Vitoantonio Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300:17–33, 2018.
- [46] Markus Braun, Sebastian Krebs, Fabian Flohr, and Dariu M. Gavrila. EuroCity Persons: A Novel Benchmark for Person Detection in Traffic Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1844–1861, 2019.
- [47] Karl Granström, Stephan Reuter, Maryam Fatemi, and Lennart Svensson. Pedestrian tracking using Velodyne data – stochastic optimization for extended object tracking. *IEEE Intelligent Vehicles Symposium*, pages 39–46, 2017.

- 
- [48] Steffen Heuel and Hermann Rohling. Pedestrian recognition in automotive radar sensors. *International Radar Symposium*, pages 732–739, 2013.
- [49] Andras Palffy, Jiaao Dong, Julian F. P. Kooij, and Dariu M. Gavrilă. CNN Based Road User Detection Using the 3D Radar Cube. *IEEE Robotics and Automation Letters*, 5(2):1263–1270, 2020.
- [50] Ole Schumann, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Semantic Segmentation on Radar Point Clouds. *International Conference on Information Fusion*, pages 2179–2186, 2018.
- [51] Roman Streubel and Bin Yang. Fusion of Stereo Camera and MIMO-FMCW Radar for Pedestrian Tracking in Indoor Environments. *International Conference on Information Fusion*, pages 565–572, 2016.
- [52] Joel Schlosser, Christopher K. Chow, and Zsolt Kira. Fusing LIDAR and images for pedestrian detection using convolutional neural networks. *IEEE International Conference on Robotics and Automation*, pages 2198–2205, 2016.
- [53] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. PointPainting: Sequential Fusion for 3D Object Detection. *Conference on Computer Vision and Pattern Recognition*, pages 4603–4611, 2020.
- [54] Ole Schumann, Christian Wöhler, Markus Hahn, and Jürgen Dickmann. Comparison of random forest and long short-term memory network performances in classification tasks using radar. *Sensor Data Fusion: Trends, Solutions, Applications*, pages 1–6, 2017.
- [55] Robert Prophet, Marcel Hoffmann, Martin Vossiek, Christian Sturm, Alicja Ossowska, Waqas Malik, and Urs Lübbert. Pedestrian Classification with a 79 GHz Automotive Radar Sensor. *International Radar Symposium*, pages 1–6, 2018.
- [56] Rodrigo Pérez, Falk Schubert, Ralph Rasshofer, and Erwin Biebl. Single-frame vulnerable road users classification with a 77GHz FMCW radar sensor and a convolutional neural network. *International Radar Symposium*, pages 1–10, 2018.
- [57] Andreas Danzer, Thomas Griebel, Martin Bach, and Klaus Dietmayer. 2D Car Detection in Radar Data with PointNets. *IEEE Conference on Intelligent Transportation Systems*, pages 61–66, 2019.
- [58] Christopher Diehl, Eduard Feicho, Alexander Schwambach, Thomas Dammeyer, Eric Mares, and Torsten Bertram. Radar-based Dynamic Occupancy Grid Mapping and Object Detection. *IEEE International Conference on Intelligent Transportation Systems*, pages 1–6, 2020.
- [59] Vijay John and Seiichi Mita. RVNet: Deep Sensor Fusion of Monocular Camera and Radar for Image-Based Obstacle Detection in Challenging Environments. *Lecture Notes in Computer Science*, 11854:351–364, 2019.



- [60] V. John, M. K. Nithilan, S. Mita, H. Tehrani, R. S. Sudheesh, and P. P. Lalu. SO-Net: Joint Semantic Segmentation and Obstacle Detection Using Deep Fusion of Monocular Camera and Radar. *Lecture Notes in Computer Science*, 11994:138–148, 2020.
- [61] Leichen Wang, Tianbai Chen, Carsten Anklam, and Bastian Goldluecke. High Dimensional Frustum PointNet for 3D Object Detection from Camera, LiDAR, and Radar. *IEEE Intelligent Vehicles Symposium*, pages 1621–1628, 2020.
- [62] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. A deep learning-based radar and camera sensor fusion architecture for object detection. *Sensor Data Fusion: Trends, Solutions, Applications*, pages 1–7, 2019.
- [63] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *International Conference on Knowledge Discovery and Data Mining*, page 226–231, 1996.
- [64] Ole Schumann, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Supervised Clustering for Radar Applications: On the Way to Radar Instance Segmentation. *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, pages 1–4, 2018.
- [65] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. A multi-stage clustering framework for automotive radar data. *IEEE Intelligent Transportation Systems Conference*, pages 2060–2067, 2019.
- [66] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Radar-based Feature Design and Multiclass Classification for Road User Recognition. *IEEE Intelligent Vehicles Symposium*, pages 779–786, 2018.
- [67] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *International Conference on Neural Information Processing Systems*, page 5105–5114, 2017.
- [68] Felix Nobis, Felix Fent, Johannes Betz, and Markus Lienkamp. Kernel point convolution LSTM networks for radar point cloud segmentation. *Applied Sciences*, 11(6), 2021.
- [69] Alessandro Cennamo, Florian Kaestner, and Anton Kummert. A Neural Network Based System for Efficient Semantic Segmentation of Radar Point Clouds. *Neural Processing Letters*, 53(5):3217–3235, 2021.
- [70] Ole Schumann, Jakob Lombacher, Markus Hahn, Christian Wöhler, and Jürgen Dickmann. Scene Understanding with Automotive Radar. *IEEE Transactions on Intelligent Vehicles*, 5(2):188–203, 2020.
- [71] Robert Prophet, Marcel Hoffmann, Alicja Ossowska, Waqas Malik, Christian Sturm, and Martin Vossiek. Image-Based Pedestrian Classification for 79 GHz Automotive Radar. *European Radar Conference*, pages 75–78, 2018.

- 
- [72] Yuwei Cheng, Jingran Su, Hongyu Chen, and Yimin Liu. A New Automotive Radar 4D Point Clouds Detector by Using Deep Learning. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8398–8402, 2021.
- [73] Emmett Wise, Juraj Peršić, Christopher Grebe, Ivan Petrović, and Jonathan Kelly. A continuous-time approach for 3d radar-to-camera extrinsic calibration. *IEEE International Conference on Robotics and Automation*, pages 13164–13170, 2021.
- [74] Robert Prophet, Anastasios Deligiannis, Juan-Carlos Fuentes-Michel, Ingo Weber, and Martin Vossiek. Semantic Segmentation on 3D Occupancy Grids for Automotive Radar. *IEEE Access*, 8:197917–197930, 2020.
- [75] Michael Meyer and Georg Kusch. Deep learning based 3D object detection for automotive radar and camera. *European Radar Conference*, pages 133–136, 2019.
- [76] Jakob Lombacher, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Potential of radar for static object classification using deep learning methods. *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, pages 1–4, 2016.
- [77] Jakob Lombacher, Kilian Laudt, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Semantic radar grids. *IEEE Intelligent Vehicles Symposium*, pages 1170–1175, 2017.
- [78] Kanil Patel, Kilian Rambach, Tristan Visentin, Daniel Rusev, Michael Pfeiffer, and Bin Yang. Deep Learning-based Object Classification on Automotive Radar Spectra. *IEEE Radar Conference*, pages 1–6, 2019.
- [79] Dominik Kellner, Michael Barjenbruch, Klaus Dietmayer, Jens Klappstein, and Jürgen Dickmann. Instantaneous lateral velocity estimation of a vehicle using Doppler radar. *International Conference on Information Fusion*, pages 877–884, 2013.
- [80] Patrick Held, Dagmar Steinhauser, Alexander Kamann, Andreas Koch, Thomas Brandmeier, and Ulrich T. Schwarz. Normalization of micro-doppler spectra for cyclists using high-resolution projection technique. *IEEE International Conference on Vehicular Electronics and Safety*, pages 1–6, 2019.
- [81] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. MulRan: Multimodal Range Dataset for Urban Place Recognition. *IEEE International Conference on Robotics and Automation*, pages 6246–6253, 2020.
- [82] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford radar RobotCar dataset: A radar extension to the Oxford RobotCar dataset. *IEEE International Conference on Robotics and Automation*, pages 6433–6438, 2019.
- [83] Arthur Ouaknine, Alasdair Newson, Julien Rebut, Florence Tupin, and Patrick Pérez. CARRADA Dataset: Camera and Automotive Radar with Range- Angle- Doppler Annotations. *International Conference on Pattern Recognition*, pages 5068–5075, 2021.

- [84] Ole Schumann, Markus Hahn, Nicolas Scheiner, Fabio Weishaupt, Julius F. Tilly, Jürgen Dickmann, and Christian Wöhler. RadarScenes: A Real-World Radar Point Cloud Data Set for Automotive Applications. *International Conference on Information Fusion*, pages 1–8, 2021.
- [85] Yizhou Wang, Zhongyu Jiang, Yudong Li, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. RODNet: A Real-Time Radar Object Detection Network Cross-Supervised by Camera-Radar Fused Object 3D Localization. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):954–967, 2021.
- [86] Mohammadreza Mostajabi, Ching Ming Wang, Darsh Ranjan, and Gilbert Hsyu. High Resolution Radar Dataset for Semi-Supervised Learning of Dynamic Objects. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 450–457, 2020.
- [87] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11618–11628, 2020.
- [88] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [89] Chen Ning, Li Menglu, Yuan Hao, Su Xueping, and Li Yunhong. Survey of pedestrian detection with occlusion. *Complex & Intelligent Systems*, 7(1):577–587, 2021.
- [90] Markus Enzweiler, Angela Eigenstetter, Bernt Schiele, and Dariu M. Gavrilă. Multi-cue pedestrian classification with partial occlusion handling. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 990–997, 2010.
- [91] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Strong Parts for Pedestrian Detection. *IEEE International Conference on Computer Vision*, pages 1904–1912, 2015.
- [92] Chunlun Zhou and Junsong Yuan. Learning to Integrate Occlusion-Specific Detectors for Heavily Occluded Pedestrian Detection. *Asian Conference on Computer Vision*, pages 305–320, 2017.
- [93] Chunlun Zhou and Junsong Yuan. Multi-label Learning of Part Detectors for Heavily Occluded Pedestrian Detection. *IEEE International Conference on Computer Vision*, pages 3506–3515, 2017.
- [94] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen. Repulsion Loss: Detecting Pedestrians in a Crowd. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7774–7783, 2018.
- [95] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Occlusion-Aware R-CNN: Detecting Pedestrians in a Crowd. *European Conference on Computer Vision*, pages 657–674, 2018.

- 
- [96] Markus Braun, Fabian B. Flohr, Sebastian Krebs, Ulrich Kreße, and Dariu M. Gavrilă. Simple Pair Pose - Pairwise Human Pose Estimation in Dense Urban Traffic Scenes. *IEEE Intelligent Vehicles Symposium*, pages 1545–1552, 2021.
- [97] X. Wang, A. Shrivastava, and A. Gupta. A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3039–3048, 2017.
- [98] M. Heuer, A. Al-Hamadi, A. Rain, and M. M. Meinecke. Detection and tracking approach using an automotive radar to increase active pedestrian safety. *IEEE Intelligent Vehicles Symposium*, pages 890–893, 2014.
- [99] A. Bartsch, F. Fitzek, and R. H. Rasshofer. Pedestrian recognition using automotive radar sensors. *Advances in Radio Science*, 10:45–55, 2012.
- [100] Sora Hayashi, Kenshi Saho, Daiki Isobe, and Masao Masugi. Pedestrian detection in blind area and motion classification based on rush-out risk using micro-doppler radar. *Sensors*, 21(10):1–14, 2021.
- [101] Mircea Paul Muresan, Ion Giosan, and Sergiu Nedevschi. Stabilization and Validation of 3D Object Position Using Multimodal Sensor Fusion and Semantic Segmentation. *Sensors*, 20(4), 2020.
- [102] Ricardo Omar Chavez-Garcia and Olivier Aycard. Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):525–534, 2016.
- [103] Simon Chadwick, Will Maddern, and Paul Newman. Distant vehicle detection using radar and vision. *International Conference on Robotics and Automation*, pages 8311–8317, 2019.
- [104] Jian Nie, Jun Yan, Huilin Yin, Lei Ren, and Qian Meng. A Multimodality Fusion Deep Neural Network and Safety Test Strategy for Intelligent Vehicles. *IEEE Transactions on Intelligent Vehicles*, 6(2):310–322, 2021.
- [105] Stefan Hoermann, Philipp Henzler, Martin Bach, and Klaus Dietmayer. Object Detection on Dynamic Occupancy Grid Maps Using Deep Learning and Automatic Label Generation. *IEEE Intelligent Vehicles Symposium*, pages 826–833, 2018.
- [106] Dominik Nuss, Ting Yuan, Gunther Krehl, Manuel Stübler, Stephan Reuter, and Klaus Dietmayer. Fusion of laser and radar sensor data with a sequential Monte Carlo Bayesian occupancy filter. *IEEE Intelligent Vehicles Symposium*, pages 1074–1081, 2015.
- [107] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. *European Conference on Computer Vision*, pages 21–37, 2016.

- [108] Hernán Badino, Uwe Franke, and David Pfeiffer. The Stixel World - A Compact Medium Level Representation of the 3D-World. *DAGM German Conference on Pattern Recognition*, pages 51–60, 2009.
- [109] Lukas Schneider, Marius Cordts, Timo Rehfeld, David Pfeiffer, Markus Enzweiler, Uwe Franke, Marc Pollefeys, and Stefan Roth. Semantic Stixels: Depth is not enough. *IEEE Intelligent Vehicles Symposium*, pages 110–117, 2016.
- [110] Thomas Hehn, Julian Kooij, and Dariu Gavrilă. Fast and Compact Image Segmentation Using Instance Stixels. *IEEE Transactions on Intelligent Vehicles*, 7(1):45–56, 2022.
- [111] Julian Francisco Pieter Kooij, Nicolas Schneider, Fabian Flohr, and Dariu M. Gavrilă. Context-Based Pedestrian Path Prediction. *European Conference on Computer Vision*, pages 618–633, 2014.
- [112] Stefan Munder, Christoph Schnörr, and Dariu M. Gavrilă. Pedestrian Detection and Tracking Using a Mixture of View-Based Shape-Texture Models. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):333–343, 2008.
- [113] António Almeida, Jorge Almeida, and Rui Araújo. Real-Time Tracking of Moving Objects Using Particle Filters. *IEEE International Symposium on Industrial Electronics*, pages 1327–1332, 2005.
- [114] Viktor Edman, Maria Andersson, Karl Granström, and Fredrik Gustafsson. Pedestrian group tracking using the gm-phd filter. *European Signal Processing Conference*, pages 1–5, 2013.
- [115] Zvonko Radosavljević, Darko Mušicki, Branko Kovačević, Woo Chan Kim, and Taek Lyul Song. Integrated particle filter for target tracking in clutter. *IET Radar, Sonar & Navigation*, 9(8):1063–1069, 2015.
- [116] Bettina Bartels and Henrik Liers. Bewegungsverhalten von Fußgängern im Straßenverkehr. *FAT-Schriftenreihe*, 268(2), 2014.
- [117] European New Car Assessment Programme. Test protocol - AEB VRU systems. <https://cdn.euroncap.com/media/58226/euro-ncap-aeb-vru-test-protocol-v303.pdf>, 2020.
- [118] Rini Sherony and Chong Zhang. Pedestrian and Bicyclist Crash Scenarios in the U.S. *IEEE Conference on Intelligent Transportation Systems*, pages 1533–1538, 2015.
- [119] Tiancheng Li, Shudong Sun, Tariq Pervez Sattar, and Juan Manuel Corchado. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with Applications*, 41(8):3944–3954, 2014.
- [120] L Ferranti, B Brito, E Pool, Y Zheng, R M Ensing, R Happee, B Shyrokau, J F P Kooij, J Alonso-Mora, and D M Gavrilă. SafeVRU: A Research Platform for the Interaction of Self-Driving Vehicles with Vulnerable Road Users. *IEEE Intelligent Vehicles Symposium*, pages 1660–1666, 2019.

- 
- [121] Joris Domhof, Julian F. P. Kooij, and Dariu M. Gavrilă. A Joint Extrinsic Calibration Tool for Radar, Camera and Lidar. *IEEE Transactions on Intelligent Vehicles*, 6(3):571–582, 2021.
- [122] Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian F. P. Kooij, and Dariu M. Gavrilă. Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022.
- [123] Karl Granström, Marcus Baum, and Stephan Reuter. Extended Object Tracking: Introduction, overview, and applications. *Journal of Advances in Information Fusion*, 12(2), 2017.
- [124] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Radar-based Road User Classification and Novelty Detection with Recurrent Neural Network Ensembles. *IEEE Intelligent Vehicles Symposium*, pages 722–729, 2019.
- [125] Heiko Hirschmüller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.
- [126] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.
- [127] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurélien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2443–2451, 2020.
- [128] OpenPCDet Development Team. OpenPCDet: An open-source toolbox for 3d object detection from point clouds. <https://github.com/open-mmlab/OpenPCDet>, 2020.
- [129] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9726–9735, 2020.
- [130] Ignacio Vizzo, Benedikt Mersch, Rodrigo Marcuzzi, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Make it Dense: Self-Supervised Geometric Scan Completion of Sparse 3D LiDAR Scans in Large Outdoor Environments. *IEEE Robotics and Automation Letters*, 7(3):8534–8541, 2022.

- [131] Kyle Genova, Xiaoqi Yin, Abhijit Kundu, Caroline Pantofaru, Forrester Cole, Avneesh Sud, Brian Brewington, Brian Shucker, and Thomas Funkhouser. Learning 3D Semantic Segmentation with only 2D Image Supervision. *International Conference on 3D Vision*, pages 361–372, 2021.
- [132] Hao Tian, Yuntao Chen, Jifeng Dai, Zhaoxiang Zhang, and Xizhou Zhu. Unsupervised Object Detection with LiDAR Clues. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5958–5968, 2021.
- [133] Fangqiang Ding, Zhijun Pan, Yimin Deng, Jianning Deng, and Chris Xiaoxuan Lu. Self-Supervised Scene Flow Estimation With 4-D Automotive Radar. *IEEE Robotics and Automation Letters*, 7(3):8233–8240, 2022.
- [134] Christopher Grimm, Tai Fei, Ernst Warsitz, Ridha Farhoud, Tobias Breddermann, and Reinhold Haeb-Umbach. Warping of Radar Data Into Camera Image for Cross-Modal Supervision in Automotive Applications. *IEEE Transactions on Vehicular Technology*, pages 1–15, 2022.
- [135] Ramin Nabati and Hairong Qi. CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection. *IEEE Winter Conference on Applications of Computer Vision*, pages 1526–1535, 2021.
- [136] Liang Xie, Chao Xiang, Zhengxu Yu, Guodong Xu, Zheng Yang, Deng Cai, and Xiaofei He. PI-RCNN: An Efficient Multi-Sensor 3D Object Detector with Point-Based Attentive Cont-Conv Fusion Module. *AAAI Conference on Artificial Intelligence*, pages 12460–12467, 2020.
- [137] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Zhou Bin, and Liangjun Zhang. FusionPainting: Multimodal Fusion with Adaptive Attention for 3D Object Detection. *IEEE International Intelligent Transportation Systems Conference*, pages 3047–3054, 2021.
- [138] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.
- [139] Zhixin Wang and Kui Jia. Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection. *IEEE International Conference on Intelligent Robots and Systems*, pages 1742–1749, 2019.
- [140] Leichen Wang, Tianbai Chen, Carsten Anklam, and Bastian Goldluecke. High Dimensional Frustum PointNet for 3D Object Detection from Camera, LiDAR, and Radar. *IEEE Intelligent Vehicles Symposium*, pages 1621–1628, 2020.
- [141] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse Fuse Dense: Towards High Quality 3D Detection With Depth Completion. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5418–5427, 2022.

- 
- [142] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal Virtual Point 3D Detection. *Advances in Neural Information Processing Systems*, pages 16494–16507, 2021.
- [143] Srimannarayana Baratam. Radar-guided Monocular Depth Estimation and Point Cloud Fusion for 3D Object Detection. *TU Delft, MSc in Robotics, master thesis*, pages 1–24, 2022.
- [144] Sean Szymkowski. Audi hangs up hopes for level 3 partial automation system. <https://www.cnet.com/roadshow/news/audi-a8-level-3-automation-traffic-jam-pilot-system/>, 2020.
- [145] Thomas Hynes and Janos Marton. Today in NYC history: The taxi riots of 1934. <https://untappedcities.com/2015/02/05/today-in-nyc-history-the-taxi-riots-of-1934-start-february-5-1934/>, 2015.
- [146] Alyssa Newcomb. Humans harass and attack self-driving Waymo Cars. <https://www.nbcnews.com/tech/innovation/humans-harass-attack-self-driving-waymo-cars-n950971>, 2018.
- [147] “Musk looks isolated as rivals embrace LiDAR”. <https://europe.autonews.com/automakers/musk-looks-isolated-rivals-embrace-lidar>, Feb 2022.
- [148] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8437–8445, 2019.





# CURRICULUM VITÆ

## Andras PALFFY

AUG 17, 1991 | Born in Budapest, Hungary.

### EDUCATION

- 2017-PRESENT | PhD candidate at COGNITIVE ROBOTICS, **Delft University of Technology**  
Thesis: Radar Based Road User Detection in Intelligent Vehicles.
- 2014-2016 | MSc. in COMPUTER SCIENCE ENGINEERING, **Pazmany Peter Catholic University**, *Graduated with honours*
- 2014-2015 | MSc. in DIGITAL SIGNAL AND IMAGE PROCESSING, **Cranfield University**
- 2010-2014 | BSc. in COMPUTER SCIENCE ENGINEERING, **Pazmany Peter Catholic University**, *Graduated with honours*

### PROFESSIONAL EXPERIENCE

- 2015-2017 | Algorithm Developer at EUTECUS/VERIZON, Budapest, Hungary  
Designing computer vision algorithms for embedded systems for Advanced Driver Assistance Systems and Intelligent Lighting Systems.
- 2014-2015 | Junior Algorithm Developer at EUTECUS, Milton Keynes, UK  
Designing and implementing an object classifier trainer and tester system (Remote job).
- 2014-2014 | Junior Algorithm Developer at EUTECUS, Budapest, Hungary/Berkeley, USA  
Working on pedestrian detection algorithm for ADAS systems.
- 2013-2014 | Developer Intern at EUTECUS, Budapest, Hungary  
Development of a car detection algorithm for embedded system (topic of BSc thesis).



---

## LIST OF PUBLICATIONS

1. **Andras Palffy**, Julian F. P. Kooij, and Dariu M. Gavrilă, “Occlusion aware sensor fusion for early crossing pedestrian detection,” *IEEE Intelligent Vehicles Symposium*, pp. 1768–1774, 2019. Author contributions: Andras Palffy recorded the dataset, devised the approach, implemented the filtering algorithm, performed the experiments, and took the lead in writing and presentation. Julian F.P. Kooij contributed to writing, advised on optimizing the algorithm, and provided guidance and supervision. Dariu M. Gavrilă provided guidance and supervision.
2. **Andras Palffy**, Jiaao Dong, Julian F. P. Kooij, and Dariu M. Gavrilă, “CNN Based Road User Detection Using the 3D Radar Cube,” *IEEE Robotics and Automation Letters*, vol. 5, nr. 2, pp. 1263–1270, 2020. Also invited for presentation at the *IEEE International Conference on Robotics and Automation*, 2020. Author contributions: Andras Palffy coordinated the data recording, devised the approach, implemented the neural network architecture, performed part of the experiments, and took the lead in writing and presenting. Jiaao Dong contributed to data preparation, implemented baselines, and performed part of the experiments. Julian F.P. Kooij contributed to writing and provided guidance and supervision. Dariu M. Gavrilă provided guidance and supervision.
3. **Andras Palffy**, Ewoud Pool, Srimannarayana Baratam, Julian F. P. Kooij, and Dariu M. Gavrilă, “Multi-class Road User Detection with 3+1D Radar in the View-of-Delft Dataset,” *IEEE Robotics and Automation Letters*, vol. 7, nr. 2, pp. 4961–4968, 2022. Also invited for presentation at the *IEEE International Conference on Robotics and Automation*, 2022. Author contributions: Andras Palffy led sensor integration and data collection, coordinated the annotation, and prepared the dataset for publication. He also implemented several feature extractors for the ablation study, performed part of the experiments, and was responsible for writing. Ewoud Pool contributed to data preparation, sensor calibration and synchronization, and created part of the figures. Srimannarayana Baratam contributed to data preparation, sensor calibration and synchronization, and performed most of the experiments. Julian F.P. Kooij and Dariu M. Gavrilă contributed to writing and provided guidance and supervision.
4. **Andras Palffy**, Julian F. P. Kooij, and Dariu M. Gavrilă, “Detecting darting out pedestrians with occlusion aware sensor fusion of radar and stereo camera,” Accepted to *IEEE Transactions on Intelligent Vehicles*, 2022. Author contributions: Andras Palffy recorded the dataset, developed the approach, implemented the filtering algorithm, performed the experiments, and took the lead in writing and presenting. Julian F.P. Kooij contributed to writing and mathematical formulations and provided guidance and supervision. Dariu M. Gavrilă contributed to writing and provided guidance and supervision.

# RADAR BASED ROAD USER DETECTION IN INTELLIGENT VEHICLES

