

Pre-Trained Driving in Localized Surroundings with Semantic Radar Information and Machine Learning

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (DR.-ING.)

von der KIT-Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M. Sc. Simon Tobias Isele

Tag der mündlichen Prüfung:

Referent:

Korreferent:

02.12.2022

Prof. Dr.-Ing. J. Marius Zöllner

Prof. Dr.-Ing. Eric Sax

KURZFASSUNG

Entlang der Signalverarbeitungskette von Radar Detektionen bis zur Fahrzeugansteuerung, diskutiert diese Arbeit eine semantischen Radar Segmentierung, einen darauf aufbauenden Radar SLAM, sowie eine im Verbund realisierte autonome Parkfunktion. Die Radarsegmentierung der (statischen) Umgebung wird durch ein Radar-spezifisches neuronales Netzwerk *RadarNet* erreicht. Diese Segmentierung ermöglicht die Entwicklung des semantischen Radar Graph-SLAM *SERALOC*. Auf der Grundlage der semantischen Radar SLAM Karte wird eine beispielhafte autonome Parkfunktionalität in einem realen Versuchsträger umgesetzt. Entlang eines aufgezeichneten Referenzfades parkt die Funktion ausschließlich auf Basis der Radar Wahrnehmung mit bisher unerreichter Positioniergenauigkeit.

Im ersten Schritt wird ein Datensatz von $8.2 \cdot 10^6$ punktweise semantisch gelabelten Radarpunktwolken über eine Strecke von 2507.35 m generiert. Es sind keine vergleichbaren Datensätze dieser Annotationsebene und Radarspezifikation öffentlich verfügbar. Das überwachte Training der semantischen Segmentierung *RadarNet* erreicht 28.97% mIoU auf sechs Klassen. Außerdem wird ein automatisiertes Radar-Labeling-Framework *SeRaLF* vorgestellt, welches das Radarlabeling multimodal mittels Referenzkameras und LiDAR unterstützt.

Für die kohärente Kartierung wird ein Radarsignal-Vorfilter auf der Grundlage einer Aktivierungskarte entworfen, welcher Rauschen und andere dynamische Mehrwegreflektionen unterdrückt. Ein speziell für Radar angepasstes Graph-SLAM-Frontend mit Radar-Odometrie Kanten zwischen Teil-Karten und semantisch separater NDT Registrierung setzt die vorgefilterten semantischen Radarscans zu einer konsistenten metrischen Karte zusammen. Die Kartierungsgenauigkeit und die Datenassoziation werden somit erhöht und der erste semantische Radar Graph-SLAM für beliebige statische Umgebungen realisiert.

Integriert in ein reales Testfahrzeug, wird das Zusammenspiel der live *RadarNet* Segmentierung und des semantischen Radar Graph-SLAM anhand einer rein Radar-basierten autonomen Parkfunktionalität evaluiert. Im Durchschnitt über 42 autonome Parkmanöver ($\varnothing 3.73 \frac{km}{h}$) bei durchschnittlicher Manöverlänge von $\varnothing 172.75 m$ wird ein Median absoluter Posenfehler von 0.235 m und End-Posenfehler von 0.2443 m erreicht, der vergleichbare Radar-Lokalisierungsergebnisse um $\approx 50\%$ übertrifft. Die Kartengenauigkeit von veränderlichen, neukartierten Orten über eine Kartierungsdistanz von $\varnothing 165 m$ ergibt eine $\approx 56\%$ -ige Kartenkonsistenz bei einer Abweichung von $\varnothing 0.163 m$. Für das autonome Parken wurde ein gegebener Trajektorienplaner und Regleransatz verwendet.

ABSTRACT

This thesis covers the complete signal processing from raw radar perception to full autonomous vehicle control with special focus on radar segmentation, SLAM and autonomous parking. The semantic radar segmentation capability of the real-world (static) environment is addressed by a radar specific neural network *RadarNet*. Achieving real-time semantic radar segmentation, the output of *RadarNet* is applied to enrich a semantic radar graph-SLAM *SERALOC*. The exemplary autonomous parking functionality parks a real-world vehicle along a recorded reference path with unprecedented robust absolute pose accuracy, solely based on radar data. This work achieves a novelty of semantic radar segmentation and localization benchmark.

In the first step, a point cloud data set of $8.2 \cdot 10^6$ point-wise semantically labeled radar detections covering a length of 2507.35 m is generated. There are no comparable publicly available data set with this annotation-level and radar specifications. The supervised training of a semantic segmentation approach *RadarNet* achieves 28.97% mIoU on six classes. An automated radar labeling framework *SeRaLF* is presented, involving multi-modal label generation via reference cameras and LiDAR, to support the radar labeling.

For the coherent mapping, a radar signal pre-filter is designed on the basis of an activation map to suppresses noise and other dynamic multi-path reflections. A specially radar-adapted Graph-SLAM front-end with radar odometry edges between sub-maps and the semantically separated NDT registration assembles the pre-filtered semantic radar sensor measurements to a consistent metric map. The mapping accuracy and data association are boosted by the separated registration of semantic-radar sub-sets. Overall, the first real-time capable semantic radar graph-SLAM is realized for arbitrary static environments.

Integrated into a real-world test vehicle, the interaction of the live *RadarNet* segmentation and semantic radar graph-SLAM is evaluated, applied for a solely radar-based autonomous parking functionality. Averaging 42 autonomous parkings ($\varnothing 3.73 \frac{\text{km}}{\text{h}}$) over an average maneuver length of 172.75 m , an absolute pose error median of 0.235 m and end position error 0.2443 m is achieved, outperforming comparable radar-localization results by $\approx 50\%$. The map accuracy of changing re-visited places along $\varnothing 165 \text{ m}$ yields $\approx 56\%$ map consistency at a deviation of $\varnothing 0.163 \text{ m}$. For the autonomous parking approach, a given trajectory planner and controller approach is combined with the presented radar localization.

ACKNOWLEDGEMENTS

Foremost, I thank my supervising professor Prof. Dr.-Ing. Marius J. Zöllner for his support on this journey. Especially the constructive criticism in our discussions on the research question focused my research. The inspiration and cooperation with him, the academic group, and the PhD peers of the institute's technical cognitive systems allowed me to discuss, develop and produce the academic standards of this work. The continuous drive and motivation from this group, supported the progress of this work significantly.

In addition, I thank my co-examiner Prof. Dr.-Ing. Eric Sax for his detailed and perceptive feedback in system comprehension, supporting essentially the refinement of this thesis. Including also the examining board of Prof. Dr. Stefan Nickel and Prof. Dr. Andreas Oberweis to enable the examination.

I also gratefully thank all my colleagues at Dr. Ing. h.c. F. Porsche AG for the support during the thesis project. First, the PhD group of the chassis development department as prodigiously bright, critical and unconventional discussion partners is one of the key factors to realize this PhD thesis work. The tremendous confidence of our supervisors and executives in the future and contribution of our work was exceptionally motivating and creating the necessary scope for development. The chance to discuss specific research topics with the experts of the whole Volkswagen Group, enriched the background knowledge, enabled multiple Patent applications and inspired future academic work.

Special thanks is dedicated to Martin Boss, Dr.-Ing. Marc Muntzinger, Dr.-Ing. Sebastian Söhner, and Dr.-Ing. Dr. rer. nat. Sascha Saralajew, for their valuable practical experience, patience and belief in my work.

Particular appreciation to the support and contribution of my supervised students on this research field. Also, I thank my PhD peer Lukas Köhrer of the Forschungszentrum Informatik (FZI) for the support during the vehicle integration of the planner and controller.

Eventually, I profoundly thank Anna and my whole family for the continuous encouragement and moral support.

CONTENTS

| | |
|--|------------|
| KURZFASSUNG | i |
| ABSTRACT | iii |
| PREFACE | v |
| 1 INTRODUCTION AND MOTIVATION | 1 |
| 1.1 Motivation | 1 |
| 1.2 State of the Art Parking Systems | 2 |
| 1.3 Problem Set and Research Question | 3 |
| 1.4 Structure of the Thesis | 7 |
| 2 STATE OF THE ART | 9 |
| 2.1 Principles of Automotive FMCW-Radar | 9 |
| 2.2 Coordinate Systems and Sensor Synchronization | 16 |
| 2.3 Principles of Supervised Machine Learning | 18 |
| 2.4 Point Cloud Registration | 33 |
| 2.5 Measurement Uncertainty | 35 |
| 2.6 Graph-SLAM | 36 |
| 2.7 Experimental Vehicle Setup for Rapid Prototyping | 39 |
| 3 SEMANTIC RADAR LABELING | 43 |
| 3.1 Motivation for Static Environment Radar Labeling | 47 |
| 3.2 Evaluation and Selection of existing Approaches: Point-Wise Labeling | 49 |
| 3.3 Data set Generation and Multi-modal Labeling Automation | 50 |
| 3.3.1 Multimodal Automatization Strategy | 51 |
| 3.3.2 LiDAR Plausibility Label Generation | 52 |
| 3.3.3 Camera Plausibility Label Generation | 55 |
| 3.3.4 Spatio-Temporal Radar Detection Tracking | 59 |
| 3.3.5 Plausibility Label Selection | 62 |
| 3.3.6 LiDAR Semantic Label Generation | 66 |
| 3.3.7 Camera Semantic Label Generation | 67 |
| 3.3.8 Visual Semantic Label Fusion | 68 |
| 3.3.9 Data Preparation for Machine Learning | 69 |
| 3.4 Data Set Evaluation | 74 |

| | | |
|----------|---|------------|
| 3.4.1 | Data Set Overview | 76 |
| 3.4.2 | Scene Evaluation | 76 |
| 3.4.3 | Labeling Process Evaluation | 79 |
| 3.5 | Section Conclusion | 84 |
| 3.6 | Section Outlook | 85 |
| 4 | SEMANTIC RADAR SEGMENTATION | 87 |
| 4.1 | Motivation for Direct Deep-Learning Radar Segmentation | 88 |
| 4.2 | Evaluation and Selection of existing Approaches: Radar Segmentation | 92 |
| 4.2.1 | 2D Image Segmentation Approaches | 92 |
| 4.2.2 | 3D Point Cloud Segmentation Approaches | 95 |
| 4.2.3 | Voxel Segmentation Approaches | 97 |
| 4.2.4 | Graph-based Segmentation Approaches | 98 |
| 4.3 | Direct Point Cloud based Radar Segmentation | 100 |
| 4.3.1 | PointNet and PointNet++ | 104 |
| 4.3.2 | PolarNet | 105 |
| 4.3.3 | Cylinder3D | 107 |
| 4.3.4 | ASAP-Net | 108 |
| 4.3.5 | Architecture Transfer form LiDAR Domain | 110 |
| 4.3.6 | Results of Architecture Comparison | 113 |
| 4.4 | RadarNet: Semantic Radar Segmentation Network | 119 |
| 4.5 | RadarNet Segmentation Evaluation | 131 |
| 4.6 | Section Conclusion | 139 |
| 4.7 | Section Outlook | 140 |
| 5 | (SEMANTIC)RADAR SLAM WITH SEMANTIC INFORMATION | 141 |
| 5.1 | Motivation | 141 |
| 5.2 | Evaluation and Selection of existing Approaches: Localization and Radar SLAM | 142 |
| 5.2.1 | Localization | 142 |
| 5.2.2 | Radar Simultaneous Localization and Mapping | 146 |
| 5.3 | Semantic Radar SLAM Method | 150 |
| 5.3.1 | Accumulation | 150 |
| 5.3.2 | Pre-Filter | 151 |
| 5.3.3 | Spatio-Temporal Filter | 151 |
| 5.3.4 | Semantic NDT Radar Scan Matching | 158 |
| 5.3.5 | Graph Front-End: Graph Construction | 164 |
| 5.4 | Experiments | 176 |
| 5.4.1 | Signal Pre-Processing | 176 |
| 5.4.2 | Semantic Spatio-Temoral Filter | 176 |
| 5.4.3 | Sub-Map Assembly | 177 |
| 5.4.4 | Semantic Registration | 180 |
| 5.4.5 | Semantic Radar SLAM Evaluation | 182 |
| 5.5 | Section Conclusion | 187 |

| | | |
|----------|--|------------|
| 5.6 | Section Outlook | 188 |
| 6 | PRETRAINED AUTOMATED PARKING | 189 |
| 6.1 | Evaluation and Selection of existing Approaches: Radar-based automated and autonomous Parking Systems | 189 |
| 6.2 | System Overview | 190 |
| 6.3 | Design of Experiments | 191 |
| 6.4 | Test Scenarios | 194 |
| 6.5 | Results of automated Parking Experiments | 200 |
| 6.5.1 | Vehicle Positioning Accuracy | 201 |
| 6.5.2 | Map Reproducibility Evaluation | 206 |
| 6.6 | Section Conclusion | 216 |
| 6.7 | Section Outlook | 217 |
| 7 | CONCLUSION | 219 |
| 7.1 | Thesis Outlook | 220 |
| A | APPENDIX | 221 |
| A.1 | Alternative Segmentation Approaches | 221 |
| A.2 | Density Plots per Semantic Class | 222 |
| A.3 | Semantic Radar SLAM Map Postprocessing | 225 |
| A.4 | Trained Parking System Integration | 226 |
| A.5 | Trained Parking Initialization Drive | 227 |
| A.6 | Trained Parking Map-Matching | 229 |
| | LIST OF FIGURES | 233 |
| | LIST OF TABLES | 243 |
| | LIST OF PUBLICATIONS | 255 |
| | Conference Contributions | 255 |
| | Co-authored Publications | 255 |
| | Supervised Theses | 256 |
| | Patents | 256 |
| | BIBLIOGRAPHY | 259 |

1 INTRODUCTION AND MOTIVATION

1.1 Motivation

Digitalization and automation in the automotive industry focuses on automating driving in order to support the driver in critical or monotonous situations. The Society of Automotive Engineers (SAE) taxonomy of automated driving discriminates 5 driver assistance functionality levels starting from level 0, not-automated but full human supervision, to level 5, fully autonomous driving at no human supervision. Level 1 covers driver assistance functions to overtake a single task (e.g. braking for an ACC system). Advanced Driver Assistance Systems (ADAS) functionalities cover level 2 or 2+ (if highly automated driving (HAD) is conditionally available under human supervision). Level 3 of conditional automation allowing the system to operate automated on in special conditions, e.g. a highway pilot. Level 4 and beyond describe fully autonomous vehicle systems.

In passenger vehicle cars, the general motivation to solve the technical challenge of increasing automation as driver support towards autonomous driving is found primarily in the safety aspect:

- Decrease of vehicle casualties by automated safety warnings and actuating systems, e.g. breaking systems, collision avoidance, human mis- or late-reaction and more.
- Second, to increase driver productivity, e.g. during a commuting situation on a highway, is the secondary goal.
- Third, the core of automation yields relief to the driver to perform un-liked or non-satisfying tasks, or even challenging tasks for humans e.g. parking or traffic jam assistance.

A measure for this are automated driving functionalities, which require first a relative localization, to know where the vehicle is located at, with respect to a very precise reference map of the environment. HD-maps are recorded with high precise reference sensors and provided as offline generated maps by special map suppliers. Having a relative ego-location, the question of path planning, where to go, needs to be solved.

Exemplary for the highway pilot level 3 system, the localization is based on live sensor data, mainly Global Navigation Satellite System (GNSS) or Global Positioning Systems (GPS) sensors, potentially paired with visual camera or LiDAR perception that is registered to precise offline reference HD-maps. The areal application limitations is structural: A HD-map needs

to be available as reference, a satellite based localization is necessary, and the potential camera or LiDAR vision requires good lighting or fair weather conditions respectively.

In this thesis, a solution solely relying on processing of on-board radar sensor data is researched, enabling an HD-map-free, satellite- and camera- or LiDAR-independent autonomous driving approach. The outcome of the radar-based localization, environment segmentation coupled with the path-planning is applicable for autonomous vehicles and other applications. For the target use-case of this thesis, an automated parking functionality, similar to the automated valet parking (AVP), but applicable to arbitrary environments is researched. This sense of technical system autonomy is applied to automate a manual parking based on a single manual maneuver execution as reference example. The conceptual independence of HD-maps enables new automated parking operational domains, e.g. private property area or any other environment for the automated parking functionality and the researched radar-based principle. Additionally, a new generation of parking assistance systems is enabled, parking a car along an arbitrary reference path (e.g. for 150 m from a front-door of a house to a garage complex) to a distant parking location, compared to the current parking systems which maneuver into a beside parking gap.

1.2 State of the Art Parking Systems

Current parking assistance systems span from passive visual support (reversing camera or 360° top-view camera projection) to actively actuating systems as lateral control (park steering assistant, PSA) or full lateral-longitudinal control (park assistant, PA). The operational domain of the PSA and PA are typically started in close proximity of a bypassed parking space, assisting on the reversing maneuver into a parking space. The systems find closeby parking lots by measuring gaps between obstacles during bypassing the scene and enables the assisted maneuvering into this bypassed parking space. Typically perpendicular, parallel parking spaces or fish-bone oriented parking lot patterns are found. The PSA overtakes steering, while the driver controls the longitudinal actuation of the car. PA overtakes both steering and longitudinal actuation of the reversing park maneuver.

A new generation of mobile device connected parking assistance systems, are called remote park assists (RPA). Coupled to a smartphone, the vehicle can be started in close proximity and be maneuvered by inputs to the smartphone. Typically the most advanced field application is yet limited to park out scenarios and visual obstacle avoidance [210], while straight reversing or straight park in motion, is lately available to get a car remote assisted into or out of a narrow parking space, requiring human supervision.

A further concept of parking automation is valet parking. Fully autonomous AVP concepts are yet rarely realized. The taxonomy of AVP concepts divides approaches into two groups [3, 61], AVP *type 1* relies on independent vehicle on-board perception while AVP *type 2* concepts rely

on external infrastructure-centralized perception processing. For AVP *type 2*, the car park infrastructure is equipped with perception sensors to detect and track vehicles and obstacles by a centralized compute-platform, which technically remote controls the automated vehicle to a free parking lot. The vehicle receives actuation signals of the infrastructure and follows the externally processed path. This concept is under test in pilot-projects [73], [74]. The external planning and actuation is a major difference of an automated system, compared to an autonomous system which integrates these tasks also.

An AVP *type 1* vehicle operates fully on-board with no infrastructure dependency. For AVP *type 1*, the whole perception, planning and actuation is processed on-board the vehicle, without any dependency of infrastructure support but relying on the vehicle sensor-set. The sole infrastructure information to be shared is potentially free parking lot locations.

For classical AVP *type 1* concepts, the environment is therefore often equipped with markers as unique landmarks (visual keys or e.g. pole-shaped landmarks) and mapped with reference sensors to high-definition maps (HD-map) as precise environment reference landmark.

As differentiation to autonomous driving applications, the autonomous parking use-case shares some same requirements and complex vehicle actuation but in a comparably save environment. Since the parking speed allows safe stops all the time, not much traffic or dynamic obstacles are expected, while the application scene conditions are appreciable similar. For example a highway pilot requires besides the localization and path following actuation also further perception modules such as driveable space estimation, object detection, semantic segmentation to recognize different objects and object types in a scene, lane detection and lane change assist, cooperative behavioral planning with respect to traffic participants, a safety redundancy for emergency cases and other sorts of special situation strategies that add to the system complexity.

1.3 Problem Set and Research Question

User Story:

The target use-case of this thesis is to realize an autonomous parking functionality, following the concept of AVP *type 1* but independent of any HD reference maps, relying only on on-board generated radar perception. Instead of being limited to standardized car-park scenarios with the AVP approach, or to generally localize to a HD-reference map, the target system of this thesis extends the application to arbitrary environments.

With this new class, called trained parking assist (TPA), it is aimed to learn from a manual reference drive to automatically re-drive the same path and maneuver, applicable to operate in an arbitrary environment. As exemplary test scenarios along the system will be evaluated

in Chapter 6, Figure 1.1 illustrates the tested use-case environments. For further details on the scenarios, see Section 6.4.

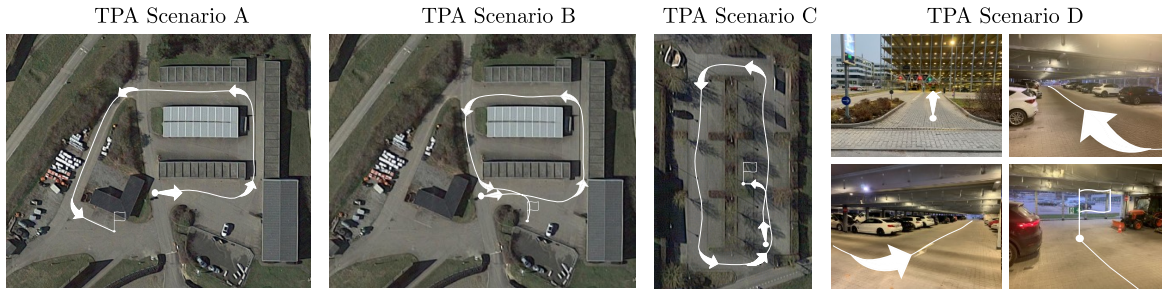


Figure 1.1: Satellite images [78, 79] of TPA test scenarios of Section 6.4 as illustration of the aimed use-case.

During an initial manual drive, called *training phase*, the system maps the environment with on-board radar sensors and saves the manual driven reference path. For the automated drive, the vehicle is manually driven to the initial starting position of the training drive, and the autonomous system performs the localization and takes over control to follow the reference path to the parking position. In order to be applicable in series cars for e.g. car parks, private underground parking or any arbitrary environment, the before introduced common assistance concepts are not applicable. Neither HD-maps nor other reference maps are available for private grounds, the system needs to be independent of lighting and weather conditions, satellite positioning is impossible for underground garages and camera localization or LiDAR sensors are denied due to volume applicability, low weather robustness and sensor cost.

The research questions focus to rely on the weather-robust and under-estimated radar sensors of a vehicle to perform semantic radar perception, radar mapping, and radar localization solely on-board.

Arbitrary environments deny HD-maps and GPS to be available and only allow a marker-free localization, due to the lack of any standard landmark types in general. Therefore the concept of semantic segmentation is chosen to be transferred to the radar domain to improve data association. Semantic segmentation of point clouds (and images), is the perception process to gain a contextual understanding and knowledge of the sensory input and segments (specific point-sets or e.g. image pixels) thereof. In this thesis, point cloud data and images are processed by semantic segmentation to leverage the sensory data to an understanding of the sensed object types and the environmental context.

Goal of the Thesis:

As overall contribution, the design of a fully autonomous parking functionality, solely based on radar, covering perception, planning and actuation, serves as proof of concept (POC) for the development of next-generation autonomous parking systems, allowing larger parking

distances compared to parking in a parking space. In the exemplary test scenarios of Section 6.4 and preempted in Figure 1.1, the developed parking system performance is evaluated, see Section 6.5 for test details.

The parking system should uninterruptedly execute the parking maneuver to test the radar-based automated driving without manual interruption, except for collision avoidance. Any manual interruption yields an invalid attempt. The path deviation is tracked to check the error along the driven path, to check the deviation progression in different scenarios, and measure especially the end pose accuracy. As basis for a further development, the deviation is required to deliver a ± 30 cm end positioning error and to avoid collisions over a longer path, see Section 6.5.1. Given the end-pose is reached automatically without collision, environment dependent deviation limits may apply along the path for further optimization.

The coupling of the problem to assemble radar sensor data to an environment map and simultaneously localize the sensor within the built map, is referred to as Simultaneous Localization and Mapping (SLAM). In this thesis, the SLAM problem is developed as radar-only mapping and guiding system, constituting the core of the parking functionality. Secondly, the radar SLAM is linked to the semantic radar segmentation on radar data to enable an accurate registration of the radar data and precise semantic environment maps.

Research Motivation:

The general field of radar-based parking assistance systems, especially the application of radar for static environment mapping is still uncommon. There exist a few works on radar base ego-motion estimation [32, 33], some works on urban localization research on automotive radar sensors for road or highway driving [151]. No specific works are covering the problem set of low-speed mapping and relocalization for parking purposes, instead only parking lot surveillance with radar sensors is known [109, 36].

From the pre-study of the presented radar mapping and data association in Section 5, the registration accuracy is found essential. Figure 1.2 shows the benefit of registration accuracy of sparse radar point clouds to accurate maps of the same environment.

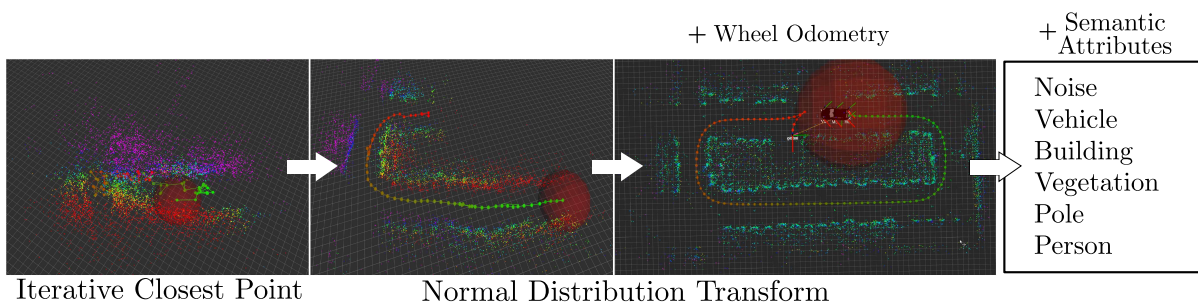


Figure 1.2: Illustration of different registration algorithms for radar point cloud association in the same environment. Radar point map colored in z-coordinate, registered poses from red to green nodes.

As novel approach on radar, the semantic radar attribute generation is motivated and researched to be applied in the registration. This data attribute yields essential registration improvements, see Section 5.3.4, allowing to build a radar parking functionality upon.

In addition, especially the focus on point cloud based approaches for the semantic radar segmentation of a multi-sensor setup is new. There exist only few works on radar and camera fusion for moving object detection [153], other works are looking into map-based segmentation of radar [226, 129]. But in general, the existence of open-source radar datasets is the main hurdle in this field to advance the research on radar, especially point-cloud based semantic segmentation. The recent radar data set RadarScenes [194] is labeled per radar point and contains vulnerable road users VRU's. As major limitation of the semantic segmentation, the RadarScenes data set labels only contain dynamic objects in several classes, no other classification e.g. of the static environment is available. Also re-labeling of the missing static labels is impossible, due to the lack of a reference sensor.

Besides, the various sensor-dependent specialties cause a lack of transferability from one radar sensor or sensor-set to another setup.

Research Questions:

The research questions of this thesis are cross-linked to the specific sections of the thesis, elaborating these questions in detail.

- **Labeling, Section 3:** How to generate point-wise semantic labels of radar point-cloud in an efficient automated, cross-sensor labeling pipeline? How is it possible to auto-generate labels for a point-wise labeled semantic segmentation data set of radar point-clouds?
- **Segmentation, Section 4:** How can the current state of the art of point-cloud processing help to solve the radar-based perception and mapping process in modern vehicles by a semantic radar segmentation? Can the deep-learning approaches on LiDAR point clouds be extended to the unknown level of sparse and noisy inputs a radar point-cloud delivers, while still yielding good semantic segmentation results?
- **Mapping, Section 5:** How can radar-based localization with multiple on-board sensors in an arbitrary static environment be solved? What mean mapping accuracy can be reached with respect to moving and occluded objects how can this mapping accuracy be measured? What mapping accuracy improvement arises in radar maps from novel direct and live semantic radar segmentation?
- **System, Section 6:** To which extent might be a radar-based localization and mapping be applicable to design and realize an autonomous parking functionality? Can the semantic radar mapping process provide an accurate scene mapping for an autonomous second passage of an automated vehicle in potentially dynamically changing situations and environments?

1.4 Structure of the Thesis

Organized in bottom-up sequence, Figure 1.3 illustrates the thesis' four essential logical consecutive main Chapters 3-6. Per chapter, the specific existing work and research of the system level is discussed, followed by an own contribution, and sectional summary.

Chapter 1: Motivation

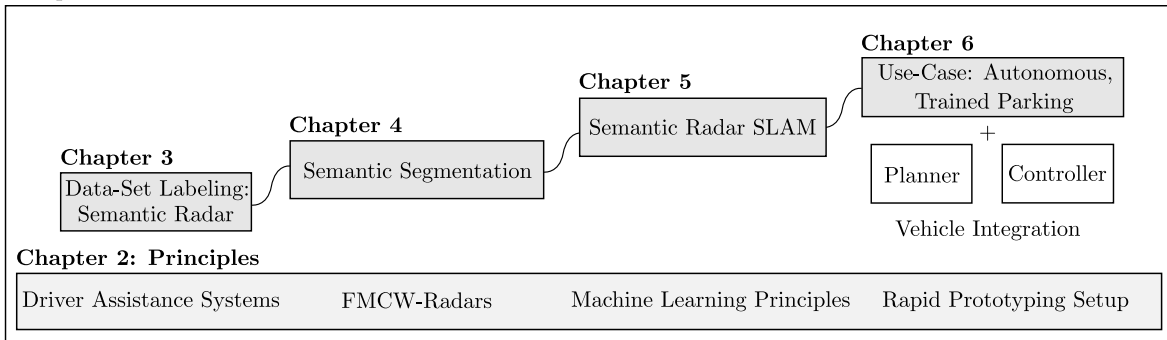


Figure 1.3: Consecutive chapter structure of the thesis.

Chapter 1 motivates the thesis, followed by **Chapter 2**, introducing the State-of-the-Art. In **Chapter 3**, a point-wise labeled radar data set is set up. Starting with an automated labeling procedure based on LiDAR and camera data, the labeling pipeline is described, resulting in a semantically labeled radar point-cloud data set. The subsequent **Chapter 4** applies the radar data-set to develop a novel semantic radar segmentation neural network. With this chapter, an artificial neural network is available to directly perform semantic segmentation on radar point clouds in real-time. **Chapter 5** develops a radar-based semantic environment mapping with on-board radar sensors and discusses different registration techniques of semantic radar data. Integrated in a real-world test vehicle, **Chapter 6** requires the essential real-time semantic segmentation with semantic radar SLAM, re-localization, trajectory planner and vehicle actuation to realize an autonomous parking functionality, solely running on the semantic radar perception. The real-world test and summary of achieved positioning accuracy closes the thesis. Finally, a general thesis conclusion and outlook is given in **Chapter 7**.

In a systematic overview, each chapters' contribution forms an essential part of the integral thesis' radar parking functionality and the realized function performance, see Figure 1.4.

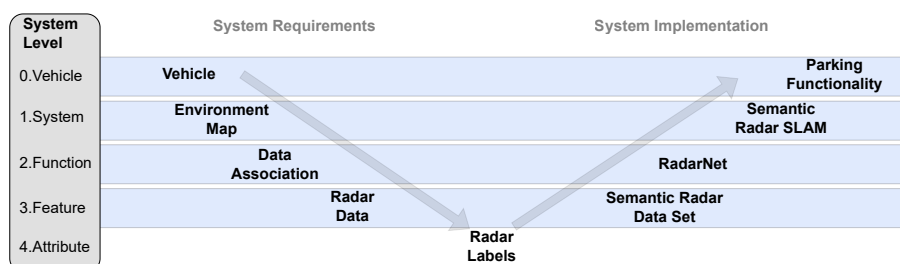


Figure 1.4: Illustration of the parking functionality structure in a system level context.

2 STATE OF THE ART

Three major topics are covered in the thesis, **radar perception**, **semantic segmentation** by means of supervised machine learning and **SLAM**, applied on radar. The theoretical principles of each topic is presented.

2.1 Principles of Automotive FMCW-Radar

Frequency Modulated Continuous Wave (FMCW) radars are the type of radar sensors commonly used in automotive applications. The focus is on 77 GHz FMCW-radars. For further details on other automotive radars, e.g. different bandwidths or other types of radars (e.g. Synthetic Aperture Radars (SAR), or Multiple Input Multiple Output (MIMO) etc.), interested readers are referred to specific works of Winner [225] and Skolnik [202].

A typical automotive radar sensor is depicted in Figure 2.1. The radar sensors are solid, robust and comparably cheap sensors. Depending on the type and application of the radar sensor, the processing of the radar echo information can deliver object detections in range and azimuth coordinates. The most significant benefit of the processing of radar signal reflections is the direct calculation of the radial relative velocity of the reflecting object [225]. With this velocity measurement, applications in the automotive domain can combine accurate position information of the detected object reflections with its relative velocity [114].



Figure 2.1: Image of an exemplary 77 GHz radar sensor, applied in the automotive context. Courtesy of Hella GmbH & Co.KGaA [114].

As a benefit of the exemplary depicted radar sensor, the sensor design allows a small packaging - which is in the automotive field an important factor. Besides the packaging advantage, the sensors can additionally be integrated in the bumpers, covered by the outer shell. Hence, the sensor setup is not visually interfering the design language of a car compared to an exposed sensor integration e.g. for cameras.

Radar as acronym describes the term *radio detection and ranging* [225], [46]. Originally applied in aviation, nautics and in the military context, the measurement principle to emit electromagnetic waves and receive a the reflected signal is adopted and transferred to multiple other tasks [202].

Ranging from people counting sensors, e.g. applied in trains or at in- and outdoor public places, to the modern application of object detection in vehicles, radar sensors are commonly met in the everyday life.

The radar principle can be described as two-stage emit-receive procedure. In the first phase, the sensor emits a short sequence of electromagnetic waves in a directed region, not uniformly. In the second phase, the sensors receives the back-scattered radiation echo signals in the sensor field of view (FoV) with a receiving antenna and processes the received echo signals on chip.

The *Radar Equation 2.1*, formulates the reflected signal power P_r depending on the range R of the sensed object, object properties and sensor properties [202, p.15]:

$$P_r = \frac{P_e G_t A_e \sigma}{(4\pi)^2 R^4} = \underbrace{P_e}_{\text{Emitted Power}} \cdot \underbrace{\frac{G_t}{(4\pi)R^2}}_{\text{Antenna Emission Rate}} \cdot \underbrace{\sigma}_{\text{RCS: Object Reflection Rate}} \cdot \underbrace{\frac{A_e}{(4\pi)R^2}}_{\text{Antenna Receiving Rate}} \quad (2.1)$$

The *Radar Equation 2.1* is grouped in four factors for an illustrative explanation. The emitted signal power P_e is quantified in the first term. As introduced before, radar beams are directed to a cone, often called *coil*, instead of emitting in a uniform sphere shape. The theoretically isotropic spread of the emitted signals over a sphere shape of radius R is formulated as denominator in the second term. But, the antenna design results in a directed beam emission, instead of a sphere-shaped emission. This beam direction design reduces the efficient denominator of the second term, formulated by the transmitting antenna gain factor G_t as nominator of the second term. The antenna design specifically influences the antenna gain factor.

The emitted radar beam is potentially reflected by an object. The reflecting objects' radar properties are described by the radar cross-section (RCS), denoted as third term σ . Depending on the object material, texture and geometrical shape, σ describes the fraction of the radar power to be back-scattered from the reflecting object with respect to the impinged intercepted radar beam.

The fourth term describes how much of the reflected signal is sensed by the radar sensor. The similar argumentation as for the emission applies. The objects' radar reflection beam is back-reflected, generally in an isotropic uniform sphere shape, therefore the same denominator appears again as for the second factor. The receiving area of the receiver antenna A_e absorbs

the reflected signals. Hence, this nominator of the fourth factor describes the fraction of the back-scattered reflection sphere.

The whole emit-receive process is repeated with high frequency, so that common automotive sensors provide a measurement rate of 15-20 Hz [225, 114].

Signal Processing to a 3D Point Cloud: The fundamental principle of FMCW radars deliver the range, azimuth and relative radial velocity of the reflecting objects. Additionally, the new generation of radar sensor applied for this work compute the elevation angle of the reflections. Hence, a full 3D representation of the radar reflections is possible, enabling the radar data representation in form of a 3D point cloud.

The applied radar sensors offer an interface to output 3D point clouds of radar detections. No specific information is available for the on-chip radar echo raw processing, the automotive sensor is manufactured by a tier-1 supplier.

The focus of this thesis is on the perception use of this radar point cloud representation. The required advanced radar reflection processing and algorithms to compute this representation are out of scope. This section introduces the working principle of FMCW signal processing, as depicted in Figure 2.2.

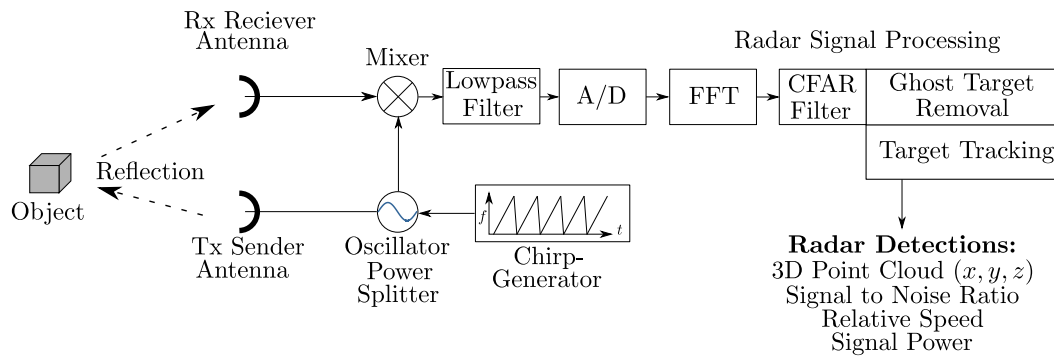


Figure 2.2: Exemplary block diagram of a typical automotive FMCW 77 GHz radar sensor.

The signal generator produces a continuous wave signal $s(t)$ of amplitude A , frequency $f(t)$ and zero phase angle ϕ_0 .

$$s(t) = A \cos(f(t) + \phi_0) \quad (2.2)$$

Without modulation, the carrier frequency f_c remains constant. Adding a linear modulation term over time, the frequency results in the depicted resulting frequency $f(t)$, see Figure 2.2 and Figure 2.3.

Although other types of modulation are researched, a linear frequency modulation

$$f(t) = f_c + \frac{B}{T_m} \cdot t, \quad (2.3)$$

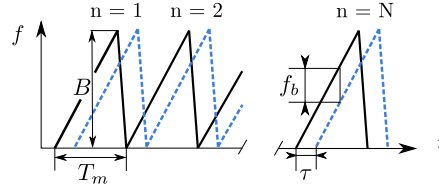


Figure 2.3: Exemplary linear modulation of an typical automotive FMCW 77 GHz radar emitted signal (black) and reflected echo (blue dotted). Illustration according to Patole et al. [160].

is common to explain the working principle. The additive term increases the base frequency linearly, to the maximum of $f_{max} = f_c + B$. The parameter B is introduced as modulation defining bandwidth, while the fraction of $\frac{t}{T_m}$ of Equation 2.3 defines the linear increasing additional modulation component.

Reformulating the modulated frequency to the instantaneous phase $\Phi(t)$ [22, 192]

$$\Phi(t) = 2\pi \int_0^t f(\tilde{t}) d\tilde{t} = 2\pi \left(f_b t + \frac{B}{2T_m} t^2 \right) + \phi_0 \quad (2.4)$$

This modulated wave is sent by the directed sending antenna [192]

$$s_{TX}(t) = A_{TX} \cos(\Phi(t)), \quad (2.5)$$

while the corresponding radar echo s_{RX} is received by the receiving antenna.

As a result of the signal travel time and meanwhile modulated frequency, the received signal s_{RX} deviates from the emitted signal in amplitude and phase [192].

$$s_{RX}(t) = A_{RX} \cos(\Phi(t - \tau)) \quad (2.6)$$

The temporal shift of τ in the received instantaneous phase $\Phi(t - \tau)$ in the received signal echo results from two effects.

$$\tau = \frac{2(R + v_r t)}{c} \quad (2.7)$$

The traveled distance $2R$ from sender antenna to the object and back to the receiver antenna causes a time delay. Plus, the potential radial velocity v_r of the reflecting object accounts additionally for a phase shift. The denominator c in Equation 2.7 represents the speed of light for radar traveling in air, generally representing the velocity of the emitted signal [192].

In the further processing steps of Figure 2.2, both emitted and received harmonic wave signals are mixed. The mixer applies a mathematical multiplication of the two slightly differing frequencies, resulting in a beat.

$$\underbrace{\cos(\Phi(t)) \cdot \cos(\Phi(t-\tau))}_{\text{frequency mixer}} = \frac{1}{2} \left(\underbrace{\cos(\Phi(t) + \Phi(t-\tau))}_{\text{high frequency: filtered out}} + \underbrace{\cos(\Phi(t) - \Phi(t-\tau))}_{\text{low frequency: beat frequency}} \right) \quad (2.8)$$

The first term of Equation 2.8, represents a high-frequency component that is filtered out in the subsequent high-pass filter. The relevant, low beat frequency f_b of Equation 2.8 is processed to a digital signal by an Analog-Digital converter.

Assuming a static object with $v_r = 0$ causing the radar reflection, Equation 2.7 simplifies to

$$\tau = \frac{2R}{c}. \quad (2.9)$$

For this case, the beat frequency f_b can be computed to a range estimate R for the radar illuminated object. The triangular similarity between the bandwidth modulation B over the modulation chirp time T_m and the beat frequency f_b over the time delay τ , can be reformulated to Schumann [192]

$$\frac{B}{T_m} = \frac{f_b}{\tau} \xrightarrow{\text{Equation 2.9}} R = \frac{f_b T_m c}{B} \frac{c}{2}. \quad (2.10)$$

For general cases, the simplification of $v_r = 0$ does not hold. Commonly, as shown in Figure 2.3, in automotive radar processing not only one chirp, but a periodic array of N_{chirp} chirps are combined to compute the positional (range) information and relative velocity. The N_{chirp} chirps are periodically repeated and form a radar *scan*. The chirps are processed altogether and constitute one radar *scan* measurement.

The N_{chirp} repetitive chirps are applied to compare the phase Φ . In subsequent chirps, stationary objects remain at the same range R , hence the corresponding phase Φ remains constant. In contrast, dynamic objects appear at a changing range coordinate R , yielding a changing phase Φ .

To identify the object reflections based on the range-frequency dependency, two subsequent Fast Fourier Transformations (FFT) are applied of the digitized mixed frequency. First, the radar echo interpretation in the (range-) FFT spectrum yields a peak per object at a certain frequency, which yields the actual range of the object. The second FFT is performed over the N_{chirp} chirps, combining the range information and the changing frequencies to compute the relative velocity, also called *Doppler-velocity*, for the moving objects.

Further mathematical details of the radar FFT analysis theory are available by the works of Suleymanov [207] and Winkler [224].

Azimuthal Resolution: Figure 2.2 illustrates only a simplified single emitting and receiving antenna, but includes the basic principle of an angular reflection of the radar signal wave front towards the receiving antenna. In detail, inside of each radar sensor a matrix-arranged pattern of multiple receiving antennas is active to compute the azimuth angle ψ . The antenna array of N_{arr} independent antennas is generally oriented parallel to the sensor surface in a equidistant spacing of d_{arr} as square array or matrix. The spacing and matrix orientation of (multiple) antenna arrays is a design parameter of the sensor. Every received radar reflection, received under an azimuth angle of $\phi \neq 90^\circ$, so off the sensor-normal, results in a measurable phase shift between the independent antennas of the sensors' antenna array

$$\Delta\Phi(\phi) = \frac{2\pi}{\lambda} d_{arr} \sin \phi. \quad (2.11)$$

Resulting from the equidistant arrangement, the phase differences between the independent receiving antennas are integral multiples of Equation 2.11. Analogous as for the range and relative velocity, a Fourier transformation is applied. The FFT covers now the N_{arr} receiver antennas to compute the azimuth angle ϕ .

To achieve an adequate azimuthal resolution for automotive applications, circa 1° angular resolution is required. The number of sensor-inbuilt antennas is often limited by the sensor ans circuit design, and consequently the accuracy is compromised. In contrast, advanced techniques are applied, such as Multiple Input Multiple Output (MIMO) radars, proposed by Li and Stoica [123]. Applying not only N_{arr} multiple receiver antennas but combining these with independent, equidistant arranged N_{em} emitter antennas, the virtual field of $N_{em} \times N_{arr}$ antennas increases the effective radar aperture A_e .

Target Detection: Depicted by the block diagram of Figure 2.2, the next processing stage includes the target extraction. Among the computed FFT of the radar echo, the radar reflections of real objects need to be filtered from clutter and amplified noise content of the signal processing. Remaining relevant radar reflections are commonly called *targets* and determine the most probable reflections of the radar echo.

For this process, the Constant False Alarm Rate (CFAR) algorithm is commonly applied [174] with sophisticated approaches to handle a variable noise level and determine significant peaks in the 3D FFT. Since the sensor processing steps are unknown, no further details of the applied radar raw signal processing can be given. Holder et al. [91] outline, that approaches such as target tracking [124], multi-scan comparison [121, 63] or high frequency estimation [55] are applied examples to mitigate shortcomings of aliasing ambiguities.

Besides the sensing ambiguities, the CFAR and target extraction can be confused by electromagnetic environment noise or signal disturbances by radar inference phenomena [9]. Especially weak reflection echos, causing a high noise floor level complicate a robust differentiation between unwanted clutter or relevant reflections. Dedicated research of Buhren and

Bin [28] on clutter occurrence and simulation yields the finding that clutter occurs as Poisson distribution. This distribution definition for clutter is later re-used in this work to model the probability of detections.

Radar Ambiguities, Artifacts and Noise:

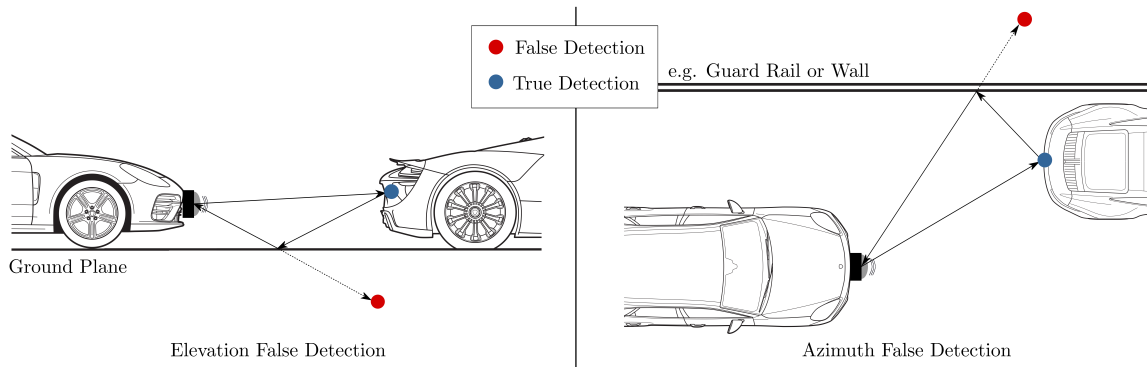


Figure 2.4: Exemplary multi-path reflection of radar detections, in horizontal view (left) and top-view (right). Illustration according to Holder et al. [91].

False radar detections are source of multiple effects, such as ambiguities, multi-path reflections and measurement clutter. Holder et al. [91] define in their work an *artifact* definition for this group of distorted and unwanted radar signals. The discrimination of real object reflections and other unwanted, radar measurement principle specific artifacts is defined as major difficulty in radar signal processing [91].

Multi-path reflections, or *mirror reflections*, result from a bouncing radar propagation, yielding ambiguities in azimuthal or elevation position and velocity measurement, while resembling to occur in the same range R .

The high frequency modulation of the emitted chirps, requires a high sampling frequency to achieve an accurate frequency resolution. Shannon [199] defines the sampling theorem, that a target frequency requires the double target frequency to be applied as sampling frequency. According to Holder et al. [91], this criteria is not met for practical sensors but aliasing effects are tolerated, causing ambiguities in angular and velocity measurements.

Similar to the formulations in Schumann [192], Holder et al. [91] outlines relative velocity ambiguities to occur in the two FFT stages. Schumann [192] derives the mathematical formulation of the errors and formulates the possible mitigations. The more chirps, the higher carrier frequency f_c or the longer the chirp duration T_m , the higher gets the Doppler velocity resolution [192].

Besides the relative velocity measurement also the angular position resolution suffers from a tight packaging of receiving antennas, resulting in a small aperture area A_e . According to Holder et al. [91], aliasing effects result in angular position ambiguities [1], depending on

the azimuthal detection angle ϕ and receiving antenna pattern [192]. Motivated from an optimal antenna spacing, Schumann [192] describes the angular resolution to depend on the physical antenna count. E.g. a double angular resolution can be achieved by a doubling of the antennas, physically or in a MIMO setup.

These ambiguities result radar reflections but represent non-plausible objects. In this thesis, these non-plausible, but measurement principle specific radar detections are interchangeably called *clutter*, *noise* or *artifacts*.

As a function of the utilized radar bandwidth, the number of FMWC chirps, the A/D sampling frequency, the antenna count and effective radar aperture, the modulation bandwidth B , the radar propagation speed, the radar detection resolution is limited [202, 225, 46, 192, 91].

As a result, the separation of closeby detections or objects suffers, weak radar reflecting objects might remain undetected, or multi-path reflections cause ghost objects that might be interpreted as obstacles. Traversable paths might become blocked by a ground reflection (e.g. a steel manhole cover) [91].

Downstream radar processing steps and applications need to take care of these phenomena.

2.2 Coordinate Systems and Sensor Synchronization

The different coordinate systems applied for sensor fusion are depicted based on their location in Figure 2.5, while Figure 2.6 depicts the hierarchical connection.

All coordinate transformations between two coordinate systems A, B are implemented as homogeneous transformation matrix $M_{A \rightarrow B} \in \mathbb{R}^{4 \times 4}$. The expression as a homogeneous matrix allows to chain coordinate transformations by a simple multiplication of homogeneous matrices and to express reverse transformations as inverse matrix $M_{B \rightarrow A} = M_{A \rightarrow B}^{-1}$. The general 4x4 isometry matrix M combines the rotation matrix $R \in \mathbb{R}^{3 \times 3}$ and translation matrix $T \in \mathbb{R}^{3 \times 1}$

$$M = \begin{pmatrix} & R & T \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.12)$$

With R as rotation matrix, summarizing the Euler angle rotation matrices in roll $R_x(\phi)$, pitch $R_y(\theta)$, yaw $R_z(\psi)$ sequence

$$R = (R_x * R_y, R_z). \quad (2.13)$$

T as 3x1 translation matrix with components T_x in x-direction, T_y in y-direction and T_z in z-direction describes linear translation components.

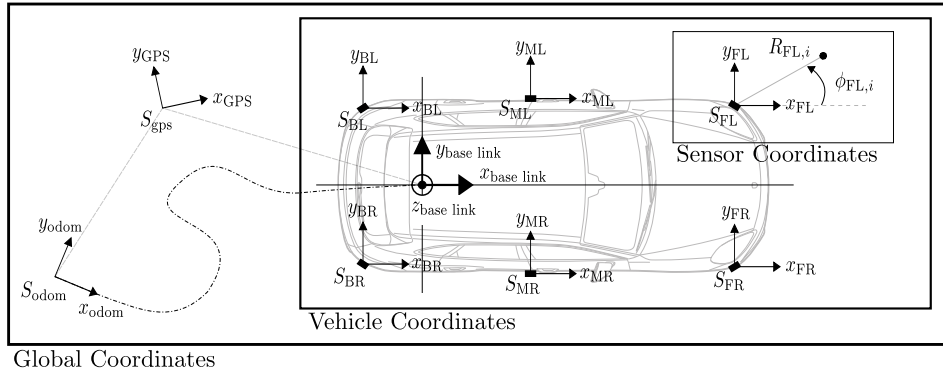


Figure 2.5: Exemplary sensor coordinate systems S_{xx}^2 of the applied radar sensor assembly with respect to the central rear-axle vehicle coordinate system $S_{\text{base link}}$. The mounting position and the resulting sensor view angle is exemplary displayed in this figure.

Each independent radar sensor \mathbf{S} , $s \in [\text{FL}, \text{FR}, \text{ML}, \text{MR}, \text{BL}, \text{BR}]$ computes radar detections i from its received echo, in relative spherical sensor coordinates $(R_{s,i}, \phi_{s,i}, \theta_{s,i})$.¹ Based on extrinsic calibration parameters, the spherical sensor coordinates are transformed into Cartesian sensor coordinates, aligned with the vehicle coordinate system $(x_{s,i}, y_{s,i}, z_{s,i})$. Applying a sensor mounting position specific translation, the radar detection coordinates of all sensors can be expressed in Cartesian vehicle coordinates, e.g. $(x_{\text{base link},s,i}, y_{\text{base link},s,i}, z_{\text{base link},s,i})$.

In accordance to ISO 8855 [102], the right-handed vehicle coordinate system is originated at the vehicles' rear axle center on ground level. Figure 2.5 illustrates the x -axis pointing in forward driving direction, y -axis to the left hand side and z -coordinate pointing upwards. This base coordinate system is assumed to remain horizontal, neglecting vehicle chassis roll or pitch motion and is applied for all later sensor fusion.

To express the vehicle motion, a space-fixed reference coordinate system is applied. S_{odom} serves as world-fixed odometry origin and origin for all SLAM maps that are processed in Chapter 5. Every vehicle motion is assumed to result in a planar motion in $x_{\text{odom}} - y_{\text{odom}}$ plane, not including lateral motion drift $\dot{y} \equiv 0$ in the considered low-speed scenarios. As a result, the rotation components reduce to a yaw-rotation component R_z .

From the reference sensors, a differential GPS position is given at every time and signal availability. Therewith, the world-fixed coordinate system S_{GPS} is defined to relate the odometry origin into global world coordinates. This relative vector is essential for the evaluation of the global positioning accuracy, in Chapter 6. The vectors $\overrightarrow{S_{\text{GPS}}, S_{\text{odom}}}$ and $\overrightarrow{S_{\text{GPS}}, S_{\text{base link}}}$ are

¹ Based on the extrinsic calibration parameters, the sensor interface delivers the azimuthal orientation of each radar detection i for all sensors directly with respect to the vehicle coordinate system.

² The subscript describes the sensor position as Back (B), Middle (M) and Front (F), plus Left (L) or Right (r) vehicle side.

applied to compare the mapped scenarios with the precise real-world position of the vehicle in differential GPS coordinates.

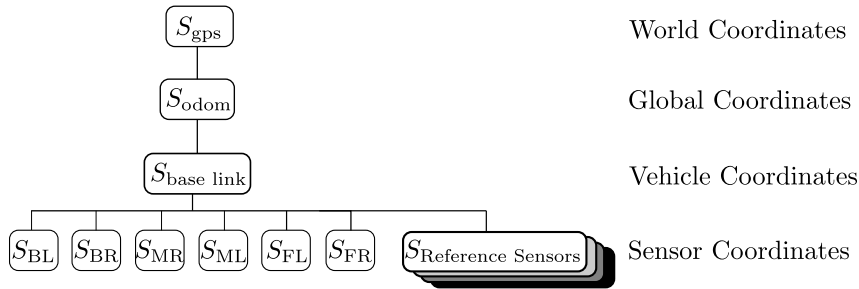


Figure 2.6: Coordinate system relation. Parental coordinates on top of child coordinate systems, in the top-down order *World* \rightarrow *Global* \rightarrow *Vehicle* \rightarrow *Sensor* as consecutive coordinate system transforms.

Sensor Synchronization: To fuse the multiple radar sensor point clouds, besides coordinate transformation, time synchronization is necessary. The $\mathbf{S} = 0..6$ individual point clouds $\mathcal{P}_{\text{radar}}$ are assembled to form a 360° point cloud, synchronized on their time stamp t_s and expressed in vehicle coordinates. Since the independent radar sensors are not commonly triggered, the non-synchronous scans are required to be expressed with respect to a common vehicle position, including a vehicle motion compensation for the sensing time gaps.

For each individual radar point cloud $\mathcal{P}_{\text{radar}}(t_s)$ at time t_s , the corresponding odometry vehicle position is synchronized $(x_{\text{base link}}, y_{\text{base link}}, z_{\text{base link}})_{\text{odom}}(t_s)$. A filter module counts and stores all six individual radar sensor point clouds with their synchronized odometry position until all six radar point clouds are available within a time range of one average frame-rate. The filter computes the ego-motion compensation for each point cloud individually and expresses the radar detections for the most recent time $t_{\text{synch}} = \max(t_s)$ and vehicle position $(x_{\text{base link}}, y_{\text{base link}}, z_{\text{base link}})_{\text{odom}}(t_{\text{synch}})$.

2.3 Principles of Supervised Machine Learning

Machine learning is generally split in three types, **supervised learning**, **unsupervised learning** and **reinforcement learning**. These three types differ along the training data set, availability of ground truth training labels \mathbf{y}_i , and the applied learning scheme to relate the input \mathbf{x}_i to an estimated output $\hat{\mathbf{y}}_i$. The learning approaches have in common, that a learn-able system is initialized, gets trained on data samples with different strategies, to be able in a prediction phase to infer an estimated output $\hat{\mathbf{y}}_i$ based on a given input \mathbf{x}_i . All three types are explained briefly, supervised learning is discussed in detail, since this method is applied.

Supervised Learning: This variant of machine learning requires a data set of $q \in \mathbb{N}$ samples of a data tuple, input data \mathbf{x}_i and corresponding ground truth output labels \mathbf{y}_i . Two

classes of problem sets are normally solved with learning approach, classification problems and regression problems.

Classification problems³ are posed, if the input data is supposed to be assigned to a discrete set \mathcal{L} of N_C individual *labels* or *classes*. N_C describes the set of c classes. The output labels \mathbf{y}_i decode in a one-hot encoding the correct class to a given input sample \mathbf{x}_i . For an exemplary binary classification case with $N_C = 2$, given a data set of q samples, the necessary data set can be formalized as

$$\mathcal{D}_{\text{cla}} = \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, \dots, q; \mathbf{x}_i \in \mathbb{R}^n; \mathbf{y}_i \in \mathcal{L}\}, \quad (2.14)$$

without a limitation to extend the formulation to a mutual exclusive multi-class classification problem with $N_C > 2$.

For the second class of regression problems⁴, the output \mathbf{y}_i describes a continuous value, e.g. of a function to approximate. The problem set formulation differs for this type of approximation problems. For an exemplary 1-dimensional output problem, the data set can be formalized to

$$\mathcal{D}_{\text{reg}} = \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, \dots, q; \mathbf{x}_i \in \mathbb{R}^n; \mathbf{y}_i \in \mathbb{R}^n\}. \quad (2.15)$$

Un-Supervised Learning: As the name suggests, the unsupervised learning⁵ method utilizes only a set of input samples \mathbf{x}_i , without any ground truth label \mathbf{y}_i information. The data set formulation simplifies to

$$\mathcal{D}_{\text{uns}} = \{\mathbf{x}_i : i = 1, \dots, q; \mathbf{x}_i \in \mathbb{R}^n\}. \quad (2.16)$$

Exemplary applications of unsupervised learning are clustering applications, which describes a search and grouping of structural similar data-patterns in given data, or dimensionality reduction. The major challenge for the unsupervised problem set are performance measures, to measure and describe the quality of the learning approach. Further information on unsupervised learning is found in Chollet [45], Goodfellow et al. [77], Ketkar [113].

Reinforcement Learning: This approach iteratively applies training and prediction as alternating steps. In abstract words, the system tries different strategies, aiming to maximize a positive feedback, formalizes as a reward. The learn-able system is called *agent*, which is trained in an simulative training environment. The agent is able to interact with its training environment, e.g. the agent executes motions or processes, which are called *actions*. With a mathematical formulation of a reward function, each action of the agent is rated. A rated action causes a reward (value) to be feed-backed to the agent. With this positive or negative

³ Subscript "cla".

⁴ Subscript "reg".

⁵ Subscript "uns".

reward, the agent adjusts or learns its parameters, to maximize the reward. This process of iteratively improve the parameter set, execute an action and adjust the parameter set based on the reward, is continuously applied to optimize the agents' strategy.

The main challenge of this approach is to formalize constraints of the explored parameter space and formalize the systems' target behaviour as reward function. Further information on reinforcement learning is found in Chollet [45], Goodfellow et al. [77], Ketkar [113].

Goals of Supervised Learning: Based on training data tuples a general applicable mapping of input to output is supposed to be achieved. The property of general applicability is named generalization and describes the potential application of the learned system on unseen, not in the training included input samples.

Depending on the training strategy, the learning parameters, and data set size, generalization might not be achieved but instead reaching under adaption, named *under-fitting*, or over-adaption (*over-fitting*). Under-fitting is commonly a result of too few training samples on which the learning approach can adapt. Over-fitting is in contrast a memorizing effect, that the learning system adapts ideally to the training data but loses the necessary model-flexibility to generalize unseen inputs to a correct output.

The sample distribution of the training data set is suggested to cover the expected distribution of inputs. As a result, under-represented training samples yields to a low generalization for this specific sample class.

Random Sample Consensus: For a given set of data points, a parameter estimation or model fitting can be mathematically formalized as quadratic optimization problem [11]. Measurement noise, outliers or other artifacts of digital signal processing can cause mis-fitting of simple quadratic optimization formulations. As mitigation, alternative formulations improve outlier robustness, e.g. the maximum-likelihood estimators or M-Estimator by Huber [98]. Especially for outlier reduction of a given data set, the iterative Random Sample Consensus (RANSAC) of Fischler and Bolles [66] is commonly applied. Based on iterative sub-sampling, potential outliers are separated to a *consensus set* and model fitting is performed on the remaining *inlier* data samples.

The steps are enumerated:

1. Random sub-sampling of the whole data set.
2. Model fitting on the sub-set of samples.
3. Detect samples of the whole data set which exceed a fixed modeling error threshold and detect the number of the remaining supporting data samples.
4. Iterate over the steps 1) – 3), find the model with the largest *consensus set* and perform a classical quadratic model fitting on the largest *consensus set*.

Goal of the sub-sampling is to remove the potential outliers from the samples set to which a model can be fitted resulting for all samples in a below-threshold modeling error.

Based on the iterative quality and outlier detection, the RANSAC algorithm is computationally expensive in complexity and runtime [66]. Nevertheless, the algorithm is commonly applied in computer vision and data fitting, without real-time requirements. In chapter 3, the RANSAC method is applied during the data set generation, finding ground plane parameters on LiDAR point clouds. Exemplary work of Schnabel et al. [188] on RANSAC methods, discusses possible performance improvements.

k-Nearest Neighbor: In this work, often the problem is posed to find for an arbitrary point $\mathbf{p} \in \mathbb{R}^n$ the closest local neighboring points from a set $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_m\}$ of m points $\mathbf{q}_i \in \mathbb{R}^n$. Formalizing the general distance of a point pair \mathbf{p}, \mathbf{q}_i as Minkowski-Norm [11] with parameter $p \geq 1$

$$\|\mathbf{q}_i - \mathbf{p}\| = \left(\sum_{k=1}^n |q_{i,k} - p_k|^p \right)^{1/p}, \quad (2.17)$$

the closest neighborhood point $\mathbf{p}_{\text{neighbor}}$ is formulated by Shakhnarovich et al. [197] as

$$\mathbf{p}_{\text{neighbor}} = \arg \min_{\mathbf{q}_i \in Q} (\|\mathbf{q}_i - \mathbf{p}\|). \quad (2.18)$$

Most commonly, and also in this work the euclidean distance is applied as distance norm ($p = 2$) between the 3-dimensional points.

For a query, searching a set of $\mathcal{K} \subseteq Q$ of $K \in \{1, \dots, m\}$ nearest neighbors (k-NN), the following mathematical conditions are required to apply [150, 146]

$$|\mathcal{K}| = K \wedge \mathbf{a} \in K, \mathbf{b} \in Q \setminus \mathcal{K} : \|\mathbf{a} - \mathbf{p}\| \leq \|\mathbf{b} - \mathbf{p}\|. \quad (2.19)$$

The computational effort to iteratively query each distance, from \mathbf{p} to all points in Q , ordering these and finding the closest neighbors yields linear complexity [197]. Sub-linear complexity is achieved by respecting the spatial distribution of the data. The search can efficiently be limited to data structure parts of kd-trees [146, 197]. Practical and efficient implementations are found in the libraries of Rusu and Cousins [178] or Zhou et al. [251]. The constraint applies, that for large data sets Q of e.g. a densely populated point cloud region, the resolution parametrization of the kd-tree affects the query efficiency.

Artificial Neural Networks: Inspired by the natural brain process of cognition in nervous systems, of both human and animals, Artificial Neural Networks (ANNs) try to artificially reproduce this working principle. The combination of single units, called "neurons" or "perceptrons" serve as state compute entities and are inter-connected to a whole network. The network setup or *architecture* typically combines multiple neurons, either arranged in parallel, constituting a network *layer*, or in subsequent order, defining the *depth* of the network.

The first layer of the ANN is called *input layer*, whereas the last layer is called *output layer*. All remaining layers in between are referred to as "hidden layers", since often ANNs are imagined as black boxes, with no possibility to comprehend their internal and intermediate status.

Based on different connection types, different ANN architectures are distinguishable. The first major class of feed-forward architecture only applies feed-forward connections which propagate the input state with each neuron layer straight through the network towards the output layer. The depth of the network can be counted by number of hidden layers.

The second type of recurrent connections enable to back-propagate the state of a later neuron layer to an earlier layer. The dynamic coupling of back-flowing state information prohibits to define a certain depth of the network, since the information can indefinitely deep flow in the recurrent connections. These network architectures are also called, deep-ANNs and constitute the research field of *deep-learning* [45, 113].

Multi-Layer Perceptron: As simple example of an ANN, the Multi-Layer Perceptron (MLP) is introduced and explained. As one entity, Figure 2.7 illustrates a single neuron, while Figure 2.8 illustrates a MLP network. As the name specifies, a MLP is constituted from chained,

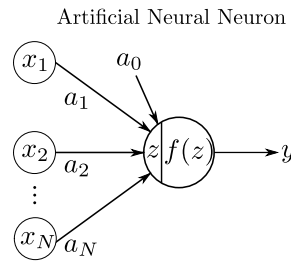


Figure 2.7: Exemplary illustration of a single artificial neural neuron.

fully connected perceptrons. The mathematical formulation of the state propagation is formalized. Given $\mathbf{x} \in \mathbb{R}^n$ as n -dimensional input of a neuron, the internal state of the neuron z is computed as sum of the component wise multiplication with a the $(n + 1)$ -dimensional weight vector $\mathbf{a} \in \mathbb{R}^{n+1}$, of which a_0 serves as additive bias to the sum [113]

$$z = \sum_{i=1}^n a_i x_i + a_0. \quad (2.20)$$

This internal state is subsequently applied as argument for a commonly non-linear activation function $f: \mathbb{R} \rightarrow \mathbb{R}$. Common, exemplary activation functions are formalized in Table 2.1 [113, 45].

$$y = f(z) = f\left(\sum_{i=1}^n a_i x_i + a_0\right) \quad (2.21)$$

This composition formulates the non-linear mapping of a single neuron for a given n -dimensional input \mathbf{x} to a 1-dimensional output y . As ensemble of the fully connected,

| Activation Functions | Formulation |
|------------------------------|---|
| Sigmoid | $\sigma = \frac{1}{1+e^{-x}}$ |
| Rectified Linear Unit (ReLU) | $\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ |
| Hyperbolic Tangent (tanh) | $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ |
| Softmax | $\sigma(x)_j = \frac{e^{x_j}}{\sum_{i=1}^N e^{x_i}}$ for $i = 1, \dots, N$ |

Table 2.1: Exemplary activation function formulations with the Softmax formulation.

multi-layer arranged non-linear perceptron mappings, the exemplary ANN architecture in Figure 2.8 allows the mathematical approximation of any non-linear function.

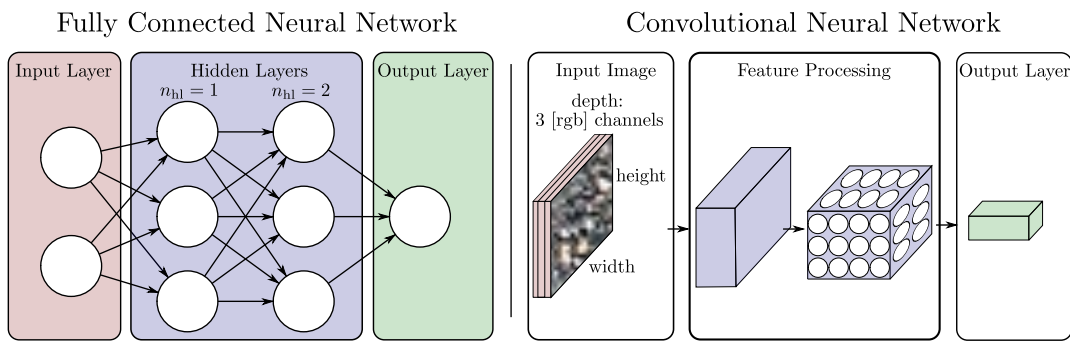


Figure 2.8: Exemplary illustration of a ANN as fully connected MLP (left), besides a Convolutional Neural Network (CNN) (right).

The application of ANNs for the discrete multi-class classification of N_C classes requires not only one single output neuron, but one output neuron per class to achieve a one-hot encoded classification output. The output vector $\mathbf{y} \in \mathbb{R}^{N_C}$ delivers an estimated score \hat{y}_i per class $i \in 0, \dots, N_C - 1$. This most probable, estimated class \hat{y}_i calculated by a combination of a class normalizing soft-max layer

$$\sigma(y_i) = \frac{e^{y_i}}{\sum_{j=0}^{N_C-1} e^{y_j}} \quad (2.22)$$

and a subsequent arg-max layer $\arg \max(\sigma(\mathbf{y}))$ to extract the most probable class.

The learn-able parameters of an ANN are constituted by all weight vectors \mathbf{a} of the network. Hence, the depth and layer count defines the trainable parameters.

Convolutional Neural Networks: The breakthrough of the application of ANNs is originally achieved in the image processing, by Convolutional Neural Networks (CNNs). Applying feature extraction, *convolutions* and *pooling* operations in composition with MLP-layers on image data, CNNs achieve unprecedented functionality.

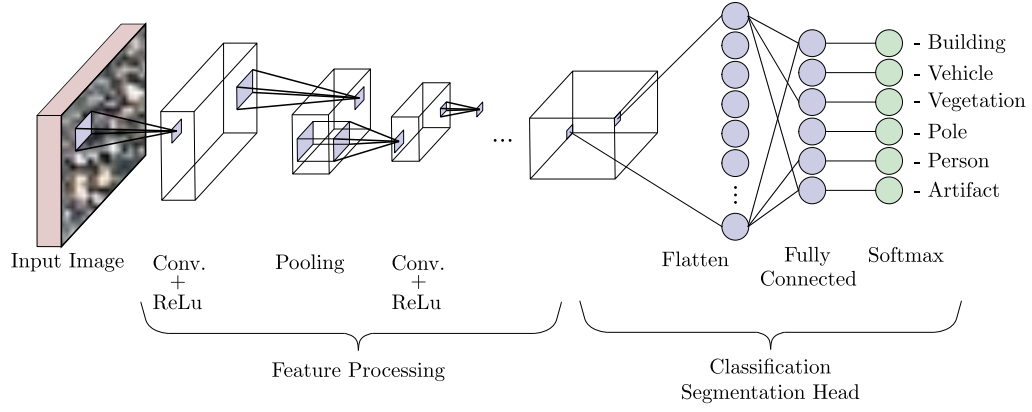


Figure 2.9: Exemplary illustration of a classical image classification CNN.

The working principle of a classification CNN is depicted in Figure 2.9. The image feature extraction by learn-able kernel-based convolutions (Conv.) are applied to extract and process local feature maps. Analogous to the neuron state Equation 2.20, given an exemplary 2D input \mathbf{X} , the calculation of the feature map \mathbf{Z} can be formulated as scalar product of the input \mathbf{X} with a filter mask \mathbf{A} , whose properties are described as kernel. For a 2D input, this scalar product yields a discrete sum

$$\mathbf{Z}_{i,j} = \mathbf{A} * \mathbf{X} = \sum_n \sum_m \mathbf{A}_{n,m} \mathbf{X}_{i-n,j-m}. \quad (2.23)$$

Additionally, a non-linear activation function, mostly the ReLU activation is applied in the convolution result, compare Figure 2.9.

As a result of the multi-dimensional tensors of size $\mathbf{Height} \times \mathbf{Width} \times \mathbf{Channels}$, the variable filter dimension $\mathbf{H}_A \times \mathbf{W}_A \times \mathbf{C}_A$, and two filter parameters *stride* s_a and padding p_a , affect the convolutions' output tensor size $\mathbf{H}_{ten} \times \mathbf{W}_{ten} \times \mathbf{C}_{ten}$. Stride s_a describes the step-size of the filter motion over the input map, mathematically expressed as increment of the sum indices in Equation 2.23, see Figure 2.10 depicting a stride of $s_a = 2$. Stride configurations of $s_a \geq 2$ downsample the input feature map. Padding as *zero padding* is depicted also in the same image. With $p_a = 1$ an empty zero-frame is applied around the original feature map. By applying a padding frame of variable value, the filter operation can be applied in edge regions of the feature map, preserving the feature map size after the convolution.

The convolutions result in an output feature map of size $\mathbf{H}_{out} \times \mathbf{W}_{out} \times \mathbf{C}_{out}$, defined by the number of applied filters per Channel \mathbf{C}_{out} and for symmetric kernel sizes by

$$\begin{aligned} \mathbf{H}_{out} &= \frac{\mathbf{H}_{ten} + 2p_a - \mathbf{H}_a}{s_a} + 1. \\ \mathbf{W}_{out} &= \frac{\mathbf{W}_{ten} + 2p_a - \mathbf{W}_a}{s_a} + 1. \end{aligned} \quad (2.24)$$

In combination with feature condensation by Pooling operations, the local information can be extracted by any arbitrary operation for further convolutional layers. Pooling operations extract information from all channels C of the size of the pooling mask. Commonly the maximum pooling is applied, but variants exist for mean or minimum pooling.

Architecture dependent feature processing can include variants of convolutions, up- or down-sampling and other layers. Commonly in the context of the feature extraction, the perceptive properties of CNN layers is interpreted as abstraction levels of features [243], ranging from low-level detail information to high-level global properties.

For the classification, the 3D tensors are flattened and connected to fully connected layers. The fully connected layer represents a convolution with a 1×1 kernel. Followed by a Soft-max layer and potentially an arg-max operation, the feature processing yields a multi-class classification [113, 45] result.

The kernel-parameters of \mathbf{W} are subject to be learned [118] and affect the feature extraction, see Figure 2.10.

Due to the feature summarizing convolutional kernel the trainable parameter count is reduced compared to fully connected layers.

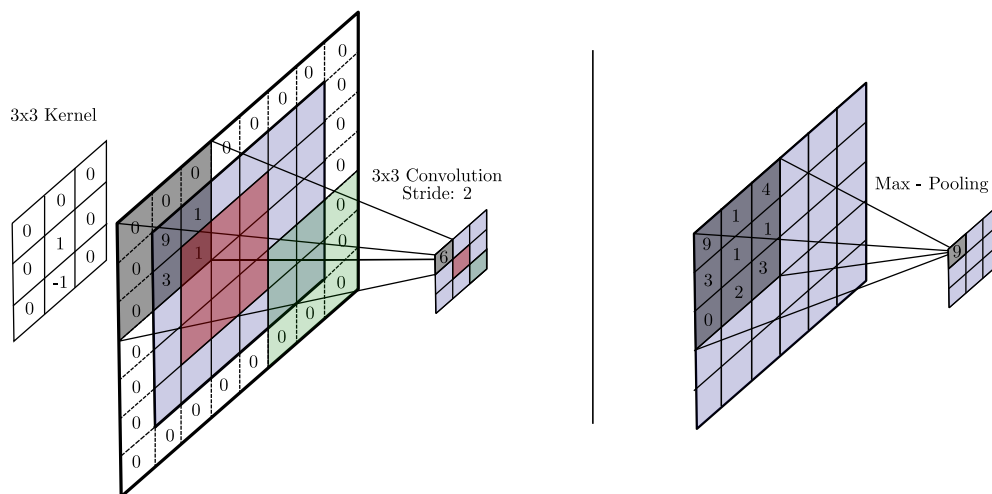


Figure 2.10: Illustration of a 3×3 kernel-based convolution with stride length 2 (left), and a 3×3 Max-Pooling operation (right) on the same image.

Figure 2.10 also illustrates the difference of Pooling compared to a Convolution. Pooling enables to compress information from different channels, or image regions to an abstract feature map, independent from local dependencies [118]. Hence, the pooling step improves robust information extraction by summarizing the most relevant information from a local feature map. Commonly, the extraction of the prominent information is performed by a maximum-pooling operation [113, 45].

The application of a CNN for classification tasks includes a flattening layer, and analogously to the before introduced MLP architecture, a softmax-layer assigns the network output to a class probability for each N_C classes and output neurons.

Neural Network Training:

The procedure to adjust the total trainable parameters $\mathbf{a}_{\text{train}}$ of an exemplary ANN is called training.

Basics: The general principle is based on gradient descent [11, 77]. First the input vector \mathbf{x} is forward processed through the network to an estimated output $\hat{\mathbf{y}}$. By a comparison of $\hat{\mathbf{y}}$ with the ground truth \mathbf{y} , a loss function calculates an output error term, which is then (back-) propagated through the network to iteratively adjust the trainable parameters .

The error term is computed between the forward calculated network prediction $\hat{\mathbf{y}}$ with respect to the ground truth label \mathbf{y} by a loss function $L: \mathbb{R}^m \rightarrow \mathbb{R}$ [77]. Several loss formulations and variants are applicable to achieve a problem and task specific network adaption, e.g. to consider edge-cases as in Aghdam and Heravi [5]. The selection of the loss-formulation affects besides the optimization direction of the gradient-based network adaption also effects of mis-classifications or outliers on the network convergence. The complexity of each specific problem, data set properties as class distribution, sample symmetry or other properties prohibit the recommendation of a universal suitable loss function [104]. Two loss functions are tested in the thesis and applied in combination.

1. **Cross-Entropy Loss:** The typical loss function for discrete multi-class classification problems is the Cross-Entropy Loss [236]. The loss function calculates an error term L_{CE} based on the mutually exclusive multi-class softmax output of the network $\hat{\mathbf{y}}$. A loss term per class i is summed to a combined loss over all N_C classes.

$$L_{CE} = - \sum_{i=0}^{N_C-1} \frac{1}{w_i} \cdot y_i \cdot \log \hat{y}_i \quad (2.25)$$

The class weighing factor $1/w_i$ is applied to realize a class-balancing of the overall loss. The weights w_i are selected as class-dependent reciprocal occurrence factors, also referred to as α -weights [49]

$$w_i = \frac{\text{Samples of class } i}{\text{All samples}}. \quad (2.26)$$

2. **Lovasz-Softmax Loss:** The second applied loss function is the Lovasz-Softmax Loss of Berman et al. [20]. This loss is inspired to achieve a differentiable formulation of the Intersection of a Union (IoU) metric, introduced later in Equation 2.44. Since the discrete definition of the IoU is not differentiable [103], the IoU performance metric can not directly be applied as loss function. Alternatively, formulated by the means of

the Lovasz extension $\overline{\Delta}_{JK}$, an average of a class-specific surrogate function is defined by Berman et al. [20]. This loss formulation allows a class-averaged IoU, denoted as mean IoU(mIoU), metric to be applied differentiable Lovasz-Softmax loss

$$L_{LS} = \frac{1}{N_C} \sum_{i=0}^{N_C-1} \frac{1}{w_i} \cdot \overline{\Delta}_{J_i}(m(\hat{y}_i)). \quad (2.27)$$

Comparable works on semantic segmentation and object detection proved that the Lovasz-Softmax loss as additional term improves the detection of rare, fine separated classes and improving the overall segmentation accuracy [20, 248, 48].

3. **Combined Loss:** In the thesis, the combination of the before introduced loss functions is tested in variants, both neglecting class weights $w_{i,CE}$, $w_{i,LS} \neq 1.0$ and including the class distribution $w_i < 1.0$. The combined loss is applied as linear combination of both loss functions

$$L_{\text{combined}} = L_{CE}(y, \hat{y}, w_{i,CE}) + L_{LS}(y, \hat{y}, w_{i,LS}). \quad (2.28)$$

Based on the forward calculated loss, the network optimization part of the training, to adjust the learn-able network parameter vector \mathbf{a} is computed. Commonly optimization algorithms apply a generalized delta rule [177], to apply an iterative gradient descent to compute adapted network parameter vectors

$$\mathbf{a}_{\text{train}}^{k+1} = \mathbf{a}_{\text{train}}^k - \eta \cdot \nabla L(\mathbf{a}_{\text{train}}^k) \quad (2.29)$$

per iteration k .

For the application of back-propagation [177], the feed-forward architecture enables an efficient computation of the loss gradient $\nabla L(\mathbf{a}_{\text{train}}^k)$.

The direction, in which the parameter vector is adapted, is defined by the calculated gradient. The exemplary adaption of a weight between neuron i and j can be formulated as

$$\Delta w_{i,j} = -\eta \frac{\partial L}{\partial w_{i,j}} = -\eta \frac{\partial L}{\partial z_j} \frac{\partial z_j}{\partial w_{i,j}} = -\eta \frac{\partial L}{\partial z_j} y_i. \quad (2.30)$$

The partial derivative of the loss function L with respect to the activation function z_j of node j is similarly re-formulated to depend on the partial derivative of the neuron output y_j . Equation 2.30 illustrates the necessity of loss functions to be derive-able. This reformulation of Equation 2.30 yields the derivative of the activation function $f'(z_j)$.

$$\frac{\partial L}{\partial z_j} = \frac{\partial L}{\partial y_j} \frac{\partial y_j}{\partial z_j} = \frac{\partial L}{\partial y_j} f'(z_j). \quad (2.31)$$

The network adaption speed, and thereby also the optimization convergence rate, is controlled by the learning rate $\eta \in \mathbb{R}^+$ step size which is commonly in the range of 0.02.

In order to improve robustness and network training speed, the training and adaption is performed on a batch of input-output pairs. Hence, batch normalization is applied on the input data [101] the gradient is averaged over the batch [25, 113, 45].

Variants of the back-propagation include robustifying measures to avoid too slow adaption for low learning rates or to get stuck in local minima. Variants to avoid "vibrating" network convergence can be achieved by dynamically adaptive, mostly decreasing learning rates. Other optimization algorithms, e.g. stochastic gradient descent (SGD) apply a momentum formulation of the learning rate, to include the latest gradient adaption also. Especially the momentum formulation is expected to avoid network convergence in local minima. As alternative state-of-the art optimizer, the ADAM optimizer [116] is applied in the thesis.

Regularization:

During the optimization of the networks' generalization and performance, different measures and modifications of the presented general learning process are applied.

- **Data Set:** Overall, the size of the data set is significant. The more samples the network training is able to learn from, the better adaption is expected. But not only the sheer data set size is important, but also the prevalent data distribution. The sampling distribution of heterogeneous examples of relevant sample events is important to enable the network to adapt to its whole operational domain. For the exemplary case of multi-class classification problems, the class occurrence should ideally be balanced over all classes either by re-sampling or cost-sensitive learning [49]. In general, deep and advanced architectures with numerous parameters require an adequate number samples to achieve convergence. A complex model with multiple degrees of freedom tends to over-fit or remains unconverged without being trained on sufficient data, failing to learn a general mapping. As one popular mitigation to avoid non-convergence, the amount of available training data can artificially be increased by creating realistic training data variants, e.g. by coordinate flipping, rotation or additional noise, referred to as data augmentation.
- **Data Set Split:** Not all available training data is applied for training the network. Instead a sub-set is applied for training, an other independent validation sub-set is applied to validate the training adaptations, and a third independent test set is not used during training but as data-pool to evaluate the generalization capability of the learning system. Therefore the available data set is split in a training data set ($\sim 70\%$), a validation data set ($\sim 15\%$), and a test data set ($\sim 15\%$).
- **Early Stopping:** The network adaption on the training data is validated throughout the training on a unknown validation data set. Therefore, if the validation performance

metrics do not improve, but only network performance on the training data set, the training is stopped to avoid over-fitting [45].

- **Dropout:** The goal of an unbiased network adaption is to achieve an balanced level of neuron influence on the output. Instead of a strong dependency by over-proportional weight of single neurons, it is beneficial to achieve balanced weights. Otherwise, the robust generalization on unseen data might not be reached. Hence, the influence of single neurons is avoided by selecting one or multiple network layers, of which a randomly selected fraction of neurons is deactivated during training [69]. As alternative, Ghiasi et al. [72] suggest drop-block, a method to apply the neuron dropout on whole contiguous regions of the network.

- **(Batch) Normalization:**

Independent feature value scales complicate the gradient descent processing. The iterative and step-wise adjustment of network parameters during the training is improved with normalization as feature pre-processing.

Mean normalization to zero

$$\bar{x}_j = \frac{1}{q} \sum_{i=1}^q x_{j,i}, \quad (2.32)$$

and standard deviation of one

$$\sigma_{x_j}^2 = \frac{1}{q} \sum_{i=1}^q (x_{j,i} - \bar{x}_{j,i})^2, \quad (2.33)$$

is applied per component, with $\epsilon \ll 1$ as stabilizing element [101],

$$x_{j,i,norm} = \frac{x_{j,i} - \bar{x}_{j,i}}{\sqrt{\sigma_{x_j}^2 + \epsilon}}. \quad (2.34)$$

The normalization reduces the input data covariance shift [101] and can be considered as regularization since the normalization is performed on noisy input data batches containing measurement noise.

- **Min-Max Normalization:** For a normalization of relative doppler velocity values, a min-max normalization is applied. Since the relative speed of a radar scan changes with the occurrence of moving objects in the scene, a normalization to fixed potential values result in a task dependent assumption. E.g. for parking, the estimated min-max values vary drastically from the expected min-max values of a driving scene on a highway with traffic. In order to remain generic, flexible, but intra-scan consistent,

without a dependency of a static speed range, for each radar scan the min-max scaling is applied with dynamic values.

$$x_{j,i,\text{norm}} = \frac{x_{j,i} - \min(x_{j,i})}{\max(x_{j,i}) - \min(x_{j,i})}. \quad (2.35)$$

Classification Metrics:

To evaluate the semantic radar segmentation, performance metrics are applied: Accuracy, Confusion Matrices and Intersection of a Union serve as major evaluation metrics. Formulas for Precision, Recall, and F1-Score are defined as well. Especially the interpretation of a confusion matrix, to find inter-class confusion can be insightful.

- **Performance Metrics:** The task to segment point clouds is independently of the sensor or data source posed as multi-class classification problem.

As evaluation measure of point cloud segmentation, commonly the Intersection of a Union (IoU) is applied.

As basis of any classification problem with input data \mathbf{x}_m , $m \in 1, \dots, M$, the class-wise interpretation of the learned system output estimation $\mathbf{y}(\mathbf{x}_m)$ compared to the true system output $y(\mathbf{x}_m)$ defines a foundation of other metrics.

For the discrete multi-class classification problem at hand, the class predictions are mutually exclusive and can be evaluated in a confusion matrix. Consider for each sample class C_c , if the networks prediction $y(\mathbf{x}_m)$ segmentation class C_p matches the examples' true ground-truth class C_{GT} . For the various combinations of true or false predictions, the common terminology is commonly applied, according to Piewak [164]:

– True Positive (TP): The sample is correctly classified, which is the considered class.

$$C_p = C_{GT} = C_c. \quad (2.36)$$

– False Positive (TN): The sample is correctly classified, but it is not the considered class.

$$C_p = C_{GT} \neq C_c. \quad (2.37)$$

– True Negative (FP): The sample is misclassified as the considered class.

$$C_p = C_c \neq C_{GT}. \quad (2.38)$$

– False Negative (FN): The sample is misclassified as any other class than the considered class.

$$C_p \neq C_c = C_{GT}. \quad (2.39)$$

With the definition of the TP , TN , FP and FN sample bins, classification metrics can be computed as different fractions of sample bins.

- **Accuracy:** To measure the precision of a classification task for a specific class, the quotient of correct classified samples over the tested samples is introduced as accuracy:

$$\text{accuracy}(C_i) = \frac{\text{correct classified samples}}{\text{number of tested samples}} = \frac{TP_{C_i} + TN_{C_i}}{TP_{C_i} + TN_{C_i} + FP_{C_i} + FN_{C_i}} \quad (2.40)$$

For an unbalanced multi-class classification data set, the accuracy metric can be misleading. Correct testing of *simpler*, more frequent classes yields good accuracy results, while especially rare relevant classes, even if they are classified incorrectly, are not significantly captured by the metric. Instead of introducing a class-weighted accuracy version, the confusion matrix is applied to interpret the accuracy.

- **Precision:** Precision measures which fraction of true positive samples are true.

$$\text{precision}(C_i) = \frac{TP_{C_i}}{TP_{C_i} + FP_{C_i}} \quad (2.41)$$

- **Recall:** Recall measures which fraction of true positive samples are classified correctly as true positives.

$$\text{recall}(C_i) = \frac{TP_{C_i}}{TP_{C_i} + FN_{C_i}} \quad (2.42)$$

- **F1-Score:** As combination measure of Precision and Recall, the F1-Score or Dice Similarity Coefficient (DSC) is defined as harmonic mean of both [54].

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (2.43)$$

- **Confusion Matrix:** To get an insight which classes the classification falsely classifies, a confusion matrix can help to interpret the behavior. With the binary label of TP or FN, an illustration of the classification distribution can be shown. The correctly classified samples define the diagonal, whereas the minor diagonal entries indicate per class, which class the classification yielded in the false predictions.

To normalize the confusion matrix, class sample independence is introduced by a normalization to the number of overall samples counts per class. As a result, the confusion matrix contains percentages instead of sample counts. A well tuned classifier yields a high range on the diagonal, at a low number or fraction of minor diagonal entries. The ideal confusion matrix is the identity matrix, without minor diagonal entries.

- **Intersection of a Union:** Using the introduced definitions, the Intersection of a Union (IoU) can be formulated as most common segmentation metric. The IoU is also called

Table 2.2: Exemplary confusion matrix evaluation of a classifier result. Illustration in reference to class 1 to denote the nomenclature of positive and negative samples. Predictions of class 2 and class 3 are mis-classifications with respect to class 1, so negative.

| | Prediction | | |
|---------|------------|------------|------------|
| | Class 1 | Class 2 | Class 3 |
| Class 1 | TP_{C_1} | FN_{C_1} | FN_{C_1} |
| Class 2 | FP_{C_1} | TN_{C_1} | TN_{C_1} |
| Class 3 | FP_{C_1} | TN_{C_1} | TN_{C_1} |

Jaccard-Index or Jaccard-Coefficient, and generally describes a similarity measure of predicted and true object boundaries or overlapping quantities [103, 45].

Originally defined for the image segmentation task, the IoU metric defines a measure for segmentation evaluation. Its metric is best explained for the image classification task. The idea is to check the fraction of the segmented image, an area of pixels of the classification at test with the ground truth segmented image. The overlay fraction of both segmentation areas is normalized and defined as IoU. The measure tests if the classification area is overlapping and matches with the ground truth area, plus describes if the inferred class matches. In brief, the IoU describes the spatial 2D position of the classification plus the classification result.

$$\begin{aligned}
 IoU(C_i) &= \frac{\text{Overlap of prediction and ground-truth sample } C_i}{\text{Union of the prediction and ground-truth sample } C_i} \\
 &= \frac{TP_{C_i}}{TP_{C_i} + FP_{C_i} + FN_{C_i}}.
 \end{aligned} \tag{2.44}$$

A score of 100% represents the total overlap and correct classification result of an ideal prediction. This can only be reached if no FP or FN are classified.

The IoU metric is formulated for one considered class $c \in N_c$ evaluation at a time. But since the semantic segmentation problem is posed as multi-class segmentation with N_c classes, averaging of the IoU scores over all classes yields a mean metric for the multi-class segmentation, the mean IoU (mIoU).

$$mIoU = \frac{1}{N_c} \sum_{C_i=1}^{N_c} IoU(C_i). \tag{2.45}$$

For point cloud segmentation, the mIoU is defined as performance metric and applied similarly to image segmentation evaluation, see Equation 2.44. Since points don't have a spatial 2D or 3D extension, their area property can not be compared. Instead, the

definition if the IoU for point clouds degrades to a measure describing per class if the classification was true or not.

The resulting IoU for point clouds can be interpreted as a False Positive or True Positive classification measure which can not be differentiated. As a result, the IoU can not be directly applies as loss function.

2.4 Point Cloud Registration

The registration of two point clouds, defines the problem to find the corresponding arbitrary spatial transformation maximizing the overlap of the source and target point cloud. In this thesis, the scale is treated as fixed, allowing only rigid point cloud transformation, as rotation or translation in 6D.

For the case of known correspondences between points, assuming Gaussian noise N_i per point i [13], the analytic formulation of the transformation is given as

$$p'_i = \mathbf{R}p_i + \mathbf{t} + N_i. \quad (2.46)$$

As general formulation, applying the quadratic error as quality function, the registration can be expressed as minimization of the target function

$$F(\mathbf{R}, \mathbf{t}) = \sum_i^N \|p'_i - \mathbf{R}p_i + \mathbf{t}\|^2, \quad (2.47)$$

to yield the optimum Rotation matrix \mathbf{R}^* and translation \mathbf{t}^* . Applying singular value decomposition (SVD) on the minimization, the rotation \mathbf{R}^* can be computed directly [11, 13]. The remaining translation \mathbf{t}^* is then given as

$$\mathbf{t}^* = p' - \mathbf{R}^* p. \quad (2.48)$$

The influence of outliers or false point-associations degrade the quadratic quality expression according to Andersen [10]. In addition, start conditions or a-priori estimates are commonly available, which can be applied for iterative procedures with k-NN approximation [162, 133]. One famous example is the Iterative-Closest Point (ICP) method [133]. According to the work of Yang et al. [232], these procedures are sensitive to the initial transformation.

Normal Distribution Transform: Besides point-based approaches solving the target function iteratively, the Normal Distribution Transform (NDT) applies an abstract representation of points for the matching. The abstraction yields increased robustness against outliers [92] and suits the application of sparse, noisy radar point clouds [139] with unknown or erroneous

associations. The presented theoretical formulation for a 2D space is given by the work of Magnusson [137].

Given the source point clouds \mathcal{P} and the target point cloud $\tilde{\mathcal{P}}$, the points of both point clouds are discretized into a grid \mathbf{G}_i , $i = 1, \dots, N_G$, analogously $\tilde{\mathbf{G}}$.

Per cell G_i the corresponding points are counted. If the point number $|\mathcal{P}_{G_i}|$ exceeds a minimum threshold n_{min} , these points are summarized by the mean vector μ_{G_i} , the covariance matrix Σ_{G_i}

$$\mu_{G_i} = \frac{1}{|\mathcal{P}_{G_i}|} \sum_{j=1}^{|\mathcal{P}_{G_i}|} p_j, \quad (2.49)$$

$$\Sigma_{G_i} = \frac{1}{|\mathcal{P}_{G_i}| - 1} \sum_{j=1}^{|\mathcal{P}_{G_i}|} (p_j - \mu_i)(p_j - \mu_i)^T. \quad (2.50)$$

Empty or too sparse grid cells are left unconverted, since a sparse NDT can yield singularities in the covariance matrix.

The adaption of Equation 2.47 to distances between the normal distributions instead of points yields

$$F_{\text{NDT}}(\mathbf{R}, \mathbf{t}) = \sum_{(k,l)}^{(|G_i|, |\tilde{G}_i|)} \tilde{\mathcal{N}}(\mu_{k,l}, \Sigma_{k,l}), \quad (2.51)$$

with mean $\mu_{k,l}$ and covariance $\Sigma_{k,l}$

$$\mu_{k,l} = \mathbf{R}\mu_k + \mathbf{t}\mu_l, \quad (2.52)$$

$$\Sigma_{k,l} = \mathbf{R}^T \Sigma_k \mathbf{R} + \Sigma_l. \quad (2.53)$$

The normal distribution $\tilde{\mathcal{N}}$ without the normalization factor is given by

$$\tilde{\mathcal{N}} = \exp\left(-\frac{1}{2} \mu_{k,l}^T \Sigma_{k,l}^{-1} \mu_{k,l}\right). \quad (2.54)$$

Compare Equation 2.56 for a full definition.

Based on the discretization in cells, the number of points per cell $|G_i| |\tilde{G}_i|$ and therewith the number of associations in Equation 2.51 is low. For an efficient search, based on an initial guess, a k-NN search can be applied to reduce the matching to respect only a sub-set of the best matching associations [137].

2.5 Measurement Uncertainty

Measurements include stochastic deviations, which need to be considered as stochastic errors. Especially important is the effects of the stochastic errors for the formulation of a localization based on measurements [26].

Based on the central limit theorem, that the mean of a sample is distributed as normal distribution, the most commonly applied error modeling is by Gaussian normal distribution. The probability density of a 1D variable x can be formalized with respect to the population mean μ_x and the populations' standard deviation σ_x as general normal distribution

$$\mathcal{N}(\mu_x, \sigma_x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left(-\frac{(x - \mu_x)^2}{2\sigma_x^2}\right). \quad (2.55)$$

The probability density can be alternatively expressed in the information form, which simplifies the formulation of optimization problems [131], see Section 5.

$$\mathcal{N}_{\mu_x, \sigma_x} = \frac{\exp\left(-\frac{1}{2}\mu^T \Omega \mu\right)}{(2\pi)^{\frac{n}{2}} \sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(x^T \Omega x + x^T \xi)\right) \quad (2.56)$$

Bases on the covariance matrix $\Sigma = \mathbb{E}[(x - \mu)(x - \mu)^T]$ the information matrix $\Omega = \Sigma^{-1}$ and potential $\xi = \Sigma^{-1}\mu$ is applied to formulate this multi-variate probability density, similar to Arras [12].

Error Propagation: The use of stochastic system inputs generally affects the system output [175]. Assuming a 1D case of a normal distributed system input x , the variance σ describes the input uncertainty, and μ_x the population mean. Given a system output y formulated as function $y = f(x)$, assuming y to be normal distributed, the system output can be approximated by a Taylor expansion (1st order) with respect to μ_x [12]

$$y \approx f(\mu_x) + \left. \frac{\partial f}{\partial x} \right|_{x=\mu_x} (x - \mu_x). \quad (2.57)$$

The output mean $\mu_y = f(\mu_x)$ and variance $\sigma_y = \left. \frac{\partial f}{\partial x} \right|_{x=\mu_x} \sigma_x$ are given analogously.

2.6 Graph-SLAM

Stachniss [205] claims the problem of vehicle self localization and environment mapping to be tightly coupled. This coupled problem set is defined as Simultaneous Localization and Mapping (SLAM) formulation [82]. A general SLAM formulation is applied and extended in this thesis. Hence, the general fundamentals of graph-SLAM problems are explained according to the detailed works of Thrun and Montemerlo [212] and Grisetti et al. [82].

The graph-SLAM formulation can be split in two parts. The front-end relates sensor measurements and constructs a graph of poses or landmarks as nodes [155, 82, 212]. The nodes are connected by edges, defining (measured) conditions to relate the two connected node position. The SLAM back-end constitutes the second part of the SLAM formulation. The back-end utilizes the graph-formulation to formulate an error function and optimizes the node positions with respect to the connecting conditions. The node positions x_i , describing the vehicle position (x, y) and orientation ϕ at time t_i , are summarized in the SLAM context as trajectory $X = x_1, \dots, x_k, \dots, x_n$. The position and orientation is limited to a planar motion and can be expressed as homogeneous transformation T . Given additional odometry measurements z_i between the vehicle positions x_i , the measurements are summarized as $Z = z_1, \dots, z_k, \dots, z_n$. Each measurement can be analogously be expressed as homogeneous transformation, with the sensor uncertainty properties as covariance matrix Σ and information matrix Ω .

The formulation of the optimization problem is expressed as maximization of the conditional pose probability

$$\begin{aligned} X^* &= \arg \max_X p(X|Z) \\ &= \arg \max_X \prod_{k,j} p(z_{i,j}|X) \prod_i p(x_k) \end{aligned} \quad (2.58)$$

The \ominus Operator in Equation 2.59 denotes the pose distance $\mathbf{T}_{x_i}^{x_j} = \mathbf{T}_{x_j}^{-1} \mathbf{T}_{x_i}$ between node x_j and node x_i .

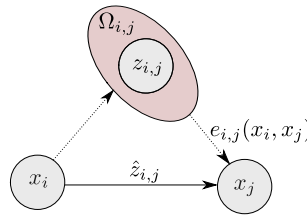


Figure 2.11: Exemplary illustration of the error $e_{i,j}$ based on the measurement $z_{i,j}$ relating node x_i to the node x_j . Illustration based on Grisetti et al. [82].

Assuming of a normal distributed measurement error

$$p(z_{i,j}) \sim \exp\left(-\frac{1}{2}\|z_{i,j} - (x_j \ominus x_i)\|_{\Omega_{i,j}}^2\right), \quad (2.59)$$

analogously the prior $p(x_k)$ is also assumed as normal distribution. Defining the error terms $e_{i,j}$ between nodes, illustrated in Figure 2.11, and an position error e_k with respect to the absolute position

$$e_{i,j} = z_{i,j} - (x_j \ominus x_i), \quad e_k = z_k - x_k, \quad (2.60)$$

the optimization of Equation 2.58 can be expressed as logarithmic expression and quadratic minimization problem:

$$\begin{aligned} X^* &= \underset{X}{\operatorname{argmin}} -\ln p(X, |Z) \\ &= \underset{X}{\operatorname{argmin}} F(X, Z) = \sum_{k,j} \underbrace{\|e_{i,j}\|_{\Omega_{i,j}}^2}_{:=F_{i,j}} + \sum_k \|e_k\|_{\Omega_k}^2. \end{aligned} \quad (2.61)$$

Neglecting the absolute positioning error e_k , the remaining argument $F_{i,j}$ in Equation 2.61 yields a local optimization function which can be solved by linearizing around an initial solution \tilde{X} [82]. The 1st order Taylor extension of the odometry error $e_{i,j}$, with Jacobi matrix $\mathbf{J}_{i,j}$ at \tilde{X} is given by

$$\begin{aligned} e_{i,j}(\tilde{x}_i + \Delta x_i, \tilde{x}_j + \Delta x_j) &= e_{i,j}(\tilde{X} + \Delta X) \\ &= e_{i,j} + \mathbf{J}_{i,j} \Delta X \end{aligned} \quad (2.62)$$

and is inserted as error approximation in the local optimization function

$$F_{i,j}(\tilde{X} + \Delta X) \approx (e_{i,j} + \mathbf{J}_{i,j} \Delta X)^T \Omega_{i,j} (e_{i,j} + \mathbf{J}_{i,j} \Delta X). \quad (2.63)$$

For the optimization step the measurement Z is constant and neglected in the argument notation of $F_{i,j}$ further on.

After multiplication of Equation 2.63, the terms are summarized

$$c = \sum e_{i,j}^T \Omega_{i,j} e_{i,j} \quad b = \sum e_{i,j}^T \Omega_{i,j} \mathbf{J}_{i,j} \quad H = \sum \mathbf{J}_{i,j}^T \Omega_{i,j} \mathbf{J}_{i,j}, \quad (2.64)$$

yielding the final optimization formulation

$$\begin{aligned} F(\tilde{X} + \Delta X) &= \sum_{(i,j)} F_{i,j}(\tilde{X} + \Delta X) \\ &= c + 2b^T \Delta X + \Delta X^T H \Delta X. \end{aligned} \quad (2.65)$$

The quadratic formulation of Equation 2.65 can be solved analytically, yielding incremental coordinates $\Delta X^* = -H^{-1}b$ and original coordinates $X^* = \tilde{X} + \Delta X^*$.

The iterative optimization procedure can be solved e.g. by the Gauss-Newton algorithm [21]. The algorithm linearizes the problem in each step k around the optimization location, which is the intermediate solution X_{k-1}^* of the iteration before. The Gauss-Newton algorithm requires the parameter space X to be an Euclidean space [21], but due to the vehicle position being included in the vehicle pose x as orientation ϕ , this assumption does not hold. Consequently, the optimization potentially results in sub-optimal results [212].

In order to approximate a local Euclidean space for the local variation ΔX , Grisetti et al. [82] suggest the Operator \boxplus to map the local variation ΔX from the Euclidean space onto a manifold. The detailed definition of \boxplus is found in the work of Grisetti et al. [82] and applied in this work without modification. General information about optimization on manifolds is discussed in detail by Hertzberg [86]. According to Hu et al. [94], the local properties of the manifold satisfy the Euclidean space requirements. To include the operator \boxplus , the error definition $e_{i,j}$ of Equation 2.62, is reformulated to

$$\begin{aligned}\hat{e}_{i,j}(\Delta \hat{x}_i + \Delta \hat{x}) &= e_{i,j}(\tilde{x}_i \boxplus \Delta \hat{x}_i, \tilde{x}_j \boxplus \Delta \hat{x}_j) \\ &= e_{i,j}(X_{i,j} \boxplus \Delta \tilde{X}) \\ &\approx \hat{e}_{i,j} + \hat{\mathbf{J}} \Delta \tilde{X}\end{aligned}\tag{2.66}$$

This modification also affects the definition of the Jacobi matrix and the target function F , which is discussed in detail in Grisetti et al. [82].

With the definition of \boxplus , the incremental adaption can be formulated as

$$\Delta \tilde{X} = \begin{bmatrix} \Delta \tilde{T} \\ \Delta \tilde{q} \end{bmatrix}\tag{2.67}$$

with T as translation vector and q as rotation quaternion. Due to the general assumption of rigid measurements, the quaternions' scaling factor q_w is neglected.

For further detailed description of other solving methods, especially covering the optimization on Cartesian manifolds, interested readers are referred to the original works [212, 4, 82].

2.7 Experimental Vehicle Setup for Rapid Prototyping

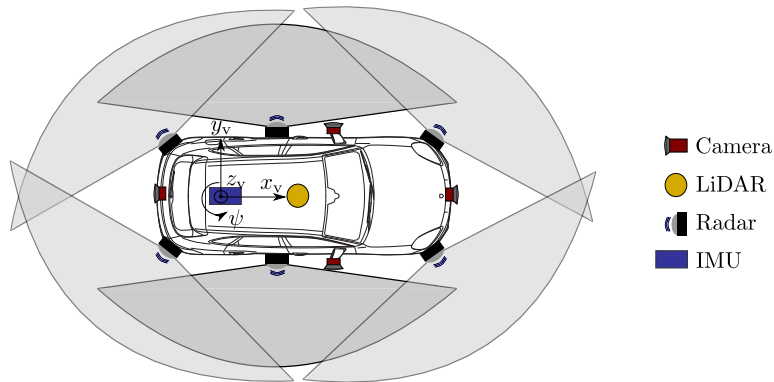


Figure 2.12: Sensor setup illustration in top view perspective with exemplary radar sensor FoV.

The test vehicle setup is an essential part of this thesis realization, due to the application of real-world measurements. Specifically for this thesis, a test vehicle setup has been designed and implemented. The general vehicle setup as automated vehicle and rapid prototyping platform resembles works on ROS-based architecture [85] and series vehicle integration [112, 2]. The major advantage is the sensor set of 360° covering radar sensors, a LiDAR, multiple cameras, and differential GPS. Compare Table 2.3 for similar autonomous vehicles according to a survey of Yurtsever et al. [240].

Table 2.3: On board sensor set of automated vehicles according to Yurtsever et al. [240].

| Platform | # 360° rotating | # stationary | # Radars | # Cameras |
|-------------------------|-----------------|--------------|-----------|-----------|
| | LiDARs | LiDARs | | |
| Ours | 1 | - | 6 | (4) |
| Nagoya Univ [240] | 1 | - | 6 | (4) |
| Boss [216] | 1 | 9 | 5 | 2 |
| Junior [122] | 1 | 3 | 6 | 4 |
| BRAiVE [27] | - | 5 | 1 | 10 |
| RobotCar [15] | - | 3 | - | 4 |
| Google car (Prius) [83] | 1 | - | 4 | 1 |
| Uber car (XC90) [203] | 1 | - | 10 | 7 |
| Uber car (Fusion) [203] | 1 | 7 | 7 | 20 |
| Berta [253] | - | - | 6 | 3 |
| Appollo Auto [2] | 1 | 3 | 2 | 2 |

Table 2.4 describes the sensor details of the vehicle sensor set. The relevant sensors mounting positions for this thesis are depicted in Figure 2.12. Figure 3.7 shows blind spot areas of the camera and LiDAR field of view. The utilized series vehicle cameras are accessed as fisheye-images and need to be pre-processed.

| Sensor | Technology | Details | f_{sample} [Hz] | Series |
|----------------------|----------------------------------|--|--------------------------|--------|
| Surround View Camera | CMOS | Position: Front, Back, Mirrors | 30 | ✓ |
| IMU | Turning Rate, Acceleration | Signal from ESP-/ABS-ECU | 16.7 | ✓ |
| LiDAR | Time-of-Flight | vertical: 40 Channels (+7° to -16°) horizontal: 360° (Resolution: 0.2°) Range: 200 m Position: Roof mounted | 10 | x |
| Radar | FMCW (77 GHz) | Position: Front, Back, B-pillar (left/right) vertical: $\pm 10^\circ$ horizontal: 160° Range: 100 m | 16.7 | (✓) |
| Gyroscope | DGPS, Turning Rate, Acceleration | Reference sensor Position accuracy: < 2 cm | 100 | x |

Table 2.4: Sensor set details.

The radar sensors cover a 360° FoV and are mounted "invisible", inside the bumper shell, compared to other setups of non-automotive radar sensors. This radar integration contributes to the general effect of dynamic uncertainty. Depending on range, azimuth and Radar Cross Section (RCS), the uncertainty of radar sensors vary. Figure 2.13 and Figure 2.14 depict the measurement based standard deviation estimation of $\hat{\sigma}_\varphi^*$ and $\hat{\sigma}_r^*$ respectively. Significant radar reflections cause higher RCS values and are detected with ease. In contrast, an increasing range of detections yields increased uncertainty, corresponding to the theoretical findings of the radar equation [202].

For the rear range detections, especially the azimuthal position towards the periphery of the FoV ($\phi = 0^\circ$ or $\phi = \phi_{max}^\circ$) affects the uncertainty to increase also. As a result of the bumper-covered integration and occasional interaction with near located metal parts, the radar antenna resolution limit yields this error. Besides the azimuthal dependency, the uncertainty in the near range can be treated constant in radial direction. This real-world set-top with partially increased error characteristic complicates the statistic detection modeling and needs to be treated in the signal pre-processing.

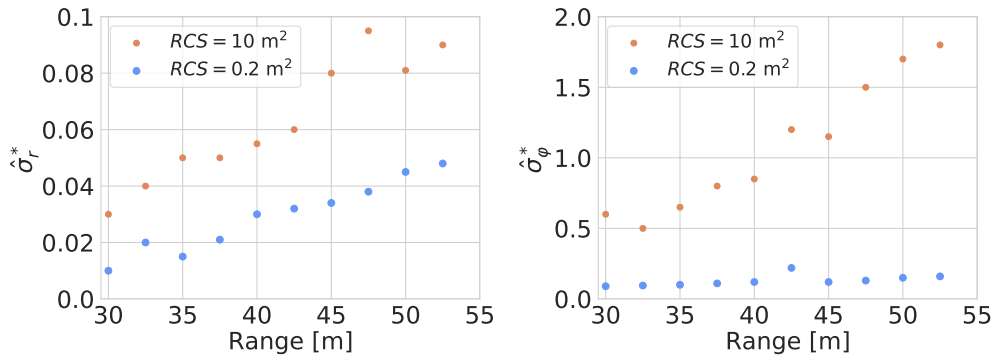


Figure 2.13: Radar sensor uncertainty in the mid-range [30 m, 55 m] for center based detections $\phi = 0.5\phi_{max}$.

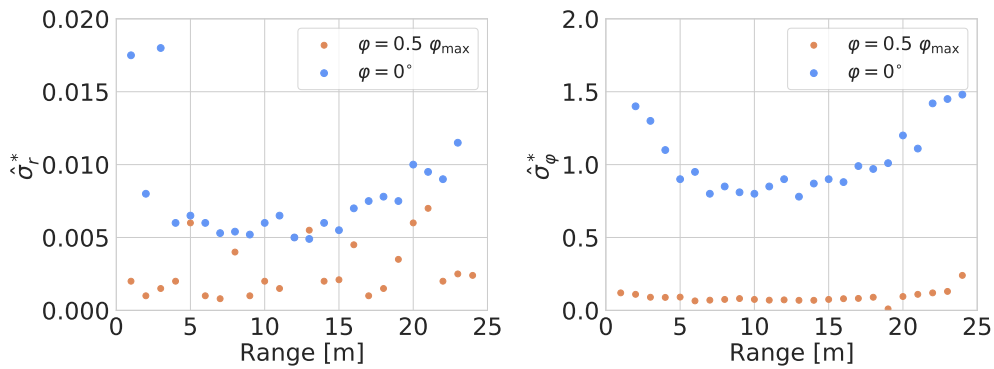


Figure 2.14: Radar sensor uncertainty in the near-range [0 m, 25 m] for a strong reflector of $RCS = 10 \text{ m}^2$, at the FoV margin $\phi = 0^\circ$ and a center detection $\phi = 0.5 \phi_{max}$.

Car-PC and Robot Operating System: Following the principle to apply the open-source environment ROS as rapid prototyping platform on a real-world test vehicle, the necessary interfaces to enable a ROS communication and actuation are implemented for this thesis. Figure 2.15 illustrates the system setup as block diagram.

Similar to the exemplary architecture in Kessler et al. [112], the core of the software integration is a CarPC, equipped with Docker and ROS melodic. The ROS platform reads vehicle sensor information from CAN or proprietary interfaces, together with external sensors e.g. from UDP-sockets. The implementation of the controller part is outsourced and deployed on a real-time capable embedded platform. This Micro Auto Box⁶ receives driving requests via a private CAN connection from the ROS system and generates actuator signals that are CAN-Mapped to actuate the vehicle.

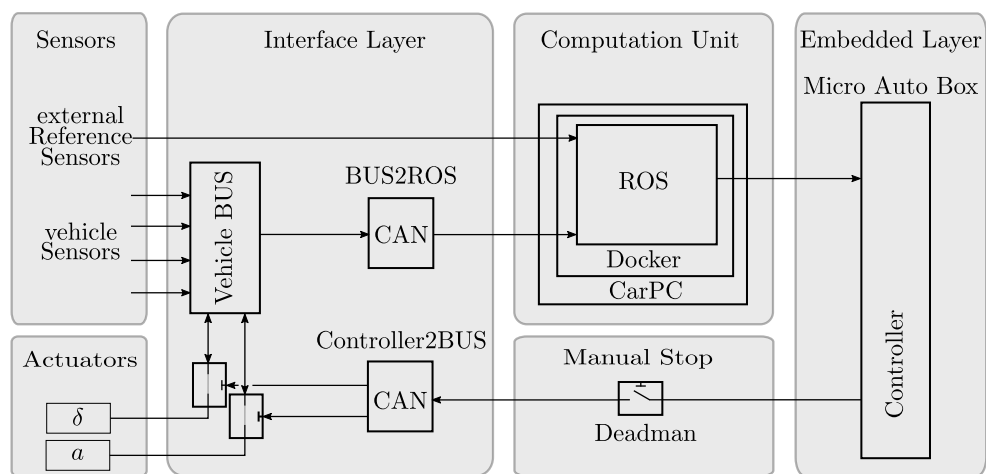


Figure 2.15: Test vehicle setup as block diagram of the different hardware devices. Signal flow from sensors to actuators shown as arrows.

The CarPC is Ubuntu⁷ 18.04 based, equipped with 2xIntel®Xeon®E5-2667 CPUs resulting in 32 kernels (3.20/3.6GHz), 16 GB RAM and a NVIDIA®Tesla V100 GPU.

⁶ Product information: <https://www.dspace.com/de/gmb/home/products/hw/micautob/microautobox2.cfm>

⁷ Product Information: <https://releases.ubuntu.com/18.04/>

3 SEMANTIC RADAR POINT CLOUD LABELING

The introduction to supervised machine learning in Section 2.3 describes the requirement of a ground truth data set with ground truth labels to train a supervised learning approach. This section outlines the point-wise radar labeling pipeline and data set structure, which is later applied in the next sections.

The task to apply machine learning is not only setting up learning architectures, but most of the time, effort and knowledge needs to be spent for data preparation, data cleaning and conversion in the appropriate form and representation format.

Numerous contributions on perception and localization nowadays rely on the availability of visual data sets, on which algorithms and neural networks are tested and developed to mine information [225, 65, 240]. Among publicly available data sets for research and academia in the domain of autonomous driving, Schumann et al. [194] summarized the Table 3.1 as extract of the relevant data sets. This table overview is extended with the details of the especially created data set of this thesis.

The visual data set availability results from the common sensor set of research and other autonomous vehicles relying mainly on multiple cameras in mono- or stereo-setup and LiDAR scanners. Since both visual sensor types, camera and LiDAR, provide rich measurements of the environment and are comparably intuitive interpretable for labeling objects, there exist multiple specific data sets and extensions for the multiple different special tasks (e.g. for depth estimation [71, 237], for object detection, for semantic segmentation [18], for instance segmentation, etc.). From visual data, context extraction and semantic processing, e.g. semantic segmentation, is a common task [65].

Radar sensors are generally included in modern passenger vehicle setups as well, but until recent, this sensor data is not necessarily applied in many functions. Since sparsity and noise detections are sensor inherent, the difficult interpretation of cluttered radar data yields until recent to an underestimation of the static environment perception capabilities [151]. Due to the point cloud learning, radar sensors are currently further researched and the next generation of potentially imaging radars is considered as future trend in vehicle environment sensing [151].

As a result, until recent there exists only a limited number of radar data sets, while each existing data set is recorded with different sensors or with a solely unique sensor setup. As

| Data Set | Size S-M-L | Radar Type | Alternative Sensors | Sequential Data | Doppler- Velocity | Number of Categories | Class Balancing | Point Annotations | Object Annotations | Varying Scenarios |
|--|---------------|---------------------------------------|---|--------------------|----------------------|-------------------------|--------------------|--|------------------------------|----------------------|
| Oxford Radar RobotCar[15] | L | Mechanically Scanning | Stereo Camera, LiDAR | ✓ | x | - | N/A | | x | + |
| MulRan[115] | M | Mechanically Scanning | LiDAR | ✓ | x | - | N/A | | x | + |
| RADIATE[201] | L | Mechanically Scanning | LiDAR, Camera | ✓ | x | 8 | ✓ | | 2D Boxes | ++ |
| nuScenes[30] | L | Low Res. Automotive | LiDAR, Camera | ✓ | ✓ | 23 | ✓ | | 3D Boxes | ++ |
| Zendar[147] | L | Low Res. Autom. (High Res SAR) | LiDAR, Camera | ✓ | ✓ | 1 | N/A | | 2D Boxes (2.5% Manual) | ++ |
| Astyx[142] | S | Next Gen. Automotive | LiDAR, Camera | x | ✓ | 7 | - | | 3D Boxes | - |
| NLOS-Radar[187] | S | Next Gen. Automotive | LiDAR | ✓ | ✓ | 2 | ++ | Point-Wise | | - |
| CARRADA[156] | S | Automotive | Camera | ✓ | ✓ | 3 | ++ | | Spectral Annotations | - |
| CRUW[219] | L | Automotive | Stereo Camera | ✓ | x | 3 | + | | Spectral Boxes (Only 19%) | - |
| RadarScenes[194] | L | Automotive | Docu Camera | ✓ | ✓ | 11 | + | Point-Wise for Objects (no static env.) | | ++ |
| Porsche InnoCampus (Ours) | M | Automotive | LiDAR, 4x Camera (Surround View) | ✓ | ✓ | 6 | + | Point-Wise (especially static) | 3D Boxes | ++ |

Table 3.1: Tabular overview of public radar data sets according to the RadarScenes publication [194]. The authors compare radar data sets for machine learning purposes with special focus for dynamic objects, referred as road users. Column *scenario variations* are considered as weather, traffic, or road types variation. *Sequential data* describes if temporally subsequent radar scans are available. Our data set contains as single data set point-wise labels for environment detections.

discussed in Section 2, there exist in the domain of radar sensors a variety of specific radar sensors which deliver a different data representation and quality. In general common, they deliver radar detections. But only some sensors allow the interpretation in the form of 2D or 3D coordinates as a point cloud, while advantageous radar specific attributes, e.g. Signal Power or Signal to Noise Ratio and relative velocity need to be reported as well.

The perception of modern radar sensors and their sensor-interfaces can vary from raw detection point clouds, to on-sensor clustered and tracked object positions which are only available as object lists. Designed for the requirements of the present radar-based automotive applications, this interface type is adequate in industry to base functionalities on the detected object lists, e.g. for a collision avoidance. But modern functionalities for advanced signal processing demand raw data for an early data fusion of either multiple sensors or comparable sensor perception data.

In addition to the specific attributes, available radar data sets can differ in the specific mounting positions of the sensors or applied sensor assemblies, see Figure 3.1: For the Oxford data set with a single roof-mounted sensor (left), with five sensors of the nuScenes setup (center), or the NLOS data set with 4 sensors at the vehicle nose (right).

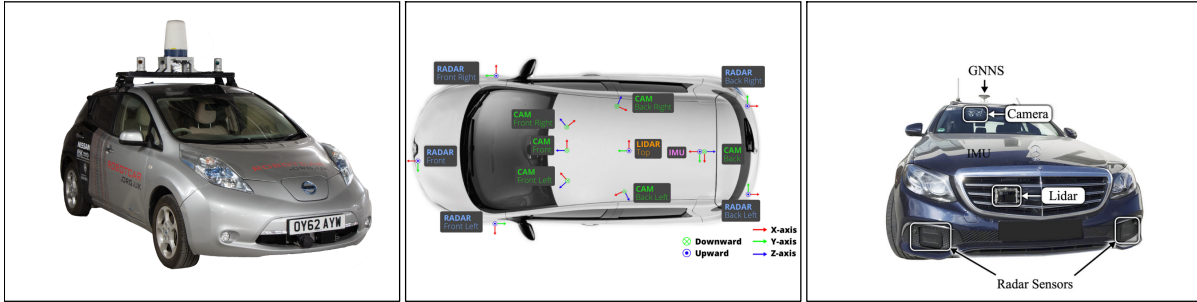


Figure 3.1: Radar Sensors of different radar sensor sets: Oxford data set [15] (left), nuScenes sensor set [30](center), and the NLOS sensor set [187](right).

Even other sensors are designed as rotating sensors [15], others are solid and smaller variants with different opening angle. In brief, the existing radar data sets are not fully comparable. Radar data is not sensor invariant as e.g. image data is.

Having radar data as 3D point cloud at hand, the labeling of radar data is a unprecedented challenge. The common radar sensor-interfaces allow three generic annotation strategies in ascending abstraction level, as Schumann et al. [194] suggest.







- *Frequency Spectrum* Annotation: As earliest possible label processing, during the calculation of detection peaks, the frequency domain can be used to classify signals. This data representation level is non-intuitive for human interpretation and a labeling approach on this level would result in complexity and require dedicated expert knowledge.
- *Point-wise* Annotation: On the sole level of 2D or 3D points, each single radar detection point is annotated with a describing label, defining its semantic information. This level of information is well human-readable. In combination with a 3D visualization, the recommended representation can be reviewed and interpreted in different 3D aspect angles, helping to understand the reflecting object shapes. In this representation stage, radar reflection points are called *detections*.
- *Object-level* Annotation: The relevant radar detections are further grouped and labeled as abstract group of detections, e.g. one infolding 3D bounding box per object, consisting of an arbitrary number of detections. This highest abstraction level is accompanied with inherent information loss, due to the abstraction to un-quantized object labels instead of detection labels.

Even on detection level, as human interpretable 3D detection point cloud, the precise and thoroughly sound manual labeling of the noisy cloud points itself is a laborous and consequently expensive process. To understand the radar signals and to determine their semantic label, still expert know-how is obligatory necessary and manual labeling is absolute.

Solely for this thesis, a quasi static radar data set was recorded of typical scenes for an automated parking scenario with varying environments and traffic conditions ¹. To enrich the recorded raw radar data with semantic labels by an efficient automated processing pipeline, an automated semantic labeling pipeline is developed to generate a point-wise semantic labeled radar data set.

Established state of the art semantic classes of Cityscapes [47] data set or the SemanticKITTI [18] data set are too detailed for radar data segmentation, see Table 3.2, and therefore consolidated to a radar applicable reduced label set $\hat{y}_{\text{sem}}(\mathbf{p}_{i,t}) \in \{\text{person } \blacksquare, \text{vehicle } \blacksquare, \text{building } \blacksquare, \text{vegetation } \blacksquare, \text{pole } \blacksquare, \text{artifact or unknown } \blacksquare\}$. This set of colors is consistently applied in this work to visualize the semantic class of a radar detection-without distinction of static or moving objects.

Table 3.2: Applied consolidation of the 22 SemanticKITTI [18] classes to six radar applicable classes.

| Radar Classes | points [%] | SemanticKITTI Classes |
|--|------------|---|
|  Vehicle | 8.39 | car, bicycle, motorcycle, truck, other-vehicle, bus |
|  Building | 7.68 | building, fence |
|  Vegetation | 4.80 | vegetation, trunk, terrain |
|  Poles | 0.60 | pole, traffic_sign, traffic_light |
|  Person | 0.18 | person, bicyclist, motorcyclist |
|  Artifacts | 78.35 | sky, road, parking, sidewalk, other-ground |

Designed as cross-sensor labeling framework, both common visual perception sensors of camera and LiDAR sensors are combined to systematically derive radar labels. In a two step procedure, first artifacts and noise is detected and labeled by the Radar Artifact Labeling Framework (RALF) [SI1], then the remaining plausible detections are processed in the second stage SeRALF [SI2] labeling step to specify point-wise the semantic radar labels. In combination with a labeling policy for subsequent manual correction, the point labels are improved to obtain a ground truth radar data set. Point-wise annotation with ground-truth semantic labels y allows to train point-wisely classifying semantic segmentation networks in supervised fashion, see Section 4.

In the supervised Master thesis of Marcel Schilling [MT2], the basis of the cross-sensor radar labeling is tested. The refined labeling concept is documented in the two conference papers, *Radar Artifact Labeling Framework (RALF)* [SI1] and *Annotating Automotive Radar efficiently: Semantic Radar Labeling Framework (SeRaLF)* [SI2].

¹ Varying weather is excluded for the generation of the data set, due to the unknown degradation of the automated labeling pipeline at harsh weather influences.

3.1 Motivation for Static Environment Radar Labeling

The point-wise semantic segmentation of radar points is addressed in this thesis to be applied in a quasi static scene environment with less moving objects. The test and data set scenes contain moving objects e.g. cars, or pedestrians, but mainly focuses on the static environment detection. Instead of only focusing on dynamic object detection and classification, as the exemplary RadarScenes [194] data set, the data set of this thesis is not intended to build a dynamic object detector. Instead, the aim is to provide a data set which can be applied for a more general application of classifying also static environment radar detection types and typical potential dynamic road users (vehicles and persons), excluding traffic scenes.

Based on the current proceedings in point cloud processing also of sparse and noisy point clouds, e.g. Rao et al. [171], Scheiner et al. [187], Schumann et al. [194], the point-wise annotation level is selected as most promising approach.

Commonly radar data is processed on three representation levels: Radar spectrum data represents the raw sensor information, which is applied in Hügler et al. [99, 100], Patel et al. [159]. In the second stage of radar signal processing significant peaks in the radar spectrum are detected and expressed as spatial 3D point clouds. This point cloud level is not very often utilized and labeled, as depicted in Table 3.5. Third level of radar data abstraction is formed by tracked point clouds or tracked objects. This level is commonly applied for ACC-assists [225] or other object level function. For further abstraction, the tracked objects are reduced to 2D occupancy grids. Occupancy grids are mainly applied for mapping or early SLAM applications [192, 129]

Beyond the detection of moving objects, this thesis aims to generate point-wise semantic information directly from the raw radar detection point cloud, independent of the object motion. With the aim to classify mainly a quasi static environment by as 3D radar point clouds scan, in direct fashion without further abstraction, the labeling of static radar detections is primarily addressed. Including potentially moving objects, e.g. vehicles and pedestrians, the approach aims to recognize these classes, independent if they are moving or static. The semantic segmentation is assumed to be perspective and situation variant, not depending on the relative velocity of an object.

Starting as data driven approach, the target application and use-case of an automated parking functionality defines further assumptions and decisions. Based on the assumption and vision to design a generalizing solution for a trained parking functionality in an arbitrary environment, e.g. on home ground, in first place, a marker-less mapping and localization approach seems inevitable. Private parking garages and home-ground do commonly not provide any sort of standardized markers or general landmarks which could be applied for radar mapping and localization. In order to provide a marker-less radar description of an arbitrary scene, the raw radar signals are assumed to deliver an accurate scene description

to apply a SLAM approach to map the scene. This radar mapping is aimed to be supported and enriched by a point-wise semantic radar segmentation. Having 3D radar raw data as semantic radar point clouds at hand during the mapping process, not only the determination of *noise* versus *plausible* can be enabled. Further, this thesis proves that the association of potentially matching structures can be confirmed by both their spatial occurrence, and additionally with respect to their semantic content. E.g. a sensed car might have a similar spatial occurrence as a bush or hedge in the same size, but could directly be distinguished by their semantic interpretation as either *vehicle* or *vegetation*. For more details of the matching and mapping process, see Section 5.

3.2 Evaluation and Selection of existing Approaches: Point-Wise Labeling

According to the overview in Table 3.1, there are three point-wise labeled data sets available: NLOS [187], RadarScenes [194] and our Porsche InnoCampus radar data set. Information on labeling of radar data on the before introduced *Frequency Spectrum* [159, 100] and *Object-level* [185, 147, 142, 143, 28] abstraction level is out of this thesis' focus.

The NLOS data set [187] is labeled based on a reference-positioning sensor and mainly focuses on the detection of moving objects from a clutter background. The work of Schumann [192] on radar segmentation is based on the RadarScenes data set [194] and comparable to the data at hand. The RadarScenes data set realizes point-wise labeling on subsequent, spatio-temporally accumulated radar point clouds of 100 *ms* windows. The purely manual labeling step requires ≈ 2 min per window, is supported by a visual inspection camera of the scene and supported by a visualization of different radar attributes. The authors describe their radar labeling to be dedicated for object detection and semantic segmentation of road users, but a separate distinction of static classes is not included.

To the best of the authors' knowledge, there is no automated labeling process, or an automatic pre-labeling support for a data set of sparse radar point clouds known. This task is discussed and inspired from the automated labeling of LiDAR point cloud data.

The general task to label a point cloud manually at scale requires vast amounts of human labor and effort. Since the 3-dimensional interpretation is necessary, occlusions can occur and the aspect angle is crucial, point cloud labeling effort and complexity exceeds the common camera image annotation by far. The advantage of modern LiDAR point clouds is their density, enabling a relatively quick and intuitive human understanding of the dense 3-dimensional scene. Problematic remains the interpretation of the sparse and noisy far-end reflections, to determine their true semantic class. The sensor sparsity and scattering of subsequent scans complicates the scene perception and labeling consistency drastically.

Exemplary automated semantic labeling procedures dedicated on LiDAR labeling apply a registered reference camera to transfer a high-quality image-based semantic segmentation to the point cloud [165]. Piwak [163] combines this automated label generation to generate a point-wise semantic LiDAR data set. Inspired from this automated labeling the following section extends the automated labeling complexity to LiDAR as second reference sensor to suggest semantic radar labels.

3.3 Data set Generation and Multi-modal Labeling Automation

Visual perception based on camera images and LiDAR point clouds are common and state of the art. As radar is often neglected in data sets, or only a sub-set of radar detections is only labeled manually [194], this thesis presents an automatic labeling process on image and LiDAR data, to yield novel semantic radar segmentation labels.

As generic method, radar detections can be interpreted and labeled by the means of common visual (LiDAR, camera) on-board sensors. Semantic labels of LiDAR point clouds Piewak [164], serve as one input for the cross-sensor labeling. The radar labeling strategy proposed in Figure 3.2, combines an independent processing of camera-based and LiDAR-based labels, to transfer labels from the visual perception. Along the elements of Figure 3.2, this section describes the developed labeling process.

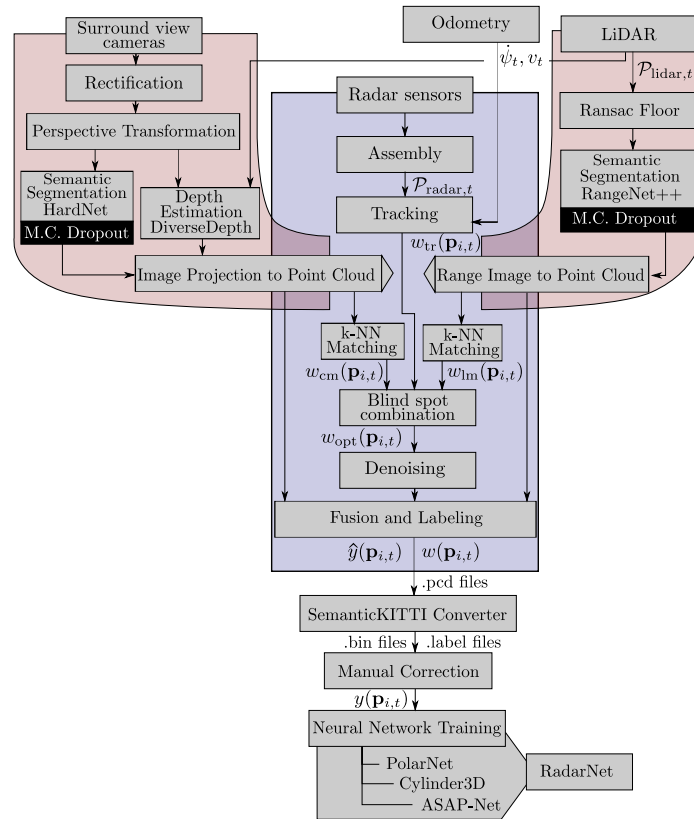


Figure 3.2: Labeling concept for the proposed automated semantic radar labeling based on reference LiDAR point clouds and camera images with subsequent data preparation steps for machine learning.

The task to label radar point can be seen as two subsequent steps which involve visual perception and spatio-temporal tracking. As first step, the determination of a radar point as *plausible* or *implausible* is performed, describing in the latter case *clutter* or *noise*. Details

are found in Section 3.3.3 for the image processing, in Section 3.3.2 for LiDAR. The spatio-temporal tracking of Section 3.3.4 produces tracking plausibility estimates which are altogether prioritized in a plausibility selection in Section 3.3.5. This overall plausibility labeling is described in an earlier publication as Radar Artifact Labeling Framework (RALF) [SI1]. As a result of this first processing step, *implausible* detections are neglected and only *plausible* detections remain.

In the second step, the "remaining" point cloud is further evaluated and split up into different semantic classes. This second semantic annotation step of the plausible detections $p_{i,t}$ by a semantic class label $y_{\text{sem}}(p_{i,t})$ is referred to as SeRaLF [SI2]. Its' concept is to process the visual perception of camera, see Section 3.3.7, and LiDAR, see Section 3.3.6, independently, each with a semantic segmentation CNN, and select from their matched semantic labels the best matching in Section 3.3.8.

In combination with a temporal signal analysis of the spatial occurrence of detections in consecutive frames, the detection of clutter and other noise detections is supported. Other than manual labeling, the proposed automated processing allows an efficient, unbiased, and consistent labeling pipeline. Consistent labeling of detection plausibility is the basis for further semantic labeling.

The labeling process can be considered as data enrichment, since the corresponding semantic class is stored as extra information, as additional attribute, to the original 3D point, similar to its other attributes. Section 3.3.9 describes a final refactoring of the data structures.

3.3.1 Multimodal Automatization Strategy

For each radar sensing cycle at time t , the sensed radar point cloud can be formalized as a feature tuple $\mathcal{P}_{\text{radar},t} = \{(\mathbf{p}_{r,1,t}, \mathbf{x}_{r,1,t}), \dots, (\mathbf{p}_{r,N,t}, \mathbf{x}_{r,N,t})\}$ of N radar detections at time t . As part of the feature vector $\mathbf{x}_{\text{radar},i,t}$, the spatial coordinates (x, y, z) describe the spatial distribution of the points. As additional feature components of $\mathbf{x}_{r,i,t}$, point attributes are stored in the feature tuple, namely Signal Power P_r , Signal to Noise Ratio SNR_r , etc. Point-wise semantic labeling adds for each radar detection $\mathbf{p}_{r,i,t} = \mathbf{p}_{r,i,t}$ a label attribute $w(\mathbf{p}_{r,i,t})$ to the feature list, namely plausibility $w(\mathbf{p}_{r,i,t}) \in [0, 1]$. The binary plausibility score $w(\mathbf{p}_{r,i,t})$ indicates if the detection is *plausible*, so representing an existing object ($y(\mathbf{p}_{r,i,t}) = 1$), or if it is rates as *implausible* and thereby rated as noise or artifact ($y(\mathbf{p}_{r,i,t}) = 0$).

The remainder of the section describes the modular perception steps of Figure 3.2 to realize the automated annotation with semantic labels. In a first stage, a plausibility score per radar detection is computed, Sections 3.3.3-3.3.5, then the *plausible* radar detections are introduced to the second stage. In this second stage further classification of *plausible* detections into semantic radar classes is performed, see Sections 3.3.7-3.3.8.

Instead of aiming to achieve an 100% accurate solution, an automatic labeling pipeline, which delivers the best possible but not necessary the optimal estimates of semantic labels for radar point clouds is developed. The estimates are inspected and corrected to ground truth quality in a subsequent obligatory final manual correction step. The sheer number of noisy and hardly interpret-able radar data, requires an initial guess to serve as starting level which subsequently can be further improved. Hence, it is a major labeling improvement to generate educated semantic label estimates in an automated, systematic and reliable procedure, instead of labeling from scratch.

3.3.2 LiDAR Plausibility Label Generation

LiDAR sensors also provide the sensed data in the form of point clouds $\mathcal{P}_{\text{lidar},t}$, but delivers only few additional point attributes besides the spatial location. Consequently, the neighborhood association between radar point cloud $\mathcal{P}_{\text{radar},t}$ and LiDAR point cloud $\mathcal{P}_{\text{lidar},t}$ can be solved in 3D space, based on a point-to-point distance measure. Depicted as k-NN block of the LiDAR-wing in Figure 3.2.

LiDAR sensors are commonly taken as reliable reference sensor for perception [240], localization [246, 42], since its range measurement and dense environment perception specifications is accurate and reliable. The high resolution and rule-of-thumb range-accuracy specification of circa 0.5 – 10 mm of laser scanners, qualifies LiDAR to be treated as ground-truth reference.

For the ideal hypothesis of overlapping radar and LiDAR reflections originating from the same objects, both point clouds can be compared by a distance measure per point. Without a relaxation of the matching distance, the matching does not yield reliable results due to differing sensor resolution and precision. Compared to LiDAR, radar signals are reflected not solely at the outer object shell of an object. Due to its' wave-length, radar reflections can occur from a partial or also total penetration of an object, reflecting at an occluded object. Whereas laser reflections occur on the direct line of sight at the first impinged surface.

With an approximately 10 times lower radar accuracy in range resolution, plausible radar detections occur in the *near* range of LiDAR reflections but must not necessarily match with those accurately. Further some assumptions are necessary for a comparison:

- No sensor degradation due to weather influences are allowed for a data set generation. In detail, no rain fog, dust or other visual degradation is allowed.
- Only planar scenes are applicable, excluding ramps, hilly environment or non-planar territories. The planar requirement allows a simplified extraction of floor-detections for floor-related radar noise detection.

Generally, automotive radar sensors should not detect reflections of the ground surface, since these reflections often occur below the general noise level of other objects. Common automotive radar sensors are specifically designed to detect significant static or dynamic objects. Ground plane radar detections can occur by chance, but are not reliable. In contrast, LiDAR reflections occur on roads and any sort of light reflecting planes also. These points are recommended to be excluded from the LiDAR point set, before LiDAR-radar point cloud comparison.

On the LiDAR point clouds, a RANSAC [66] ground estimation is performed to extract the ground plane LiDAR detections. With the ground-plane corresponding LiDAR point set, the radar point cloud is checked for radar detections in close proximity of the estimated plane. The corresponding radar detections are directly labeled as implausible $y(\mathbf{p}_{r,i,t}) = 0$.

The remaining *plausible* radar points and the ground-plane reduced LiDAR point set are compared to find corresponding neighbors. To find and identify matching radar detections, a k-Nearest-Neighbor (k-NN) clustering implementation [251, 178], which is denoted in Algorithm 1, is applied.

Algorithm 1 Proposed LiDAR matching by Isele et Al. [SI1].

Require: $\mathcal{P}_{\text{radar},t}, \mathcal{P}_{\text{lidar},t}$

Ensure: $w_{\text{lm}}(\mathbf{p}_{r,i,t})$

```

1: for  $i, t = 1, \dots, N_{\text{radar},t}$  do
2:    $\mathbf{q} \leftarrow \text{K-NN}(\mathcal{P}_{\text{lidar},t}, \mathbf{p}_{i,t}, K)$ 
3:    $d \leftarrow 0$ 
4:   for  $l = 1, \dots, K$  do
5:      $p_{x,l,t}, p_{y,l,t}, p_{z,l,t} \leftarrow \mathcal{P}_{\text{lidar},t}.\text{get\_point}(\mathbf{q}[l])$ 
6:      $r_{l,t}, \varphi_{l,t}, \vartheta_{l,t} \leftarrow \mathcal{P}_{\text{lidar},t}.\text{get\_features}(\mathbf{q}[l])$ 
7:      $\sigma_{d,i,l} \leftarrow \text{MODEL}(r_{r,i,t}, \varphi_{r,i,t}, \vartheta_{r,i,t}, r_{l,t}, \varphi_{l,t}, \vartheta_{l,t})$ 
8:      $d \leftarrow d + \sqrt{\frac{\Delta p_{x,t}^2 + \Delta p_{y,t}^2 + \Delta p_{z,t}^2}{\sigma_{d,i,l}^2 + \epsilon}}$ 
9:   end for
10:   $w_{\text{lm}}(\mathbf{p}_{r,i,t}) \leftarrow \exp(-\beta_{\text{lm}} \frac{d}{K})$ 
11: end for

```

By the k-NN clustering, for each radar detection $\mathbf{p}_{\text{radar},i,t}$ an associated set of K nearest neighbors of the LiDAR scan $\mathcal{P}_{\text{lidar},t}$ is found. Due to the higher resolution of LiDAR scans, the associated K LiDAR neighbors indicate for increasing $K > 1$, the robustness of a radar detection.

Radar $(r_{\text{radar},i}, \varphi_{\text{radar},i}, \vartheta_{\text{radar},i})$ and LiDAR detections $(r_{\text{lidar}}, \varphi_{\text{lidar}}, \vartheta_{\text{lidar}})$ are measured in sensor specific local sphere coordinates but are expressed in a standard vehicle reference coordinate system, see Section 2.7.

To express the uncertainty based on the L^2 norm d , and to propagate an error measure on the spatial coordinates of the radar and LiDAR detections, a measurement model is formulated in Cartesian coordinates:

$$\begin{aligned} h_x &= r_{\text{radar},i} \cos \vartheta_{\text{radar},i} \cos \varphi_{\text{radar},i} + {}^v x_{\text{radar}} - (r_{\text{lidar}} \cos \vartheta_{\text{lidar}} \cos \varphi_{\text{lidar}} + {}^v x_{\text{lidar}}), \\ h_y &= r_{\text{radar},i} \cos \vartheta_{\text{radar},i} \sin \varphi_{\text{radar},i} + {}^v y_{\text{radar}} - (r_{\text{lidar}} \cos \vartheta_{\text{lidar}} \sin \varphi_{\text{lidar}} + {}^v y_{\text{lidar}}), \\ h_z &= r_{\text{radar},i} \sin \vartheta_{\text{radar},i} + {}^v z_{\text{radar}} - (r_{\text{lidar}} \sin \vartheta_{\text{lidar}} + {}^v z_{\text{lidar}}). \end{aligned} \quad (3.1)$$

The assumption of uncorrelated and independent measurement accuracy of different sensors allows to formulate an error propagation model in Equation 3.2. Radar detection coordinate uncertainties are denoted $\sigma_{r,\text{radar}}, \sigma_{\varphi,\text{radar}}, \sigma_{\vartheta,\text{radar}}$, LiDAR Time-of-Flight uncertainty is denoted as $\sigma_{r,\text{lidar}}$.

$$\sigma_{d,i,l}^2 = \left(\frac{\partial d}{\partial r_{r,i}} \right)^2 \sigma_{r,\text{radar}}^2 + \left(\frac{\partial d}{\partial \varphi_{r,i}} \right)^2 \sigma_{\varphi,\text{radar}}^2 + \left(\frac{\partial d}{\partial \vartheta_{r,i}} \right)^2 \sigma_{\vartheta,\text{radar}}^2 + \left(\frac{\partial d}{\partial r_l} \right)^2 \sigma_{r,\text{lidar}}^2 \quad (3.2)$$

In Algorithm 1 denoted as MODEL, this error propagation of error in Equation 3.2 relates the obtained neighbor distance to its uncertainty measure. The partial derivatives of the euclidean L^2 distance d between corresponding neighbors, with respect to the radar range r , azimuth angle φ , elevation angle ϑ , and the LiDAR range allows to relate the neighbor distance to the uncertainty measure. By this, the k-NN matching and distance measure incorporates the sensor specific accuracies to find corresponding LiDAR radar pair.

The modeling of plausibility $y(\mathbf{p}_{r,i,t})$ of radar detections based on the distance to neighboring LiDAR points is reached by an exponential decay function for increasing distance d .

$$\begin{aligned} \mathbf{p}_{\text{radar},i,t} &= w_{\text{lm}}(\mathbf{p}_{\text{radar},i,t}) \\ &= \exp(-\beta_{\text{lm}} \frac{d}{K}) \end{aligned} \quad (3.3)$$

First to notice, a higher number K of neighbors increases plausibility score. At low distances d , close to an intersection of LiDAR and radar detections, the plausibility score is high but decreases by the exponential decay factor $\beta_{\text{lm}} \in \mathbb{R}^+$ with increasing k-NN matching distances d .

3.3.3 Camera Plausibility Label Generation

The processing steps are depicted in the Camera-wing of Figure 3.2. To integrate the environment perception of the series-vehicle fish-eye cameras, the raw camera images are un-distorted, perspective transformed and processed with a depth-estimation CNN, allowing to express the camera image as 3D point cloud, see Figure 3.3. A depth estimating CNN delivers a relative depth image or depth map as output. This depth map needs to be re-scaled to a metric depth map by LiDAR sampling points as reference. Finally, a continuous camera-based depth estimation can be compared to the radar point cloud by applying a k-NN clustering. Similar to the procedure for the LiDAR plausibility check in Algorithm 1, also a L^2 Euclidean distance measure is applied to indicate radar detections in larger distance of the depth estimation as *implausible* outliers or *noise*. Depicted as k-NN block of the Camera-wing in Figure 3.2, here the parameter β_{cm} replaces the LiDAR specific β_{lm} .

Hence, as secondary visual sensor, the series cameras of the vehicle are applied to determine *plausible* from *implausible* radar detections. The application of series camera sensors is chosen to showcase the rich capability of already integrated vehicle sensors and reduce the effort of a physical integration of alternative reference cameras. But obviously, to successfully apply pre-trained (open-source) image processing modules, to the real-worlds image data, the lower the domain shift between data-set and raw fish-eye sensor-image, the better performance can be achieved. Open-source works and data sets for fish-eye image processing as Woodscapes [238] or the artificial CARLA Simulator [57] generated data sets exist, but most existing image works are based on regular rectified, horizontally pointing camera data sets. In this thesis, it is aimed to integrate off-the-shelf solutions, originally designed for rectified images.

The cameras applied for series vehicles deliver a top-view perspective of the vehicle periphery as auxiliary driver information and proximity surveillance. Their downwards FOV covers parts of the ego-vehicle and parts of the close environment, compare Figure 3.3.

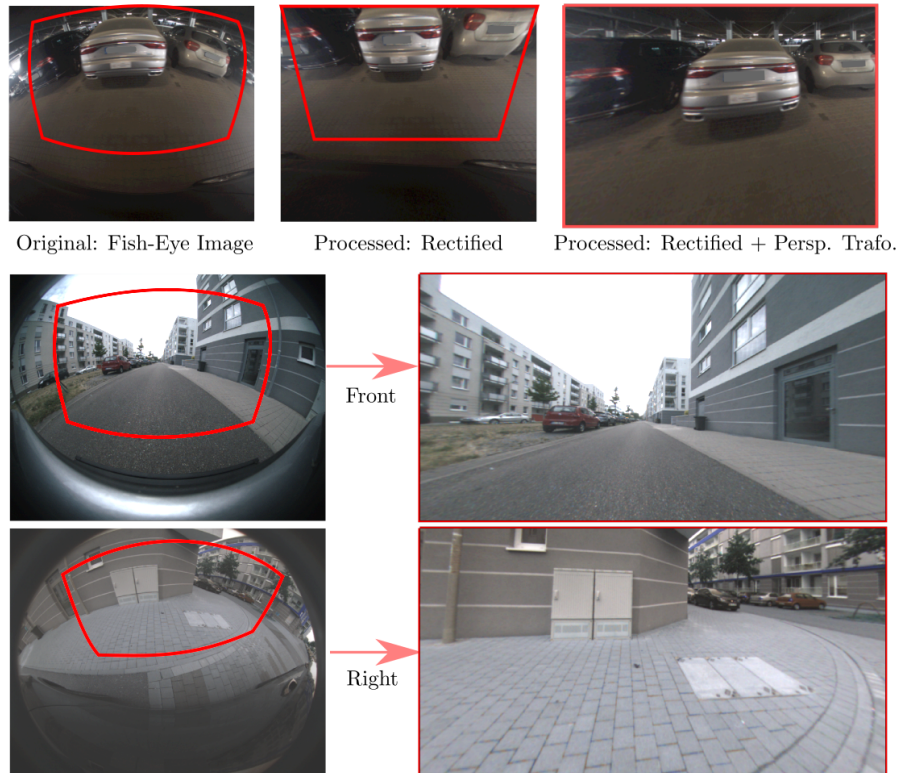


Figure 3.3: Illustration of the top-view camera image processing steps. Raw fish-eye images (left), rectified (center) and perspective transformed image (right), with exemplary frame (red) of the effective image region. Camera images from [MT2], figure modified.

Besides the informative top-view scene surveillance, the raw images of the mono cameras deliver fish-eye images of the environment. For the driver assistance functionality of the top-view projection, only fractions of the fish-eye images is used to project the top-view. The image range of visually detected space around the vehicle, covers large parts of the environment, but heavily distorted by the fish-eye optics.

After pre-processing the fish-eye image as illustrated in Figure 3.3, the processed image resembles a common regular, rectified camera image view, with a horizontally pointing camera perspective - not anymore ground-oriented as the initial raw image. In order to process this pre-processed image as if it originates of a regular camera, the original fish-eye camera-matrix $\mathbf{A}_{\text{fish-eye}} \in \mathbb{R}^{3 \times 3}$ was modified to be available for the rectified and perspective transformed image $\mathbf{A}_{\mathbf{t}} \in \mathbb{R}^{3 \times 3}$. For the un-distortion, the camera modeling of OcamCalib of Scaramuzza et al. [182] is applied.

With the knowledge of the camera intrinsics parameters $\mathbf{A}_{\mathbf{t}}$ and their mounting position as extrinsic parameters $\mathbf{E}_{\mathbf{t}}$, the 3D radar detections $\mathbf{p}_{r,i,t}$ can be projected onto the image plane. Similar to the previous section, the range information from the visual reference sensor, now camera-based, is taken as measure to determine closeby *plausible* radar detections from *noisy* radar detections. For a comparison of mono-camera depth perception, described as

image-based pixel maps, with a 3D radar detection point cloud, a continuous image pixel representation of the estimated depth needs to be processed of the same metric scale.

The reconstruction of a scene from mono camera images is the research field of Structure-from-Motion (SfM) approaches, of e.g. Schönberger and Frahm [190]. Based on extracted image features in subsequent frames, an estimated relative motion of the camera can be detected. Based on this motion estimation, a scenes' features can be arranged in 3D space, according to their reconstructed location of the 3D scene perception. In low speed scenarios, e.g. in narrow parking situations, but also for feature-less camera image perspectives, e.g. close to a feature-poor wall, the environment reconstruction quality degrades due to close perspective and less context, potentially yielding failure and SfM results in a warped or erroneous environment reconstruction, according to own experimental findings. In general, the reconstructed scene only consists of the 3D arranged image features, which might not necessarily correspond with radar detections. Consequently, the comparison of radar detections with the reconstructed scene anchor points is impractical.

As alternative scene recognition, a continuous depth estimation can directly be processed of an input image by a pre-trained depth-estimation CNN. For the generalization of the depth estimation CNNs, especially for other views or other applications, the training data set limits the transferability. Whereas most automotive image data sets exclusively contain a single front facing camera perspective view, every network solely trained on this perspective degrades substantially for the application on imaged of other view directions.

Figure 3.4 illustrates the four camera perspectives of the test vehicle. Each perspective, front-facing, left-, right- and back-facing cameras are integrated and need to be processed to generate a depth-estimation image.

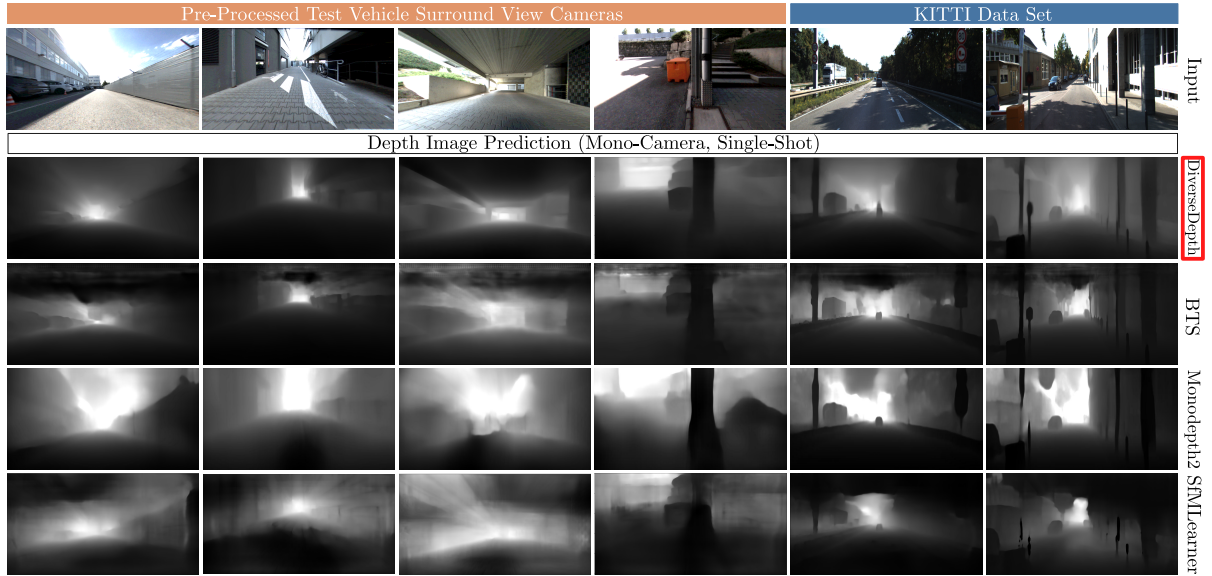


Figure 3.4: Illustration of rectified top-view camera images and exemplary KITTI Benchmark[71] images, applied for depth prediction with a selection of mono-camera depth estimation CNNs [120, 75, 217, 237]. Estimated relative depth difference in the images are decoded as brightness. Results of [MT2], figure modified.

To determine the best suitable, open-source available mono-image depth estimation CNN, a couple of tests are performed to evaluate their performance on real camera images of the vehicle setup. Figure 3.4 illustrates the advantage of a diverse dataset over the common front facing views: The exemplary four mono-camera depth estimation CNNs, SfM-Net [217], Monodepth2 [75], BTS [120] and DiverseDepth [237] trained on common data sets of deep-learning tasks in an automotive context e.g. KITTI [71] are outperformed by a general purpose depth estimation CNN DiverseDepth [237], measured visually by detail-richer and sharper depth-maps in all four top-view camera perspectives. Even if the independent front-view results might be promising, side-view depth maps under-perform. The objective to apply the same depth estimation CNN on all four cameras, in order to consistently estimate depth values for all views, yields DiverseDepth [237] as best performing CNN.

With the consistent and dense depth map representation for all four mono cameras of the test-vehicle, the relative depth maps need to be scaled to true, real-world metric scale. Since LiDAR is considered in the thesis as delivering ground truth range information, the LiDAR point cloud is referencing a scaling factor s_{lidar} . To derive the scaling factor s_{lidar} , again point cloud projection is applied. The LiDAR points are projected onto the depth estimation image.

In the overlapping sensor FoV regions of cameras and LiDAR, the LiDAR points with their metric range attribute r_{lidar} , can directly be compared to the image-based depth value $r_{\text{cam,depth}}$. By this comparison, the LiDAR points serve as sparse scaling factor sampling points of the metric real-world depth.

Experimental results yield multiple local scaling factors $s_{loc, lidar}$ to outperform a single global scaling factor s_{lidar} per depth map, as a systematic cause of the decreasing resolution towards image edges of rectified and perspective transformed fish-eye images. Robust parameter estimation by a RANSAC estimation [66] is applied to find the best suitable global scaling factor s_{lidar} . Local scaling factors $s_{loc, lidar}$ allow to compensate remaining fish-eye artifacts. The local scaling factor procedure yields more accurate metric-scaled depth maps.

Due to the ordered structure of the LiDAR point cloud, the projection of LiDAR points into the depth image results in equidistant projection locations. Applying an ordered distribution of sampling points, an equidistant grid of depth pixel locations is selected to find for each of these image pixels the closest LiDAR neighbor point. The closest LiDAR neighbors are found by common k-NN clustering.

The local image depth map is locally rescaled to the near LiDAR projection depth value, as Equation 3.4 describes

$$d_{lidar} = d_{scaled} = s_{loc, lidar} \cdot d_{depth\ map} \quad (3.4)$$

Applying the local scaling factors, the depth estimation map is ready to project the camera image into a 3D pixel point cloud. In this representation as projected 3D pixel point cloud, the camera depth map results in a dense 3D point cloud, similar to the LiDAR point cloud. Hence, the radar detections can be compared with this camera-based 3D point cloud to determine camera-based, which radar detection is *plausible* and which radar detection is rated *implausible*. Following the explanation in Section 3.3.2, Algorithm 1 is analogously applied, but with the error propagation MODEL in Cartesian coordinates expressed for the depth estimation uncertainty.

3.3.4 Spatio-Temporal Radar Detection Tracking

With an approximate cycle time of 50 *ms*, subsequent radar scans are sensed. From the analysis of ego-motion compensated radar point clouds, the context inspection of subsequent independent measurements allows to identify *noise*. Accumulating independent measurements of a moving vehicle by the motion-compensated measurement positions, it is found, that noisy radar detections can be identified as outliers of the resulting spatio-temporally accumulated point cloud. The exemplary spatio-temporal accumulation of 3 subsequent raw, unfiltered radar scans is applied in this thesis and illustrated in Figure 3.5. This process is depicted as Assembly and Tracking blocks in Figure 3.2.

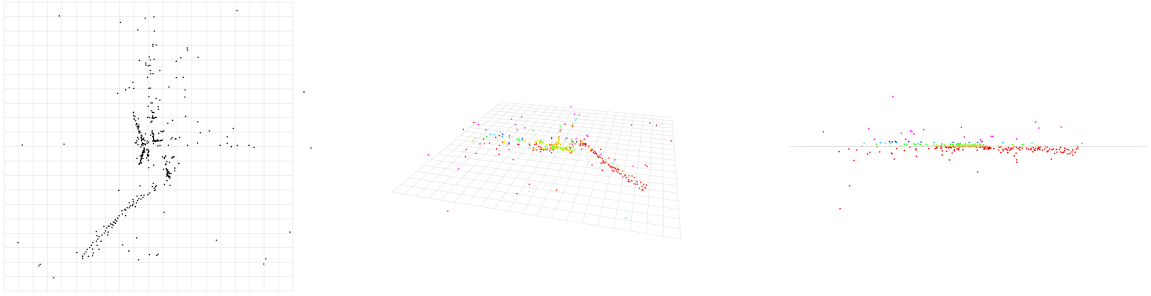


Figure 3.5: Illustration of 3 accumulated, subsequent 360° radar point cloud assemblies (3 sensing cycles), covering approximately 150 ms and a 45x45m grid: Top-view perspective (left), side view (middle) and horizontal view (left) are colored according to the points' z -coordinate $\in [-0.5m, 5m]$. Please note the interpretation difficulty of even accumulated radar scans.

In contrast, *plausible* radar reflections accumulate at the same areas, resulting in densely populated regions. This thesis assumes *clutter* and *noise* to occur mostly as randomly distributed outliers, but *plausible* detections are assumed to be modeled as Poisson Noise [28].

As Figure 3.6 illustrates, an accumulated inspection over multiple measurement cycles (approximately 100 scans accumulated in standstill) helps to identify noisy real detections from clutter and multi-path reflections. Based on the spatial re-occurrence of radar detections, a discrimination can be formulated.

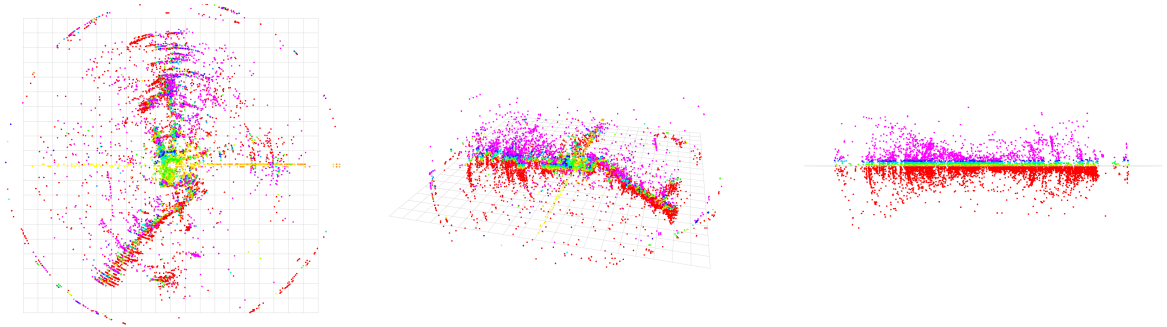


Figure 3.6: Illustration of approximately 100 accumulated clouds on a 45x45m grid. Colors according to z -coordinate $\in [-0.5, 5]$: Top-view (left), side-view (middle), and horizontal-view (right).

In the pipeline of the data set generation, labeling nevertheless is a post-processing step. Hence, at each inspection of a radar point cloud $\mathcal{P}_{\text{radar}, t_k}$ of timestamp t_k , its previous point cloud $\mathcal{P}_{\text{radar}, t_{k-1}}$ at t_{k-1} and following point clouds $\mathcal{P}_{\text{radar}, t_{k+1}}$ at t_{k+1} can be included for the inspection. This idea, to check the spatial re-occurrence, specially accounts for static environment reflections, whereas dynamic objects might not hold the assumptions. Instead, dynamic moving objects occur in the accumulated point cloud as spread clusters and might be treated thereof as *implausible* if their location changes substantially over subsequent radar measurement cycles. Since the focus of this labeling is on a labeling of the quasi-static environment, the effect of mistreating dynamic objects is neglectable for the automated

labeling. The manual correction step allows to correct potential shortcomings on spread dynamic object clusters to be labeled correctly in a frame-by-frame inspection.

For the point accumulation not only a single leading and subsequent timestamps is compared, but a set of $n_b \in \mathbb{N}$ sensor cycles earlier and later than the reference timestamp t_k is considered. For the inspection of a driving scene, the spatial accumulation needs to be ego-motion compensated. In this thesis, the relevant scenes cover low or medium speed, driving in planar scenes without much maneuvering at standstill, allowing to apply a kinematic single-track motion model based on wheel odometry [223].

The motion model gets ego-vehicle sensor data as input. The ego yaw-rate $\dot{\psi}$ measurements from on-board vehicle sensors, as well as the longitudinal vehicle velocity v , are available from the vehicle CAN network. In order to integrate the velocity during the radar measurement cycle, e.g. k to $k+1$, the motion model requires the time stamp difference Δt of the CAN messages.

$$\begin{bmatrix} x_v \\ y_v \\ z_v \\ \psi \end{bmatrix}_{k+1} = \begin{bmatrix} x_v \\ y_v \\ z_v \\ \psi \end{bmatrix}_k + \Delta t \begin{bmatrix} v \cos \psi \\ v \sin \psi \\ 0 \\ \dot{\psi} \end{bmatrix}_k \quad (3.5)$$

Assuming a planar motion, neglecting roll and pitch movement, the rotational vehicle motion can be expressed as yaw rotation matrix $R_z(\psi)$ with rotation angle ψ .

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

The resulting ego-motion compensation for each accumulated radar point is given as

$$\begin{aligned} \tilde{\mathbf{P}}_{\text{radar},i,t_{k+j}} &= R_{z,\psi_k}^{-1} \left((x_v, y_v, z_v)_{k+j}^\top - (x_v, y_v, z_v)_k^\top \right) \\ &+ R_{z,\psi_{k+j}-\psi_k} \mathbf{P}_{r,i,t_{k+j}}. \end{aligned} \quad (3.7)$$

So that for the batch of the additional $(2 \cdot n_b)$ radar point clouds $\mathcal{P}_{\text{radar},t_{k+j}}$ for every $j \in \{-n_b, \dots, -1, 1, \dots, n_b\}$, every radar detection can be spatially expressed with respect to the reference point cloud at time t_k .

The tuning of the parameter n_b , should be selected in regard of the scene to be tested. As mentioned before, in quasi static environments a larger n_b can be selected. For dynamic

object detection, the accumulation batch size should be decreased. Otherwise, moving objects do not result in denser regions, indicating *plausible* detections, but produce tail-shaped sparsely distributed clusters per object which are difficult to interpret and require frame-by-frame inspection. Inspecting multiple consecutive frames, only moving dynamic objects produce tail-shaped accumulations along their motion path. This view enables to determine and follow the global motion path of moving objects along these tails.

As found by Schumann [192], to detect or label moving objects in a time accumulated representation, the Doppler-Velocity of the detections needs to be considered in Equation 3.7 and visualized for the detection plot as well. Schumann [192] applies a visualization of the dynamic moving direction to consistently identify detections of moving objects.

Taking the error propagation in the spatial radar measurement equations into account, see Equation 3.2, the evaluation of the accumulated point clouds results in a batch of distance measures d_j per point.

A simple strategy to combine the distance measures for each point would be averaging. In edge-cases of short radar sequences of only a few sensing cycles, if a low parameter n_b is selected, potentially relevant detections might remain undetected in the environment of noisy detections. Instead, sorting the n_b distances d_j in ascending order is performed. As the closest distances contributes with a higher probability to the interpretation of a radar detection to be *plausible*, an exponential decaying weighting factor β_{tr} is applied on the ascending distance list $y(\mathbf{p}_{\text{radar},i,t}) = \exp(-\beta_{tr} * d_j)$. As a result, rated with a larger weight, accumulated radar detections in close distance to the reference scan, increase the *plausibility* score $y(\mathbf{p}_{\text{radar},i,t})$ of each radar point. Accumulated points with neighbors only at a higher distance get automatically de-rated towards an *unplausible* score $y(\mathbf{p}_{\text{radar},i,t}) = 0$.

3.3.5 Plausibility Label Selection

In the previous sections, the three parallel modules have been introduced: First the **LiDAR branch**, Second a **camera branch**, Third a **spatio-temporal tracking module**, as depicted in Figure 3.2. These three modules are combined to yield a single plausibility label $y(\mathbf{p}_{\text{radar},i,t})$ in the blocks *Blind Spot Combination* and *Denoising* of Figure 3.2. In the first step, the plausibility label of both camera and LiDAR perception branch, is selected based on the sensor FoVs. The resulting perception-based plausibility scores $w_{lm}(\mathbf{p}_{\text{radar},i,t})$ and $w_{cm}(\mathbf{p}_{\text{radar},i,t})$ are then compared to a spatio-temporal tracking module value $w_{tr}(\mathbf{p}_{\text{radar},i,t})$.

Field of View Sensor Selection: For the sensor FoV based plausibility selection, a selection scheme is applied. The sensor mounting position and each specific sensor FoV is respected in the label fusion process. For areas of only single sensor coverage, the available plausibility labels are applied directly. E.g. the blind spot area of the LiDAR $V_{\text{bs, lidar}}$, see grey areas in Figure 3.7, yields only camera and tracking plausibility as input.

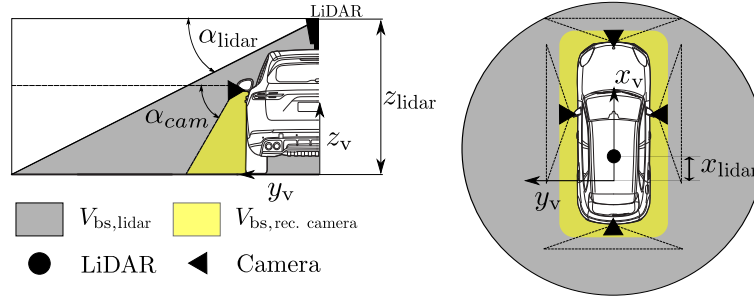


Figure 3.7: Exemplary LiDAR and camera blind spot regions from roof mounted position and rectification.

As Figure 3.7 illustrates for a schematic mounting positions ($x_{lidar} > 0, y_{lidar} = 0, z_{lidar} > 0$), the near proximity of the car is not covered by the roof mounted LiDAR reflections, depicted as grey area. The floor can not be detected in a straight line of sight with the schematic LiDAR opening angle of $\alpha_{lidar} > 0$.

$$V_{bs,lidar} = \left\{ \mathbf{p}_i = (p_{radar,x,i}, p_{radar,y,i}, p_{radar,z,i})^T \in \mathbb{R}^3 \mid p_{radar,z,i} \in [0, z_{lidar}] \wedge \sqrt{(p_{radar,x,i} - x_{lidar})^2 + p_{radar,y,i}^2} \leq \frac{z_{lidar} - p_{radar,z,i}}{\tan \alpha_{lidar}} \right\} \quad (3.8)$$

Close to the car only the camera plausibility is available, so the camera-based rating is applied, as Equation 3.9 describes, the subscripts cm, lm denote "camera matching" and "lidar matching".

$$w_{opt}(\mathbf{p}_{radar,i,t}) = \begin{cases} w_{cm}(\mathbf{p}_{radar,i,t}) & \text{if } \mathbf{p}_{radar,i,t} \in V_{bs,lidar} \\ w_{lm}(\mathbf{p}_{radar,i,t}) & \text{otherwise.} \end{cases} \quad (3.9)$$

Equation 3.9 describes further, that at the far range LiDAR plausibility is preferred over the camera rating. The far range of an image-based depth estimation degrades in accuracy with increasing range, the LiDAR based plausibility rating remains the reliable source of radar detection plausibility. For intermediate ranges with overlapping sensor FoV's camera and LiDAR based ratings are available, but LiDAR plausibility is experimentally found superior. In brief, cameras deliver the plausibility rating in close vehicle proximity, excluded the very close yellow region of Figure 3.7.

In the overlapping area, the result of both parallel independent visual processing branches (camera and LiDAR) are compared and checked for congruent plausibility labels. Both independent branches ideally deliver the same plausibility rating for each radar point cloud. Fusing both channels' ratings in case of different ratings, was experimentally found to degrade the overall accuracy of the plausibility labeling. It was found, based on the tested image depth estimation CNNs, that wherever a LiDAR rating is available, it is trustworthy to rely on LiDAR. In perspective of further developments of mono-depth estimation networks, this

statement could be relaxed. But for the tested setup, especially towards the image edges, depth estimation CNNs degrades and the statement to prefer LiDAR is found valid.

Finally, it should be noted that relying on LiDAR or camera data, processed by independent CNNs, their potential failure and potential erroneous results need to be detected. By checking for inconsistent ratings for the overlapping sensor FoVs, or significant deviations of radar ratings in the overlapping range, failure modes can be detected and flagged for later manual inspection.

Visual Perception and Spatio-Temporal Tracking Selection: A method to decide, if the visual perception or the tracking module needs to be prioritized to determine the plausibility of a radar detection, is applied.

For the radar sensors of the vehicle, their specific mounting position in covered packaging positions behind the bumper shell, is responsible for specific deflections and multi-path reflections. The denominator and sensor reliability factor $\gamma_s(\varphi_{\text{radar},i,t})$ in Equation 3.10 models the sensor accuracy for each radar sensor, exemplary illustrated in Figure 3.8.

$$\begin{aligned}\hat{y}(\mathbf{p}_{\text{radar},i,t}) &= H(w(\mathbf{p}_{\text{radar},i,t}) - w_0) \\ &= H\left(\frac{\alpha \cdot w_{\text{opt}}(\mathbf{p}_{\text{radar},i,t}) + (1 - \alpha) w_{\text{tr}}(\mathbf{p}_{\text{radar},i,t})}{\gamma_s(\varphi_{\text{radar},i,t})} - w_0\right),\end{aligned}\quad (3.10)$$

The sensor reliability curve of Figure 3.8, is uniquely available for every single radar sensor. Due to the specific mounting position and structural effects of the close-by packaging, azimuthal sectors of potential radar inaccuracies can be seen as peaks, hence increasing the denominator of Equation 3.10, promoting a radar detection to be rated as *implausible*. For every radar detection, Equation 3.10 outputs a binary final artifact labeling ($y = 0$) or a plausible detection artifacts ($y = 1$).

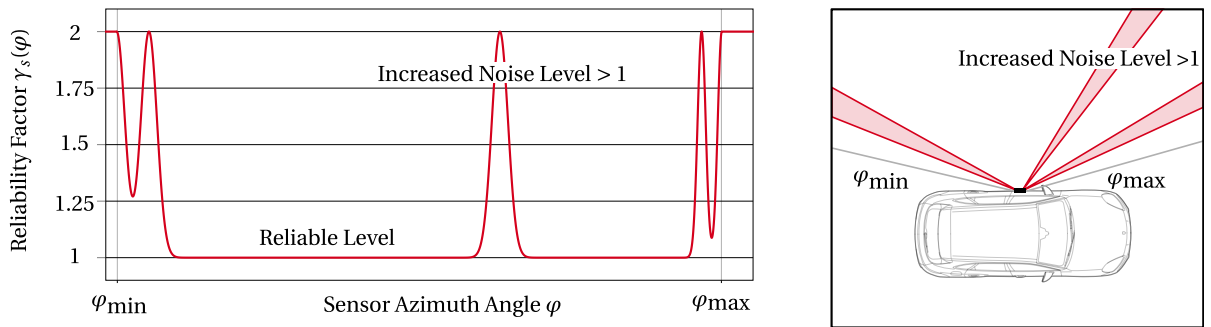


Figure 3.8: Exemplary radar reliability factor γ_s over the azimuth angle φ with affected sensor regions (right).

This decision formulation prioritizes between tracking and perception result and can be tuned by the weighting parameter $\alpha \in [0, 1]$. The selection is formalized in Equation 3.10 as a parameterized Heaviside function $H: \mathbb{R} \rightarrow \{0, 1\}$. The parametrization is reached by the term $w(\mathbf{p}_{\text{radar},i,t})$, understanding $w_0 \in [0, 1]$ as a threshold value.

The flexible parametrization of α in Equation 3.10 balances the selection of which the visual branch a plausibility label is preferred. With $\alpha = 1$, radar detections are only plausible if supported by a nearby visual detection from LiDAR or camera. With this parametrization, the influence of the tracking branch is respected but only the visual rating remains active.

The other extreme, $\alpha = 0$ suppresses the visually generated plausibility ratings, resulting in an tracking-only plausibility rating. Such a filter helps to detect static detections which are occluded by foreground objects. E.g. the radar reflections behind a vehicle could be detected as static detections of a building for example.

With this parametrization possibility, the case of a quasi-static scene labeling is addressed. For a quasi static scene, the tracking of stable and stationary objects is essential. In a localization context, the radar signature is an advantageous feature compared to visual localization. While visual sensors perceive only the outer shell of objects, radar sensors generate a radar signature also from the inner structures of a detected object or building.

With $\alpha = 1$, the radar signature would be cut to only return the reflections of the outer object shell.

The described plausibility rating based on visual perception and tracking can be fine-tuned by the parameter set $\alpha, \beta_{\text{lm}}, \beta_{\text{cm}}, \beta_{\text{tr}}, n_{\text{b}}$ and K .

Based on experimental results and manual tuning, the following values have been applied:

Table 3.3: Parameter set applied for the labeling pipeline.

| α | β_{lm} | β_{cm} | β_{tr} | n_{b} | K |
|----------|---------------------|---------------------|---------------------|----------------|-----|
| 0.8 | 0.7 | 0.009 | 2.3 | 3 | 2 |

For a large scale automated labeling, the labeling evaluation metrics help to fine-tune the process parameters in Table 3.3. With a scene specific, or even labeling policy specific parameter set, the outlined plausibility labeling procedure can detect relevant from noisy radar detections.

In the next steps, the remaining *plausible* detections are subject to further labeling. Qualified as *plausible*, subsequent processing steps assign a specific semantic class to each radar detection to further describe its semantic meaning.

3.3.6 LiDAR Semantic Label Generation

Jumping back to the LiDAR branch of Figure 3.2, the semantic label generation is added. The LiDAR point cloud of the reference sensor is processed with a pre-trained CNN to achieve semantic labels, depicted in the process chart of Figure 3.9. The pre-trained CNN RangeNet++ [145] is trained to semantically segment the LiDAR point of the SemanticKITTI [18] automotive scene data set. The generalization of RangeNet++ for similar driving scenes of this thesis is assumed.

Converting the LiDAR point cloud $\mathcal{P}_{\text{lidar},t}$ into a range image $\mathcal{I}_{\text{lidar},t}$, the organized point cloud is expressed as image data. Advantageous for such a representation is the common property of LiDAR to deliver the same number of points in an ordered consistent fashion. Based on this ordered structure, each single point can be converted to a specific image pixel.

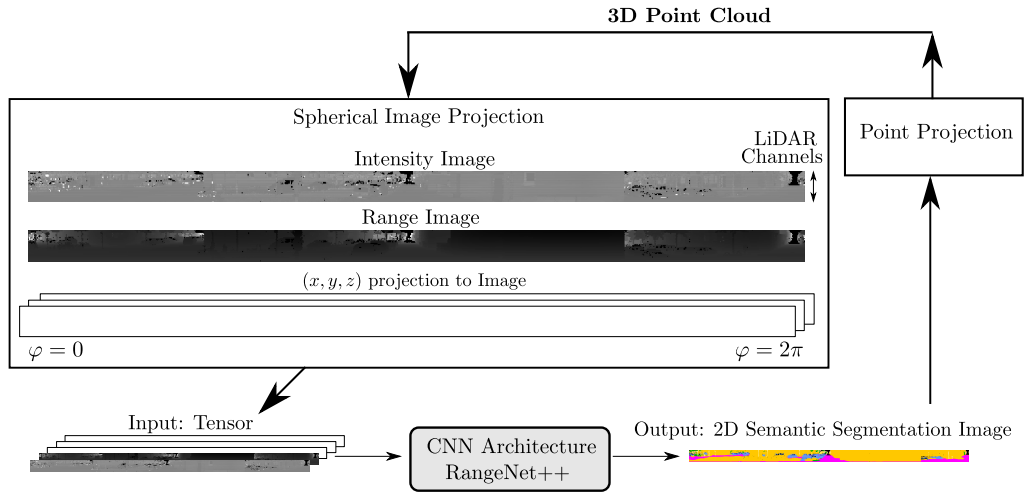


Figure 3.9: Exemplary LiDAR semantic segmentation pipeline. Data conversion to a range-image and inferred semantic classes on the range image.

The CNN inference step delivers semantic labels for each image pixel, which gets remapped to the corresponding point in 3D coordinates. Hence, each LiDAR point gets annotated by a semantic label $\hat{y}_{\text{sem},\text{lidar}}(\mathbf{p}_{\text{radar},i,t})$.

The remapping into 3D coordinates is skipped for the semantic radar point cloud labeling. Instead, the intermediate semantic range image representation of the CNN output is used as semantic label map, onto which the radar point cloud is projected.

Based on the sensor extrinsics, the same mapping from 3D points to the 2D image plane is applied for the *plausible* rated radar detection points.

With the sensor FoV and extrinsic parameters for both sensor types, first the FoV comparison is performed. *Plausible* radar detections in the LiDAR FoV, are considered only. The remaining radar detections, e.g. close to the car in the LiDAR blind spot of Figure 3.7, or radar

detections that are labeled as *im-plausible*, are generally assigned with the semantic label unknown. The semantic label unknown, depicted by the color in further semantic point cloud illustrations, is applied e.g. for background radar detections that occur behind other e.g. LiDAR matched *plausible* radar detections with $y(\mathbf{p}_{\text{radar},i,t}) = 1$. Hence, the background radar detection is not visible or occluded for the visual sensor and cross-sensor labeling is infeasible for these radar points.

For the projected radar points in the FoV of the LiDAR, the image location onto which the radar detection is projected, delivers a label mapping from the segmented LiDAR range-image to each single radar detection.

3.3.7 Camera Semantic Label Generation



The pre-processing of the fish-eye camera in the Camera branch of Figure 3.2 delivers an un-distorted, rectified and perspective transformed image of the scene. The semantic image segmentation in the Camera branch of Figure 3.2, is implemented parallel to the depth-estimation module. The semantic image segmentation of the pre-processed image is performed by the open-source available CNN HarDNet [35]. The semantic labeling process is not limited to this specific network architecture, any other semantic segmentation tool can easily be integrated. The comparison and consistency of the inferred semantic image labels with the semantic labels of the LiDAR semantic segmentation CNN is important. Since most semantic segmentation tasks include the 19 classes of the KITTI [71] data set, the only requirement to apply an other segmentation tool is to deliver the same semantic classes. Any network which delivers a suited semantic segmentation image can be applied alternatively.

In order to reduce the domain shift of this image based semantic segmentation task, a pre-trained CNN on typical automotive scenes is selected. Being trained on the Cityscapes data set [47], the HarDNet [35] CNN generalizes well on the pre-processed camera images for the application in parking scenarios and on-street scenes.

The semantic segmented image pixels serve as reference for the radar point labels. With respect to the camera mounting position as extrinsic parameter and the intrinsic camera matrix of the transformed perspective $\mathbf{A}_t \in \mathbb{R}^{3 \times 3}$, a projection of radar points onto the image is applied. Depending to which FoV of the cameras each radar point $\mathbf{p}_{\text{radar},i,t}$ suits best, the radar point gets projected onto the camera image. To which camera each point is projected is decided by a distance measure of the projected point to the corresponding image center. The closest distance to the image center is selected as projection target image. This voting takes care, that radar detections in the overlapping camera images are projected to the better available image of both camera perspectives. It is important to mention, that the un-distorted

images might still contain a little distortion towards the image borders, so a radar point projection onto image margins, potentially might include an image projection error.²

Independent of the semantic labels which are relevant for the radar segmentation, the Hard-Net++ CNN [35] discriminates 19 semantic classes on which it is trained on. With the radar points projected onto the image, each radar point gets the semantic label $\hat{y}_{\text{sem,camera}}(\mathbf{p}_{\text{radar},i,t})$ of its corresponding image pixel assigned.

The only image label that is excluded from the radar label assignment is the image label for sky . Since radar reflections describe reflective object surfaces, the sky will never occur as radar reflection. All radar points projected to sky labeled pixel areas are re-mapped to the semantic label unknown .

3.3.8 Visual Semantic Label Fusion

Fusing the two visual semantic perception branches stands for voting either for the LiDAR semantic $\hat{y}_{\text{sem,lidar}}(\mathbf{p}_{\text{radar},i,t})$, or to prefer the camera semantic $\hat{y}_{\text{sem,camera}}(\mathbf{p}_{\text{radar},i,t})$. The *Fusion and Labeling* block of the labeling process Figure 3.2 finalizes the automated labeling of the radar point cloud.

Since both CNNs of image semantic segmentation and LiDAR semantic segmentation do not deliver confidences or other quality measures directly, the result of the CNNs is further probed to deliver reliable labels.

As the data set generation is an offline process, the data is present and can be processed runtime independently. Similar to the batch-accumulation of radar point clouds for the spatio-temporal tracking, this idea to compare multiple time-steps is transferred to the image processing.

In order to increase robustness of a CNN, generalization measures are applied, e.g. L2-regularization to regulate weights from growing or a dropout is introduced as regularization measure for the training. Similar to the generalization by dropout, Kendall and Gal [111] and Loquercio et al. [132] suggest the Monte Carlo Dropout (MCD) on the inference.

Requiring M multiple inference steps, the MCD application defines an uncertainty measure. This type of uncertainty describes how well the CNN models perform. This model uncertainty depends on the systematic model-induced errors and is called epistemic uncertainty [111].


With having the epistemic uncertainty, comparing M multiple inference results, model failures on an image sample can be detected. For each image and LiDAR point cloud, the segmentation CNNs are processed for a set of M inferences. Comparing the output images

² This error can not be reduced, due to the camera optics.

with respect to semantic consistency, a reliability score per pixel is calculated. Consistent areas indicate a reliable segmentation, whereas uncertain image pixels might be segmented inconsistently over the M processings.

This reliability score is calculated by the MCD and can be visualized as binary black (unreliable) and white (reliable) mask of each processed image, or respective range-image in the LiDAR case.

For the semantic label fusion, each camera pixel with its corresponding reliability score as pixel map is compared to the corresponding LiDAR label with its' reliability score. A rule-based decision is applied:

- For consistent semantic labels of both sensors, this semantic label is applied.
- In case of conflicting, contra-dictionary semantic labels delivered by both sensors, the reliability mask is applied as decision score. The semantic label with higher reliability score overrules the other sensors' semantic label.
- If conflicting semantic labels with same reliability score appear, an indifferent semantic label unknown  is applied.

With this decision, for each radar point $\mathbf{p}_{\text{radar},i,t}$, the introduced labeling pipeline defined if is an *im-plausible* detection $\hat{y}(\mathbf{p}_{\text{radar},i,t}) = 0$, or a relevant *plausible detection* $\hat{y}(\mathbf{p}_{\text{radar},i,t}) = 1$ and which semantic label this *plausible* radar detection most probably represents.

The subsequent steps towards an automated labeled data set includes manual correction and a data format conversion.

3.3.9 Data Preparation for Machine Learning

In the steps described before, the additional semantic information from the labeling pipeline is attached to each radar point as new point attribute. To process the attributes to apply pattern recognition on the generated data set, the data format of each point cloud is converted into binary files. The grey blocks in Figure 3.2 illustrate the refactoring process of the labeled radar point clouds.

Data set Format Conversion: According to idea to train a semantic segmentation network on the generated semantic labels, the refactoring of point cloud segmentation models is enabled by a data-structure conversion workflow. Instead of customizing a radar specific data set structure, the existing SemanticKITTI [18] data structure is extended to be applicable for the radar data.

As a result, existing semantic network architectures, which have proven to work for LiDAR point clouds, can simply be reused on the radar data. The input tensor sizes of the model

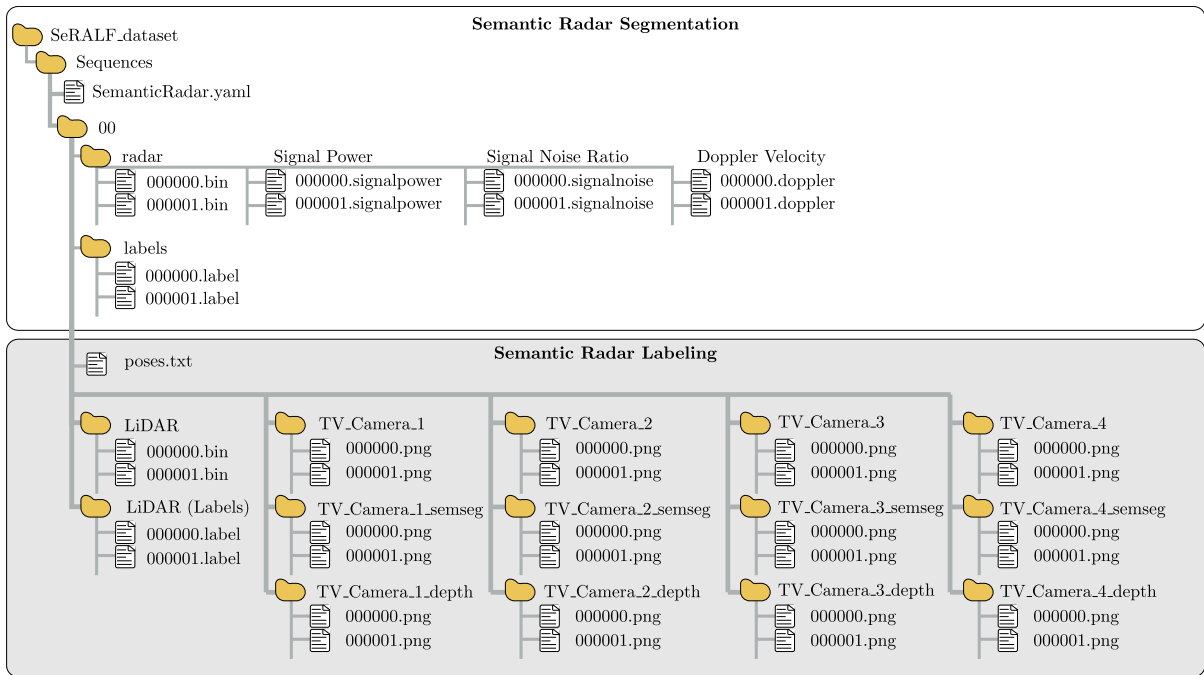


Figure 3.10: Data set format inspired by Behley et al. [18], with extension for radar specific channels *Signal Power*, *Signal to Noise Ratio* and *Doppler Velocity* in additional binary files per channel.

data-loaders are adjusted for a flexible radar attribute extension. Point based operations are specifically adjusted for each network modification, resulting in minor changes. Yielding a reduced subset of classes, the potential network architectures can also be streamlined to a reduced set of output neurons or classes.

The native ROS measurements as ".bag"-files are saved as individual point clouds ".pcd"-files per measurement cycle by the labeling pipeline. This labeling output is converted in the *SemanticKITTI Converter* of Figure 3.2 into the more generic binary data format of binary (".bin"-) files enables the real-world measurements to be converted into the generic data set format of SemanticKITTI [18]. As baseline of the modified process serves the common Kitti2Ros converter, which is extended to be applicable on the radar specific point clouds.

For this conversion process, the six separate radar point clouds of the test vehicle are ego-motion compensated and assembled to a single synchronized 360° point cloud and time-stamp. This 360° cloud is then converted into binary representation and stored as .bin file.

Instead of processing each 360° scan independently, each two consecutive 360° scans are again ego-motion compensated and concatenated to one dense point cloud. The underlying assumption is a low ego-motion drift in a short sequence and an outbalancing advantage of denser point clouds instead of sparse 360° radar point clouds. Due to the concatenation, the resulting data set frame-rate is reduced to half the original sensing frame-rate. This concatenation increases the density of the point clouds, enabling local feature extraction over a reduced frame-rate. For the semantic segmentation approach, which relies on point

neighborhood relations, the disadvantages are rated as non-essential. Depending on the application in scope, the drawbacks of an accumulation approach need to be discussed separately.

The binary files of a point cloud efficiently only store the spatial information of x, y, z coordinates. Each relevant other radar point attribute from the ".pcd"-file is converted into an additional binary file and saved independently. See *SemanticKITTI Converter* of Figure 3.2, each point label attribute is extracted as binary ".label" file, encoding each points' class label.

Per radar point cloud of a sample sequence, the files are named with a unique identifier code. With this structure, the data-loader in a machine learning pipeline is able to efficiently load only the relevant signals or attributes from specific files by the unique identifier code, or is able to shuffle the data during training. As such, the required memory of the network and data set is kept to a minimum, at the highest flexibility to include or exclude certain radar attributes during the network design.

Manual Correction Tool: Beneficial of the SemanticKITTI data set structure are the common evaluation methods e.g. the SemanticKITTI-API or especially the point labeler [18].

Post auto-labeling processing and binarization, the consolidated semantic labels of Table 3.2 are manually checked by a visualization tool, extending the point labeler, to reach ground truth quality level. This step is illustrated as *Manual Correction* block in Figure 3.2. Hereby, all labels beside the automatically determined *clutter* are corrected by manual inspection and re-labeling. The conclusion of valid *clutter* labels is drawn from the requirement of the spatio-temporal tracking filter in the automated labeling pipeline. This filter stage indicates non-stationary objects to be *noise*. Hence, most multi-path reflections etc. are filtered to *noise* and do not need to be revisited by manual labeling.

The labeling and visualization tool is enriched to visualize the radar points with their semantic class as color-code of Table 3.2 with the LiDAR point cloud as grey value reference sensor data. Either in single radar scan overlay, or for a sequence of radar scans being projected together along the odometry-based trajectory, the semantic labeling of the radar points is facilitated, see Figures 3.11-3.13. With the chance to check the projected radar labels for sensed objects over a sequence, especially spatio-temporal semantic consistency is ensured.

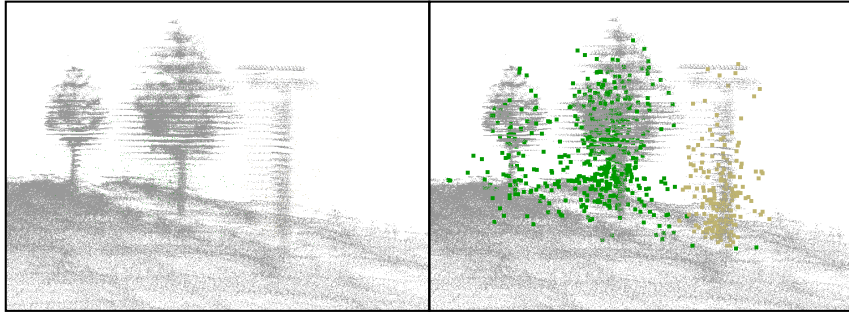


Figure 3.11: Exemplary sample of trees (green) and poles (yellow), depicted as dense grey LiDAR scan assembly (left) and manually corrected sparse radar point cloud (right) without noise.

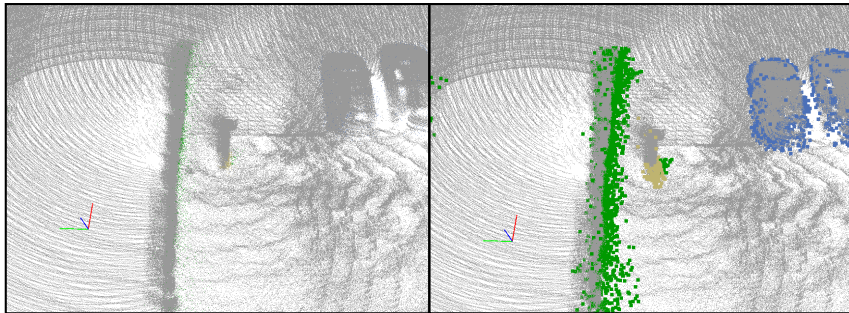


Figure 3.12: Exemplary sample of a hedge (green), poles (yellow), and cars (blue), depicted as dense grey LiDAR scan assembly (left) and manually corrected sparse radar point cloud (right) without noise.

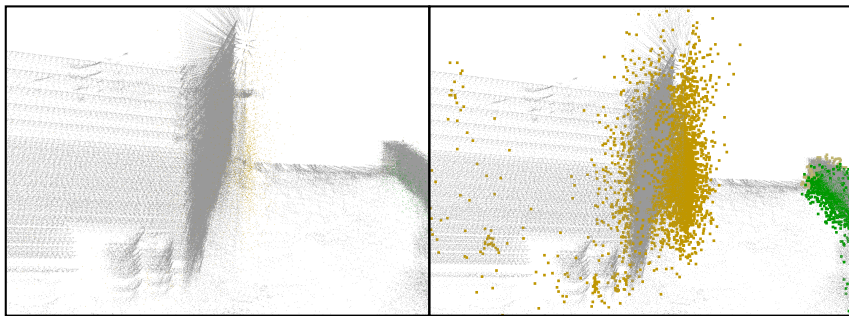


Figure 3.13: Exemplary sample of a hedge (green) and wall structure (yellow), depicted as dense grey LiDAR scan assembly (left) and manually corrected sparse radar point cloud (right) without noise.

For small objects, similar to poles or persons, this is essential. Besides the above mentioned spatio-temporal inspection scan by scan, scenes containing moving objects need special manual inspection to label the moving objects correctly. From the LiDAR sequence projection, the moving object paths are visible as tailing tracks. Indicating the path of an object is helpful for the manual labeling per scan. Hence, each radar scan needs to be labeled independently, with scene comprehension which of the potential automatically as clutter labeled detections might be resulting of the considered moving object.

In order to realize a consistent manual labeling, a labeling policy is set based on which the LiDAR point cloud helps to correct the semantic radar label.

1. Labeling of outer building structures only: Internal building structures are labeled as *clutter*.
2. Radar detections inside a car are all labeled as *vehicle* and not excluded.
3. Any sort of fences or barriers are non-driveable and are labeled as *building*.
4. Flat vegetation (e.g. grass) is not labeled (normally not reflecting), but bushes, hedges as trees.
5. Dynamic objects are labeled in frame-by-frame manner.

3.4 Data Set Evaluation

The data set is generated on 20 different driving sequences which resemble typical parking or suburban driving scenarios, recorded at low ego-vehicle speed. Based on the low ego-motion, subsequent sensor readings (radar, camera and LiDAR) overlap by a large fraction. Overlaying subsequent sensor scans facilitate the context-aware label cross-check and ensure a spatio-temporal consistent semantic labeling.

In the intermediate first step of the automated labeling, all radar points are reduced by 80% to the *plausible* detections. Figure 3.14 illustrates *implausible* red detections in overlay with the blue *plausible* radar detections, in comparison to the exclusive subset of blue *plausible* radar detections. The LiDAR point cloud (grey) serves as reference context background. The removed 80% are classified as radar artifacts (red) and not further labeled.

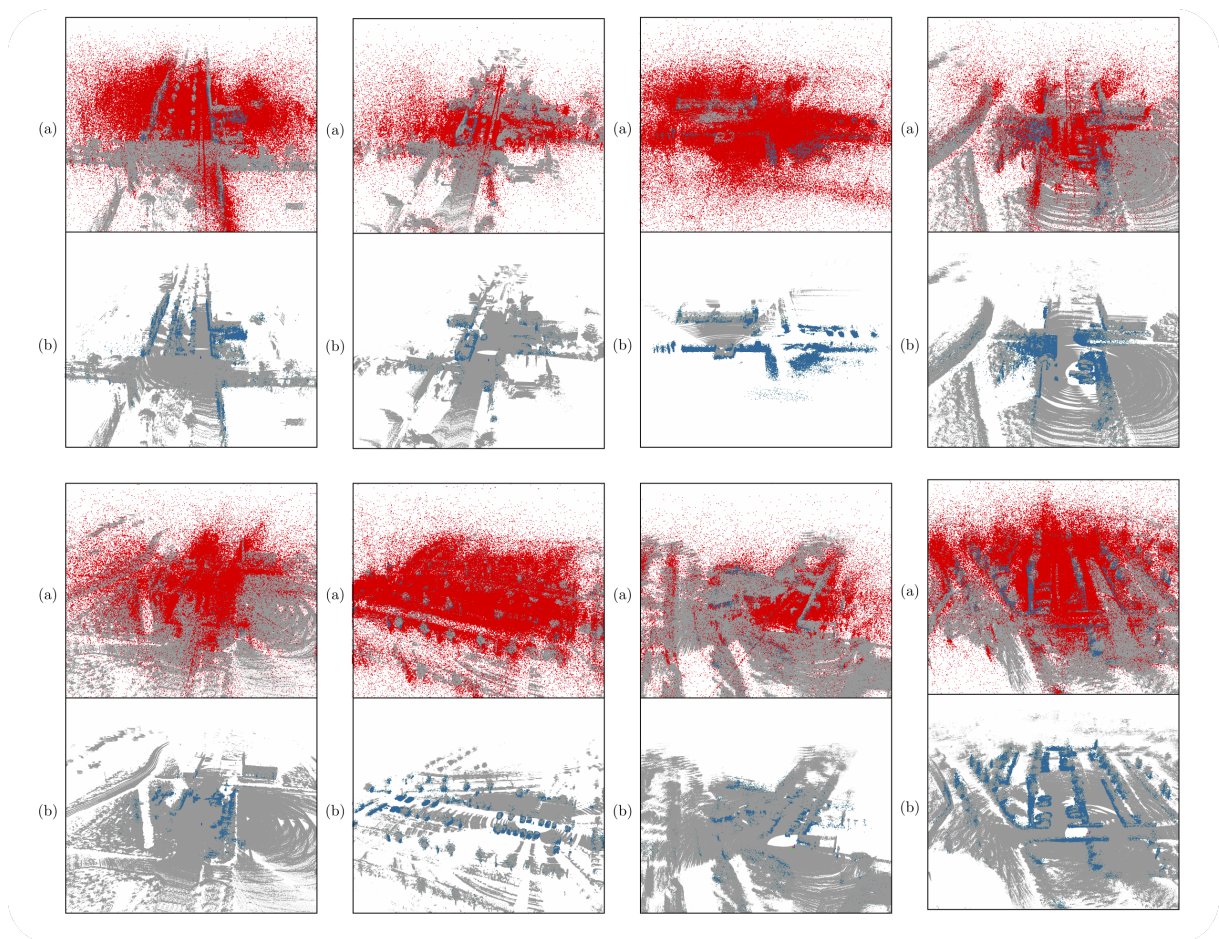


Figure 3.14: Exemplary data set sequences 00-07 with grey LiDAR reference points, of Isele et Al. [SI1]. All *noise* radar points (red) and *plausible* radar points (blue) in the top figures (a), while the bottom images (b) show the same sequence, showing exclusively the remaining *plausible* radar detections.

In the second step, the remaining *plausible* radar labels are specified into six meaningful semantic radar labels, compare Table 3.2 for the details of the semantic radar classes. An

exemplary scene illustration of the semantic labeling of *plausible* radar detections is found in Figure 3.15.

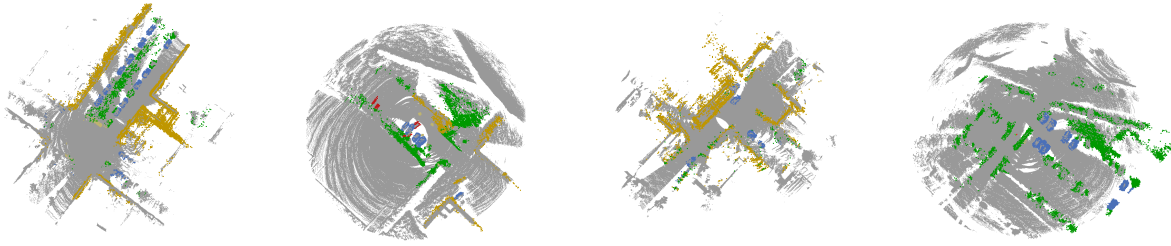


Figure 3.15: Exemplary scenes with resulting semantic radar labeling of *plausible* radar detections, illustrated without noise, class-wise colors according to Table 3.2. Figure of Isele et Al. [SI3].

Cars and other *vehicles* are mainly stationary. *Vegetation* in form of bushes, hedges and other trees is present. The third major object class is *buildings* and other man-made structures, which are consolidated to one class. Walls, fences or other non-through-passing man-made structures are commonly labeled as *building*. In this class, the radar detections can still vary broadly: Reaching from a steel- or metal made fence to a wooden fence to even all various types of walls, boards or cemented house artifacts, the class is set to *building*.

Figure 3.16 illustrates the data set content as generic scenes with less road traffic.

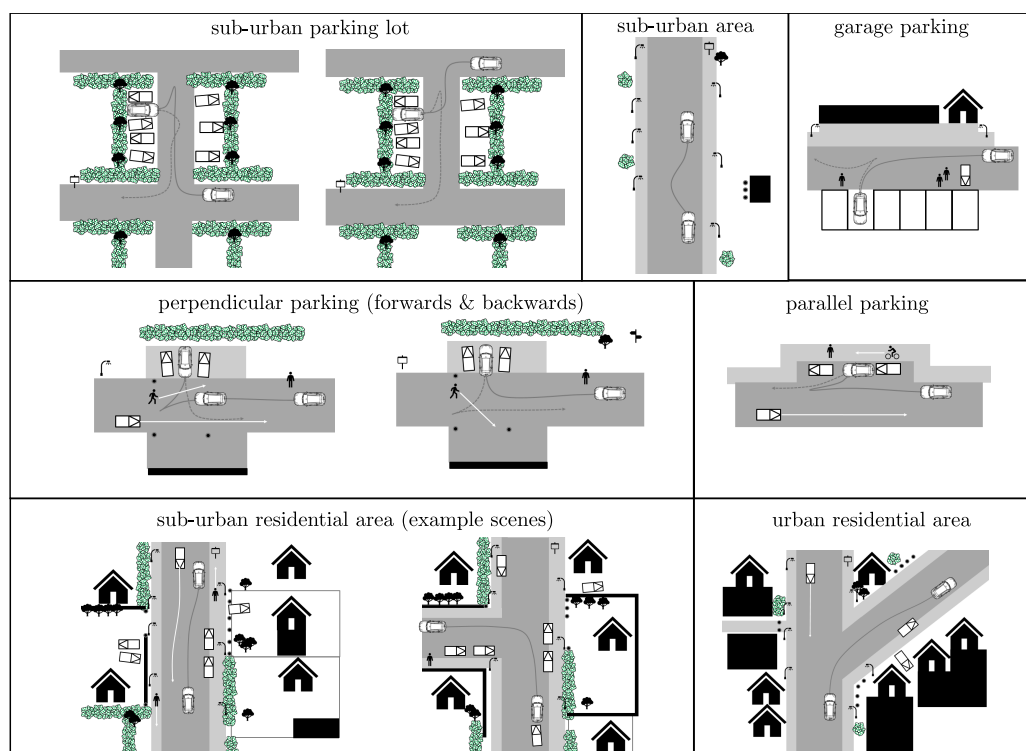


Figure 3.16: Exemplary scenes of the data set with ego-vehicle and its park-in motion as solid (park-out as dotted) grey line in typical environment structures. Black blocks represent buildings, black circles represent poles, white vehicles are either parked or their path is indicated as white arrow.

3.4.1 Data Set Overview

Details of the 20 driving scenes are found in Table 3.4. The data set is comprised of 8405 radar scans, resulting in $\approx 8.2 \cdot 10^6$ radar detections, covering a driven path of 2507.35 m at $\approx 15 \frac{km}{h}$. Overall, the data set consists of ≈ 600 seconds of radar recordings.³

Table 3.4: Data set overview of semantically labeled radar data.

| Name | Description | Path Length [m] | Radar Scans |
|-------------|--|-----------------|-------------|
| Sequence 00 | urban residential area | 58.74 | 245 |
| Sequence 01 | perpendicular parking | 31.66 | 290 |
| Sequence 02 | sub-urban residential area | 20.34 | 101 |
| Sequence 03 | sub-urban parking lot | 80.65 | 400 |
| Sequence 04 | garage parking | 51.92 | 332 |
| Sequence 05 | urban residential area | 29.78 | 163 |
| Sequence 06 | perpendicular parking | 25.71 | 170 |
| Sequence 07 | sub-urban parking lot | 77.92 | 442 |
| Sequence 08 | urban residential area | 62.19 | 256 |
| Sequence 09 | sub-urban residential area | 17.81 | 82 |
| Sequence 10 | sub-urban residential area | 27.83 | 232 |
| Sequence 11 | sub-urban parking lot | 92.92 | 474 |
| Sequence 12 | parallel parking | 31.06 | 243 |
| Sequence 13 | sub-urban residential area | 14.98 | 215 |
| Sequence 14 | sub-urban area | 16.15 | 45 |
| Sequence 15 | sub-urban residential area (loop) | 480.66 | 969 |
| Sequence 16 | sub-urban residential area (double loop) | 613.60 | 1452 |
| Sequence 17 | sub-urban residential area (loop) | 410.65 | 1203 |
| Sequence 18 | sub-urban residential area (loop) | 362.69 | 1082 |

3.4.2 Scene Evaluation

The sequences are an exemplary collection of typical objects and average scenarios to be met in real-world. This collection of sequences is recorded in typical parking scenarios in different settings and environments, to generate a real-world based data basis for the research in this thesis. Systematically radar data delivers an un-balanced data set, considering the total number of detections per class. Without claim of completeness, the data set represents an exemplary, general collection of real-world radar samples of e.g. *poles* or *person* occurring with lower regularity, compared to exemplary *vehicle* radar detections. The specific overall class distribution over all classes is illustrated in Figure 3.17. A complementary occurrence rate or definition of relevant objects is hereby not implicitly formulated.

Table 3.5 shows the radar point count of the binary classification into *im-plausible* $\hat{y}(\mathbf{p}_{\text{radar},i,t}) = 0$ and *plausible* $\hat{y}(\mathbf{p}_{\text{radar},i,t}) = 1$ radar detections.

³ As requirement for the automated labeling pipeline, the corresponding 4x30Hz camera data, CAN-Data, and 20Hz LiDAR point clouds need to be recorded and processed. The overall recording and post-processed data exceeds 5 – 10TB.

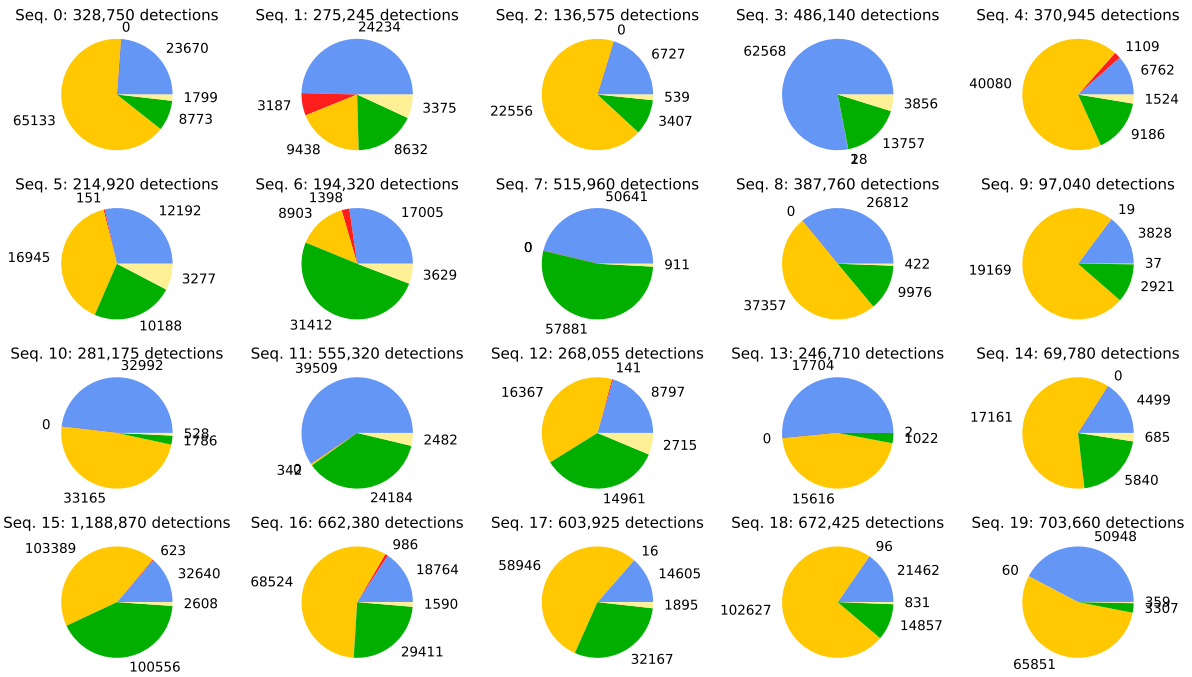


Figure 3.17: Data set label distribution as pie chart, visualizing semantic radar classes in the colors according to Table 3.2, clutter excluded.

Table 3.5: Data set details for the 00-10 sub-set, as in Isele et Al. [SI1].

| | $y = 1$ | $y = 0$ |
|---------------|-----------|---------|
| $\hat{y} = 1$ | 2 432 440 | 268 869 |
| $\hat{y} = 0$ | 157 076 | 430 418 |

The remaining *plausible* detections are split in the six semantic classes. As visualized in the pie-charts before, the number of detections per class are not balanced. But also the spatial occurrence of the detections is of interest. The overall detection range of 90 *m* is visualized in Figure 3.18, showing a heatmap of all detections with respect to the vehicle.

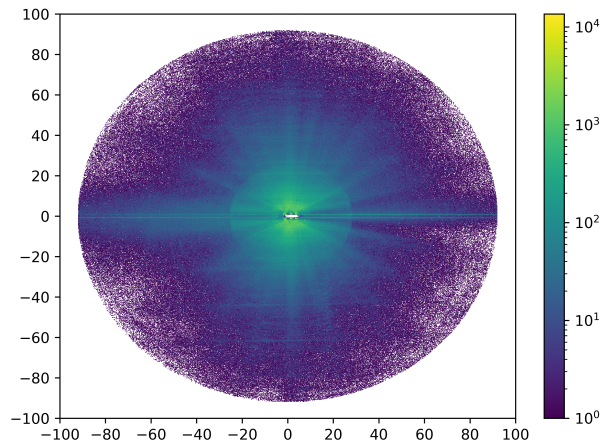


Figure 3.18: Heatmap of spatial occurrence count of all radar detections in a 90 m radius around the vehicle independent of their semantic class.

From the overall visualization in Figure 3.18, the covered areas are spread and no distinct concentration can be found. Therefore, the same spatial occurrence heatmap is illustrated but isolated per semantic label in Figure 3.19. This illustrates the different spatial occurrence of radar detections per semantic class as summary over all recorded sequences of Table 3.4.

For the following discussion of semantic labels, only detections in the range of 50 m are depicted and discussed. Only detections in this range will be consequently utilized in the later segmentation in Section 4, mapping in Section 5 and parking system in Section 6. From the class-wise spatial occurrence heatmaps in Figure 3.19, straight passages of cars and streets stand out. Similar to other data set distributions, e.g. SemanticKITTI [18], the forward passage of scenes is recorded. For a generalization to other use-cases or other raw data-distributions, e.g. specifically crossing traffic at intersections, the data set does not contain such scenes without augmentation. For the training of the semantic segmentation network, data augmentation is applied to achieve independence of the spatial occurrence of the data, see Section 4.4.

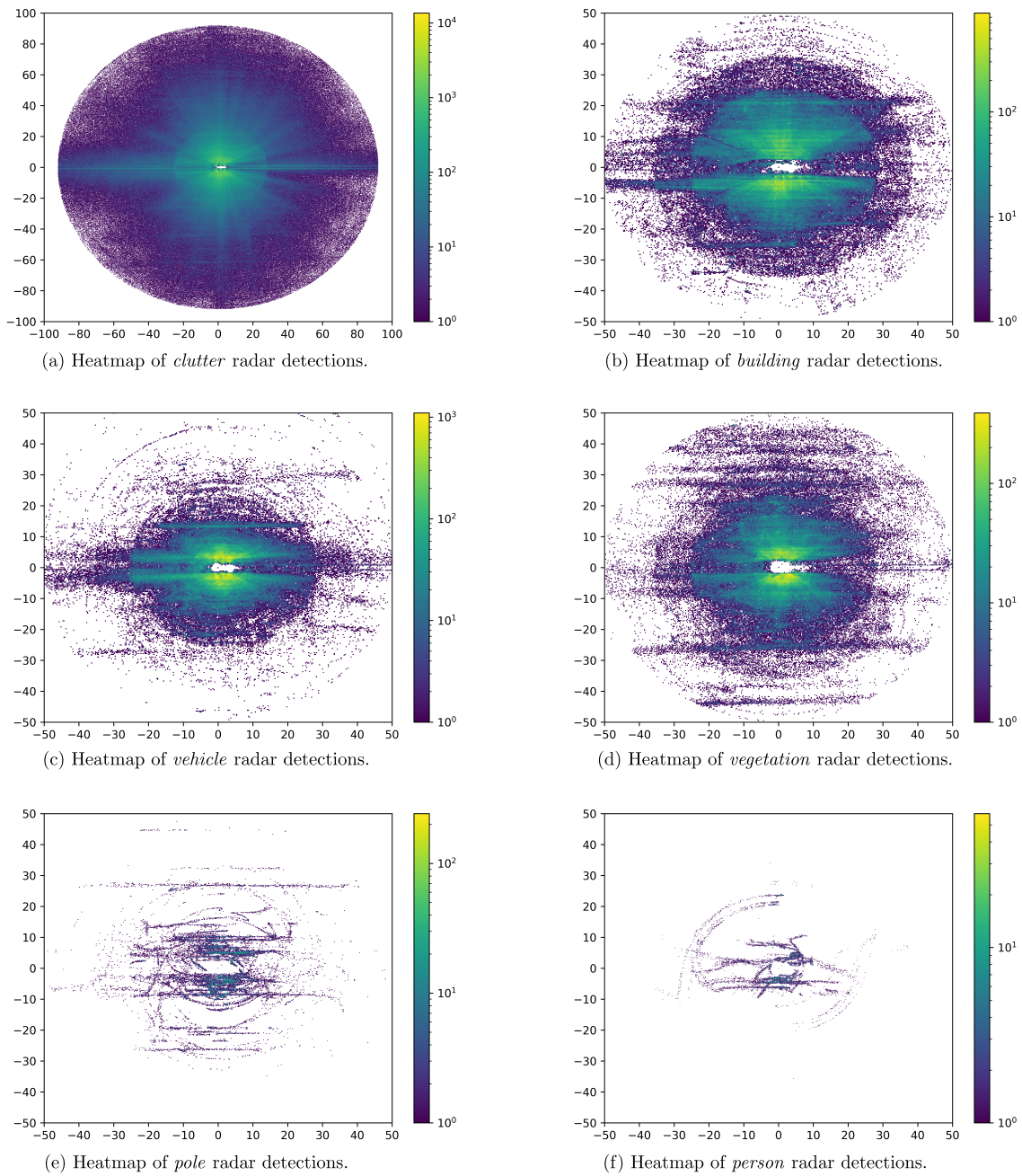


Figure 3.19: Heatmaps of the spatial occurrence of radar detections per semantic class in a 50 m radius: *Clutter*, *building*, *vehicle*, *vegetation*, *pole* and *person*.

3.4.3 Labeling Process Evaluation

The two-stage process of labeling *plausible* detections and further split these detections into semantic classes can be evaluated per step. First, the plausibility labeling result of Section 3.3.5 is discussed, then the overall semantic labeling result of the process, see Figure 3.2 and Section 3.3.8 is evaluated.

Table 3.6: Tested plausibility label data set of $M = 11$ sequences with bold marking of min. and max. metric scores.

| ID | $\emptyset N$ | $\emptyset Acc$ | $\emptyset Precision$ | $\emptyset Recall$ | F1 plausible | $\emptyset IoU$ | IoU plausible | IoU artifact |
|----------|---------------|-----------------|-----------------------|--------------------|--------------|-----------------|---------------|--------------|
| Σ | 2704 | 0.873 | 0.826 | 0.779 | 0.675 | 0.682 | 0.510 | 0.854 |
| 00 | 245 | 0.848 | 0.833 | 0.793 | 0.722 | 0.687 | 0.565 | 0.810 |
| 01 | 290 | 0.883 | 0.796 | 0.781 | 0.646 | 0.673 | 0.477 | 0.869 |
| 02 | 101 | 0.878 | 0.839 | 0.822 | 0.740 | 0.720 | 0.588 | 0.852 |
| 03 | 400 | 0.885 | 0.706 | 0.761 | 0.622 | 0.662 | 0.451 | 0.873 |
| 04 | 334 | 0.927 | 0.863 | 0.859 | 0.765 | 0.768 | 0.620 | 0.917 |
| 05 | 163 | 0.910 | 0.869 | 0.845 | 0.757 | 0.752 | 0.609 | 0.895 |
| 06 | 170 | 0.776 | 0.771 | 0.678 | 0.537 | 0.554 | 0.367 | 0.742 |
| 07 | 422 | 0.860 | 0.808 | 0.739 | 0.616 | 0.644 | 0.445 | 0.824 |
| 08 | 265 | 0.850 | 0.759 | 0.771 | 0.622 | 0.640 | 0.451 | 0.829 |
| 09 | 82 | 0.845 | 0.813 | 0.776 | 0.684 | 0.667 | 0.520 | 0.814 |
| 10 | 232 | 0.874 | 0.844 | 0.798 | 0.715 | 0.703 | 0.556 | 0.850 |

Stage 1 -Plausibility: To determine a qualitative measure for the accuracy of the *plausible* detections, the data set is manually inspected and corrected. For the manual correction of individual detections, the labeling tool presented in Section 3.3.9 is applied. The evaluation of the automated labeling is performed on the sequences 00-10 of the data set, the remaining sequences are not evaluated.

As *Classification Metrics* in Section 2.3, Accuracy, Precision, Recall, F1 score, and IoU are defined and applied for the data set labeling evaluation. Table 3.6 presents the metrics score for the sequences 00-10.

Based on the Accuracy, a mean error of only 12.95% is achieved. With this pre-filter, most of the *noisy* and transient detections can be filtered out. This first step of labeling is overestimating the plausibility on purpose.

It is found beneficial for the subsequent semantic labeling, to have a precise region information where the detections labeled as *plausible* occur. The overestimation of the relevant region allows the subsequent semantic labeling steps, both the automated and manual inspection step, to refine this region and detections herein.

The automated process yields an overall average Precision of 82.6% with average Recall of 77.9%. Hence, the detections to label semantically is successfully reduced to the most probable *plausible* ones by the automated labeling process. The semantic labeling step further refines the semantic information of these detections.

The 11 different scenes in Table 3.6 showcase the range of labeling difficulty. To mention Sequence 06 explicitly as the lowest Accuracy and lowest average IoU, the difficulty of this scene is found in the narrow passage of a garage in the beginning of the scene. Obviously, the depth estimation fails e.g for a very close wall where the LiDAR reference shows basically no overlap with the depth image. Compare the field of view in Figure 3.7. This example showcases that too close walls or too narrow passages are difficult to label.

Stage 2 - Semantic Labeling: The comparison of the labeling result is performed after manual correction of the sequences 00-10.

The labeling accuracy determines how precise the semantic labels are automatically annotated to the *plausible* radar detections. Best measured is this by inspecting the confusion matrix of Figure 3.20, taken from Isele et Al. [SI2]. According to its' introduction as *Classification Metric* in Section 2.3, the confusion matrix shows correct segmentation results on its diagonal. The percentage of hits and the corresponding number of samples is given for each tile.

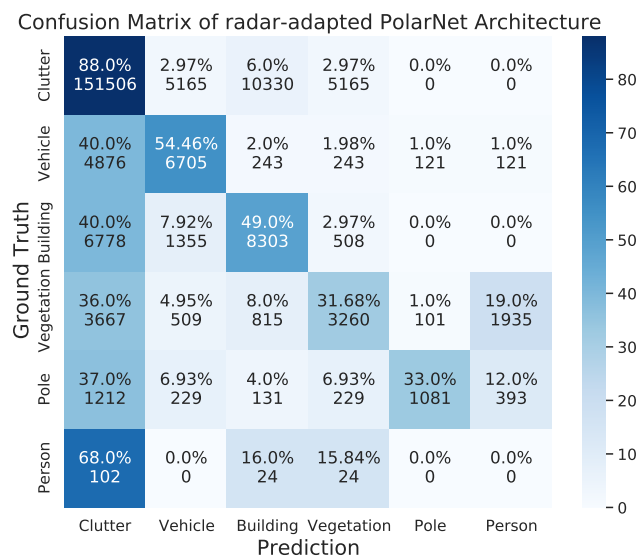


Figure 3.20: Confusion matrix of automated semantic labeling vs. manually corrected semantic labels.

Mentioned before, an over-estimating region of *plausible* detections allows a refinement by the semantic labeling step [SI1, SI2]. The manual inspection and refinement is facilitated by refining the object boundaries, instead of increasing an underestimated shape. As a matter of over-estimation, the confusion of all classes with *clutter* can be explained.

Approximately 10 persons and circa 100 poles of different material, size and diameter occur in the data set, compare the number of hits and total samples per class. The rare semantic classes *pole* or *person*, see Table 3.7, are semantically hard to label because of the structurally

less detections per object. A human reflects with circa 4-10 radar detections per sequence, pole reflections are denser but vary with their material and diameter. In this perspective, the data set lacks examples of radar detections of persons and poles in different ranges and aspect ratios. A mis-classification, either by over-estimation or by mis-perception causes a hard decrease in percentage of accuracy and percentage of the confusion matrix. The mis-classification of the rare classes, also includes the shape-variant *vegetation* class. Especially *persons* are mis-classified as dense *vegetation*, similar to a hedge, mainly explainable by similar sparsity and radar reflection properties. A larger data set of reference examples, especially for *persons*, *poles* and different kinds of *vegetation* is expected to drastically improve the detection performance of these classes, but remains open for future work.

The corresponding table of average semantic label Accuracy, Precision, Recall and F1 score is found in Table 3.7.

Table 3.7: Dataset evaluation of the automated semantic labeling, according to Isele et Al. [SI2].

| Class | [%] | ∅Acc | ∅Precision | ∅Recall | F1 | ∅IoU |
|------------|-------|-------|------------|---------|-------|-------|
| Average | - | 0.861 | 0.618 | 0.501 | 0.542 | 0.406 |
| Artifact | 78.35 | - | 0.899 | 0.940 | 0.920 | 0.850 |
| Vehicle | 8.39 | - | 0.682 | 0.482 | 0.565 | 0.394 |
| Building | 7.68 | - | 0.676 | 0.643 | 0.659 | 0.491 |
| Vegetation | 4.80 | - | 0.648 | 0.315 | 0.424 | 0.269 |
| Pole | 0.60 | - | 0.313 | 0.139 | 0.192 | 0.106 |
| Person | 0.18 | - | 0.490 | 0.488 | 0.489 | 0.324 |

Having introduced the spatial spread of classes in the heatmaps of Figure 3.19, the spatial occurrence is evaluated with respect to the IoU over detection range. Figure 3.21 illustrates the IoU curves. In combination with Figure 3.19, the IoU drop of all classes to 0% IoU except *noise* in the range beyond 40 m, is explained by the absence of detections in this range - compare with Figure 3.19. The same explanation mechanism accounts for the increase in the near-field. With a quasi stable mean IoU of approximately 38% and noise IoU beyond 80%, the automatic semantic labels serve as well educated, consistent guess.

Labor Savings: As last conclusion, the facilitation of manual inspection and correction step is shortly discussed. Extensive manual labeling from scratch requires expert knowledge, a solid capacity to think in three dimension, plus having a feeling and expertise in radar sensor details. The labour effort that is saved by the automated semantic pre-processing is rated to reduce the manual effort by a significant factor (estimated: 4-8) of intensive 3D point cloud inspection time. Additionally, the invaluable and immeasurable effect of consistent semantic labeling, what the automated pipeline guarantees, is a major advantage of the

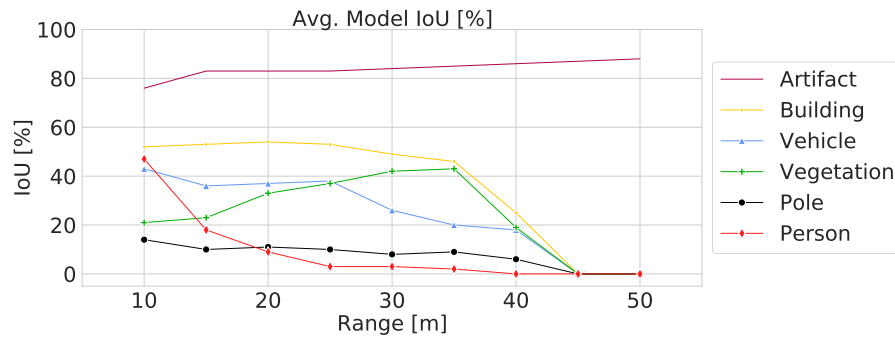


Figure 3.21: SeRaLF Results of Isele et Al. [SI2]: IoU over distance to the sensor.

automated radar labeling pipeline. Compared to other works, Schumann et al. [194] discuss the effects of manual labeling to be significant.

3.4.3.1 Failure Potential

The presented automated labeling delivers an estimated guess of the semantic radar labels with the help of reference sensor interpretation. Hence, reference sensor data artifacts can cause erroneous label suggestions, as the two most important are discussed here.

Monocular Depth Estimation: Image-based perception generally can be compromised from a variety of camera and lens effects in real-world conditions. Illumination, shadows, lens flare, dust and mist, or rain and snow are generally causing camera images to contain exceptional effects. These effects should not be neglected for a general evaluation. Nevertheless, they are out of scope for this thesis. Actively avoiding such conditions, the potential effect on the scenes at hand are rated as neglectable. No bad weather conditions, nor a blinding illumination are recorded.

Resulting from the LiDAR FoV blind spots, see Figure 3.7, not all of the radar detections can be compared to the LiDAR point cloud. Instead, a possibility to apply monocular depth estimation on the series fish-eye cameras is implemented. The key to facilitate the plausibility labeling based on image depth estimation, is to apply an adequate image pre-processing. Since pre-trained CNNs are trained on rectified rectangular images, the domain gap between original training samples and the pre-processed real-world fish-eye images need to be reduced to a minimum. This image pre-processing remains a potential error source, e.g. from un-distortion, perspective transformation or 3D point cloud projection onto the image.

Furthermore, indeterminable if and to which extent, as a direct result of the image pre-processing or as independent systematic depth estimation error, the depth estimation can yield erroneous results. These shortcomings can elegantly be reduced by an overlap with the LiDAR FoV, allowing the local scaling of the depth estimation image to metric scale.

Experimentally it is found that objects of low or medium height objects below ≈ 30 cm suffer in the depth estimation to be detected as outstanding shape. The overview of the different depth estimation results in Figure 3.4 illustrates the different depth estimation CNNs to under-perform at the small pole-like structure.

Planar Scene Violation Assuming a planar motion, the single-track model degrades in non-planar motion. As a result, the ego-motion compensations is compromised.

Furthermore, on the LiDAR point cloud, a RANSAC floor plane estimation is applied to extract the LiDAR ground points. Including ascents or other ramps, independent of their inclination if positive or negative, the plane estimation can result in artifacts. Not fully removed LiDAR ground points result in radar detections near the ground to be rated *plausible*. As a consequence, there remain a higher fraction of plausible radar detections to be labeled semantically. In such cases, e.g. flat regions with lawn areas or low bushes, are estimated as *plausible* and relevant detections - in contrast to general radar detections of the ground being *implausible* in this labeling approach.

3.5 Section Conclusion

The section describes the Semantic Radar Labeling Framework (SeRaLF) [SI2] as an automated semantic labeling process for radar point cloud data based on reference sensors, with subsequent manual inspection and data refactoring. A process-chart of the subsequent steps is found in Figure 3.2, which summarizes the *labeling* research questions of Section 1.3 how to generate and automate a radar labeling pipeline. Based on real world hardware measurements, the presented labeling pipeline of Figure 3.2 processes available image data from surround-view cameras and a reference LiDAR sensor to fuse both sensor modalities. Two parallel optical multi-sensor pipelines of optical semantic image segmentation and LiDAR point cloud semantic segmentation are combined to compare and fuse point-wise semantic labels for radar point cloud. The general procedure is covered by two patents [Pat3] [Pat4].

Designed for a general application, the use-case is a point-wise labeling of a quasi static environment perception by radar point clouds. The resulting data set of semantically labeled radar point clouds consists of a set of 19 sequences, see Table 3.4. Recordings from suburban areas, to rural village scenes, to environments in typical parking scenarios are labeled as Figure 3.15 illustrates and Figure 3.16 categorizes. Designed especially for the purpose to train neural networks for semantic radar segmentation, the presented dataset provides $\approx 8.2 \cdot 10^6$ radar detection points in total.

The semantic annotation of radar detections requires a radar-compliant label set. Inspired from the general semantic classes of CityScapes [47] and SemanticKITTI [18], only a subset of classes only is applicable for radar classification, see Table 3.2. This radar-applicable

semantic classes are labeled in the data set: *Artifact, vehicle, building, vegetation, pole* and *person*. There is no distinction between static and dynamic instances of a class.

The precision of the manually corrected semantic radar labels, including unavoidable erroneous labels, is expected to bound the upper limit of the achievable mIoU of a semantic classifier. The immeasurable impact of different labeling accuracy of multiple manual labelers cause a systematic difference of especially noise labels, far range detections or the labeling of sparse and scattered detections. Especially for geometrically small objects, resulting in rare radar detections, e.g. *person* or *pole*, or highly scattered reflecting objects such as *vegetation*, even a few label deviations of the manual labeling can cause difficulties to train a generalizing classifier. The lower the count of examples set per class, the higher the potential inconsistent labeling and harder classification task.

This availability of this semantic radar attributes plays a major role in this thesis. Considering the bottom-up system components of Figure 1.4, the semantic attributes provide the basis system level to constitute a feature level semantic data set. The following steps to generate a live segmentation on function level applies this feature level data set.

3.6 Section Outlook

The presented method serves as initial semantic radar data set generator and basis for the following section. Utilizing this data set as basis to train an semantic segmentation network in the following Section 4, this network could be utilized in return as additional semantic radar labeling channel to leverage the quality of the automated labeling process to refine the semantic radar data set, and scale its size. In such an approach the data set size and e.g. a trained segmentation network performance is expected to grow in parallel.

With the utilized visual modules of camera segmentation, mono depth-estimation, and LiDAR segmentation as interchangeable tools, any further improvement of these tools yield an improved labeling framework. But still, manual supervision of the results is expected to be required.

Due to the modular framework, also the extension other classes or e.g. dynamic objects and instances can be integrated by the application of available segmentation tools for the reference sensors. A labeling of dynamic objects is expected as useful further extension of this labeling framework. In order to develop a direct radar segmentation, dedicated to dynamic object segmentation, this extension might be necessary.

The application of the labeling framework on a large data basis is implemented as sequential processing of the different channels, but parallelization yields further shorter processing time but requires parallel GPU capacity and a parallel read-write data container.

4 SEMANTIC RADAR SEGMENTATION DEEP LEARNING NETWORKS

The preceding Section 3 introduced point-wise semantic radar labels. These semantic labels form the basis to train a learning approach for point-wise semantic radar segmentation. This section of the thesis aims to fill the gap between raw radar detections and a live semantic radar segmentation perception in an arbitrary structured environment. Therefore, first radar appropriate learning frameworks are presented, secondly a selected approach is extended, applied, and evaluated on the presented radar data. The section is closed by a discussion of the approaches' potential in the automotive domain, and challenges of the discussed frameworks and networks.

Commonly applied for dynamic object clustering [186, 225] or radar ego-motion estimation [32, 33, 8], radar sensors are underestimated for the static environment description. New approaches utilize radar maps for localization [73], but not for semantic segmentation. For a live semantic segmentation of radar sensor data, including dynamic and static vehicles or environment objects, no algorithmic solution is yet available. Most of the existing radar classifiers are specifically designed to focus on dynamic object detection [186]. Nevertheless, the increase of automatization level in robotics and automation requires rich and redundant from live object segmentation to semantic mapping functionalities. Therefore, the noisy radar data can be enriched by a semantic classification to be interpreted and potentially arranged in space as interpretable environment representation. With known semantic classes, the semantic segmentation of radar detections solves a classification problem. As specific target use-case of the developed semantic radar segmentation, the data association in exemplary radar graph-SLAM mapping or radar point cloud scan matching in general, applying enriched sensor data with additional semantic attributes is addressed.

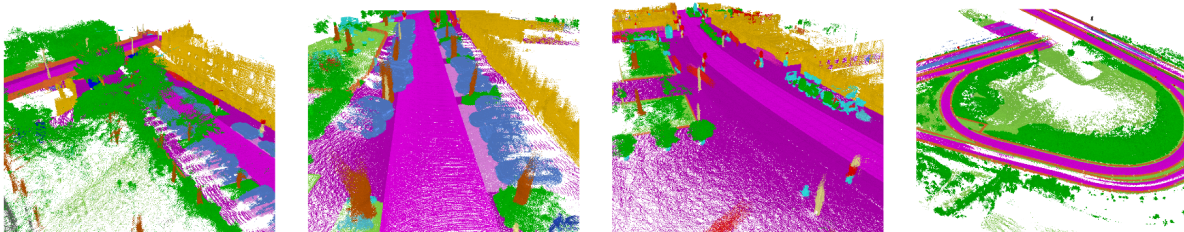


Figure 4.1: Illustration of segmented LiDAR point clouds of street scenes, assembled by a SLAM approach from the KITTI Vision Benchmark [71], published as the semanticKITTI data set of Behley et al. [18] with class-wise colors of the semanticKITTI data set.

A generic semantic radar segmentation for live and real-time application is not presented in literature for automotive radar sensors. Only few works touch this capability of direct sensor data classification, but focus on dynamic object detection and classification, e.g. RadarScenes of Schumann et al. [194]. Generic live semantic radar segmentation approaches, covering dynamic and static environment classification are not known. The closest comparable semantic radar segmentation is an offline classification of an environment map.

In comparison, applied research in the LiDAR sensor domain has proven live point cloud segmentation to be applicable for any kind of static or congested environments, e.g. Suma++ [39]. These LiDAR approaches include the fine-grained but generic 22 SemanticKITTI [18] classes, illustrated exemplary in Figure 4.1.

In this section, a method is derived to achieve real-time capable semantic segmentation in generic structured environments. According to the system modeling and dependencies in Figure 1.4, the segmentation is required to be real-time capable and achieve good semantic labels to be further applied as input generator for an exemplary target use-case functionality of semantic radar SLAM, see Chapter 5.

In perspective to achieve at a similar semantic segmentation on radars, Chapter 3 introduced a radar label generation and class consolidation, see Table 3.2. Applied on these radar labels, this section describes the transfer of selected promising and comparable deep learning point cloud segmentation CNN approaches to the radar domain. Training different neural network architectures in a supervised fashion on real-world radar data, allows to select and fine-tune a deep learning architecture for the specific semantic segmentation of radar point clouds.

The different segmentation architectures are implemented by the supervised Master thesis of M.Sc. Fabian Klein [MT3]. Partial content of the architectural comparison in this section can also be found in the conference paper, *Learning Semantics on Radar Point Clouds* of Isele et Al. [SI3]. Further research on alternative segmentation architectures beyond *RadarNet* can be found in the supervised Master thesis of Daniel Rotärmel [MT7].

4.1 Motivation for Direct Deep-Learning Radar Segmentation

The environmental perception on 3D point clouds - ranging from LiDAR sensors, over Radar sensors, to other sensing modalities, find their application in autonomous system pipelines [225, 240]. To boost the information content and functional usage of this rich raw radar data, additional radar perception strategies are required. The more data attributes are perceived from the raw radar sensor data, the better downstream data analysis modules or a situation interpretation is able to assemble a consistent scene and context perception. Redundancy across different sensing modalities increases system safety and reduces misperception.

As radar sensors are able to provide 3D coordinates (x, y, z) of reflections directly with a reference velocity of the reflection, called Doppler Velocity (v_D), this data provides a rich information content. Nowadays, such automotive radar data is referred to as *4D radar sensing* [142, 136]. In combination with other radar specific measured variables namely *Signal Noise Ratio (SNR)*, or *Signal Power (P_{Sig})*, the interpretation of radar data can be extended to six dimensions (*6D*). Leveraging higher order dimensional interpretation beyond rule-based classification suggests to apply learning methods. Deep learning based supervised learning promises to yield a live, real-time capable semantic segmentation solution by a neural network. Therefore, the previous Chapter 3 introduced the dedicated labeling of the data set which serves as training data set in this Chapter 4.

The high dimensionality of radar sensor data is challenging to interpret due to two systematic but significant sensing characteristics: A characteristic low density of the radar detection point clouds in the sense of scattered data with only a few neighboring points [65]. Dependent on the environment the radar detections, a characteristic sparsity is found in radar point clouds. In addition to the low overall number of detections per scan, the count of detections in subsequent scans is dynamically changing. In addition, the reflections contain noise, clutter from multi-path reflection or general radar reflection artifacts [91]. Detailed

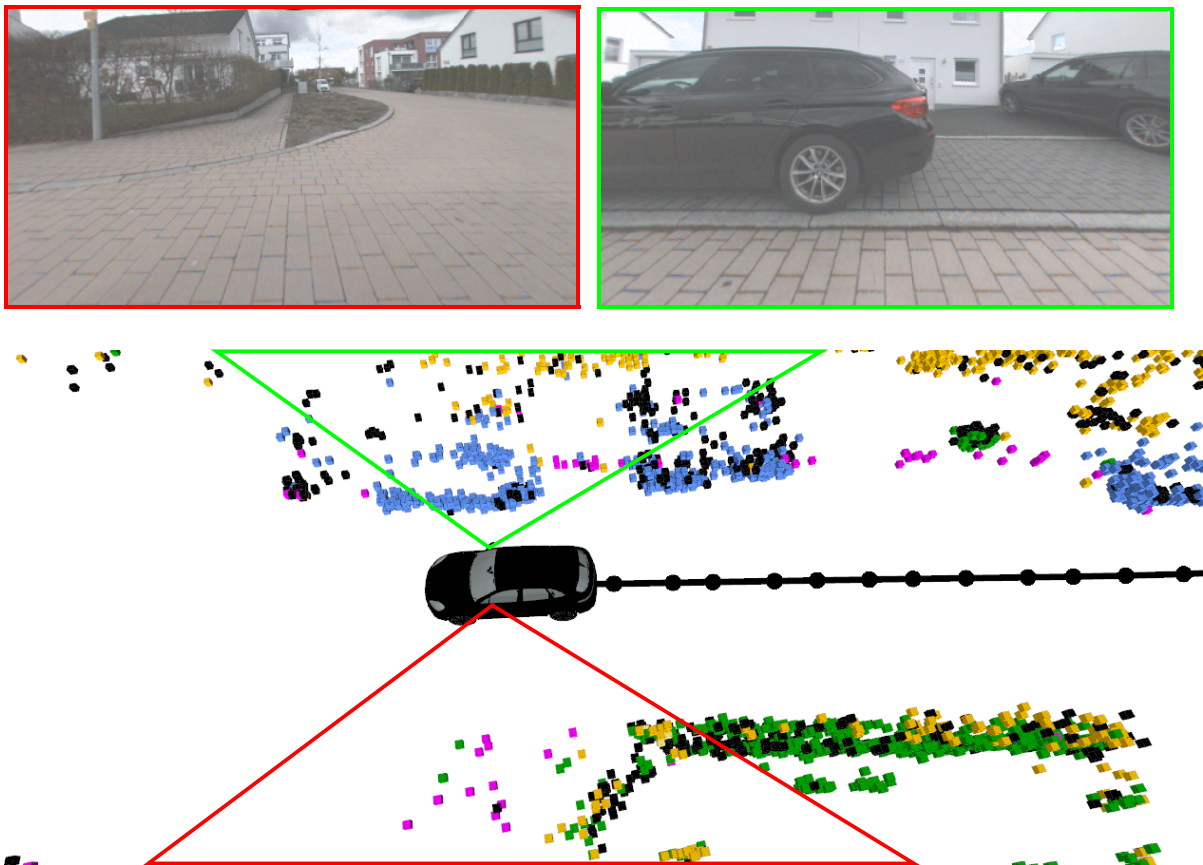


Figure 4.2: Semantic map building with 360 deg. automotive radar SLAM; mirror camera images (left/ right) for visual scene inspection. Figure of Isele et Al. [SI4], color encoding according to Table 3.2.

information on radar typical noise and clutter types are discussed in the work of Holder et al. [91].

As a result, local point neighborhood interpretation, which is common for dense point cloud processing, e.g. point set normal-vector calculation for plane detection or other local feature descriptors [179, 181], are impossible or deprecate for sparse radar point clouds. The multi-path reflections and specifically the remaining artifacts of multi-path reflections introduce the requirement to denoise the perceived point cloud before processing.

In order to cope with these challenges to find systematic solution the development of learning based solutions in this thesis is performed based on the existing and established reference data set framework SemanticKITTI [18]. This framework is set up as comparative data set for LiDAR based point cloud perception, e.g. segmentation, pan-optic segmentation and other point cloud based challenges, solved by different artificial neural networks or other machine learning techniques.

The SemanticKITTI reference data set framework established a data set structure which adopted in this thesis for the development of semantic radar segmentation, see Section 3.3.9. With the radar training data set structured in the same form of the SemanticKITTI benchmark data set, semantic segmentation network architectures for originally LiDAR semantic segmentation can be transferred with ease to the radar domain and be tested for applicability on real-world radar data.

The **problem definition** of point-wise semantic radar segmentation is formulated as multi-class classification problem. A raw radar point cloud \mathcal{P} , consists of $N \in \mathbb{R}^{n_i \times 6}$ points $p_{\text{radar},i}$ and is described by a label vector $y_i \in \mathbb{Z}^{n_i}$. Each point p_i is a data tuple of six attributes: Three spatial coordinates (x_i, y_i, z_i) and three radar specific measurands $(\text{SNR}_i, P_{\text{sig},i}, \nu_{D,i})$.

The describing label $y_i \in \mathbb{Z}^{n_i}$ is a semantic segmentation label for each point $p_{\text{radar},i}$ of \mathcal{P} . Given the input point cloud \mathcal{P} , the semantic segmentation function $f(\cdot, w_{\text{trainable}}, w_{\text{fixed}})$ predicts a semantic point-wise label \hat{y}_i for each point in the given input cloud \mathcal{P} . The trainable parameters $w_{\text{trainable}}$ in the semantic segmentation are learned to minimize a formalized difference between \hat{y}_i and y_i .

Existing solutions for this problem definition are found in a different sensor domain of dense, 4D LiDAR data. Classically, LiDAR delivers 3D spatial coordinates (x, y, z) and a fourth reflectivity or remission value. The transfer of LiDAR architectures to radar requires an adequate semantically labeled data set and two substantial architecture adaptations:

1) Radar Attributes: Specific network architecture adaptations are required to encode the additional radar specific input dimensions in the data loader structures as additional feature channels. Utilizing the informative quality of the specific radar attributes, the additional

dimensionality, from 3D LiDAR to 6D for radar point clouds, plays a major role in the network architecture transfer and applicability test. The additional channel data is read from extra binary files, normalized and applied as point attribute: *Signal Power* P_{sig} , *Signal Noise Ratio* SNR , and *Doppler Velocity* v_D .

2) Radar Applicable Classes: Based on the coherent and consistently reliable evaluation data set, SemanticKITTI serves as generic data structure to train and evaluate point-based segmentation models.

But as the SemanticKITTI benchmark data set utilizes 22 semantic classes, the radar applicable set of specifically radar-applicable classes needs to be consolidated. The remaining radar specific and radar detectable subset of six remaining classes serves as minimal class set of generic semantic radar classes. A detailed definition of this consolidation process is found in Table 3.2. The label-map is modified to yield for all semantic radar segmentation networks only the six radar applicable classes, encoded per one hot encoding per point. This network configuration addresses the final layer of the segmentation backbone, which compresses to the number of N_C classes to N_C output neurons. A visualization is found in Figure 4.14.

After establishing the two architectural adaptations and the class consolidation, the radar-consolidated six semantic classes pose a multi-class classification problem. Both two classes *building* and *pole* share static structural man-made elements, which could be interpreted interchangeably, but are preferably labeled as *pole*. In the application of mapping, poles are specifically relevant since they provide well determinable fix-points or so called landmarks. Further information to radar typical noise and clutter types can be found in the work of Holder et al. [91].

With the adaptations realized, a quick transfer and check of approved LiDAR segmentation models and network architectures for the semantic radar segmentation can be achieved. To avoid a domain shift, the networks are trained from scratch on a dedicated real-world semantic radar data set, no pre-trained weights have been tested to avoid the domain shift.¹

With this Section of the work, these contributions are addressed:

1. A first model transfer of point cloud based generic (static & dynamic) semantic segmentation from LiDAR sensor domain into semantic radar segmentation domain.
2. Data structure transfer between sensors to unify a common point cloud data standard, instead of non-comparable, domain specific model adaptation and data-loaders.
3. A systematic comparison of semantic segmentation models on radar point cloud data.

¹ An overview of currently available public radar data sets is found in Section 3.1. The above mentioned radar signal attributes are not consistent with the available automotive radar data sets. Also, the applied radar sensors differ from the data sets in their mounting position and point cloud density.

4.2 Evaluation and Selection of existing Approaches: Radar Segmentation

For semantic segmentation tasks, multiple approaches have been applied until recent. As first introduction to this task, an overview of general semantic segmentation methods is presented in the remainder of this section.

There exist four general network input modalities, as Schumann [192] describes: 2D Images, 3D Point Clouds, 3D Voxels or a Graph-based input. Each input modality is presented briefly in the Sections 4.2.1-4.2.4, plus their appropriateness and required conversion is evaluated from a radar point cloud perspective for each modality. With the selection of the most promising modality, Section 4.3 derives a structural architecture from reference examples, which is further optimized in Section 4.4 and evaluated in Section 4.5.

4.2.1 2D Image Segmentation Approaches

For point cloud applications, image-based architectures are referred to as *indirect* point cloud approaches since a conversion into an image is necessary.

The field of image based semantic segmentation is based on the progress of CNNs. With the transfer of CNN based approaches from images to other grid-represented sensor data of e.g. camera, based on data projection to images, the CNN application also for non-imaging sensor data advanced.

Image retrieval from point clouds involves projection of multi-dimensional data to a 2D image grid, encoding a third dimension in the rgb- or gray-scale image pixel values. Classical image processing steps, e.g. CNN approaches, can be applied after the point cloud projection as 2D image. Sheeny et al. [200] suggest to represent the radar data as grey-scale image, in order to apply image processing by CNNs. This approach yields a 35% success rate for a single object, being tested in ideal conditions. Aerial radar data of Synthetic Aperture Radar (SAR) images, is processed as image with classic CNNs [97]. And Ouaknine et al. [157] utilizes multiple radar histogram representations in a multi-input CNN with encoder-decoder structure architecture to detect three dynamic classes (car, cyclist, pedestrian, background).

In top view projections of the radar data, the spatial distribution of points in x-y coordinates is discretized to a fixed resolution image grid. The pixel resolution of the image representation defines the spatial discretization.

With a spatial top-down projection yielding an image pixel location, the remaining third dimension of the image pixel value remains open to be defined. There exist approaches of multi-view heat maps representing a radar spectrum as image data [157], or also (normalized) density probability functions [125].

If such radar images are generated and accumulated as map-images, besides the classification of objects, the environmental properties can be distinguished. Lombacher et al. [129] is the first work, researching static object detection in automotive radar data and identifies 17 static object classes. The authors propose to benefit of the static properties by accumulating the radar detections over time and motion in order to represent the radar data as grid map. The classification is suggested to be applied on radar grid map patches, being classified by a CNN architecture, but does not yield experiments on the total suggested set of static object classes. In a later work, Lombacher et al. [130] classify the static classes *car*, *other* and *unlabeled* with the same CNN per grid approach, but the authors suggest to extend the set of classes due to promising classification results.

Werber et al. [222] refines the work on radar grid map processing, achieving suitable maps based on radar cross section and occupancy for radar self-localization tasks. Lombacher et al. [130] applies the mentioned radar grid maps, containing accumulated static radar detections, to be cell-wise classified semantically by means of a CNN architecture. The static classes *car*, *other* and *unlabeled* are distinguished in this work, but the authors suggest to extend the set of classes due to promising classification results.

Extending this approach, Prophet et al. [168] implement the idea of a clustering free direct radar classification by the means of modified image segmentation CNNs to predict semantic grid maps. Tested on static real-world grid maps, the proposed network classifies 5 static classes (Background, Street, Barrier, Car and Small Obstacle). The authors claim the method to require more training data to be further improved.

Schumann et al. [193] suggests a separated processing of dynamic and static radar detections. Combining the grid map accumulation for the static objects, grid patches of the radar grid map are classified and the corresponding 3D points annotated. In parallel, the dynamic branch processed the radar point cloud directly to a neural network for semantic segmentation. Fusing both dynamic and static points together, the combined point cloud is semantically separated.

An alternative projection method is applied for e.g. circular LiDAR sensor data. A sensor field-of-view specific projection of (x, y, z) -coordinates yields for a 360° sensing circular point cloud a cylindrical panoramic image. This projected images commonly encode the range of the 3D detection coordinate as brightness value of a pixel. This result is alternatively named cylindrical *range image*. As bidirectionally unique projection, the mapping benefits from static (x, z) vector for each discrete azimuthal sensing. The fixed resolution grid-structure in (x, z) dimension is inherently given by rotational axis z , resulting in no overlapping points at a 360° revolution.

Exemplary the RangeNet++ [145] architecture, which is applied for LiDAR processing in Section 3.3.6, generates a cylindrical range image from the point cloud data and further processes

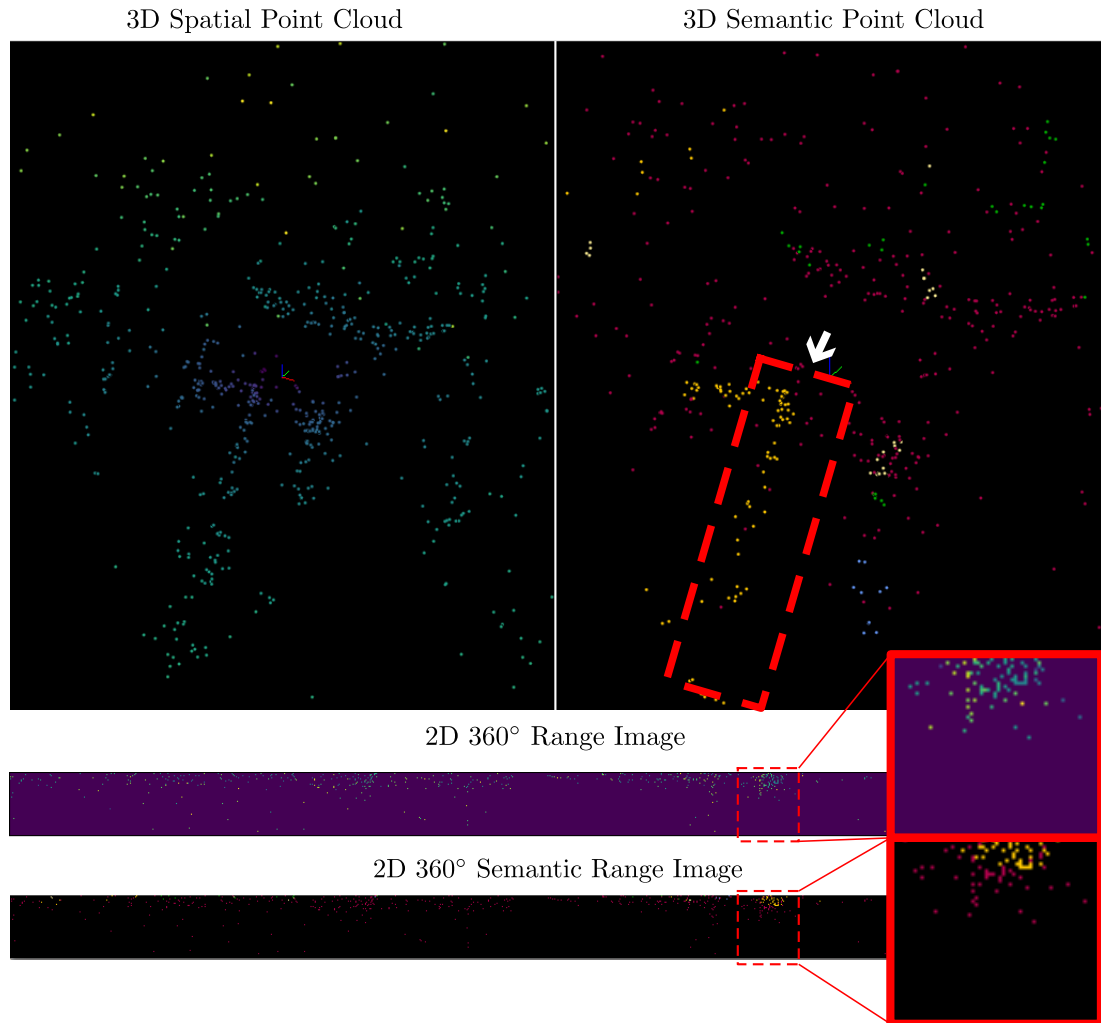


Figure 4.3: Exemplary conversion example of the same 3D point cloud in top view with spatial attributes (top left) or with semantic attributes (top right), to the corresponding spatial 2D range images (top landscape image) and semantic 2D range image (bottom landscape image.) The 3D spatial point cloud and the projected 2D range image depicts the range as rgb value. The semantic 3D point cloud and the 2D semantic range image color the semantic class as rgb-channel according to Table 3.2.

this image with a CNN. Applications of this approach are commonly applied for dense LiDAR point clouds, e.g. RangeNet++ [145], SalsaNext [48], RandLA [95], etc.

Image-based Radar Segmentation Examples:

Evaluation for Radar Segmentation: Dense point clouds allow a reasonable, dense image representation, which is beneficial to process local context image features by pixel convolutions. From an information-theoretical point of view, an image or heat map based representation structurally includes an information loss. The information dimension of a 2D image is inferior to the arbitrary general 3D representation of its multi-dimensional attribute vector.

With the sensor setup of this thesis, described in Section 2.7, the assembly of six radar sensors delivers theoretically up to 6×600 points per cycle, compared to a centuple number of points

per LiDAR scan. But under real-world conditions, the sensor assembly delivers less detections of approximately 1200 points per cycle. The pixel-wise projection is yet not a run-time limiting factor in terms of computational overhead, since the projection effort is low. But the general projective overhead of a potential image representation remains.

Radar point cloud sensings are a data form of unordered points. The ray-tracing projection approach for the line of sight is not bidirectionally unique for radar point clouds or radar range images. Points can be overlapping due to multi-path reflections and be accumulated in space or range, resulting in un-determinable pixel clusters. From an inspection of the images, radar range images are empirically estimated to be inappropriate. An exemplary illustration is found in Figure 4.3.

4.2.2 3D Point Cloud Segmentation Approaches

Point-based *direct* point cloud learning architectures directly apply the 2D or 3D data of the point set as input. From the given set of multi-dimensional input attributes, a feature vector representation is computed which is applied for further processing. Compared to an *indirect* representation as images, by skipping the necessary 2D projection, the limited input dimensionality of 2D (rgb- or grayscale-) images can be overcome to an unlimited-dimensional input. Each 3D position is flexibly equipped by an additional description vector containing an unlimited count of point attributes. Compare this to an image pixel of only two pixel coordinates (u, v) , with maximum three potential signals per rgb-channel. Hence, the input form of images is generally limited to 5-dimensional inputs if not applied as Tensor.

Applicable to any form of point clouds, direct point cloud processing also allows sparse and multi-dimensional point attributes of point clouds to be utilized and processed in the feature extraction. From point clouds, the 3D point structure, shape and relative local neighborhood information is directly processed to express feature vectors or input tensors for a neural network architecture. A flat image can hardly encode a 3D shape at all.

The pioneering PointNet of Qi et al. [169] introduced a method to extract features from the unordered local neighborhood of 2D or 3D points and aggregate a feature representation of a point cloud in a fixed-size vector. Capable to deal with unordered data of arbitrary order, extracting information from the local spatial relative context of points, while extracting features independent of the input point cloud orientation, this work delivers a robust basis for feature extraction.

In subsequent works Qi et al. [170], suggest PointNet++, which enriches the feature extraction of PointNet to respect hierarchical levels of point features. The idea of generating hierarchical point sub-sets and encode the local set in representative features is pioneered in this work. PointCNN [126] and other architectures apply a 2D or 3D convolution on extracted feature tensors in order to capture the general local consistency [220, 62].

Since the perception of images implicitly depends on the pixel location of an image feature, the generalization to other image pixel locations is not given in general and requires further steps, e.g. data augmentation. In contrast, the feature extraction of a point cloud, is rotation and orientation invariant, if the local point context is extracted as feature, e.g. by PointNet [169].

Mainly focusing on improving the spatial relative context of local structures in the point cloud in an encoding, the feature extraction step is discussed in a variety of works.

Direct Point Cloud Strategies: Scheiner et al. [185] and Scheiner et al. [186] detect six dynamic classes (Pedestrian (+Group), Bike, Car, Truck, Garbage), not including any type of static objects, as vulnerable road users (VRUs), from the sparse radar data. The authors suggest a classifier ensemble, applying in each classifier a recurrent neural network which learned a set of 98 features from a labeled data set, to recognize these features in a radar scan, yielding a clustering and two-stage clustering respectively of "known" scan content to VRUs.

Wöhler et al. [226] extends the work of Scheiner et al. [185], by comparing a random forest classifier to a long-short term memory (LSTM) neural network to classify the six classes of VRUs. The findings indicate the sparsity of radar data as drawback of direct classification. Hence, the random forrest classifier is trained with feature vectors over a sliding window, whereas the LSTM is trained on a sequence length of 8 consecutive radar scans. Overall, the LSTM achieves slightly better classification results.

Danzer et al. [50] suggests to apply a PointNet [169, 170] to generate 2D bounding box hypotheses from the sparse radar data, but projected onto a top-view 2D plane. The method is trained to distinguish between *cars* and *clutter*, given a 4D input vector of point coordinates (x, y) , Doppler velocity v_D and radar cross section RCS of one single radar scan. Cennamo et al. [34] suggests the RadarPCNN network for the similar task.

Dreher et al. [58] achieves significant run-time improvements with a YOLOv3 [173] inspired grid-based CNN. Therefore the sparse radar point cloud is transformed into a grid structure. Moreover the grid-based approach achieves real-time inference times, whereas direct point cloud methods based on PointNet achieve higher classification accuracy on five classes (Pedestrian, Object, Two-Wheeler, Car, Truck).

The work of Schumann et al. [193] introduces a direct processing of radar point cloud with a fully connected neural network, but only for the dynamic object detection, excluding the static environment.

Evaluation for Radar Segmentation: Sparsity and missing or occluded parts of a point cloud is often taken as degradation example of an ordered point cloud, LiDAR sensed objects as well as other point representations [76]. In contrast, radar detections in the form of detection point clouds deliver only sparse data and are hard to interpret. As such, the radar point cloud processing is expected to be challenging even under ideal radar sensing conditions.

In Addition, compared to LiDAR sensors, the radar detection point number is variable in subsequent scans. With this, the representation of a radar scan as ordered structure e.g. a range-image results in a significantly changing input for a semantic segmentation module, if there is no e.g. normalizing or padding module for missing or few detections in the feature extraction.

Secondary, even if the radar scans are sampled with a low time delay (50 *ms* per scan), the spatial distribution of detections can vary significantly. To cope with the spatial noise in subsequent scans, the dynamically changing number of detections and e.g. multi-path reflections of the sensor data, an abstract but constant input for the segmentation is necessary. A fixed-size feature vector of the point cloud feature extraction delivers this property.

4.2.3 Voxel Segmentation Approaches

Binning points by their 3D coordinates into a 3D grid cells, voxelation describes to organize the unordered point cloud into smaller sub-sets per volume. Splitting the whole point cloud in smaller volumes results in a 3D grid discretization of voxels. Voxel approaches follow the divide and conquer approach, to contain the local point cloud context per voxel. The idea to sub-divide a point cloud into neighborhoods of a voxel which can be treated independently, assumes dense point clouds, stable spatial neighborhoods and distinctive point sets per object. For the whole volume, a unique representation is calculated per volume representation, not for individual point value representation. The regular 3D structure of the voxel grid subsequently allows to apply 3D convolutions on the feature vectors per voxel.

Evaluation for Radar Segmentation: Considering a sensor with potentially high sensing range, the major drawback gets obvious. Depending on the scene and sensed objects, numerous voxels remain empty for an open environment scene. The drawback to process empty or sparsely occupied voxels causes computational inefficiency [227].

Second shortcoming is the necessity to discretize the sensed volume. VoxelNet [252] applies a object detection directly per voxel. The selection of a voxel resolution is non-trivial and requires intense application tuning or dense point samples in the respective RoI to detect and segment the point cloud adequately. As an alternative approach, Kd-networks [117] share transformations in the local the Kd-tree structure instead of a formalized grid structure, but are not commonly applied. Exemplary 3DContextNet [245] applies this approach for segmentation and classification.

Summarizing, the sparsity of radar point cloud prohibits to apply a voxelization approach. The diversity of object shapes and object sizes is too sparse to fine-tune a volume discretization but still contain sufficient neighborhood information in each voxel.

4.2.4 Graph-based Segmentation Approaches

Originating from the knowledge representation of graph-similar structures [183], graph neural networks (GNNs) model the node connecting edges as dependencies between nodes and e.g. predict a label of a node. Preserving the context information modeled by edges, the context representation is similarly applicable to point cloud data. TGNet Li et al. [127] suggests a geometric Graph CNN, which exploits a graph pyramid of hierarchically grouped points convolutions. A Gaussian weighted Taylor Kernels and a pooling layer is applied to extract the features of the point graph representation.

Chen et al. [37] find their suggested local information and hierarchical information preserving hierarchical attentive pooling graph network HAPGN superior to other common classification methods and networks. Especially interesting for a potential radar application is the methods' superior robustness against missing point data and gaussian noise, compared to a classical PointNet-based segmentation. But, the drawback of the presented method is its non real-time capable run-time compared to PointNet or PointNet++.

Also considering point attention in an GNN, AGNet [105] proposes an attention approach to increase the topological information of the extracted point features. Especially interesting is the test on decreasing input point density, yielding decreasing classification accuracy. A similar significant deprecation is expected for a sparse radar point cloud, since the AGNet is only tested on synthetic dense point cloud objects.

Similar, RMGnet [67] represents the local point context in a graph representation, but based on handcrafted 3D shape features of multiple scales. Achieving rotation invariant point cloud segmentation results, the intermediate representation in a graph enables graph-convolutions and graph-downsampling to extract also multiple-scale information. But, the method is only tested for synthetic point-sets.

Also tested on synthetic data, the HyNet [198] applies a given mesh instead of points to be converted to a hybrid graph. The nodes in the proposed graph describe the geometrical mesh-face properties in 8 dimensions. Hence, the method suffers from sparse input point-sets or mesh. With attention layers and an MLP classifier, the network processes the graph represented mesh features to the segmentation. Generally, also other mesh based approaches[209] do not transfer to the sparsity of e.g. radar point clouds.

Zhang et al. [247] propose to combine the voxel feature extraction of a sparse 3D-convolution U-Net, with a direct point feature processing in a joint GNN network. Both feature extraction branches are supposed to extract different local point context information, which the authors process in a final conditional random field. The total inference time of the overall process is by far not real-time capable.

Zhao et al. [249] suggest multi-scale supervoxels which are connected to an inverse node graph. The graph is divided into parts according to the edge connections and processed by a conditional random field. This segmentation is processed with a random forest classifier and refined in a second higher-order conditional random field processing. Despite good segmentation results, the method is not designed as online real-time segmentation, but as offline segmentation.

Designed for instance segmentation of a given point cloud with a describing query sentence, Huang et al. [96] apply a text-guided GNN not for the feature extraction but for the segmentation reasoning. Extracted semantic masks from a 3D U-Net feature extractor and a gated recurrent unit output of the query sentence, are processed jointly with an text-guided GNN to predict the instance positions. Different from the other approaches, the GNN is not applied for feature extraction but for the final reasoning step, as segmentation head.

Instead of relying on radar point clouds as graph, Meyer et al. [144] represent the earlier radar processing stage of the radar spectrum tensor as graph. Combining the GNN as local context extraction, a subsequent 2D Pyramid ResNet is applied as backbone model to generate 3D bounding boxes. Hence, the focus of this supposed architecture addresses specifically dynamic objects and yields an improvement over classical CNN architectures.

Evaluation for Radar Segmentation: As conclusion of GNN approaches, the most methods involve a graph representation of the input point cloud which is treated with graph-convolutions. Similar to the range image projection overhead, GNN methods require the graph-representation as feature extraction step, and therewith a longer processing time compared to a direct point cloud feature extractions. Also, the sparsity of radar point clouds complicates this information representation as graph, e.g. due to sparse geometrical data, but the only work on radar finds an improvement compared to classical CNNs [144]. But, the approaches applying a graph as hierarchical point-feature connecting element is assumed to be similar effective as classical attention-aware point operations.

Table 4.1: Evaluation of different semantic segmentation approaches, rating the applicability to perform on sparse radar point clouds.

| Approach | Dimension > 2D | Order Independent | Input Adaption | Bijective |
|-------------|----------------|-------------------|------------------------|---------------|
| 2D Image | x | x | 2D Projective Overhead | not for radar |
| 3D Point | ✓ | ✓ | - | ✓ |
| 3D Voxel | ✓ | ✓ | Volume Discretization | x |
| Graph-based | ✓ | - | Graph Association | (✓) |

4.3 Direct Point Cloud based Radar Segmentation

Based on the Sections 4.2.1-4.2.4, a network architecture from the summarizing Table 4.1 is selected and summarized in the following. The native data structure of sparse radar point clouds are unordered point clouds. This data structure results from two effects. First, the dynamic number of reflection points that a radar sensor samples each sensing cycle. Second, compared to LiDAR sensors perceiving point clouds as ordered structure, the radar sensing does not perceive points along a static sampling pattern. Radar point clouds arrive as unordered data structure with a variable number of points and variable point distribution for each scan. Instead of images or discretized voxels, the direct processing of point cloud data as individual 3D points is chosen to be evaluated and tested in this thesis, avoiding any input representation pre-processing.

Inspired from the LiDAR semantic segmentation on 3D point basis, the sparse radar point classification addresses the long tail of sparse, noisy and multi-dimensional point cloud processing. As sensor systematic challenge, for the sparse radar data at hand a dense spatial context of dense 3D point clouds is not available. The additional multi-dimensional radar specific attributes are assumed to balance this drawback. The aim is to achieve a new level of data segmentation and semantic classification performance for radar, directly processing the sparse spatial radar point cloud, boosted by including the multi-dimensional radar specific attributes.

A point-based approach is selected to be investigated in this thesis, based on the advantages of the direct and point-wise semantic segmentation. Further focus is on classification accuracy and approach appropriateness of this selection in the context of radar data.

The overview of radar segmentation examples in Section 4.2 reveals, besides the multitude of radar-based approaches, the lack of a point-based direct classification of the static environment. Besides the work of Schumann et al. [193], which provides point-wise semantic radar labels, no general applicable semantic segmentation approach comparable to a LiDAR semantic segmentation, e.g. RangeNet++[145], is known. The limited static object detection approaches lack detailed classes and more-over a labeled point-wise annotated data set. Since the publication of the RadarScenes [194] data set, focusing on dynamic object and instance detection, the research on this problem increased significantly [128], but leaving the static segmentation often unconsidered.

A selection of three LiDAR solutions is presented in the next sections and tested on the data set of Chapter 3. Among a set of three selected architectures PolarNet, Cylinder3D, and ASAP-Net, introduced in Section 4.3.2, Section 4.3.3, and Section 4.3.4 respectively, a radar-adaption is for these architectures is described in Section 4.3.5. The basis of the further steps is the semantic radar data set of Chapter 3. The data set in the format of SemanticKITTI allows a quick architecture adaption to radar-specific feature-channels, pre-processing, training methods

and the application of the same benchmark performance evaluation methods. Consequently, the adaptations and architectures' segmentation performance is tested in Section 4.3.6. Finding the most promising setup, based on the segmentation performance, a single architecture is selected to be further optimized. Finally, the selected network architecture is further fine-tuned and hyper-parameters adjusted in Section 4.4.

Network Selection: To achieve an independent model architecture, applicable for general application, the tested network architectures are selected to be single-frame approaches. For a robust behaviour, single-frame approaches offer the most flexibility, not being limited to a specific data fusion or collection pipeline. Directly processing the raw point cloud data promises the most flexible application, also run-time wise, if the semantic classification is required to be real-time applicable since the output is further processed in perception modules.

The semantic segmentation network is supposed to run in live-operation, integrated in the test vehicle setup. Hence, on-board live inference requires a high inference rate and disqualifies slow architectures in advance. With the selection of direct 3D point approaches, the processing of the few sparse points ensures a low processing time, plus a potentially larger projective overhead e.g. for image-based approaches is systematically avoided.

At the time of network approach selection, the semanticKITTI leader-board was considered as approach selection catalogue, providing a fair evaluation of promising and available semantic segmentation architectures. Table 4.2 provides a non-necessarily complete overview of the available network performances.²

² Developed performance advantages of other architectures in the meantime are not considered in this table.

| Network | Year | Point-based | Image-based | Voxel-based | mIoU SemKITTI | mIoU nuScenes |
|--------------------------------------|------|-------------|-------------|-------------|---------------|---------------|
| AF2S3Net [43] | 2021 | ✓ | x | ✓ | 69.7 | 78.3 |
| Cylinder3D [250] | 2020 | ✓ | x | ✓ | 67.8 | 77.9 |
| SPVNAS [208] | 2020 | ✓ | x | ✓ | 66.4 | 77.4 |
| JS3C-Net [230] | 2020 | ✓ | x | ✓ | 66.0 | - |
| SalsaNext [48] | 2020 | x | ✓ | x | 59.5 | - |
| KPConv [211] | 2019 | ✓ | x | x | 58.8 | - |
| SqueezeSegV3 [229] | 2020 | x | ✓ | x | 55.9 | - |
| PolarNet [248] | 2020 | ✓ | x | x | 54.3 | 69.4 |
| RandLA-Net [95] | 2020 | ✓ | x | x | 53.9 | - |
| RangeNet++ [145] | 2019 | x | ✓ | x | 52.2 | - |
| ASAP-Net (Backbone: PointNet++) [31] | 2020 | ✓ | x | x | 35.3 | - |

Table 4.2: Overview of public semantic segmentation networks on the SemanticKITTI leaderboard with public implementation. Selection based on performance on SemanticKITTI [18] and nuScenes [30] data.

For a systematic comparison of structural elements as feature extraction and segmentation heads, the non comprehensive selection of tested network architectures include three structurally similar architectures. As comparison, Figure 4.4 illustrates the structural setup of the selected architectures.

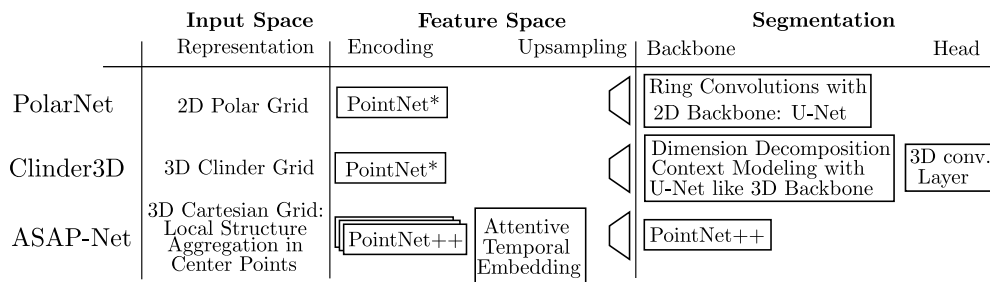


Figure 4.4: Evaluated models architectural network comparison: PolarNet[248], Cylinder3D [250], and ASAP-Net [31]. The authors of these works apply a PointNet-inspired feature extractor, mainly implemented as fully-connected MLP layers.

Based on their architectural similarity and good performance on the SemanticKITTI semantic segmentation challenge data set, see Table 4.2, three point-based, single-shot approaches are selected: PolarNet [248], the Cylinder3D [250] Network, and the ASAP-Net [31] with local

feature extraction and temporal feature aggregation. These three architectures are introduced in the Sections 4.3.2/ 4.3.3/ 4.3.4 respectively. The selected architectures gradually vary in their feature extraction and feature processing steps, allowing a comparison of effectiveness on sparse radar data.

To cope with the sparse nature of radar point clouds, the feature extraction needs to be efficient but also robust for radar-typical noise in subsequent scans. As established feature extraction method in general point cloud applications, specifically PointNet-based applications, has been tested and applied in comparable radar learning works [192]. Nevertheless, an extensive test on real-time segmentation of sparse 360° radar point clouds has not been tested yet.

PolarNet [31] serves as basis model and constitutes the first concept architecture. KPConv [211], RandLA-Net [95] and PolarNet [248] are also point-based, but rely on Cartesian coordinates, whereas PolarNet discretizes the input points in polar coordinates. The benefit of polar coordinates is rated to yield beneficial results for sparse radar data, compared to superior mIoU of KPConv.

Cylinder3D [250] extends the 2D polar discretization of PolarNet with a z-coordinate to a 3D approach. Similar to SPVNAS [208], JS3C-Net [231], Cylinder3D is based on Sparse 3D Convolutions, allowing a 3D CNN to be applied as backbone model. Compliant to the PolarNet baseline model, also polar grid discretization is applied, which seems suitable for radar data. Cylinder3D network also outperforms the other variants and is selected as second model to be tested.

Third, the ASAP-Net [31] architecture incorporates temporal consistency by a concatenated feature map of previously processed frames. Technically remaining a single-frame network with a memory of the previous feature-map the ASAP-Net is tested for comparison. ASAP-Net applies different backbones in their publication. For a consistent comparison, the PointNet++ backbone is selected.

Architecturally similar structures of feature extraction of 3D points in the form of PointNet++ variants, provide an abstract and radar point-count independent input tensor per radar input cloud. Feature processing in the Form of 2D or 3D U-Net [176] backbone network with multi-scale convolution and bypasses, extract context features for the segmentation backbone and feature embedding. Per approach different segmentation heads result in the different semantic segmentation performances of the three network architectures to be compared. With the selection of these three architectures, a systematic comparison of the networks' suitable transfer to radar is tested.

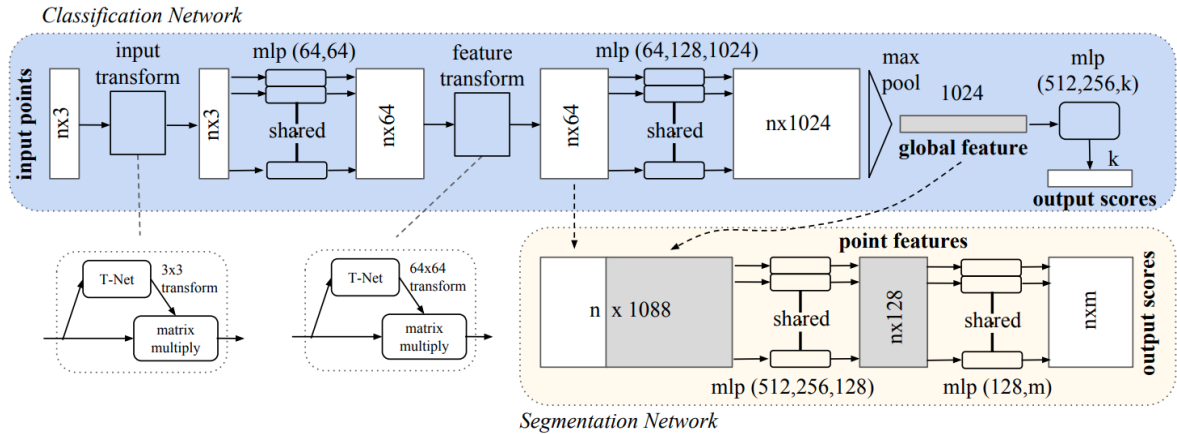


Figure 4.5: Illustration of the PointNet architectural principle, illustration from Qi et al. [169].

4.3.1 PointNet and PointNet++

PointNet: Qi et al. [169] proposed a network architecture which inputs a set of points and computes a fixed size feature vector based on the 3D information of (x, y, z) coordinates. To select informative points of an unordered input, a symmetric max-pooling extracts only the maximum activated point per kernel convolution. With subsequent fully-connected layers, the pooled data is processed to a global descriptor of the input point cloud. The symmetric max-pooling allows to remain independent of the order, but rely only on the activation magnitude, expressing local information.

The general idea of PointNet is to learn representative feature vectors from multiple local spatial regions in the point cloud, and further process these local feature vectors to a global point cloud signature.

One major advantage of this architecture is the capability of variable number of inputs points. This technical advantage facilitates to vary the input, independently of the subsequent output-processing modules. Not even for the number of points, but also for the number of features per point. Originally designed to only process (x, y, z) coordinates, the processed channels can be adapted to process also additional channels per point.

PointNet++: PointNet++ [170] improves the local spatial structure encoding. Based on recursively applied PointNet feature extractions on convoluted input grouping stages, the feature extraction is achieved on multiple scales of the point cloud. Contextual information is hereby extracted in close neighborhood, processed and grouped so that also the larger scale of the scene context is represented.

In PointNet, max-pooling operations achieve the aggregation of local features to a global point cloud features. Drawback of the pooling operation, significant local attributes are not detected by the maximum selection. Especially to improve the local feature content,

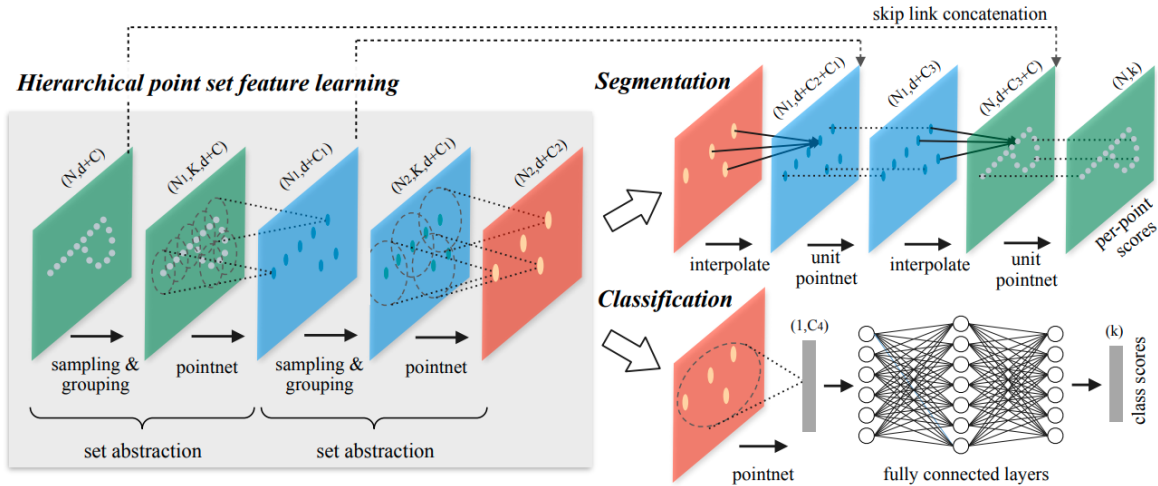


Figure 4.6: Illustration of the PointNet++ architectural principle of hierarchical feature extraction for 2D points, illustration from Qi et al. [170].

PointNet++ introduces the step-wise hierarchical grouping and new feature extraction on a new grouped set of elements. This procedure can operationally be expressed in three steps.

- **Sampling:** The set of input points is clustered to local centroids.
- **Grouping:** The local neighborhood of each centroid is constructed as local point set.
- **Feature Extraction:** The local point sets are computed by a PointNet to a local feature vector per centroid.

One drawback remains, the explicit neighborhood search in the grouping step needs to be performed. This neighborhood search is rated time-consuming since it resembles a computationally expensive k-NN search [197, 150].

4.3.2 PolarNet

Based on a polar grid with range ρ and azimuth angle θ , the input 3D point cloud discretization (x, y, z) yields a 2D description (ρ, θ) , but independent of the points' z -coordinates. Polar grids are well suited for circular point clouds e.g. LiDAR point clouds, due to the quasi-native representation of signals in azimuth-angle and range coordinates. Other than in Cartesian grid representation, the polar grid cells reduce the amount of empty cells [248].

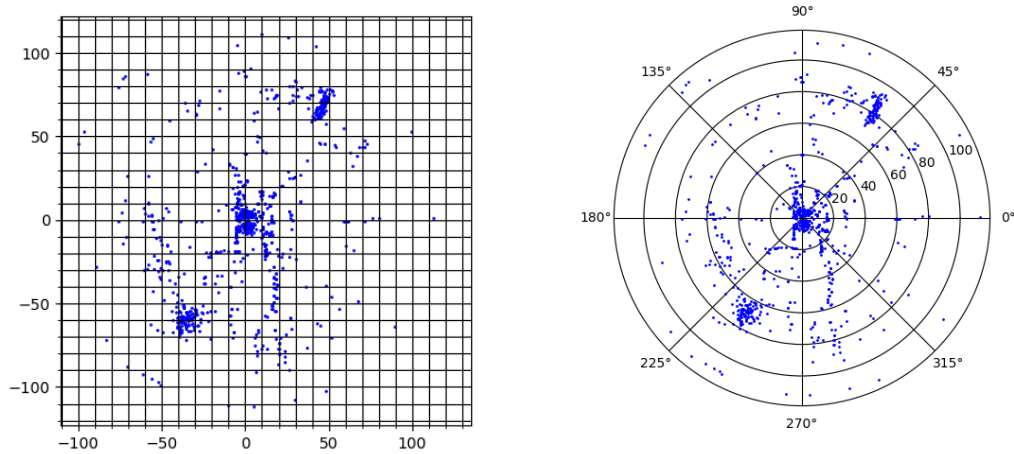


Figure 4.7: Illustration of a Cartesian grid discretization and a polar grid discretization of an exemplary point cloud $\mathcal{P}_{\text{radar}}$. Compared to the static cell area of an cartesian grid, the polar grid yields area-wise increasing regions with increasing radius. Hence, less polar grid cells remain empty compared to a regular cartesian grid.

By the radius increasing grid cell area, an improved homogeneous distribution of radar points over the grid cells is achieved. Contrary, Cartesian grids of fixed size would result in higher point counts near the sensor and sparse and empty cells in the far distance. For input tensors for neural networks, padding or zero fields can lead to biases. Ideally normal distributed data is preferred, hence the more homogeneously distributed data in grid cells is beneficial.

Per cell of the polar grid, the discretized points are processed to feature vectors by a PointNet inspired feature extracting MLP. Since the vectors contain a N_C dimensional feature vector, for the classification of N_C classes, max-pooling is applied to extract the dominant feature per class in each cell feature vector. This max-pooling allows a nearest neighbor free, order independent feature extraction. The cell resolution specifically influences the potential segmentation resolution, if multiple points fall into the same 3D grid volume.

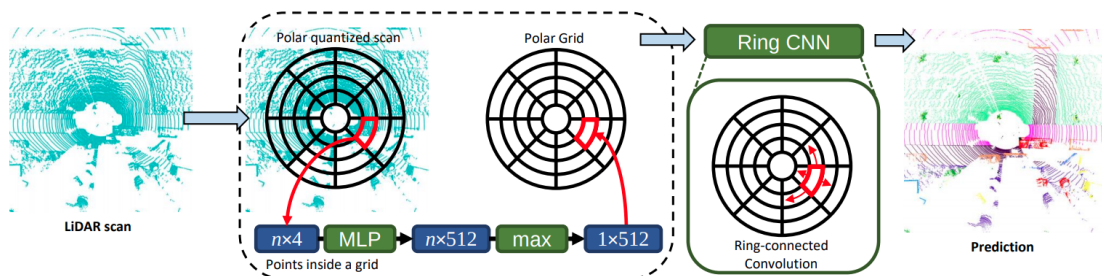


Figure 4.8: Illustration of the PolarNet architectural principle, illustration of Zhang et al. [248]. The depicted MLP block is inspired from the PointNet [169] architecture but simplified to a fully connected MLP. The ring convolutions are applied in a not depicted U-Net [176] CNN backbone with skip-connections.

The resulting maximum features per cell are subsequently processed by a convolution layer, which is adapted as convolution, according to the polar grid, in ring form. The convolution kernel propagates the information in azimuthal direction, allowing neighboring cells to propagate a gradient information in azimuthal direction. Originally designed for circular properties of LiDAR point clouds, this ring convolution emphasizes the technical principle of radial point neighborhoods.

Subsequently, the classification head in form of a U-Net [176] architecture block is fed. The scale-independent feature down-sampling, skip-connections and re-concatenation in U-Net is applied to further extract connections of local attributes of the max pooled feature vectors and relate it to a global context of the point cloud.

A final fully connected layer and soft-max operation delivers a label of the available N_C classes per cell. Projecting the cell' label to the corresponding 3D original input points, the segmentation attribute is attached and related to the points per cell.

4.3.3 Cylinder3D

Zhang et al. [248] propose to further enable spatial information extraction in real 3D coordinates. This architecture complements the idea of a 2D polar discretization in (ρ, θ) with additional z -coordinate to a 3D grid. Similar to the PolarNet, per cell a feature extraction to a fixed-size feature vector is performed by the application of a PointNet variant as 4 layer MLP with BatchNorm and ReLu. Details can be found in the original publication of Zhou et al. [250].

Working with a 3D grid, a special form of sparse 3D convolutions [80] is applied to utilize a 3D kernel, parameterized with Stride = 2. Special property of the applied convolution is the asymmetric residual block. Originally designed for automotive application with LiDAR data, with special interest to detect the asymmetric shapes of vehicles and cut computational efforts, the authors motivate this residual block.

The spatial context of the down- and up-sampled features is further processed in a dedicated Dimension-Decomposition based Context Modeling (DDCM) block. Since arbitrary objects yield different shapes 3D LiDAR point clouds, the authors suggest a low-rank context to be computed from the high-rank context based on height, width and depth properties. With low-rank convolutions, the contextual information is aimed to be processed with at higher computational efficiency, while the context information is extracted with respect to the local object shape. As segmentation head, the authors propose a light-weight 3D convolution layer with a $3 \times 3 \times 3$ kernel.

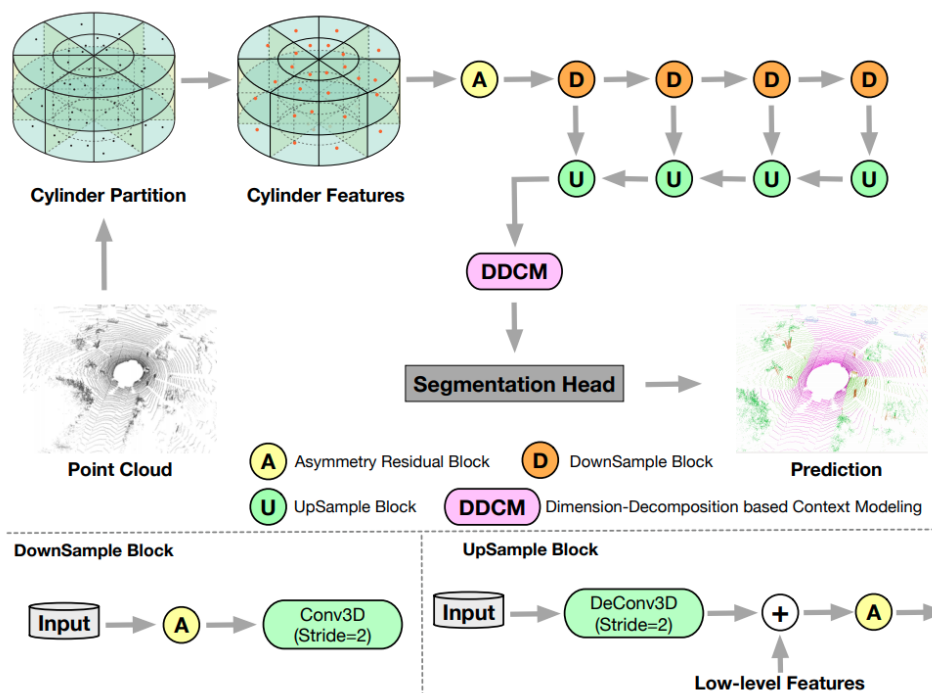


Figure 4.9: Illustration of the Cylinder3D architectural principle as presented by Zhou et al. [250]. The 3D input discretization is processed with sparse 3D convolutions in a 3D U-Net backbone, followed by a Dimension-Decomposition based Context Modeling (DDCM) block and a segmentation head.

The Cylinder3D architecture works similarly to PolarNets' feature extraction, applies a similar feature backbone, but performing feature downsampling and upsampling by 3D sparse convolutions. The segmentation includes a special dimension decomposition block but similar segmentation head. A combination of Lovasz-softmax loss [20] and cross-entropy is applied with class weights.

4.3.4 ASAP-Net

Different from PolarNet and Cylinder3D, the ASAP-Net not only processes one single point cloud scan at a time, but includes additional attention and temporal context.

Concurrent to PolarNet and Cylinder3D, the feature extraction step of single 3D input point clouds is performed with a PointNet++ variant. The authors address this as backbone block. Alternatively the authors propose SqueezeSegV2 [228] as backbone variant, which results in a spherical grid projection and convolutions of 3D feature vectors.

For the input cloud, the centroid points of the extracted feature vectors is found. The authors propose to store this representation to be available for the comparison with subsequent input feature maps.

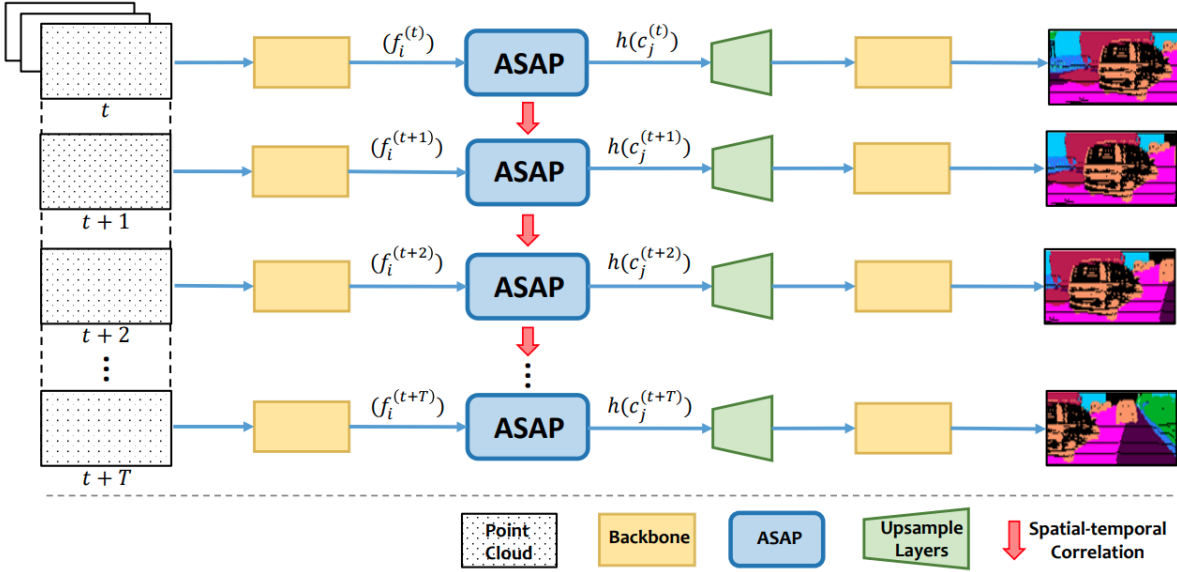


Figure 4.10: Illustration of the ASAP-Net architecture from original publication of Cao et al. [31]. The concatenation of multiple single-scan feature maps is realized by the ASAP block. With a PointNet feature extractor and backbone, the network segments the point cloud with improved spatio-temporal correlation.

For an available set of N_{ASAP} input feature maps of N_{ASAP} subsequent scans, the authors propose to associate the feature maps. The proposed association method is based on the centroid points of the first frame from farthest-point sampling. This sampling is applied only once for one point cloud, and kept static to associate the remaining N_{ASAP} feature maps based on this spatial information.

Denoted from the authors as Attentive Temporal Embedding (ATE), the individual point clouds are related to contain temporal and spatial information in two steps. First by concatenating the associated centroids and propagating the centroids by a MLP to a new centroid map, the spatial information is incorporated.

Second, an attention mapping is applied to add temporal information. Therefore two associated features are concatenated, and compressed in a MLP. The softmax output thereof is considered as attention scalar. Hence, applying this scalar factor on both of the associated features and summing both components, the final feature map is updated by this weighted attention.

The output of the ATE block yields an updated feature position and an attention weighted feature map. This feature map is upsampled to the point clouds' original dimension, processed with the backbone model as segmentation head, resulting in the semantic segmentation per point.

But especially for radar, the changing number of detection points in subsequent scans complicate a robust association.

Table 4.3: Data set overview of for the architecture test of LiDAR segmentation networks on radar data. Sequences 00-10 used for training, while Sequence 05 exclusively applies as test sequence for evaluation.

| | Name | #Scans | Length[m] | Description |
|----------------------|-------------|--------|-----------|----------------------------|
| Architecture Testing | Sequence 00 | 245 | 58.74 | urban residential area |
| | Sequence 01 | 290 | 31.66 | perpendicular parking |
| | Sequence 02 | 101 | 20.34 | sub-urban residential area |
| | Sequence 03 | 400 | 80.65 | sub-urban parking lot |
| | Sequence 04 | 332 | 51.92 | garage parking |
| | Sequence 05 | 163 | 29.78 | urban residential area |
| | Sequence 06 | 170 | 25.71 | perpendicular parking |
| | Sequence 07 | 422 | 77.92 | sub-urban parking lot |
| | Sequence 08 | 265 | 62.19 | urban residential area |
| | Sequence 09 | 82 | 17.81 | sub-urban residential area |
| | Sequence 10 | 232 | 27.83 | sub-urban residential area |

4.3.5 Architecture Transfer form LiDAR Domain

The selected LiDAR architectures of PolarNet, Cylinder3D and ASAP-Net are structurally adapted for radar attributes and trained on semantic radar data.

Network Training: The selected architectures are trained as the native benchmark data set semanticKITTI suggests, splitting the data set of 2732 radar scans into a training sub-set (90,3%, Sequences 00-10), excluding unseen data (9.7%, Section 08) as test section. Table4.3 depicts the data set on which the architectures are compared. This preliminary test is to compare general applicability of the architectures on sparse radar data, not for a detailed network optimization. Based on the results of the different architectures, the best performing architecture is selected for further radar-specific adaption and fine-tuning.

For the architecture comparison, the models are trained on the same 10 sequences of the data set, excluding sequence 08, which is used for test, validation. Little parameter tuning is performed to achieve convergence of the network training, but the best settings from the original publications are applied.

Testing the models as live inference modules is omitted for this initial architecture comparison, since the comparison and selection of the general structure is in the focus of the performance evaluation on validation data. The testing is performed on unseen sequence data.

In the following, the radar specific network adaptations to comply with radar specifics are discussed. First the input discretization and second their feature embedding.

Input Discretization: For PolarNet, the input point cloud discretization remains in 2D space, independent of the points' z-coordinates or respect the 3D distribution along the height axis. Polar grids are generally well suited to radar point clouds, due to the quasi-native representation in range ρ and azimuth-angle θ .

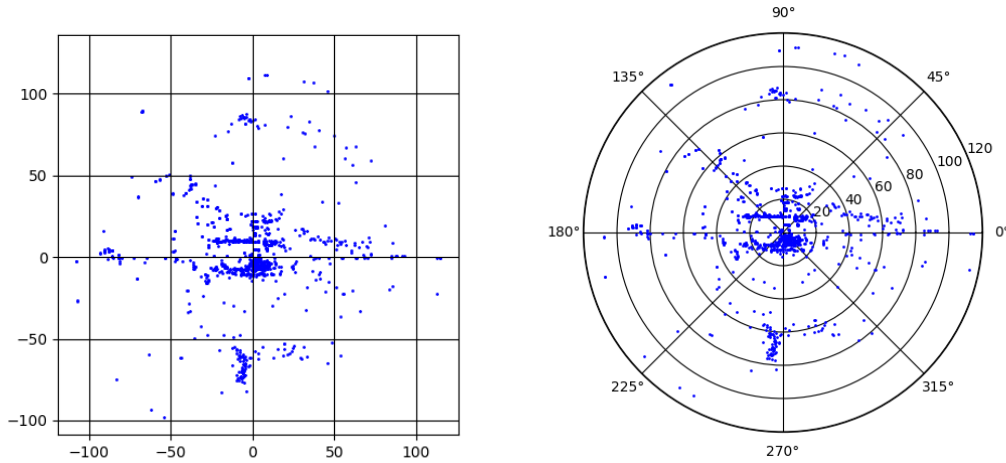


Figure 4.11: Illustration of a the Cartesian grid discretization (left) versus a polar grid discretization (right). The selection of a Cartesian resolution is non-trivial, whereas the polar discretization matches the radar point distribution of the exemplary radar point cloud better. Figure of Isele et Al. [SI3]

With increasing radius, grid cell areas grow and yield an improved homogeneous distribution of radar points over the grid. Contrary, Cartesian grids with a fixed cell size result in increased point population per cell near the sensor while only sparsely populated or empty cells in the far distance. For input tensors of neural networks, padding or zero fields can lead to biases and inefficient computational effort for empty cell convolutions. Ideally, normal distributed point count per cell is preferred. As relaxation of this requirement, an improvement towards homogeneously distributed data by polar grid cells is beneficial.

In Table 4.4, the effect of varying grid sized is depicted for a PolarNet variant. The grid sizes for Cylinder3D and ASAP-Net are not varied, since their convergence was not sufficiently achieved for a detailed grid-size study. Instead, Cylinder3D and ASAP-Net are trained with a fixed grid discretization of [75, 75, 32].

With the polar coordinates, the input discretization directly complies with the LiDAR specific ring-form of sensor readings. Explicitly designed for concentric LiDAR rings, PolarNet [248] introduces ring-convolutions from which the radar sensor benefits as well. Radar detections tend to blur azimuthal along concentric rings, rather than blurring in range coordinate. Technically, the blurry sensor tails are a beneficial property considering a ring-shaped blur per object.

Radar Feature Channels: Other than for LiDAR point clouds, the radar point clouds consist of variable number of points, only a maximum of 600 detections per sensor cycle is defined. After a feature extraction step, the networks feed their fixed size dimensional feature vector to different backbones, sampling blocks, and segmentation heads.

The compared network architectures in this thesis rely on a similar feature extraction method, namely a PointNet inspired MLP, which enables a fixed-size input feature. Due to a max-pooling operation on the extracted features, yielding a constant dimension output vector, the sparse and dynamic radar points density does not require architectural network changes.

Specifically adapted for all tested networks in this thesis, the feature extraction is extended to include also the relevant radar attributes Signal Power P_{sig} , Signal to Noise Ratio SNR and Doppler Velocity v_D besides the classical (x, y, z) -coordinates. Since the adaptations only cover the feature extraction, the gross of the network architecture to remains unchanged. The radar specific attributes Signal Power P_{sig} , Signal to Noise Ratio SNR and Doppler Velocity v_D are included and tested in different combinations. In order to provide normalized values, Signal Power P_{sig} and Signal to Noise Ratio SNR are linearly normalized to a maximum value of 1. The Doppler velocity v_D is min-max normalized by constant values $v_{D_{\text{min}}} = -140 \frac{m}{s}$, $v_{D_{\text{max}}} = 140 \frac{m}{s}$.

For the additional radar channels, their signal value is plotted in the histograms of Figure 4.12, after normalization to $[0, 1]$.

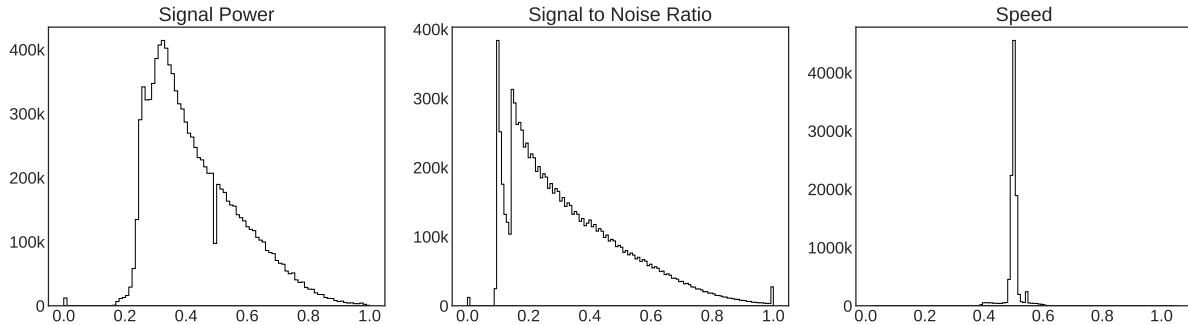


Figure 4.12: Histogram of Signal Power P_{sig} (left), Signal to Noise Ratio SNR (center), and Doppler Velocity v_D (right).

The effect of additional specific radar channels and different channel combinations on the overall average network performance is found in Table 4.4 and discussed.

Signal Correlation: In order to check the relevance and discrimination content of the available data, a correlation study was performed beforehand, to find the most relevant attributes on a data-driven basis.

The radar Equation 4.1 is repeated here again, to compare the correlation findings.

$$P_{\text{sig}} = P_{\text{recieved}} = 10^{-2kr} \cdot RCS \cdot \lambda^2 G^2 \frac{P_{\text{send}}}{(4\pi)^3 R^4} \quad (4.1)$$

Correlations are tested based on the Spearman correlation coefficient [204]. As similar measure for correlations, often the Pearson correlation [161] is tested. Compared to Pearson [161], Spearmans' theory is also applicable for non-linear, monotonous correlation, but with the prerequisite of strict monotony. Assuming monotony, the tested correlations are found:

- Radar Signal Power P_{sig} and Signal to Noise Ratio SNR are naturally strongly correlated. See the radar Equation 4.1, proving the assumption formalized, as the recieved signal obviously linearly depend on the sent signal power P_{send} .
- The 4-th power of range R as denominator in Equation 4.1 shows a significant radius dependency of signal power P_{sig} . Based on the expected strong correlation and importance for classification, the expected classification information of P_{sig} decreases dramatically with increasing range. Consequently, the classification accuracy for detections at higher range coordinate is expected to decrease similarly.
- The Doppler velocity ν_D is found to be not correlated to other signals, displaying an independent and essential information content.

4.3.6 Results of Architecture Comparison

The baseline network architecture comparison is performed with suggested original LiDAR settings. The training procedure for all variants is limited to 30 epochs to avoid over-fitting.

Feature Ablation Study: By a variation over different configurations of input attribute combinations, Signal Power P_{sig} , Signal to Noise Ratio SNR , and Doppler Velocity ν_D , an ablation study is performed to evaluate the effective result on the overall segmentation performance under same training conditions. From the signal correlation check, the three most potential and uncorrelated radar information channels are chosen to be varied.: With this forward feature engineering, the effective input channels for an improved output performance is tested to select the output-optimal feature combination for further fine-tuning. It is aimed to find a beneficial radar-specific feature channel combination and feature extraction to make up the structural drawback of radar sparsity in order to learn to distinguish clutter and noise from semantic classes.

As common 4D input tensor, the models are fed with common (x, y, z) -coordinates, substituting the fourth LiDAR reflectivity channel with the radar channel signal power P_{sig} . Varying combinations were tested analogously, substituting the 4th channel or extending the feature

extraction model to a 5D- or 6D-tensor including the Doppler velocity ν_D or the signal to noise ratio SNR .

| Model | Var. 1 | Var. 2 | Var. 3 | Var. 4 | Var. 5 |
|------------|------------------|------------------------|--------------------------------|---------|--------------------------|
| | P_{sig} | $P_{\text{sig}} + SNR$ | $P_{\text{sig}} + SNR + \nu_D$ | ν_D | $\nu_D + P_{\text{sig}}$ |
| PolarNet | 49.47 | 49.18 | 49.49 | 46.90 | - |
| Cylinder3D | 39.42 | 42.83 | 41.27 | - | 43.35 |
| ASAP-Net | 35.16 | 35.06 | 35.05 | - | - |

Table 4.4: Ablation study over test IoU [%] of the three classes *clutter*, *building* and *vehicle*. The other classes' IoU are not included in the reported average IoU since they often remain unrecognized at 0.0%³ and do not occur in the exclusive test sequence 05.

Unsurprisingly, the model performances' on radar data under-perform compared to the native LiDAR application, especially for the rare classes, see Figure 4.13. With manifold reasons, the performance drop can be explained: Effects of noise in the radar compared to LiDAR, a very long-tailed class distribution, manual annotation inaccuracies, similar classification on neighbor classes, and mainly a small data set with very high class imbalance.

Across the tested 4D, 5D or 6D input variants, the significance of additional radar specific input channels are not significant compared to the baseline 4D model. The channel specific IoU gains or IoU degradation is inconsistent across model architectures. In contrast, Table 4.4 illustrated the performance differences of the architecture transfer on the radar data set.

Both, PolarNet and Cylinder3D achieve convergence over all input variants. Noteworthy, the Cylinder3D network needed more of parameter testing to achieve convergence. Even if the sparse 3D convolution is designed for sparse grids, the absence of points seems relevant. Only variants with coarse grid size achieve the reported results.

For the ASAP-Net and Cylinder3D, convergence can only be achieved for a coarse grid discretization of [75, 75, 32] for a 50 m range point cloud, but not for all channel combinations the convergence is achieved. The required coarse grid seems a natural finding, since the sparsity of point clouds complicates the ATE block to find consistent weights. With dedicated fine-tuning of the ASAP-Net, the attention resolution could be modified to cope with the sparsity and spatial changes of subsequent scans. Testing the Doppler velocity as 4th dimension channel on the Cylinder3D architecture does not converge also. Potentially, the networks are confused by the velocity in general. Considering the 4D PolarNet variant with Doppler velocity points at the same finding.

³ Compare Tables 4.6-4.8 to find the excluded classes unrecognized.

Comparing the overall best performing architecture variant of Table 4.4, the 6D input tensor PolarNet configuration is found the overall best network variant.⁴ Although the other architectures promise theoretical advantages of the feature extraction, the PolarNet was found the most robust, but simple Network architecture, even outperforming the other variants. The ablation study over channel variation did not yield a specific superior combination, besides the relevance of Signal Power P_{sig} .

In detail, the configuration with the 6D radar attributes, the network inference is performed on spatial coordinates in (x, y, z) , Signal power P_{sig} , Signal to Noise Ratio SNR and Doppler velocity v_D .

For the best PolarNet variant, the corresponding Confusion Matrix is displayed in Figure 4.13. As expected from the data set balance, the rare classes pose the hard cases to be determined correctly. For all classes it is found, that the main confusion is with the most occurring class of clutter or noise.

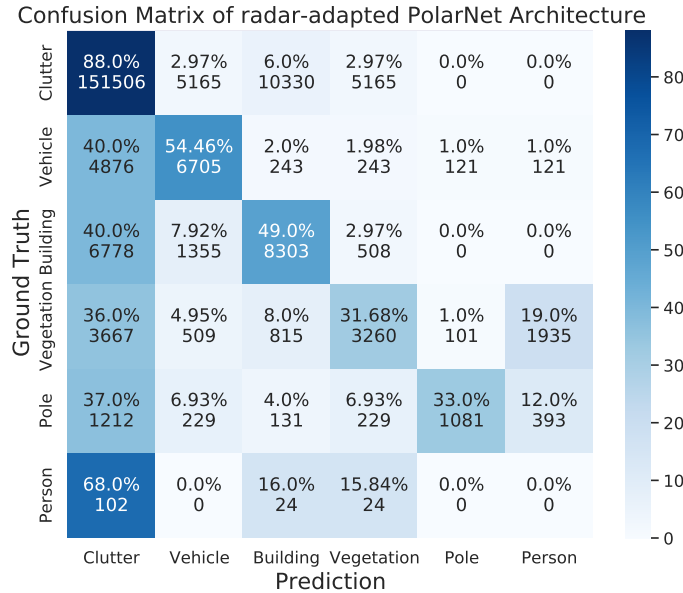


Figure 4.13: Illustration of the confusion matrix for the best PolarNet variant.

z -Coordinate: Differences of 2D Grid based PolarNet architecture and 3D based Cylinder3D is found in the grid indices and maximum pooling over the cells. For PolarNet, the 2D Grid is sub-sampled in unique (x, y) -cell coordinates, describing in which cells points occur. As input tensor, the PolarNet ingests (x, y, z) coordinates as polar coordinates (ρ, θ, ϕ) ,

⁴ For the evaluation of all networks, the architecture evaluation is based on their classification performance on the offline data set. They are not all tested systematically in real-live performance. Due to good performance on the data set, only PolarNet is selected to be implemented as live inference network.

the unique cell indices (u_1, u_2) and the radar specific additional radar attribute channels $(P_{\text{sig}}, \text{SNR}, \nu_D)$. Every single point, described by a 11-dimensional input vector, is processed by an MLP to a feature vector of size 512. This per point feature vector is processed similarly as the PointNet feature extraction. As a result, per occupied unique cell index multiple feature vectors can occur, each representing original point measurements. Along this 3rd axis of the grid size discretization, a convolution layer compresses the point feature vectors from 512 into the dimension of 32. Per unique cell in 2D, the PolarNet architecture processes this "feature compression" layer with maximum pooling layer to represent multi-occupied cells by a single PointNet feature vector of size 32. The resulting feature vector per unique cell and its combination with all other pooled feature vectors in the grid represents the feature map of the input point cloud. This feature map has a 3D grid size before it gets processed in a U-Net.

Hence, for the PolarNet architecture the height attributes of the points are encoded in the z -coordinate grid discretization. It is not respected by the feature extraction of PointNet, but respected in the 2D maximum pooling layer and ring-convolution layer.

The theoretical advantage of Cylinder3D to include the z -coordinate in the sparse 3D convolution, and thereby include the elevation explicitly in the feature vector, does not materialize in a superior overall IoU. Compared to Figure 3.6, the z -coordinate is limited by the FoV of the sensor according to Figure 2.12. Consequently, the network can not expect the same spatial z -information over the whole sensor range.

Segmentation Heads: PolarNet suggests only ring convolutions as segmentation head. Cylinder3D applies 3D convolutions as segmentation head to extract data with unsymmetrical residual blocks. According to the authors, the unsymmetrical residual blocks improve the detection of (large) vehicles. It is questionable, if the sparse radar data sufficiently detects medium-far object regions dense enough to deploy the residual blocks on the radar detections. The radar detections occur scattered and sparse in single scans, so an interpretation based on the *shape* is comparably harder as for dense LiDAR. Additionally, Cylinder3D integrates a dimension decomposition for context modeling and a convolution with a $3 \times 3 \times 3$ kernel. The FoV of a radar sensor in elevation opening limits the range in which objects can be sensed to a point set with reasonable spread in z -coordinate. For very close detections to the sensor, the FoV of the sensors do not allow a significant z -coordinate, but received detections occur on a flat plane. Hence, near radar detections could be miscalculated by the 3D convolution with empty cells.

The segmentation head of the ASAP-Net is similar to the feature extraction a PointNet++ backbone. This segmentation backbone is estimated not to deprecate at radar data, but the grouping stages can yield instabilities.

Temporal Consistency: ASAP-Net extends the pre-mentioned network architectures by the temporal dimension. Aiming to track the past feature map and align the current features

with the knowledge over the feature maps seen in the past, ASAP-Net introduces an Attention Temporal Embedding (ATE) block to extract temporally consistent features. This block orchestrates different feature maps to be inter-frame associated by concatenation and fusion. The suggested ATE block systematically decouples the spatial consistency and the temporal consistency as independent feature embedding, independent from the applied segmentation backbone.

The feature maps of subsequent scans are processed temporally and spatially decoupled in the extraction step. By concatenation, the subsequent ATE block recombines the PointNet++ features from the current forward pass with subsequent older measurements, which are available from a buffer. The network applies a buffer so save a number of these feature maps for previous point cloud.

This attention map is applied to a second MLP layer with a pre-defined number of sensor scans. In brief, the aggregated points describe the spatial context. The feature fusion of subsequent scans by the means of the attention map covers the temporal information.

Technically fusing multiple sensor scans by the ATE block in the feature embedding, ASAP-Net is theoretically able to systematically integrate temporal information in an overall similar structure as a single scan approach.

Major drawback of this attempt is the dependency of previous scans. Hence, similar to the idea of a memory block as in LSTM⁵ networks, the current network performance depends on former states. This can cause instability, if the concatenated adjacent feature maps vary much.

As shown in Table 4.4, some variants of the ASAP-Net would not converge for certain sets of input channels. Especially the 4-dimensional Doppler Speed v_D variants do not converge. As explanation attempt, v_D values of subsequent frames introduce a noisy feature map per frame in the ATE block. Concatenating this feature maps causes unstable attention weights and thereby also diverging gradients of subsequent training steps in the network training.

Grid-Cell Discretization: Since the model formulation addresses a multi-class classification problem, a final arg-max layer finds the most likely predicted label per grid cell yielding a classification label per cell. As a result of the voxel based arg-max classification, there is a trade-off between label accuracy and label discretization of closeby neighbor classes.

For Cylinder3D, the grid resolution of [75, 75, 32] for the 50 m ranging sensor signals is tested. Model convergence of Cylinder3D is only achieved with this coarse discretization. A too fine discretization of the input cloud, leaving neighboring cells empty, is found to introduce problematic convergence behaviour with the sparse radar point cloud input. Empty 3D voxels

⁵ Long-Short Term Memory networks are introduced by Hochreiter and Schmidhuber [89], and mainly applied for time-series application.

of sparse point clouds in combination with 3D context modules, which explicitly search neighborhood context, are empirically found to degrade in model convergence. Similar to the findings of PolarNet, that coarse grid discretization technically prohibits to accurately distinguish closeby or neighboring objects, the context modeling of Cylinder3D is found inappropriate for sparse point clouds, too.

Architecture Evaluation: The performance of the compared network architectures does not correspond to the expectations. Expecting PolarNet to yield baseline results, while both other architectures suggest to yield superior results due to their context modulation blocks. In contrast, Table 4.4 shows the inverse performance ranking of the tested architectures. The hypothesis of superior context extraction based on 3D convolutions in Cylinder3D, or the spatio-temporal attention map from ASAP-Net to improve the performance, does not hold. Instead, the 2D PolarNet architecture yields best results.

A possible explanation can be found in the sparsity of the data. Instead of incorporating the height information directly in 3D convolutions, it might be beneficial to have a 2D convolution in PolarNet to describe the neighborhood of feature vectors in the feature space.

Besides the detection of larger objects, the classification of rare classes does not achieve good result. This phenomena is directly coupled to the occurrence of these rare classes. The data set is heavily imbalanced, and the few samples are too less to allow the network an adaption to classify those and not treat them as outliers which vanish in the sheer point-mass.

Consistent and substantial speed-up of the classification time is reached by polar discretization over a cartesian grid discretization, see Table 4.6. The tested PolarNet architecture yields real-time capability compared to the sensor sensing rate. As the other approaches fall behind in classification performance, their inference-time is not tested.

Further can be concluded from the empirical architecture evaluation, that a 6D radar attribute configuration outperforms 4D or 5D approaches.

As final evaluation, Table 4.5, depicts a summary of the findings of the architecture comparison of this section, e.g. the weighted IoUs of Table 4.4 or inference time.

Table 4.5: Architecture comparison, rating from positive (++) to negative (- -), with weighted wIoU: Considering only *clutter*, *building*, and *vehicles*.

| Model | wIoU | Speed | Robustness | Radar Spatial Context | Temporal Context |
|-----------------|-------|----------|------------|-----------------------|------------------|
| best PolarNet | 49.49 | 12.6[ms] | ++ | + | N/A |
| best Cylinder3D | 43.35 | N/A | -- | - | N/A |
| best ASAP-Net | 35.16 | N/A | -- | -- | (+) |

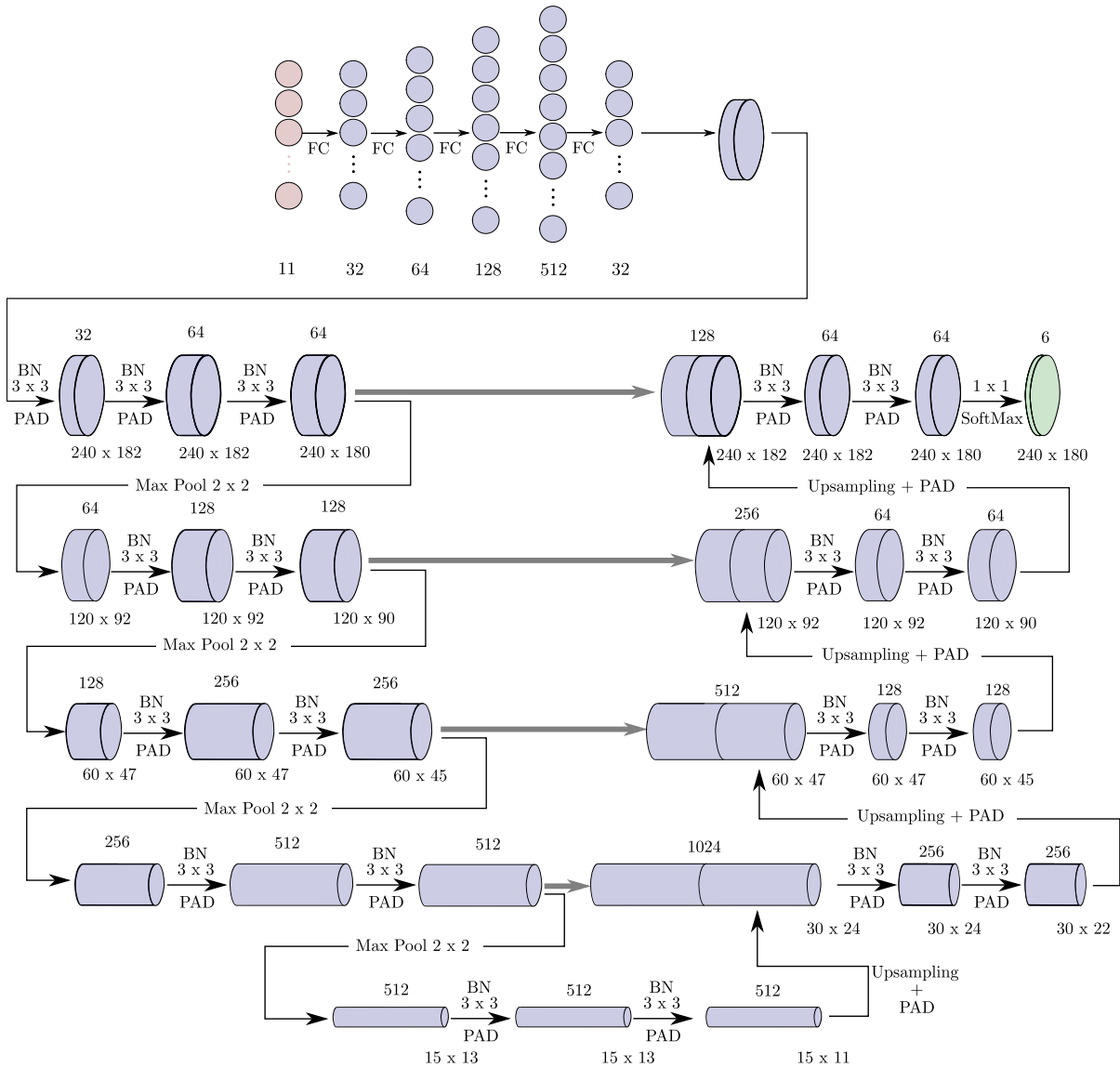


Figure 4.14: Illustration of the RadarNet architecture: Additional radar feature channels and reduced output dimensionality.

4.4 RadarNet: Semantic Radar Segmentation Network

Outperforming other architectures in the pre-test of Section 4.3.6, the PolarNet architecture is chosen to be fine-tuned for a radar domain live inference application on sparse radar data. This radar-adapted and fine-tuned architecture is referred to as *RadarNet* in this work.⁶ This section reports the specific radar adaptations and architecture optimization. The next section reports the results of the network training and the best performing RadarNet.

⁶ The RadarNet architecture is tuned and implemented as live inference module for direct deployment. As ROS module, it subscribes to the live or replayed sensor data of the test vehicle and publishes the data semantically annotated.

Architecture Adaption: Originally designed for a LiDAR point cloud as input, the best performing PolarNet variant is modified to take an 11 dimensional input, including three additional radar channels for Signal Power P_{sig} , Signal to Noise Ratio, SNR and Doppler velocity v_D . According to the class consolidation Table 3.2, the final softmax layer is also adapted. According to Table 3.2, only the relevant $N_C = 6$ classes as predicted by N_C output neurons, instead of the original 22 SemanticKITTI classes.

Figure 4.14 illustrates the layer structure of the proposed radar segmentation network.

Hyper-Parameter Optimization: Found in the sections before, the experimental results proves the architecture of PolarNet to cope comparably well to the sparse radar data. This section describes how this architecture is refined, retrained and the adaptations discussed in detail. For point-based networks, the features and hyper-parameter optimization require fine-tuning. Common things such as feature normalization, batch-size, batch normalization, learning-rate, dropout or weight regularization are considered. Momentum is not considered since the optimizer SGD and Adam are being challenged with different settings. Early stopping is manually checked from the training results. For a sub-set of hyper-parameters, their effect on the network performance is displayed and discussed.⁷

The fine-tuning of the semantic radar segmentation is performed without special dependence of a utilizing system, e.g. an application of automated parking, but with the following range limitation. From an application point of view, the region of interest (ROI) of the radar point cloud is limited to a range of 50 meters of an planar environment. With respect to the quasi static sensor mounting position on the vehicle, the network is trained in fixed volume mode. Excluding the potential further detections, this mode limits the maximum range of input points and limits also the z-coordinate of the points. For the parking application, the relevant and reliable sensor range limit is assumed to 50 m at a z-coordinate limitation to the range between $z_{\min} = -0.5 m$ and $z_{\max} = 3.0 m$. Hence, the semantic segmentation radar input is cut to this ROI window.

Grid Size: For the fixed volume to be classified in 50 m range and relevant z-coordinate $z \in [-0.5 m, 3.0 m]$, the tested extreme grid resolutions are [240, 180, 16] as coarse discretization to the finest resolution [480, 360, 32]. Variation of the polar grid to a resolution which represents a coarse grid cell discretization of 0.2 m in radial distance to a fine radial cell discretization of 0.1 m . According to the estimated double standard deviation 2σ of the radar sensing accuracy, illustrated in Figure 5.4, the lateral uncertainty $\hat{d}_{y,2\sigma} = 0.28 m$ and radial uncertainty $\hat{d}_{r,2\sigma} = 0.06 m$ of detections at 40 m range are estimated from Equation 5.7. Hence, the radial grid discretization of the 50 m range is sampled between 460 cells (0.104 m) to

⁷ Not included as hyper-parameter is the accumulation of point clouds. In the labeling process, the necessary density is only reached by accumulating three radar point clouds. The accumulation parameter is out of variation and set fixed for the training.

Table 4.6: Model study of PolarNet with variations in discretization, features and network architecture.

| Model | Grid | # Cells | Inference [ms] | weighted IoU | Vehicle IoU | Person IoU | Artifact IoU | Building IoU | Vegetation IoU | Pole IoU |
|--------------------------------------|------------------------|---------|----------------|--------------|--------------|--------------|--------------|--------------|----------------|--------------|
| PolarNet | Cartesian:[200,200,32] | 40k | 33.6 | 0.486 | 0.362 | 0.000 | 0.822 | 0.273 | 0.000 | 0.015 |
| PolarNet | polar: [150,150,32] | 22.5k | 11.7 | 0.457 | 0.313 | 0.000 | 0.802 | 0.256 | 0.002 | 0.076 |
| PolarNet | polar: [240,200,32] | 48k | 12.5 | 0.500 | 0.366 | 0.000 | 0.817 | 0.317 | 0.000 | 0.000 |
| PolarNet | polar: [300,300,32] | 90k | 16.6 | 0.504 | 0.372 | 0.000 | 0.824 | 0.315 | 0.000 | 0.017 |
| PolarNet | polar: [480,360,32] | 172k | 26.2 | 0.516 | 0.389 | 0.000 | 0.826 | 0.334 | 0.025 | 0.025 |
| PolarNet | polar: [250,250,32] | 62k | 13.5 | 0.520 | 0.408 | 0.000 | 0.824 | 0.327 | 0.002 | 0.040 |
| PolarNet | polar: [200,200,32] | 40k | 12.2 | 0.523 | 0.437 | 0.000 | 0.823 | 0.305 | 0.000 | 0.043 |
| 4D PolarNet: (x, y, z)+SNR | polar:[200,200,32] | 40k | 12.3 | 0.502 | 0.389 | 0.000 | 0.830 | 0.287 | 0.000 | 0.067 |
| 4D PolarNet: (x, y, z)+ P_{sig} | polar: [200,200,32] | 40k | 12.2 | 0.548 | 0.460 | 0.002 | 0.835 | 0.350 | 0.004 | 0.007 |

comply with one standard deviation ($\frac{\hat{d}_{y,2\sigma}}{2}$) to 150 cells, resulting in 0.33 m radial cell size. The angular polar grid resolution is sampled from 150 cells per revolution (2.4° resolution) to the originally proposed fine 1° resolution of 360 radial cells per ring.

The visualization of different discretization performance can be found in Table 4.6. Figure 4.15 plots the grid size variation influence on the network inference time and segmentation performance.

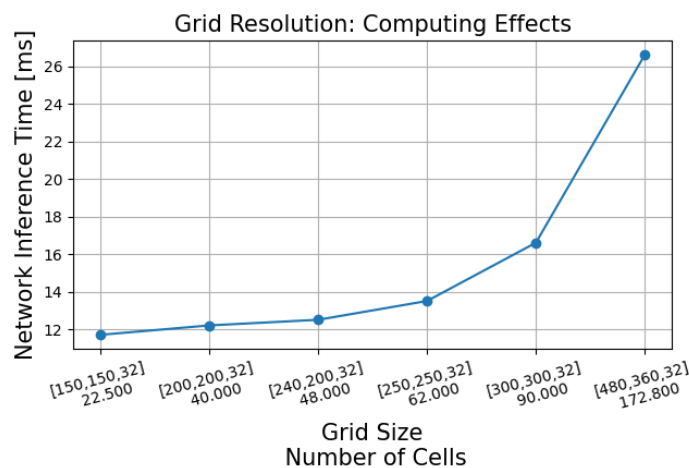


Figure 4.15: Illustration of the PolarNet variant with additional radar feature channels. Blue: 19 class, 6D PolarNet variant from architecture comparison.

Table 4.7: Repetition of Table 3.4, now as data set split overview of training data for *RadarNet* training, validation (light grey marked) and testing (dark grey).

| | Name | Scans | Length [m] | Description |
|------------------------|-------------|-------|------------|--|
| Architecture Testing | Sequence 00 | 245 | 58.74 | urban residential area |
| | Sequence 01 | 290 | 31.66 | perpendicular parking |
| | Sequence 02 | 101 | 20.34 | sub-urban residential area |
| | Sequence 03 | 400 | 80.65 | sub-urban parking lot |
| | Sequence 04 | 332 | 51.92 | garage parking |
| | Sequence 05 | 163 | 29.78 | urban residential area |
| | Sequence 06 | 170 | 25.71 | perpendicular parking |
| | Sequence 07 | 422 | 77.92 | sub-urban parking lot |
| | Sequence 08 | 265 | 62.19 | urban residential area |
| | Sequence 09 | 82 | 17.81 | sub-urban residential area |
| RadarNet Training Data | Sequence 10 | 232 | 27.83 | sub-urban residential area |
| | Sequence 11 | 437 | 92.92 | sub-urban parking lot |
| | Sequence 12 | 243 | 31.06 | parallel parking |
| | Sequence 13 | 215 | 14.98 | sub-urban residential area |
| | Sequence 14 | 45 | 16.15 | sub-urban area |
| | Sequence 15 | 969 | 480.66 | sub-urban residential area (loop) |
| | Sequence 16 | 1452 | 613.60 | sub-urban residential area (double loop) |
| | Sequence 17 | 1103 | 410.65 | sub-urban residential area (loop) |
| | Sequence 18 | 1082 | 362.69 | sub-urban residential area (loop) |

Data Set Extension: For the pre-tests and general architecture comparison, only sequences 00-10 have been applied. At the time of the *RadarNet* optimization, a larger data set was available as fully semantically labeled data set extension. For the fine-tuning and training of the RadarNet, a larger data set of additional 8 sequences, sequences 11-18, are applied, see Table 4.7. The data set size of sequences 00-10 ($\approx 3.28 \cdot 10^6$ points, covering a length of $484.64m$) is augmented by 250% to an overall count of $\approx 8.2 \cdot 10^6$ radar points and a length of $2507.35m$ in all sequences 00-18, enabling an improved training for better generalization.

The training data and class balance is found in Figure 4.16, including the *clutter* detections and in Figure 4.17. The latter shows the clutter-removed pie-charts of the class content per sequence of Table 4.7, in order to check the class distribution over the sequences besides the dominant portion of *clutter*.

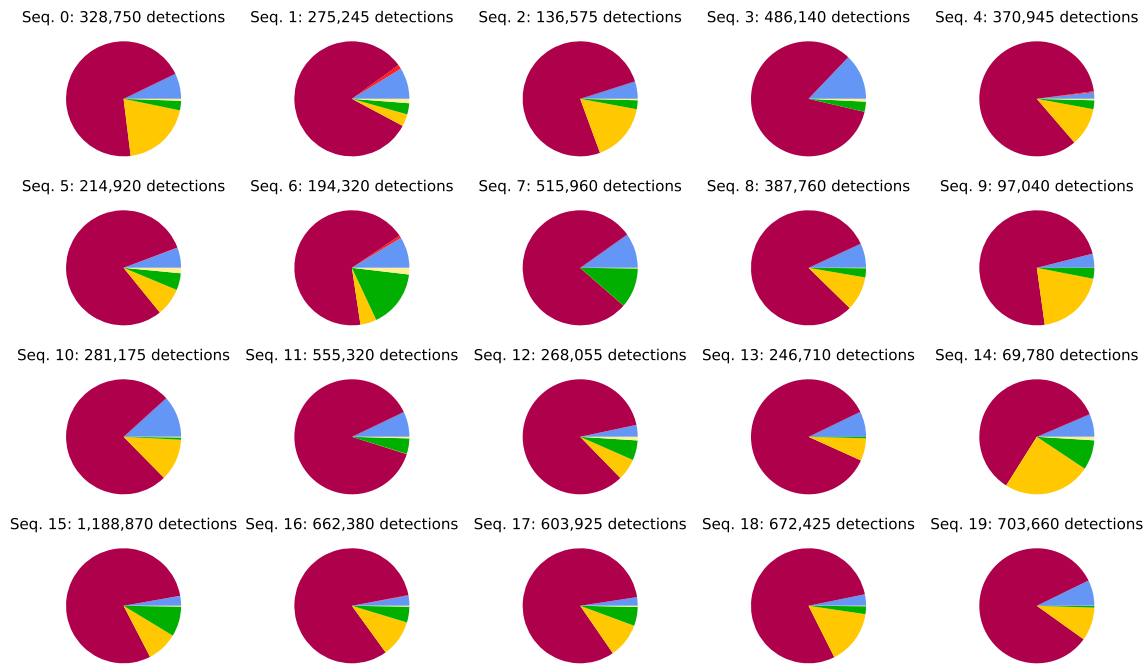


Figure 4.16: Illustration of semantic class balance in the data set sequences including *clutter*, visualized as pie charts. Color encoding: *building*, *vehicle*, *vegetation*, *person*, and *pole*.

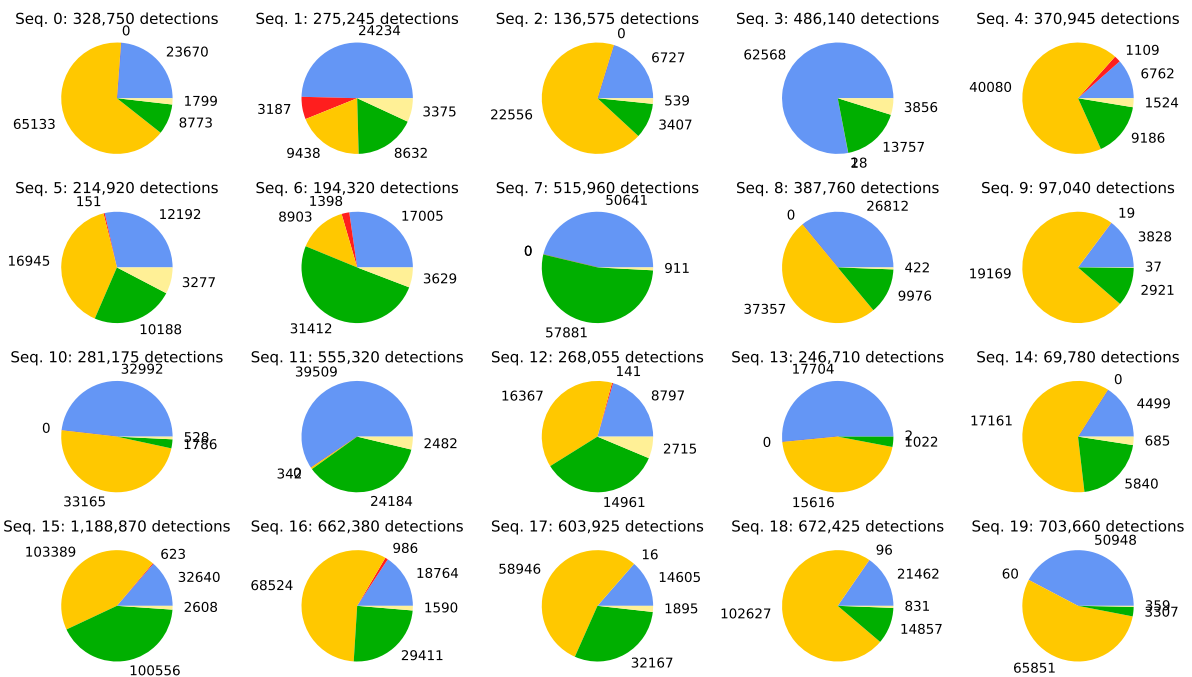


Figure 4.17: Illustration of data set and distribution of radar semantic classes without clutter, visualized as pie charts with class-specific counts per label. Color encoding: *building*, *vehicle*, *vegetation*, *person*, and *pole*.

Data Augmentation: To increase the amount of available training data and to achieve rotation invariance and order independent network training, data augmentation is applied. Data augmentation describes to increase the data set artificially, by creating realistic data variants, e.g. by flipping, rotation or additional noise. In this work, the training data is shuffled across training sequences and arbitrary rotated. Also to a percentage of 25%, a random coordinate flip on the (x, y, z) axis are applied, similar to the rotation invariance of LiDAR data in PolarNet [248], the rotation invariance of radar detections is legitimate. It is avoided to add e.g. Gaussian Noise as measurement noise on the data, due to the inherent radar noise. Besides the value normalization of input features, the model is trained in *fixed-volume* mode, clipping the input point cloud to a maximum range of 50 m with $z \in [-0.5 \text{ m}, 3 \text{ m}]$. By clipping, also scale-invariance is achieved [248].

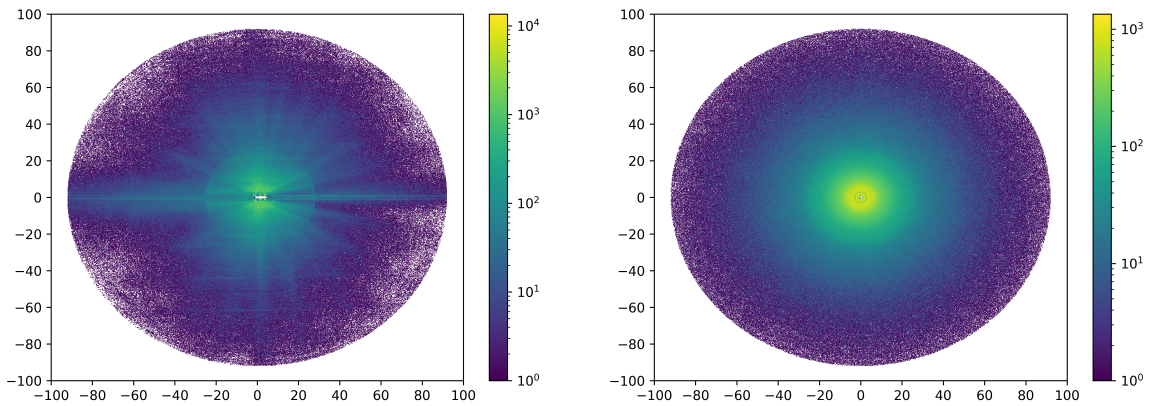


Figure 4.18: Illustration of data augmentation result for the total of all spatial locations. Original heat map (left) compared to the spatial distribution with data augmentation (right).

The resulting training data is visualized in the heat-map for all classes summarized. Detailed plots per class are found in the Appendix, section A.2. The beneficial effects of the data augmentation can be impressively visualized in the even distributions of Figure 4.18.

Loss Function: The IoU or Jaccard index, is commonly applied for semantic segmentation problems to express the fraction of correctly predicted labels, compared to the objects surface area. For point cloud semantic segmentation, the IoU measure distinguishes a segmentation classification only as *false* or *true*. The original PolarNet architecture of Zhang et al. [248] is trained with Cross-Entropy as loss function. In contrast ASAP-Net generalizes better on also rare classes, which is estimated also as a result of the application of the Lovasz-softmax loss [20]. This loss is alternatively applied for segmentation tasks, allowing a direct optimization of the Jaccard index by a surrogate formulation of the discrete IoU formulation. Lovasz-softmax loss commonly achieves a fine-tuning compared to cross-entropy loss. In

order to respect the data set class imbalance, individual class weights w are implemented and applied on the loss function.

Cross-entropy Loss: For a multi-class classification problem with mutually exclusive classes $N_C > 2$, the separately calculated loss per class gets summarized to an overall measure. For the semantic segmentation in this thesis, set $N_C = 6$. The ground truth label $y_{o,C}$ is binary and defines the true class of point o as one-hot encoding.

$$L_{wCE} = w_{CE,C} \cdot \frac{1}{N_C} \sum_{C=1}^{N_C} y_{o,C} \log\left(\frac{1}{\hat{y}_{o,C}}\right) \quad (4.2)$$

The formulation uses a binary status y , if the observed point o was correctly classified as class C . The network prediction $\hat{y}_{o,C}$ is the output of the networks' final soft-max layer, describing the predicted probability of point o to be of class C according to Yessou et al. [234].

The optional static weight w_{CE} per class C is defined by Equation 4.5 to balance the data set as proposed by Cui et al. [49] and applied in Pihur et al. [166] or Ho and Wookey [88]. In the general case of ideally balanced data, no weighing is applied and $w_{CE,C} = 1$, this is referenced as L_{CE} Loss. Based on the high class-imbalance for a multi-class classification problem, introducing a weighing factor is beneficial to penalize mis-classifications of rare classes by higher consideration. Hence, the loss-based optimization recognizes rare classes better, see WCE-Loss in Table 4.8.

Lovasz Loss: Alternatively to the cross-entropy loss, the Lovasz-softmax loss L_{wLS} is applied for multi-class $C \in N_C$ segmentation networks. For a detailed derivation refer to the original paper of Berman et al. [20]. This loss function directly optimizes the mIoU value during training by a differentiable loss surrogate function $\overline{\Delta}_{J_C}(m(C))$ of the Jaccard index Δ_{J_C} , using the hinge loss vector m .

In combination or as alternative to class-weights, the Lovasz-softmax Loss significantly increases the classification performance of rare classes and can also be weighted with class weights $w_{Lovasz,C}$.

$$L_{wLS} = w_{Lovasz,C} \cdot \frac{1}{N_C} \sum_{C=1}^{N_C} \overline{\Delta}_{J_C}(m(C)). \quad (4.3)$$

In the general case of ideally balanced data, no weighing is applied and $w_{Lovasz,C} = 1$, this is referenced as L_{LS} Loss.

Combined Loss: Both Lovasz and Cross-Entropy (CE) loss are tested and applied for the network training. As the authors of the Lovasz-Loss [20] state, the optimization of the mean IoU (mIoU) for a data-set depends on the number of classes and the training batch-size. Practical applications suggest to pre-train a segmentation network with cross-entropy loss and fine-tune with the Lovasz-softmax Loss.

The most promising approach for *RadarNet* is to train the network with a combination of both loss functions







$$\begin{aligned} L_{\text{combined}} &= L_{wLS} + L_{wCE} \\ &= w_{CE,C} \cdot \frac{1}{N_C} \sum_{C=1}^{N_C} y_{o,C} \log(\hat{y}_{o,C}) + w_{\text{Lovasz},C} \cdot \frac{1}{N_C} \sum_{C=1}^{N_C} \overline{\Delta}_{J_C}(m(C)). \end{aligned} \quad (4.4)$$

Both loss terms of Equation 4.4 could further be weighted to define an importance for each term, but it is found impractical to weight the Lovasz-softmax.⁸

Table 4.8 displays the behaviour of a PolarNet variant, trained with different loss functions, yielding different classification IoU results. The most improvement potential is found for the rare classes, which is the most difficult problem to generalize over the very few data samples. Based on this findings, measured by the IoU values, a rating of loss effectiveness to the radar segmentation is possible.

Most interestingly, the combination of two losses yield the best classification of **artifact**-labels, achieving the best de-noising quality, but rare classes are found best by the Lovasz-Softmax loss. The combined loss formulation improves sensitivity to classify rare classes, while keeping the effective Cross-Entropy loss for training convergence. Further, class weights applied on Cross-Entropy contributes to improve the rare class recognition.







Table 4.8: PolarNet architecture radar variants as comparison of different Loss-functions for semantic segmentation: Network variant 4D PolarNet with radar channel signal power P_{sig} and grid size [200, 200, 32], trained on Sequences 00-10.

| Model | Inference [ms] | mIoU | Artifact IoU | Vehicle IoU | Building IoU | Vegetation IoU | Pole IoU | Person IoU |
|------------------------------------|----------------|--------------|---|---|---|---|---|---|
| | | |  |  |  |  |  |  |
| PolarNet (L_{CE}) | 12.4 | 0.243 | 0.831 | 0.402 | 0.244 | 0.000 | 0.000 | 0.000 |
| PolarNet (L_{wCE}) | 12.4 | 0.275 | 0.670 | 0.420 | 0.350 | 0.097 | 0.099 | 0.016 |
| PolarNet (L_{LS}) | 12.4 | 0.289 | 0.702 | 0.448 | 0.327 | 0.106 | 0.125 | 0.025 |
| PolarNet (L_{combined}) | 12.2 | 0.276 | 0.835 | 0.460 | 0.350 | 0.004 | 0.007 | 0.002 |

Class Weights: Given the sparsity of a radar point cloud, smaller objects are systematically detected by a lower number of radar reflections. Hence, tall but narrow structures such as poles and also humans are systematically under-represented by only a small fraction of a

⁸ Future work is open, to test a dynamic loss weight in later training stages. E.g. dependent on the gradient size to fade the fine-tuning capabilities of the Lovasz-Loss.

Table 4.9: Proposed class weights w_i per class C to balance rare classes in a reduced data set for architecture tests (sequences 00-10) and for the RadarNet training on the full data set (sequences 00-18).

| Radar Class | Architecture Tests | | RadarNet Training | |
|---|--------------------|--------|-------------------|--------|
| | Percentage [%] | Weight | Percentage [%] | weight |
|  Vehicle | 8.39 | 14.17 | 6.54 | 15.28 |
|  Building | 7.68 | 12.96 | 7.77 | 12.87 |
|  Vegetation | 4.80 | 20.89 | 4.75 | 21.04 |
|  Poles | 0.60 | 133.42 | 0.34 | 282.63 |
|  Person | 0.18 | 750.23 | 0.13 | 774.22 |
|  Artifact | 78.35 | 1.19 | 80.42 | 1.24 |

datasets overall detections. Given this example, its obvious that a real world data set is covering larger structures more frequently compared to the detection count per smaller objects. This applies here to *buildings*, or larger objects such as *vehicles*, compared to a lower number of detections per *person*, per *pole* and per *vegetation*.

Solving a multi-class classification problem, the detection of a rare class can be supported by class weights per loss term. Knowing the expected detection probability, a-priori class weights are a measure to artificially balance the data set during training [49, 88]. By multiplying the individual semantic class weights on the class-corresponding loss, the network learns to penalize the rare, but equally important wrong classifications of under-represented classes similarly to a wrong classification of a more frequent class.

For the given radar data set, the class imbalance is translated into loss class weights by the reciprocal occurrence, resulting in the class weights of Table 4.9. As weights, the number of detections per class over overall detection count is defined [49], see Equation 4.5

$$w_i = \frac{n_i}{N_{tot}} \quad \text{for all classes } i \in C . \quad (4.5)$$

For the hyper-parameter optimization, these weights are applied on both loss terms of Cross-Entropy of Equation 4.2 and the Lovasz-softmax loss of Equation 4.3. It is experimentally found beneficial to apply weighs only on the Cross-Entropy, leaving the Lovasz-softmax loss unweighted.

Batch Size: Based on the architecture plot in Figure 4.14, batch normalization (BN) is applied in multiple layers, normalizing to standard deviation of 1.0 and zero mean per batch, see Section 2.3 and Equation 2.34.

The tested *RadarNet* models are also trained with different batch sizes, in order to test for robustifying generalization or increasing IoU. Batch sizes with increasing or decreasing segmentation performance effects are not found. The training is performed on a Tesla-V100 GPU, allowing to test large batch sizes also. But memory limitation was only achieved for batch sizes beyond 64, resulting in larger memory allocation as 32 GB VRAM. Training with larger batch sizes achieve an efficient training procedure of approximately 6 hours per 30 Epochs of the whole data set.

Learning Rate: Decaying the learning rate is found to stabilize the validation loss, but too low learning rates prevent any further network adaption. The training and validation loss, remain constant in that case, while the network accuracy stagnates.

Training at a fixed or high learning rate $\eta = 0.01, \dots, 0.02$ yields significantly increasing cross-validation loss after a dip at approximately 30-50 epochs, depicted in Figure 4.22. Counter-intuitively, the network accuracy further increases especially for the detection of *clutter*, whereas the *vegetation* detection decreases. In contrast, the IoU scores of the other classes saturate, see Figure 4.23-4.24. For a further discussion, refer to Section 4.5-paragraph **Over-fitting**.

Dropout Regularization: Instead of applying Dropout [69] at specific layers, RadarNet is trained with drop-blocks [72] which are found to significantly increase generalization for deep networks with numerous parameters. The drop-blocks in the CNN part of the U-Net is applied on regions, to dropout these in the skip connections of the network. The dropblock parameter is applied to 0.5.

Initialization Seed: In order to achieve reproducible results for varying training runs, the same initialization seed is applied to train RadarNet from scratch. Four models are trained in the same configuration and with the same initialization seed to be evaluated in Section 4.5. With this averaging possibility over multiple independent model trainings, the model plots in Section 4.5 reveal a stable training and stable training results toward a global optimum. Over-fitting is unlikely to result in so similar results.

Optimizer: As alternative optimizer beyond the common Adam Optimizer [116], Stochastic Gradient Decent (SGD) [77, 45] is applied to optimize the gradients. The optimizer momentum is set to 0.9 with the above described decaying learning rate.

From the comparison of the hyper-parameter tests with both mentioned optimizers only marginal differences emerge. SGD results in marginally higher accuracy.

Training, Validation and Test Split: In order to report robust and reproducible results, the data set split into exclusive sub-sets for independent network evaluation is essential, as explained in Section 2.3 The markings in Table 4.7 illustrate the exclusive training (unmarked), validation (light grey), and third test (dark grey) sub-set. For unevenly balanced data sets, all data splits should reflect the original class balance of the training data. But especially the validation split is of interest, since mostly this performance is considered if a model over-fits during training.

For the problem at hand, balancing of class examples is not trivial, as a manual constitution of a validation set would require to hand-pick radar point clouds to be excluded from the training data which satisfy the training data class balance. Instead, the cross-fold validation scheme is applied. The left plot of Figure 4.19 illustrates the training loss of a model zoo of the 20-fold cross-validation. In comparison, the right plot of Figure 4.19 depicts the validation loss for the same models. In consideration with the class balance of the different sequences, the network performance can be discussed.

In the Figures 4.19-4.20, the result of a 19-fold cross-validation training is shown. 19 Models are trained for 50 epochs, with one sequence (00-18) excluded from the training, but applied as validation sequence. Figure 4.20 (left) shows the mIoU on the different validation sequences of the same network. The right plot of Figure 4.20 shows the corresponding accuracy curve. Comparing different cross-validation sequences along their IoU, the training sequence differences are obvious. Scenes with apparently higher IoUs indicate similar data set compared to the training data, especially in terms of class balance. Other models, yield low IoU curves and accuracy scores, indicating more difficult scenes of different class balance or difficult scene content, e.g. a larger dynamic content or increased *vegetation*. Hence, according to the definition of Meyer et al. [144], there exist a variety of *hard* to *simple* radar classification samples.

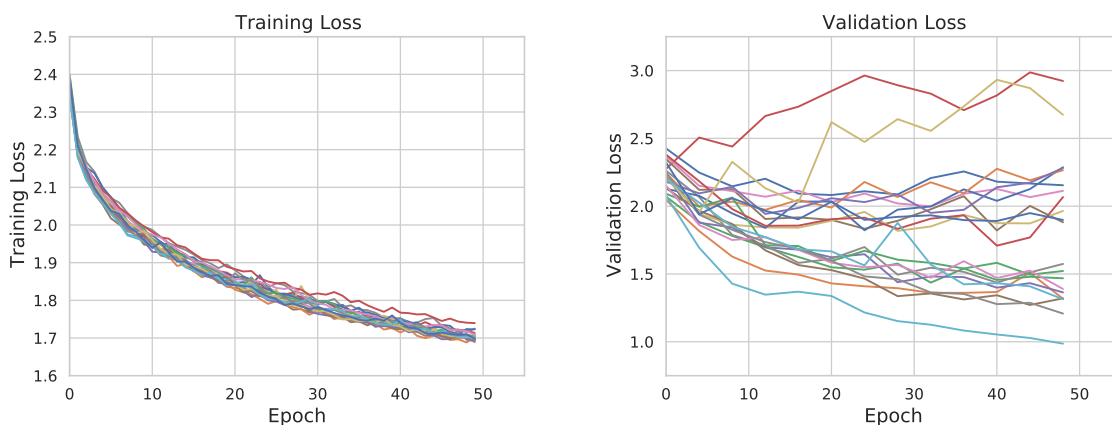


Figure 4.19: Training loss (left) versus validation loss (right) for every possible cross-validation sequence variant from independent training runs for 50 epochs with same network configuration.

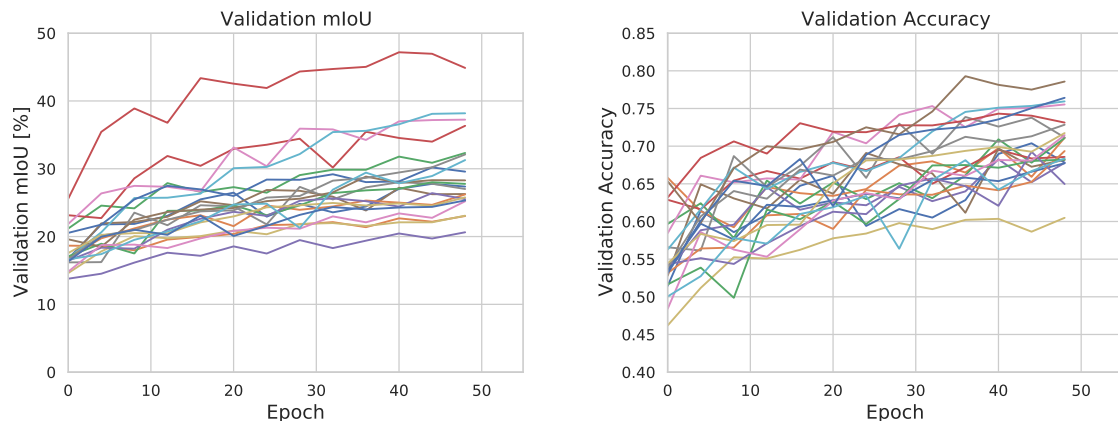


Figure 4.20: Validation mIoU (left) versus validation accuracy (right) for every possible cross-validation sequence variant from independent training runs for 50 epochs with same network configuration. The differences of the specific scene validation mIoU can be explained by the ground truth data. Besides the physical scene differences, see Section 3.4.1, the occurrence of each label type is uniquely distributed per sequence. Hence, the validation of the network is impossible without a special selection of a training-resembling validation data set.

As mitigation and to avoid cherry-picked results, the cross-validation sequences for the model optimization are chosen specifically to contain a different label proportion and check for an average performance on different validation sequences. The selected validation sequence set (05,08,09,12) reflects approximately the same class distribution compared to the overall training data split. As test sequence, Sequence 06 is chosen, due to the occurrence of all classes in a similar count, compare Figure 4.17.

4.5 RadarNet Segmentation Evaluation

As a result of the parameter optimization and radar-specific adjustments, the best performing RadarNet variant is found. The model variant yields reproducible performance, trained from scratch with the same initialization seed. The remainder of the evaluation is dedicated to generalization testing on unseen data in live-operation on a test vehicle.

In order to reproduce the model performance, four independent models with the same initialization seed have been trained independently for 200 epochs, to check the model convergence robustness. Based on the comparable result of the independent parallel models, their performance is discussed. Over-fitting is discussed as well, due to the increasing combined validation loss curve in Figure 4.22. The combined validation loss curve, left plot of Figure 4.22, shows the combination of the Lovasz-softmax loss and Cross-Entropy loss which needs to be discussed independently, as depicted as single component plots in the right of the same figure.

Utilized training parameters for the best performing RadarNet variant: Learning rate $lr = 0.01$, decaying by 25% every 10 epochs, training from scratch without pre-trained weights but same initialization seed, α -class weights for the Cross-Entropy loss but not for the Lovasz-loss, SGD optimizer and validation steps every 5 epochs.

The qualitative evaluation of the network segmentation performance is measured with the mean IoU over all classes. The Figure 4.21 displays the adaption of the network performance over the training steps.

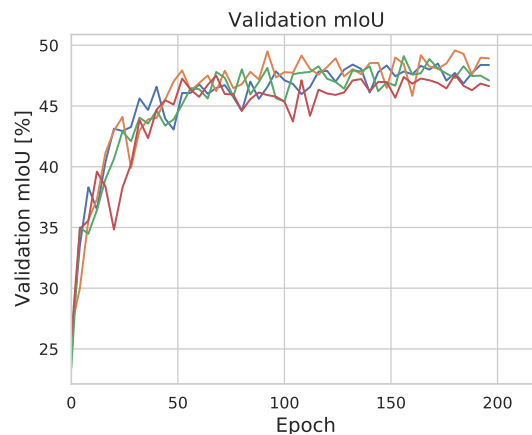


Figure 4.21: Overall validation mIoU (classes: *Artifact*, *Building*, *Vehicle*, *Vegetation*, *Pole*, *Person*) of the best RadarNet variant. These four independently trained model runs are compared.

Seemingly, all four independent network variants seem to achieve a similar mIoU between 45% and 50%, averaged to 47.6%.

How the optimization of the network parameters with respect to the loss function evolves during the training is depicted by the training loss and validation loss in Figure 4.22.

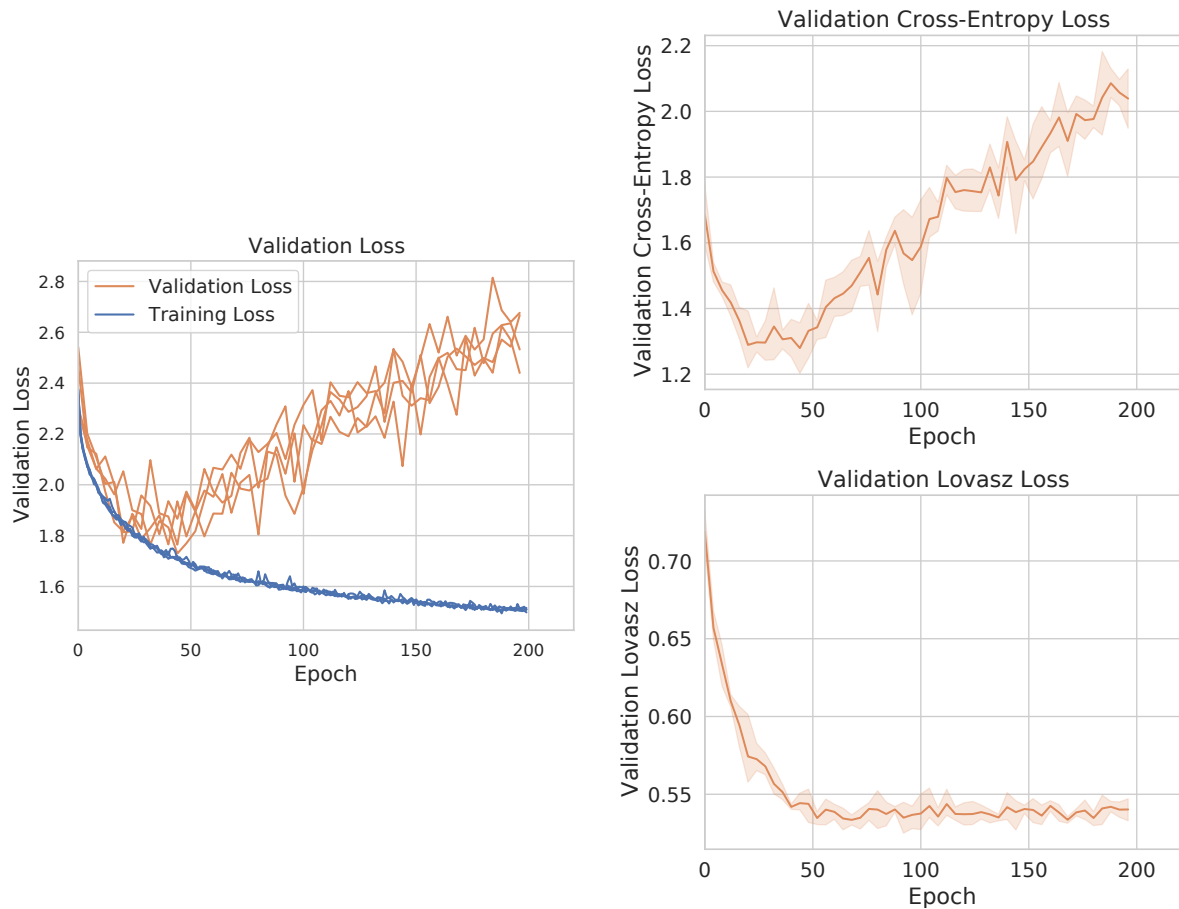


Figure 4.22: Training loss and validation loss of best RadarNet variants of four independent models (left) in comparison to the averaged loss components (right): Robustly decreasing validation Lovasz-Loss (top) versus increasing Cross-Entropy validation loss (bottom).

As expected, the Cross-Entropy loss supports the training to converge by a decrease of approximately 20%. But as special behaviour⁹, the Cross-Entropy loss continues to increase after this dip, in a nearly linear ascent, whereas the additional Lovasz-softmax loss term continues to first decrease, then remains saturated. In case of over-fitting, the Lovasz-Loss would be required to increase similarly. Consistent to the expectation to stabilize rare class segmentation, the decreasing and saturating Lovasz-softmax Loss component as sum over the six classes is capable to further optimize the mIoU of the network. This network adaption can be tracked by the consideration of the class-wise IoUs, especially of the *clutter* improvement in Figure 4.23 and the increase of the *person* class in Figure 4.24 to the cost of the *vegetation* class in the same figure.

⁹ This behaviour resembles an over-fitting tendency, but can be explained from unspecific examples of rare and non-trivial semantic class labels instead, compare with Figure 4.23-4.24.

Both Figures depict the increasing mIoU curves per class and reveal the saturation of all classes, except for the *clutter* and the *vegetation* class, seemingly as protagonist and antagonist respectively.

With the data set class split, the first three classes *clutter*, *vehicle* and *building* are represented with more samples, expect-ably yielding better segmentation results for this sub-set of classes.

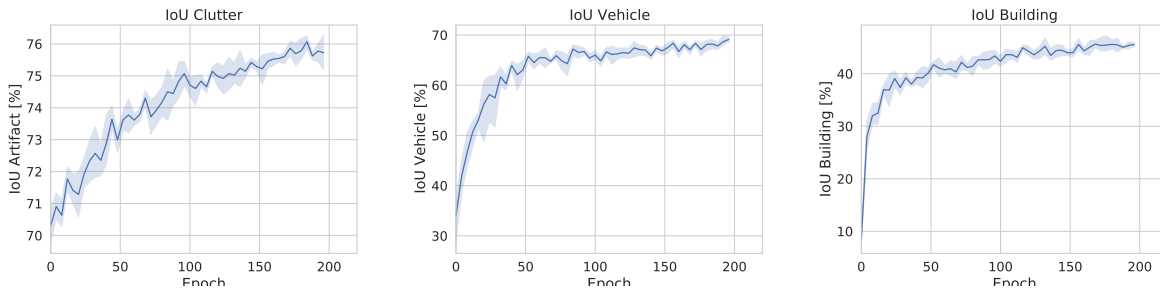


Figure 4.23: IoU of classes *clutter* (left), *vehicle* (center), and class *building* (right).

With the class-weights and the Lovasz-softmax loss, the remaining three rare classes *vegetation*, *person* and *pole* also yield fair results, despite the high data set imbalance, see Table 4.9. For a very low number of detections which the training could learn from, but enforced by class-weights and a class-sensitive loss term for the combined loss, RadarNet is able to generalize for these rare classes. As the architecture comparison reveals, and the theoretical findings of the authors of the Lovasz-Loss [20] claim, the Lovasz-softmax Loss significantly improves the mIoU, especially for the rare class samples.

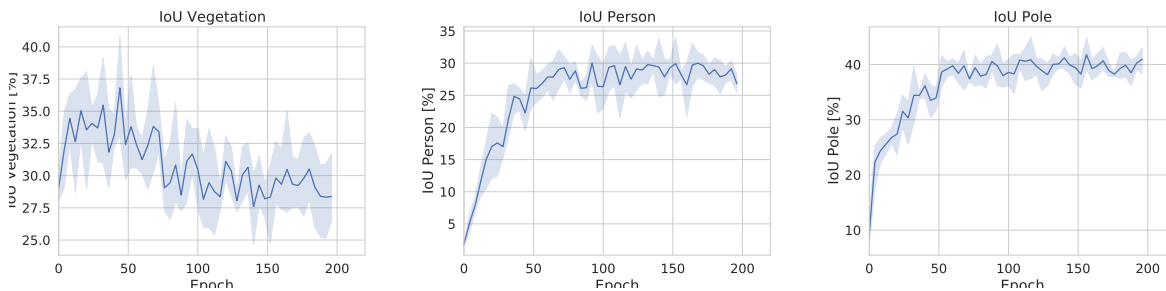


Figure 4.24: IoU of classes *vegetation* (left), *person* (center), and class *pole* (right).

Over-fitting: In Figure 4.22, the combined validation loss of Lovasz-softmax loss and Cross-entropy loss dips at approximately 30-50 epochs and increases with further training.

It is indetermineable if this behavior results from over-fitting or a result of rare or inconsistencies of the manually corrected labels. Motivated by the loss definition, cross-entropy loss increases for predictions with lower activation margin in the soft-max function. First, with the data being manually labeled, the manual correction step might introduce label inconsistencies. In the labeling process, only *plausible* detections are further split into semantic

classes. Based on the assumption of a correct *implausible* labeling, it is rated not beneficial - from a labeling perspective - to further inspect the *implausible* labels also. This assumption is bypassed for the *vehicle* class. *Vehicles* are treated as a detection accumulation, independent if detections occur inside or only from the outer shell. With this assumption, the detection robustness of this class is increased by re-labeling the *implausible* detections inside a *vehicle* to the *vehicle* class.

Secondary effects might result from the sensor itself. Based on the noisy sensor properties, it can not be guaranteed to have 100% accurate labels - noise and random clutter are systematic. There is a high chance of ambiguous classes or detections seemingly of a class, but being actually *clutter*. As a result, with such labels the network learns to be predict classes at a lower certainty margin due to the semantic classes being not fully separable clusters. Especially for the highly-scattered *vegetation* detections, the decreasing IoU supports this finding.

While the validation Cross-Entropy loss decreases in Figure 4.22 only until 25k samples, the Lovasz-softmax loss still continues to decrease. Comparing the IoU gain of the rare classes, the remaining training results in approximately 5% – 10% IoU gain per class, which the cross-entropy would leave un-optimized.

A test run of a very long training for 2000 epochs, continues the IoU saturation per class, achieving a final mean IoU of approximately 67%. Testing this models and intermediate models still achieve a generalization on completely new and unseen data. Hence, the slight model over-fit is specifically true for the *clutter* and *vegetation* prediction, the other classes are generalized sufficiently, compare Figure 4.27 for a generalization example.

The confusion matrix in Figure 4.25 results illustrate the similarity and confusion of the rare classes *person* and *pole* with the *vehicle* class. Revising the data set reveals the few *person*-samples to mainly occur besides vehicles - so the confusion with vehicles is traceable to this data set characteristic. To trace the confusion of *poles* with *vegetation* could either indicate trees, while the confusion with *vehicles* might originate from a similarity of metallic street signs and metallic reflections of a vehicle.

But more important, the confusion matrix also reveals *clutter* to be confused with *vegetation*. The confusion between these two classes results in the decreasing IoU and broader jitter-band of *vegetation* of Figure 4.24 ($\approx -4\%$ from epoch 30-200), while the *clutter* IoU of Figure 4.23 increases ($\approx +2\%$ from epoch 30-200). As weak indication, both IoU curves also reveal a similar jitter-band pattern. See also Figure 5.24 for an example of confused labels on a scene or Figure 5.25 on a larger scenario (plot c) versus plot d)).

Based in the confusion of little and rare classes, the motivation of an attention-based approach can be derived, to achieve increased context awareness of small objects. From the few examples, the network should predict static poles with respect to their always constant

Doppler velocity. In contrast, persons obviously resemble a pole shape, but are mostly moving in the data set instead of standing still.

Based on the spatial data distribution of the training data after coordinate flipping and rotation augmentation, no general spatial dependency can be learned by RadarNet. Consequently, the semantic segmentation performance is expected to generalize to the 360° sensed environment.

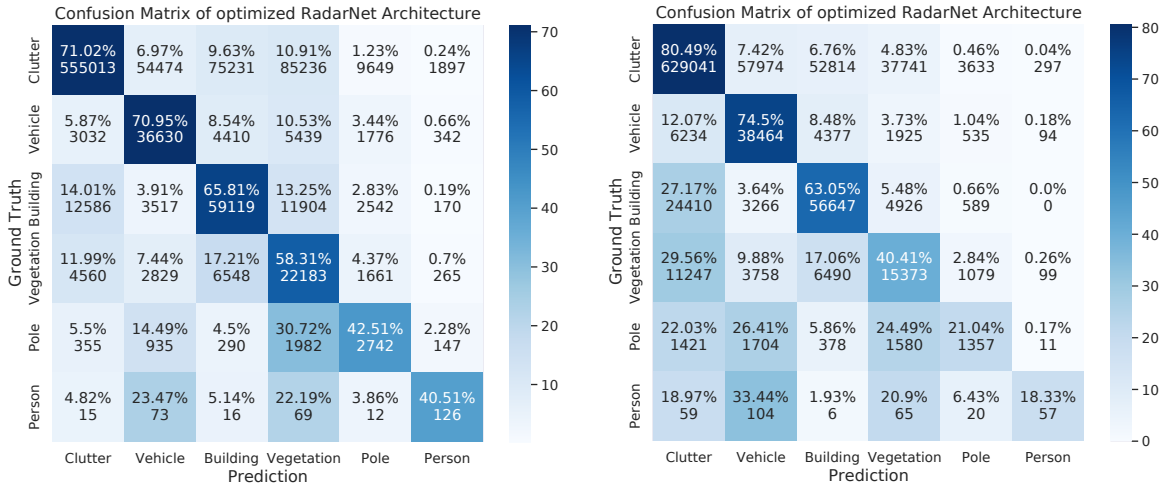


Figure 4.25: Illustration of the *RadarNet* confusion matrix at 52 epochs (left) and at 180 epochs (right).

Range Accuracy versus Near Field Inference: The class-prediction is also evaluated with respect to the sensed range of the object in Figure 4.26. For a subsequent mapping module, matching subsequent radar point clouds based on the semantic classes, it is important to evaluate in which spatial region the semantic prediction achieves accurate results.¹⁰ Especially for a moving platform, the classification result over the range yields important findings when point clouds of the same environment are sensed from different ranges, but need to be compared and matched.

From the spatial occurrence of classes, with increasing range also density of the classes decrease. Congruently, it was assumed that classification accuracy decreased with range due to a lower number of detections per object, and the decaying signal power P_{sig} with the fourth power of the range. Hence, the decaying IoU per class in Figure 4.26 behaves as expected. The *clutter* IoU increases with range due to the spatial and mostly exclusive occurrence in the far-range. As a result, the segmentation performance of e.g. *vehicles* and other classes depend on the spatial range of the corresponding radar detections. Radar detections in medium

¹⁰ RadarNet is trained in *fixed volume* mode, clipping detections beyond a 50 meter range from the training and inference. This configuration is chosen with respect to the use-case of trained parking, limiting the relevant sensor range to a 50 m radius.

detection range yield a correct semantic segmentation whereas far-range and closeby detections might be mis-classified. Either with more training data, distributed uniformly over the spatial range, or with increased sensory detection per object in the far-range, the perception performance can be increased. But no network modification can mitigate this data set and sensor resolution shortcoming.

The only potential network adaption would be an adaptive grid cell refinement, similar to a kd-tree refining the grid where it is populated. As such, the arg-max function in the feature extraction process would no neglect smaller or multiple objects at far range in polar-grid dependent large cell areas.

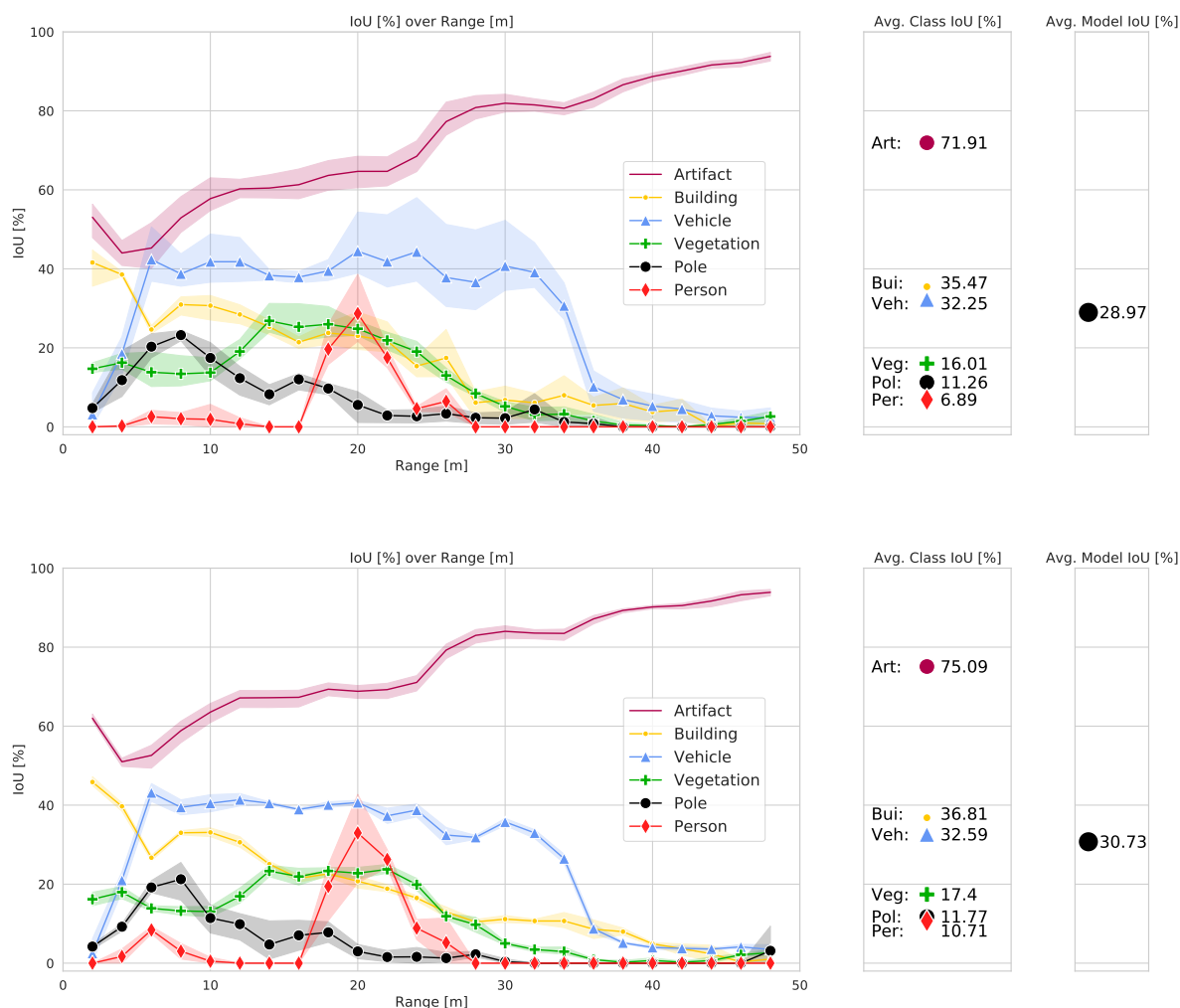


Figure 4.26: Illustration of the IoU per class over range for the same *RadarNet* model but at alternative training epochs evaluated in 2m range steps for $R \in [2, 50]$. The early epoch model (trained for 52 epochs) is depicted on top, with the same model, but trained for additional 128 epochs (180 epochs in total), in the bottom plot.

Segmentation Result Evaluation: Since the automatic labeling process generates labels which were manually corrected to ground truth labels, perfect labeling accuracy can not be guaranteed. The sensor noise yields in difficult labeling situations, also for manual correction by an experienced labeler. Technically, remaining noise in the data set results in an overall sub-optimal network adaption to a non-optimal certainty of predictions. This non-optimal certainty is found in the Cross-Entropy loss plot over the training epochs. First decreasing to a global minimum at circa 30-50 epochs, the validation loss increases while the network IoU and validation IoU continues to increase also. This is not rated as over-fitting but rather expresses the certainty margin of the Cross-Entropy loss to increase due to noisy labels. The reported results of the *RadarNet* are selected to be most reliable for the early model results, trained for approximately 50 Epochs.

The presented semantic segmentation network is aimed to be applied in a system architecture with a spatio-temporal filter module to achieve temporal consistent semantic segmentations, which are subsequently semantically matched in a SLAM module to a semantic radar map. Hence, a perfect segmentation is not necessarily required, but a better *RadarNet* performance yields a more accurate semantic radar scene map. In the context of this system performance, discussed in the next chapters, the achieved *RadarNet* generalization is proven sufficient to yield a new state-of-the art.

RadarNet Deployment : The online direct inference of the *RadarNet* on unseen data is tested as live implementation in the test vehicle. Implemented as module, subscribing to live sensor data, a direct inference and further application of the segmented data is realized. It is important to mention the required real-time capability of the network, since the signal routing and framing implementation of the required pre-processing requires additional computation time.

Compared to a validation scene in Figure 4.27, the inference of live data can not be compared to ground truth data. Visual inspection is the only possible measure and proves the generalization capability of *RadarNet*, similar as in Figure 4.27.

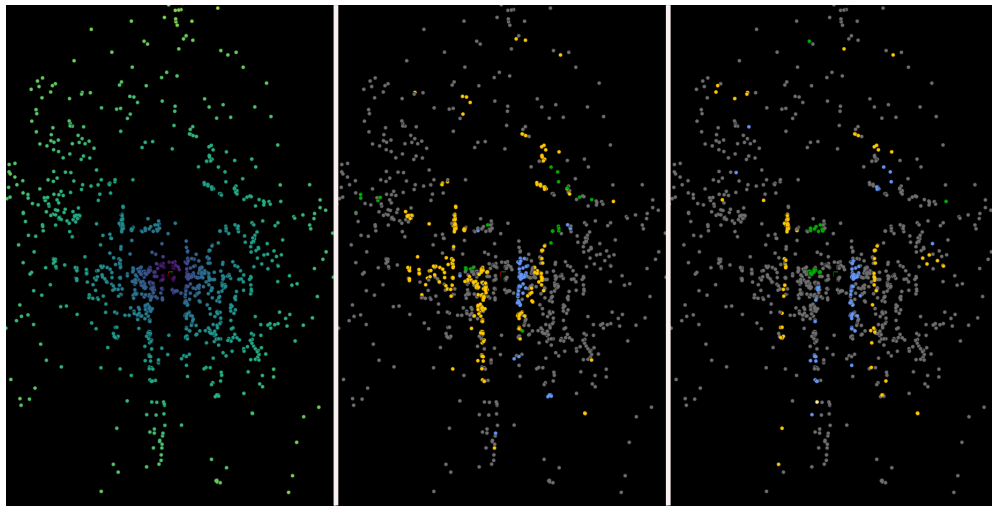


Figure 4.27: Top-View Illustration of the RadarNet deployment on the test data set: Raw radar point clouds (left), the semantic predictions of the best performing *RadarNet* (center) variant, compared to semantically labeled ground-truth radar detections (right).

Radar Scan Assembly: The assembly and accumulation of the radar clouds is an essential pre-processing step. The multiple independent sensor readings are assembled to a single 360° point cloud. As introduced for the data set labeling, a subsequent number of $N_{acc} = 3$ 360° radar scans are accumulated and treated as processed as single sensor reading.

4.6 Section Conclusion

This section covers the comparison of three state-of-the-art semantic segmentation network architectures, PolarNet [248], Cylinder3D [250] and ASAP-Net [31], which are originally designed and developed for dense 3D LiDAR point clouds, but are transferred to the sparse radar point cloud domain. The network architectures are adapted to radar data input, trained and evaluated on an own labeled and recorded real-world semantic radar data set covering $8.2 \cdot 10^6$ radar detections over a length of 2507.35 m, which is organized in semanticKITTI benchmark data set structure. Applying common performance metrics for LiDAR semantic segmentation on the radar data inference, the comparison of architectures, input channels and radar specific network adaptations is discussed. The potentially most radar-suitable architecture PolarNet is selected to be fine-tuned. Based on the PolarNet architecture, radar specific adaptations and extensive hyper-parameter tuning is applied and yields the direct radar point cloud segmentation network *RadarNet*. With this study, the radar point cloud segmentation and radar deep-learning potential of the *segmentation* research questions of Section 1.3 are answered.

Achieving 28.97% mIoU a new benchmark for semantic radar point cloud segmentation after training for 52 epochs, *RadarNet* is able to denoise and classify in real-time raw radar point clouds in six semantic classes including the static environment (*Clutter* 71.91%, *building* 35.47%, *vehicle* 32.25%, *vegetation* 16.01%, *pole* 11.26% and *person* 6.89%). Yielding a new level of data enrichment for automotive radar, an orchestrated sensor set of synchronized radar sensors can be processed in real-time with *RadarNet* to improve the environment perception.

The described findings address specifically a live semantic segmentation application for parking applications in low speed of an arbitrary structured, local environment.

As contribution of this chapter can be formulated:

- Structured architecture comparison and evaluation of measures for sparse point cloud feature extraction and semantic segmentation.
- Design of an independent real-time capable semantic segmentation network for radar point cloud segmentation.
- Deployment of the proposed segmentation network on a test vehicle and testing under arbitrary real-world conditions.

In the systematic context of Figure 1.4, the developed function *RadarNet* constitutes the indispensable basis of the semantic SLAM of Section 5 as system level block.

4.7 Section Outlook

The outlook covers a data set increase, potential feature improvements of the network architecture and an object-based labeling.

Data Set Extension: Foremost, the data set size applied for training *RadarNet* could be extended with specific samples of the rare classes (*person* and *pole*), to mitigate the high class imbalance. As other works on radar classification state [144], there occur harder and easier samples. Hence, the enrichment of the data set to additional situations and a plenty of data is key to improve generalization.

Attention-based local Feature Extraction: An attention-based approach is expected to improve inter-label consistency especially for rare classes as *pole*, *person*, and for the scattered *vegetation*. An attention based approach, extracting convolutional information from neighboring cells is expected to increase class discrimination. Applying the convolutional extraction in 3D coordinates with 3D convolutions, the advantages of the PolarNet 2D grid representation is expected to be boosted for the semantic segmentation.

Derived from the attention idea, also an attention-based label morphing step is suggested to compare labels in their local occurrence of a point cloud. Similar to the multi-scale interpretation of U-Net, it is suggested to formulate a grid or spatial consistency loss term to penalize mis-predictions in impossible cases, e.g. mixing a *person* label into a point cluster of an other structure, e.g. *building* or *vehicle*.

Panoptic Segmentation and Labeling: According to similar works on radar object detection, the course of clusters over time improves the temporal consistency [192]. With the above mentioned extension of the data set, also the count of dynamic objects samples should be increased, so that the combination of object detection and semantic segmentation towards panoptic segmentation can be tested.

The utilized data set can be labeled as extension to this thesis with additional instance labels for static objects and moving clusters. By this, a new benchmark for radar data sets and direct point cloud segmentation can be established and separation of neighbor clusters can be trained to be improved. The loss function could include an other penalty term for mixed or missed instances. Especially the mixed-class region or space is expected as beneficial, since the IoU or Cross-Entropy only considers each single point of a point cloud without a context or neighborhood consideration. E.g. every single detection of a *person* or a *vegetation* inside a *vehicle* cluster is rated with the same loss penalty as a mis-classification else-where. Including a greater context or a spatial loss-averaging and multi-scale comparison could further reduce semantic mis-classifications to a greater label consistency.

5 SEMANTIC RADAR SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)

This semantic registration formulation to enrich the radar SLAM to respect semantic features is developed in collaboration during the supervision of the master thesis of Fabian Haas-Fickinger [MT5]. As summary of this development, fractions of this chapter are published in the conference paper *SeRaLoc: SLAM on semantically annotated radar point clouds* [SI4], showcasing the essential accuracy gain of the resulting environment map. This chapter gives a detailed overview of the developed semantic radar SLAM, a signal filter, the front-end construction and the semantic radar registration method.

5.1 Motivation

Specifically in the environment mapping context for automotive applications, Radar is not commonly applied. But static detections of multiple radar sensors, be assembled to a driving vehicle, can be combined and assembled over time. The driven path enables to build a radar map of the driving scene. With advantageous combination of robust sensing properties, independence of outer lighting or weather conditions, paired with the direct measurement of relative velocity makes radar sensors inevitable as future mapping sensor. This section covers the general mapping problem setup, also known as Simultaneous Localization and Mapping (SLAM), and its formulation as graph problem to a graph SLAM. Furthermore, the graph SLAM front-end is improved by the use of additional specific radar information and constraints for the special radar application are integrated.

Online on-board mapping of the static environment, based on live radar detections, allows a new dimension of scene interpretation. Relating the current scene with an initial position guess, the ego-position can be matched and refined by a registration with a pre-recorded map. Typically, radar sensors provide information in a mid-range of up to 60 m [114, 160, 225]. For automated systems, having a online radar reference map at hand, such a sensor range can be useful to determine the possible path to drive and orientate the ego-vehicle with respect to the environment.

The radar layer bridges between short range information of ultrasonic sensors or short range cameras on the short end, to the gps-based localization in the coarse driving context. Having a local radar map of the mid-range environment, automated parking functionalities and

potentially other driving functions can be developed and deployed based on this on-board sensors and perception. As final result of this chapter, an accurate and metric radar map of an arbitrary environment is available.

5.2 Evaluation and Selection of existing Approaches: Localization and Radar SLAM

In this section, first the fundamental topic of localization research is introduced. Secondly, localization and mapping in the focus of radar applications is presented.

5.2.1 Localization

The research on localization can generally be tailored for different applications and various sensor types. E.g. 3D visual information of RGB-D, stereo or monocular cameras [148] or classical LiDAR point cloud SLAM [246, 218] is common state-of-the-art. With this thesis focusing specifically on the benefits of radar point cloud processing, and this section discussing the localization problem with radar, the current radar-specific research or transferable approaches are introduced and compared to their potential radar localization application.

In the context of this thesis, the radar-based localization is required to work robustly for an autonomous parking use-case in static environments, without prior knowledge of the environment features, objects, or overall geometry. In order to yield a general applicable solution, discussing radar localization along three potential scales of radar processing from descriptors, over objects to the whole scene geometry, the localization state-of-the-art is discussed and related to the problem at hand. This overview helps to understand the basic problems, the subsection of radar-SLAM state-of-the-art needs to cope with.

Radar Descriptors: Radar descriptors can be considered as landmarks, based on which localization can be performed. To find correspondences or associations between different data-patterns e.g. for localization relative to a specific pattern, a distinctive and characteristic description of the pattern is classically applied. Such descriptors for point clouds can be hand-crafted, based on a manually defined rule-set or algorithm, but can alternatively be subject to learning approaches.

As descriptor, local characteristics of the data is abstracted. A classical measure are histograms, representing a set of local characteristics. Examples are point-feature histograms [179] and the rotation invariant extension of the SHOT descriptor [181], which rely on the local surface. The RoPS approach of Guo et al. [84], applies rotation projection statistics of the local environment of a point cloud to generate a histogram-based descriptor. Similar, the

work of Scheiner et al. [184] describes the combination of the Cumulative Binary Occupancy (CBO) grid with a convex hull area for moving radar object detection.

Enriching the manual descriptors with a learned, data-driven extension based on a grid discretization of the point cloud, Zeng et al. [244] achieve robust descriptors, but at the price of long processing times.

The most prominent and efficient learning approaches of descriptors of point clouds are based on the feature processing of PointNet[169] and PointNet++ [170], discussed in Section 4.3.1 as feature extraction step for the semantic segmentation. But in the context localization and environment descriptors, missing local geometric relation are a drawback, since convolutions are not included. Further deep learning approaches are presented in Sections 4.2 - 4.2, but with no special focus on landmarks or descriptors, as the following works focus on.

Building on PointNet, in the works PPFNet of Deng et al. [53] and PPF-FoldNet of Deng et al. [52], the local context is included in the feature extraction. Multiple independent feature extraction PointNets are applied on local point cloud patches, which are re-combined by a max-pooling and concatenation of the extracted local PointNet-features to a global point cloud feature set.

For 3DSmoothNet, the authors Gojcic et al. [76] suggest a learned feature extraction and robust point cloud matching by a siamese deep learning CNN. The authors propose as feature extraction a voxelization of the input point cloud to a normalized single value decomposition(SVD) 3D voxel grid, serving as input to a proposed CNN, the two given point cloud segments are matched. This approach states well performance on sparse LiDAR data (tested until 12.5% density), but also states a systematic deprecation the sparser the data gets.

In general, manual descriptors rely on the properties of the local neighborhood and deprecate significantly for sparse point clouds, geometrical as well as statistical. Especially for the radar data at hand, e.g. surface planes or even surface normals can not be processed due to the lack of radar detection density. Learned descriptors are generally more robust to sparsity, since their local context is computed by convolutions or concatenation. Nevertheless, due to the variable sparsity of the radar point cloud in real-world conditions, the descriptor distinctiveness is unknown. As a result, it is assumed not to rely on descriptors for a robust localization.

Radar Object Detection: Instead of relying on point cloud descriptors for a robust localization, the larger context of the point clouds is considered. Aiming to recognize a landmark, either in the context of localization, or more generally to recognize an object in raw sensor data, there exist a variety of approaches beyond simple descriptors. Ranging from sensor data to be represented in a grid structure or image introduced in Section 4.2.1, to the other end of direct processing of the sensor data by deep-learning methods in Section 4.2.2, the

solutions, mainly applied for radar-based object detection are diverse and might even be combined with other raw data processing steps.

Considered in the context of the localization problem, the limited static object detection approaches lack detailed classes and lack real-time capability to be applied for the static radar environment perception and localization. Hence, the object scale of radar perception cannot fulfill the requirements for a radar-based localization.

Point Cloud Scan Matching: Instead of recognizing specific objects, the whole point cloud can be considered for localization. The state-of-the-art point cloud matching ranges from classical mathematical distance formulations to learning systems, similar to the already mentioned 3DSmoothNet [76].

As classical standard solution, the Iterative Closest Point Algorithm (ICP) is commonly applied in the context of SLAM, e.g. in ORB-SLAM [59] or Suma++ [39]. With an initial guess, the most overlap of two point clouds is calculated by an iterative quadratic distance error minimization. This registration process relies directly on points to be associated. The ICP is simple to implement, quick, but tends to yield sub-optimal registration results at presence of noise or outliers. The quadratic error formulation might yield convergence to local minima in such cases.

Global optimal convergence is achieved by Go-ICP [233], independent of the initial position. By searching the whole parameter space by a Branch-and-Bound procedure, the optimal solution can be found even for noisy data. Drawback of the Go-ICP is the non-real-time run-time, arising from solving iteratively multiple registration processes for sub-sets of the point clouds.

In to improve the data association, the related work of Chen and Medioni [41] suggests point abstractions to add robustness to the data association. A form of abstraction is the representation as a distribution. Magnusson [137] suggest the Normal Distribution Transform Matching (NDT) as alternative registration class and utilizes the procedure to register loops [138]. Discretizing the target point cloud in a 3D voxel or 2D grid, the core idea is to represent the points per grid cell as normal-distribution and match this target distribution with the source point cloud by a k-NN search. With the mathematical formulation of normal distributions as representation of the point set, the effect of outliers and noise is systematically reduced.

Hong and Lee [92] extends the normal distribution discretization to the source point cloud, yielding a Model-to-Model matching of two normal distributions, achieving improved robustness against outliers.

Similar to the problems of sparse neighborhoods for descriptor processing, sparsity in point clouds yields problematic normal distribution representations. The more points per cell are available, the more precise the normal distribution is describing the set. Schulz and Zell [191] adapt the NDT to sparse LiDAR point clouds of an occupancy grid. Jun et al. [107] avoid

empty or over-populated cells, geometric representation of whole objects, and improve run-time performance by proposing dynamic voxel sizes, depending on the sensed geometry in the point clouds. This method is tested on RGB-D and LiDAR point clouds.

As radar-specific related work on distribution matching, the efficient multi-resolution Correlative Scan Matching (CSM) [154] is modified by Li et al. [125] to be applied in a radar SLAM context, matching the radar scan to a radar grid map. As *Scan-to-Map* procedure, the suggested work applies a mapping of subsequent radar scans to a map, in order to compare this *dense* map as distribution with the radar scan. The suggested method is applied on a single front-facing radar.

Similar to the 3DSmoothNets' registration CNN and applicable to the extracted features proposed by 3DSmoothNet, the Teaser++ [232] registration algorithm outperforms the registration of 3D point clouds with Go-ICP. Formulated as Truncated Least Squares (TLS) problem, the graph-theoretic formulation decouples the estimation of scale, rotation, and translation. Achieving real-time applicable run-time, significant outlier robustness of assumed correspondences, the necessary of correspondences disqualifies the methods from sparse radar scans which pose the previous problem to define a feature extraction from radar scans.

Learnable approaches such as the Deep-ICP [134] work on this gap, combine feature extraction based on PointNet with the ICP registration with SVD. The hard feature correspondences are weighted based on a saliency map in order to emphasize strong correspondences. Besides achieving an improved overall registration error compared to ICP and Go-ICP, the registration precision of the end-to-end trainable Deep-ICP decreases at an overall increased run-time.

Yew and Lee [235] propose RPMNet to improve the problems of local ICP convergence due to the effect of noise, outliers on hard correspondences. RPMNet applies one network to find soft correspondences in form of a correspondence matrix from geometric and local information, and proposes a second network to estimate annealing parameters. The method increases a robust initialization, handles missing correspondences and copes well with occluded point clouds at reasonable run-time. Applying hybrid features, the RPMNet architecture utilizes handcrafted features 4D point pair features (PPF) [179, 51] as additional input to the PointNet feature extraction of the point cloud. These handcrafted features are again sources of severe instability, based on the radar point cloud sparsity.

OverlapNet of Chen et al. [38] works with dense LiDAR point clouds in a SLAM context which typically involves subsequent scans, typically overlapping by a large fraction. Achieving fair registration results on dense and accurate LiDAR data, especially for loop closures [40], the application to low-density point clouds is questionable. OverlapNet applies RangeNet++ [145], which relies on a cylindrical projection of the point cloud as range image. Similar to the semantic segmentation in Section 4.2.1, this representation is unsuitable for sparse radar data.

The presented state-of-the-art registration procedures reveal the problematic transfer of these methods to an application relying only on sparse radar point clouds. Providing robustness to radar noise and radar outliers, being independent of point cloud sparsity in the local neighborhood, the NDT registration promises the best applicability.

Learn-able registration algorithms have the potential to add robustness to the registration procedure but are not yet tested for sparse or noisy data. Especially a real-time capable feature extraction of normal distributions and additional radar attributes such as radar cross section, semantic segmentation labels, or others appear to be promising future research.

5.2.2 Radar Simultaneous Localization and Mapping

With the before introduced radar features and registration schemes, the single radar scans can be spatially related in order to form a radar map of the environment. With the focus on radar, the general foundation of SLAM state-of-the-art is not presented in the section, but instead exclusively specific difficulties of radar SLAM is discussed. Interested readers on SLAM basics are referred to Thrun et al. [213], Grisetti et al. [82], while Cadena et al. [29] summarizes the past of SLAM developments. Specific radar SLAM approaches will be presented in this section, but also general SLAM extensions to respect semantic information in the SLAM context.

Marck et al. [140] implements a landmark indoor localization radar SLAM, based on a Grid Mapping Particle Filter (GMPF) [81], as one of the first radar SLAM works. The authors prove the general applicability of radar SLAM, using one rotating FMCW radar sensor. Converting the radar data to a 2D image, the ICP registration is coupled with a particle filter to refine the ego-motion estimates.

Extending the sensor-set to four FMCW radar sensors mounted on a real-world vehicle, Schoen et al. [189] combine the 360° 2D point cloud and construct an occupancy grid map as environment representation. Achieving high positioning accuracy of 1.12 *m* over an "8"-shaped 400 *m* test track with loop closures, at real-time capability, this approach serves as successful real-world test.

Schuster et al. [195] develops a first BASD [172] descriptor landmark-based radar graph-SLAM on similar data as Schoen et al. [189]. The achieved mapping performance is rated as successful, while the localization performance yields comparable results of a mean error of 1.0 *m* over an "8"-shaped 400 *m* test track with loop closures, but fails to reconstruct the scene partially.

Hong et al. [93] test the radar SLAM capabilities under harsh weather conditions, such as heavy rain and snow. The applied high resolution, rotating radar data is applied from the Oxford RobotCar Radar Data Set [15] and MULRAN [115] data set and represented as 2D

radar image data. Image feature extraction based on BLOB-features is paired with graph-based outlier-detection to detect and exclude dynamic objects. An ICP is applied to register these features and to detect loop-closures from radar image-derived histogram descriptors. The method is compared to state-of-the-art LiDAR approaches as SuMa++ [17], and achieves comparable results of averaged $\approx 2 m$ per a $100 m$ segment.

Data association based on power-range spectrum defined landmarks and geometric inter-landmark properties, respected in the matching by Cen and Newman [32]. In order to estimate the ego-motion, the radar detections of a rotary radar sensor are assembled according to the inverse estimated ego-motion. The subsequent works of Cen and Newman [33], Aldera et al. [8] and Barnes and Posner [14] estimate the radar odometry as well, based on gradient radar image key-point feature matching. Further works address the radar-odometry estimation by weakly-supervised attention learning [7].

Mainly focused on imagery radar-based place recognition are radar-image based CNNs, e.g. the NetVLAD [180] or the radar image sequence-based fully connected network of Gadd et al. [68].

The PhaRaO framework [158] suggest also to represent the radar data as 2D image and estimates the radar odometry. But instead of feature extraction, the image is further converted into the frequency domain. Applying a Fourier-Mellin-Transformation on subsequent radar images, the rotation and translation estimation is decoupled. Achieving real-time run-time, the achieved positioning error yields $\approx 13 m$ over a $2000 m$ test track. The procedure depends on a dense radar point cloud, in order to generate a significant Fourier domain image and estimate a robust radar motion estimate.

Holder et al. [90] is the first landmark-free formulation of a radar graph-SLAM. Besides the radar point cloud based ICP registration, also the wheel-based odometry estimate is included for the graph construction to guess the vehicle motion. Also, the radar point cloud sparsity is overcome by a windowing approach to accumulate subsequent radar point clouds to a *sub-map*. Similar to the finding of Li et al. [125], the authors find that radar sparsity causing erroneous registration can be improved by matching of accumulated sections or sub-maps, achieving higher registration accuracy. Moreover, Holder et al. [90] combines the radar image transformation to extract GLARE-features [87], similar to the radar SLAM of Hong et al. [93]. The authors have tested a single front-facing radar in a real-world scenario, mapping the environment to a point cloud map. To focus on specific radar features is questioned by the authors. Due to the variety of radar reflections, the model-free, landmark-free SLAM solution is emphasized.

Narula et al. [151] proposes an automotive radar-only localization with respect to a given urban radar environment map and achieves a localization error below $0.5 m$. The point cloud

registration is based on batching subsequent radar scans and apply correlation-maximization to yield globally-optimal registration even for automotive radar noise and clutter.

Among these state-of-the-art SLAM works, none includes semantic information of radar.

Semantic Simultaneous Localization and Mapping (SLAM): Besides the 2D/3D spatial sensor data, additional data dimensions e.g. semantic labels, are a valuable environment description for SLAM systems.

Early works represent the environment geometry as semantic *quadrics* in a graph-SLAM formulation [152], but do leave the semantic information as future constraint of higher order geometric constraints on matching quadrics.

Bowman et al. [23] includes the association of semantic landmarks as expectation maximization problem, coupled to the second problem of a classical SLAM optimization. The semantic key-point information of a camera image is incorporated as semantic feature dimension into the existing visual GTSAM graph-SLAM framework. The extension work of Bowman et al. [24], further researches multiple semantic key-point features per objects and general environment. The authors prove that the additional semantic feature dimension does not add SLAM complexity on the underlying factor graph-formulation, but improved the overall positioning accuracy.

Doherty [56] combine the before mentioned works with a marginalization of the SLAM graph, yielding a practical semantic SLAM evaluation based on visual data.

Other visual SLAM approaches e.g. DS-SLAM [239] are mainly based on semantic key-point features and extend the works of the ORB-SLAM [149] family.

As discussed earlier, key-point similar radar-descriptors or radar features are instable due to the radar point cloud sparsity.

The works of SuMa [17] and SuMa++ [39], incorporate the semantic information fundamentally different and are developed on LiDAR point clouds. Instead of considering the semantic information as condition, the authors apply a semantic weighting directly on the ICP registration. In the ICP registration, the point-correspondences are weighted directly, based on their semantic class. Hence, no abstract descriptor is necessary, but the point cloud with semantic labels from RangeNet [145] itself can be applied.

Similar to the semantic weighting, the work of Zaganidis et al. [241] and Zaganidis et al. [242], both suggest to support the registration process by an inclusion of semantic attributes in the NDT registration. Additionally, the potential application of a semantic histogram descriptor as loop-closure indicator is applied, similar to the loop-closure detection by a histogram feature of Holder et al. [90].

Approach Evaluation: The advances in radar data processing include a first semantic segmentation, which enables to combine the radar point cloud SLAM with this semantic information as next step. To the authors best knowledge, the combination on a multi-radar sensor set delivering 3D radar detections with the direct semantic segmentation of e.g. *Radar-Net* is a comparably new research field. The transfer of *classical* SLAM approaches e.g. from LiDAR, to solve a radar SLAM are not trivial, due to the severe differences of the sparse radar data. Radar descriptors, radar features or radar object detection are not applicable or deprecate due to the low radar density.

Expecting the noisy and low density radar point clouds to be efficiently represented by normal distributions, the NDT approach is selected to be paired with the semantic segmentation labels of the radar. Selecting at a model-free, generally applicable radar SLAM approach by a combination of a semantic NDT registration, wheel-based odometry estimation, combined in a pose-graph SLAM formulation, a robust and semantically consistent data-association is assumed. By a radar specific extension of the mapped regions, the idea of sub-map matching is aimed to improve the SLAM loop closure and avoid local wheel-odometry induced SLAM drift. As further contribution, the applicability of the semantic weighing or a semantic seperation of the NDT-based radar registration are discussed.

5.3 Semantic Radar SLAM Method

In this section SLAM modifications are introduced, which are evaluated along experiments in Section 5.4: The radar raw signal filtering cascade is designed as three stage pre-processing. For the integration of the live semantic segmentation inference discussed in Chapter 4, Section 5.3.1 introduces a radar sensor accumulation to be applied before the segmentation network enriches this ego-motion compensated radar point cloud with point-wise semantic labels.

The subsequent regional pre-filter of Section 5.3.2, defines a general region of interest (ROI) in which the mapping-relevant radar detections are located. This pre-filter is designed as pass-through filter, including e.g. a threshold of SNR , and its effects are found in the experiments, Section 5.4.1. Furthermore, a second spatio-temporal filter module of Section 5.3.3 reduces remaining clutter, contained multi-path reflections and other artifacts to a minimum as Section 5.4.2 shows. Filtering spatially unstable detections builds the basis for a robust static environment mapping. With the semantic labels available, this filter step also considers the semantic consistency to remove semantically inconsistent radar detections.

Section 5.3.5 explains the construction of the graph SLAM formulation of different edges, partly evaluated in Section 5.4.3, followed by the discussion of two semantic NDT registration schemes in Section 5.3.4. Two registration schemes to weight or separate the semantic classes in the NDT registration, see Section 5.3.4.2 or Section 5.3.4.1 respectively, are tested in Section 5.4.4.

An overall evaluation of the developed semantic radar SLAM is found in section 5.4.5.

5.3.1 Accumulation

In order to reduce point cloud sparsity in a scan, the vehicle odometry is applied as ego-motion compensation to spatially concatenate multiple temporally subsequent radar scan sequences. For the independent $i = 1..6$ radar sensors delivering per sensing cycle at time t_i a radar point cloud $\mathcal{P}_{\text{radar},i}(t_i)$ in sensor coordinate system S_i , ego motion compensation is applied to assemble the single sensor point clouds $\mathcal{P}_{\text{radar},i}(t_i)$, with respect to the ego-vehicle motion. Based on wheel odometry, a pose transformation of the radar scans to a reference frame V_{ref}

$$\mathbf{T}_{S_i}(t_i) = \mathbf{T}_V(t_{\text{ref}})^{-1} \mathbf{T}_V(t_i) \mathbf{T}_{S_i}, \quad (5.1)$$

is applied for each single radar sensor point cloud. Then, the point cloud is shifted by this transformation

$$\mathcal{P}_{\text{radar},i,V_{\text{ref}}} = \mathbf{T}_{S_i}(t_i) \mathcal{P}_{\text{radar},i,S_i}. \quad (5.2)$$

For the accumulation, the latest of all $i = 6$ radar scans is chosen as *reference* scan which defines $t_{ref} = \max(t_i)$ for $i = 1, \dots, 6$.

With this reference projection, the resulting assembled point cloud is formalized as

$$\mathcal{P}_{\text{radar}, i, V_{\text{ref}}} = \bigcup_i \mathcal{P}_{\text{radar}, i, V_{\text{ref}}}. \quad (5.3)$$

In order to apply the wheel-based ego-motion compensation, a local consistency of the wheel-based odometry is assumed. Based on Werling [223], this assumption is valid at a sampling frequency at 10 Hz and ego velocity below 15 kmh.

5.3.2 Pre-Filter

The first filter stage is rule based. Motivated from a sensor integration simulation of the bumper-integrated sensors, systematic deflections in very close proximity of each sensor occurs, see Section 2.1. Hence, filtering too close detections and detections out of a reasonable range removes systematic misleading detections. For a mapping of the environment, especially considering parking scenarios, the range is limited to 40 m to map the near-range of the vehicle first, before increasing the scene range.

Summarized in Table 5.1, the applied parameters are combined as logical AND to define a ROI for the mapping.

Table 5.1: Rule based pre-filter parameters.

| Attribute | Minimum | Maximum |
|------------|---------|---------|
| Radius r | 0.2 m | 40 m |
| Height z | -0.1 m | 3 m |
| SNR | 20 | - |

The result of the rule-based pre-filter is shown in the experiments section of this chapter, see Figure 5.15.

5.3.3 Spatio-Temporal Filter

Core of the filtering is based on the assumption of static radar detections in global space over temporal subsequent scans, based on sensor motion compensation. As filter core idea, the spatial occurrence of static detections is addressed. Considering subsequent ego-motion compensated sensor scans, the global position of the areas in which radar detections occur, gain probability to describe a real object with each nearby subsequent detection. Similar to

the *plausibility* label in the automated labeling Section 3.3.2, a threshold based point-wise binary qualifier $\hat{y}_{\text{filter}} = 1$ allows a spatio-temporal stable radar detection to pass the filter. As distinguishing characteristic, multi-path reflections or clutter occurs scattered in spatial and temporal dimension and gets filtered.

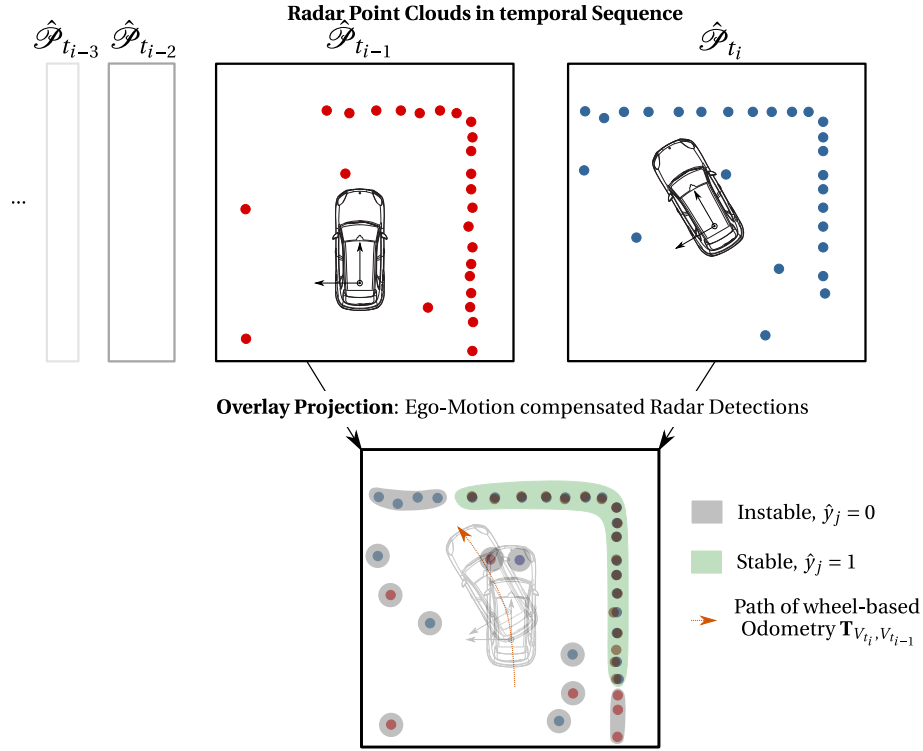


Figure 5.1: Spatio-temporal filter principle with binary label $\hat{y}_{\text{filter},j}$ to determine static, stable radar detections and filter instable detections.

Three modules comprise the filter implementation. First, the *detection model* discretizes the radar point cloud $\mathcal{P}_{\text{radar}}$ in a Cartesian grid representation. For each grid cell a detection probability $p_d(G_i, t)$ is defined, based on the points per grid cell G_i .

The second *activation model* defines a memory functionality to memorize active areas. The detection probability $p_d(G_i, t)$ per cell is processed to an activation probability $p_a(G_i, t)$ per cell.

As final stage, the *perception model* compares the radar point cloud $\mathcal{P}_{\text{radar}}$ to the perception probability function per cell $p_p(t, G_i)$. If the detection probability $p_d(t, G_i)$ exceeds a threshold, the binary qualifier $\hat{y}_{\text{filter}} = 1$ is granted to pass the filter finally.

1. Detection Model: The 3D radar point cloud coordinates (x, y, z) are discretized into a 2D grid in the Cartesian coordinate system V_f to (x_f, y_f) with k semantic layers, see Figure 5.1. The goal of this grid discretization is to calculate per point cloud $\mathcal{P}_{\text{radar}}$ a grid \mathbf{G} with a detection probability $p_d(G_{i,k}, t)$ per grid cell G_i of semantic class k . The 2D Cartesian grid

origin of V_f is fixed to the moving vehicle reference coordinate system "base link". The grid origin of V_f is moving with the vehicle odometry, but its orientation remains globally fixed with respect to the global coordinate system "odom".

In order to enable a later separation of the radar point cloud into its semantic components, each grid cell G_i of the grid \mathbf{G} , contains six semantic dimensions $k \in [0,5]$. According to its semantic class k , each radar point is associated with its corresponding grid index i and semantic dimension k , see Figure 5.2.

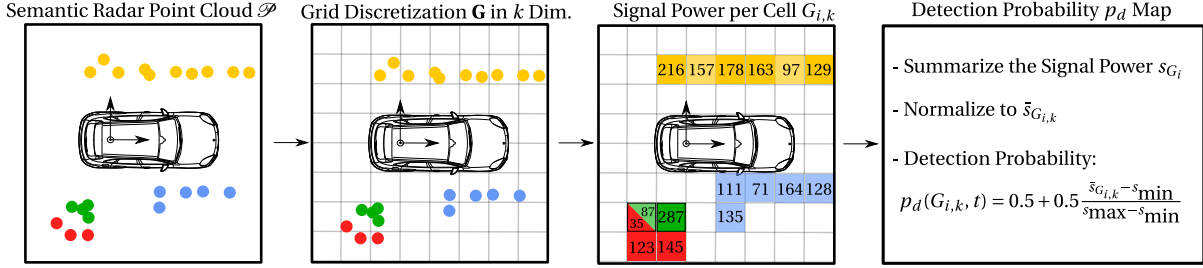


Figure 5.2: 2D cell discretization of the 3D radar detection points with processing steps to the calculation of the detection probability $p_d(G_{i,k}, t)$. Colors according to the semantic radar classes of Table 3.2.

Over all radar points $p_{\text{radar}}(G_{i,k}, t)$ per cell $G_{i,k}(t)$, the summarized signal power $s_{G_i}(t)$ can be calculated as

$$s_{G_i}(t) = \sum_{p_{\text{radar}}(G_{i,k}, t) \in \mathcal{P}_{\text{radar}}(t)} P_{\text{sig}}(p_{\text{radar}}(G_{i,k}, t)). \quad (5.4)$$

As the signal power is correlated according to Equation 4.1, the summarized signal power $s_{G_i}(t)$ can be interpreted as reflectivity indicator per grid cell G_i .

To yield a normalized value per cell, this normalization to the interval $[0.5, 1.0]$ is introduced

$$\bar{s}_{G_{i,k}} = \begin{cases} s_{\min} & \text{for } s_{G_{i,k}} < s_{\min} \\ s_{G_{i,k}} & \text{for } s_{\min} \leq s_{G_{i,k}} < s_{\max} \\ s_{\max} & \text{for } s_{\max} \leq s_{G_{i,k}}. \end{cases} \quad (5.5)$$

With the normalized signal power per cell, the detection probability $p_d(G_{i,k}, t)$ is defined. In order to yield interpretable values in the interval $[0.5, 1.0]$, Equation 5.6 is applied. The selection of this interval is motivated by the definition of an activation model.

$$p_d(G_{i,k}, t) = 0.5 + 0.5 \frac{\bar{s}_{G_{i,k}} - s_{\min}}{s_{\max} - s_{\min}} \quad (5.6)$$

With $p_d(G_{i,k}, t)$, at time t a formal definition of a detection probability per cell G_i for each semantic class k is given. A minimum of $p_d(G_{i,k}, t) = 0.5$ represents the highest uncertainty if the detection in this cell are plausible or artifacts. With $p_d(G_{i,k}, t) = 1.0$, the highest detection

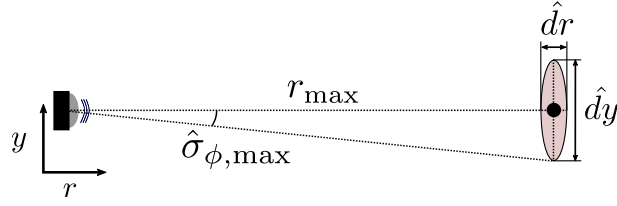


Figure 5.4: Position uncertainty illustration of the lateral \hat{d}_y and radial \hat{d}_r uncertainty components. Exemplary plot of a point detection, based on the sensor error of Figures 2.13 and 2.14.

probability is reached, defining plausible detections per cell. An exemplary illustration of the normalization and limitation of the detection model can be found in Figure 5.3.

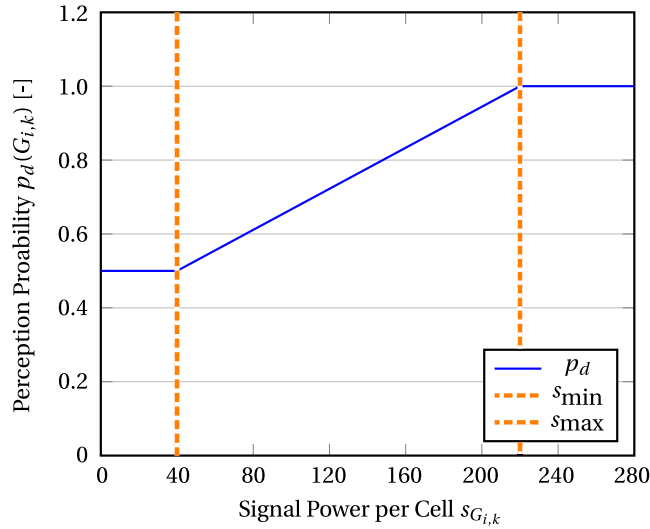


Figure 5.3: Illustration of the perception probability $p_d(G_{i,k})$ per cell-wise accumulated signal power $s_{G_{i,k}}$, plotted for an exemplary discretized filter cell $G_{i,k}$, with $s_{min} = 40$, $s_{max} = 220$.

The definition of Equation 5.5 depends on the limit values of s_{min} , s_{max} and the grid size d_x of the cells $G_{i,k}$. Deriving a probability measure of a detection, the sensor specific measurement uncertainty is taken as reference. The maximum azimuth angle $\hat{\sigma}_{\phi,max}$ deviation is specified as 1.5° .

Assuming normal distributed measurement errors, a double standard deviation yields $\pm 2\hat{\sigma}_{\phi,max} = \pm 3.0^\circ$ while still covering 95% of all detections. Based on these assumptions, for the maximum detection range of $r_{max} = 40\text{ m}$ for the SLAM, a maximum lateral uncertainty $\hat{d}_y = 0.28\text{ m}$ is calculated:

$$\hat{d}_y = 2 \cdot \tan(\hat{\sigma}_{\phi,max}) r_{max}. \quad (5.7)$$

Analogously, the uncertainty in radial distance $\hat{\sigma}_r = 0.03 \text{ m}$ at $r_{\max} = 40 \text{ m}$ yields a value of $\hat{d}_r = 0.06 \text{ m}$. As a result, a plausible and spatially stable radar detection needs to comply with the following Equation:

$$\hat{d}_x > \max(\hat{d}_r, \hat{d}_y) = \max(0.06 \text{ m}, 0.28 \text{ m}). \quad (5.8)$$

This evaluation defines the grid cells to be discretized to $d_x = 0.3 \text{ m}$.

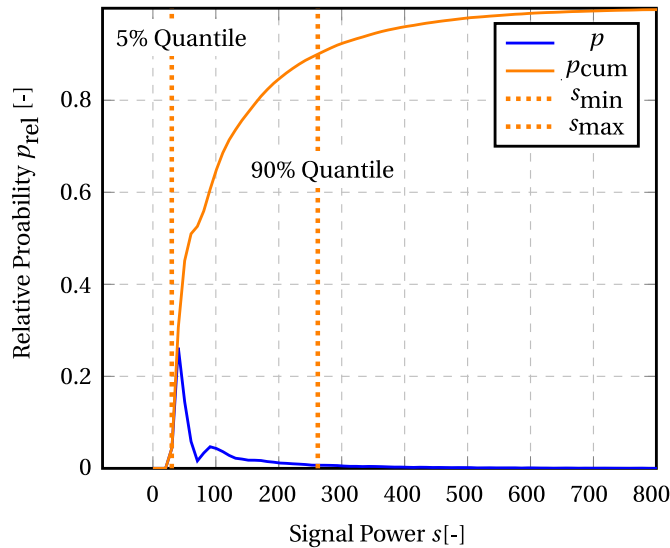


Figure 5.5: Probability of Signal Power s intensity (blue) and accumulated curve (orange) of all detections of a typical recorded scene. s_{\min} at $p_{\text{cum}} = 5\%$, s_{\max} at $p_{\text{cum}} = 90\%$ cumulative frequency. Figure extended from [MT5].

In similar fashion, the signal power normalization values are derived. Figure 5.5 illustrates the signal power range over all received radar point clouds in an exemplary driving scene. From the relative occurrence of the signal power values, the lower 5%-quantile defines $s_{\min} = 20$, whereas the 90%-quantile defines the maximum signal power at $s_{\max} = 270$. Based on the plausibility detection method of Gamba [70], the higher saturation of the upper-end formalizes a higher detection plausibility of densely populated and strong reflective cells.

2. Activation Model: This module receives the detection plausibility and removes temporally instable detections. Based on the assumption that stable detections occur potentially in the same region as in the previous scans, so called *active areas* are tracked. The activation probability $p_a(G_{i,k}, t)$ formalizes the a-priori knowledge from previous scans.

The grid discretization of the detection model is re-used and extended to an activation dimension. In each cell, the temporal course of the resulting detection probability $p_d(G_{i,k}, t)$ is tracked over subsequent radar scans. Hence, temporally consistent high detection probabilities increase the activation score of a cell, whereas absent radar detections in a cell, detection

probability of $p_d(G_{i,k}, t) = 0.5$, continuously decreases the activation score per cell. For each missing detection, and decreasing activation score of a cell, the expectation to receive plausible detections in this cell decreases. Equation 5.9 formalizes the behaviour.

$$p_a(G_{i,k}, t) = \min \left(K_t^d \cdot K_{G_i}^{d,s} \cdot p_a(G_{i,k}, t-1) + \log \frac{p_d(G_{i,k}, t)}{1 - p_d(G_{i,k}, t)}, 1 \right). \quad (5.9)$$

As Logit of the detection probability, the second term contains the filter input [16]. For each time step t with a high detection probability $p_d(G_i, t)$, the activation probability $p_a(G_i, t)$ increases.

The first term describes the dynamic decay of the activation probability. This decay formalizes two independent mechanisms which arise with the temporal decay and the semantic segmentation of the radar detections.

The temporal decay factor $K^d \in (0, 1)$ is consistently applied for all cells with $p_a(G_i, t) > 0$ per time step. Whereas semantic discontinuities between the the current radar scan and previous activation grid are tracked. Inconsistencies per cell are punished with the additional dynamic decay factor $K^{d,s} \in (0, 1]$. If in a cell G_i all radar detections are consistently labeled as class C , the semantic decay remains inactive $K^{d,s} = 1.0$. In the case of multiple different semantic detections are counted per cell, the decay factor $K^{d,s} < 1.0$ is set for this cell. Equation 5.10 formalizes the penalty factor.

$$K^{d,s}(G_i, t) = \begin{cases} K^{d,s} & \text{for } \sum_{k=1}^{C_N} |\{p_a(G_i, t, k) > 0\}| > 1 \\ 1 & \text{else.} \end{cases} \quad (5.10)$$

As a result, for all not perceived classes in a cell G_i the decay factor is increased. If radar detections with different semantic label occur at the same cell overt time, the formulation of decay yields a decreasing activation probability. The additional semantic radar information allows to add a semantic check to the spatial consistency, both considered with time dependency.

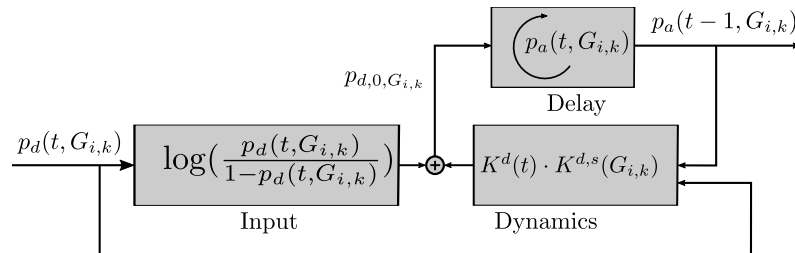


Figure 5.6: Temporal activation model depicted as modular blocks: Input-, Delay- and Dynamics-Component as illustrated in [MT5].

This formulation, visualized in Figure 5.6, causes a delay. Hence, the output of the activation model delivers an activation probability $p_a(G_i, t - 1)$ for the time step $t - 1$. This activation map serves as a-priori input for the subsequent perception module.

The delay results from the ego-vehicle motion and a necessary grid cell shift according to the vehicle motion. With the selection of a vehicle-fixed, but global orientation constant Cartesian coordinate system, an unambiguous association for each cells is possible. According to the vehicle motion model in 2D coordinates and equal tile sizes, the overlap and association of shifted cells results in a pure translation. Equation 5.11 describes the resulting translatory motion t_a , discretized by the grid cell size dx , based on the continuous vehicle translation components t_V .

$$t_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \frac{t_V(t) - t_V(t-1)}{dx} \quad (5.11)$$

A discretization in polar coordinates would result in unambiguous association due to e.g. different cell sizes. Similar, if the grid orientation would move also with the vehicle motion, the rotated cells would result in a similar unambiguity of overlaps and cell association.

3. Perception Model: As final stage of the spatio-temporal filter, the perception model processes the detection probability $p_d(G_{i,k}, t)$ with the activation probability $p_a(G_{i,k}, t)$ to a perception probability $p_p(G_{i,k}, t)$

$$p_p(G_{i,k}, t) = p_a(G_{i,k}, t-1) \cdot p_d(G_{i,k}, t). \quad (5.12)$$

This perception probability is projected back to each individual radar point j in the point cloud $\mathcal{P}_{\text{radar}}(t)$, yielding a binary filter score $\hat{y}_{\text{filter},j}(t)$. As constant threshold value $p_{p,th} = 0.5$, the perception filter per point j , in a cell G_i , for the semantic class k , can be formalized:

$$\hat{y}_{\text{filter},j}(t) = \begin{cases} 1 & \text{for } p_p(G_{i,k}, t) \geq p_{p,th}, \\ 0 & \text{else.} \end{cases} \quad (5.13)$$

In case of a high detection probability, the case $p_p(G_{i,k}, t) = 1$ results in a filter bypass. This bypass allows highly reflective areas, cells with high cumulated signal power, to pass directly. These grid tile are probable to contain according to the *detection model* a physical object. The qualification by a saturated value s_s , allows to bypass the temporal delay characteristic of the spatio-temporal filter.

The resulting temporal equalization yields a filter interpretation as a discrete *PT-element* as in control theory. With $p_a(G_{i,k})$ being limited, and both factors $K_t^d, K^{d,s}$ in Equation 5.9 to decay asymptotically, the filter is temporally stable [46].

5.3.4 Semantic NDT Radar Scan Matching

As basis for graph based SLAM problems, an estimation of the relative motion between two subsequent poses is necessary. Independent from which sensor the relative transformation and rotation is derived, a robust and accurate SLAM input is desired. In the world of point cloud based motion estimation or pose estimation, the relative transformation between two corresponding rigid scans is registered [137, 138, 139]. This registration of point clouds is mainly based on a distance measure, describing the necessary spatial translation or rotation of the source point cloud to match the target point cloud. Rigid point cloud registration is mainly applied for dense LiDAR point clouds or other sensor data, describing environmental objects with a very high accuracy and low noise [137, 138, 139]. In contrast, the described radar sensor specifics (e.g. noise, clutter, ...) complicates a robust and precise radar point cloud registration. Hence, the registration of the defective radar point clouds yields sub-optimal results and local minima, instead of reaching the global registration minimum. As a result, the potential registration results yield a non-conforming vehicle motion.

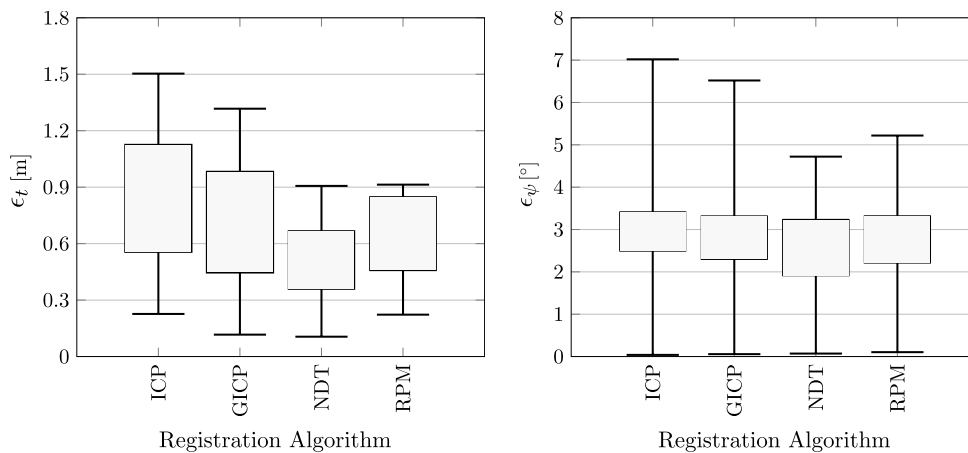


Figure 5.7: Comparison of translational and rotational error of commonly applied point cloud registration procedures (ICP [133], GICP [196], NDT [137], and RPM [235]) for an exemplary radar point cloud registration.

Figure 5.7 illustrates a comparison of spatial error measures for commonly applied point cloud registration methods (ICP [133], GICP [196], NDT [137], RPM [235]). For the test setup, the initial guess of the relative translation sampled with stochastic deviation of significant 3σ around the true relative translation. The NDT registration outperforms other methods on real-world noisy and sparse radar point clouds [137, 138, 139]. Based on these findings, the NDT registration is selected to be applied in this work.

To improve the data association of the registration step, Suma++ of [39] proposes the inclusion of the point cloud's semantic information content in the registration of associated point pairs. Specifically developed for dense LiDAR point clouds, the applied point cloud registration procedure is not applicable to sparse radar point clouds [139]. Nevertheless, the general idea can be transferred and gets in this work applied on radar point clouds. Based on a semantic segmentation of a point cloud, each point contains a semantic label property which is used to find corresponding points of the same label. The semantic label association is independent from the initial relative transformation guess or optimization step.

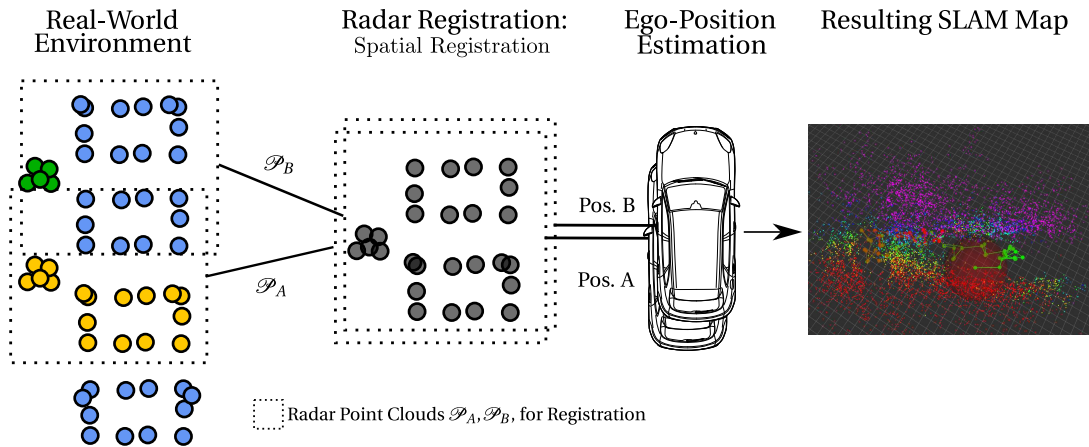


Figure 5.8: Illustration of the classical spatial radar point cloud registration and exemplary results.

Figure 5.9 displays how semantic labels help to find corresponding structures in a reference point cloud, compared to the pure spatial registration concept of Figure 5.8. With this example, also an example is given for a misleading registration based only on spatial information. Especially repetitive structures might be sensed as a resembling, confusable and not assignable subset in point clouds. Registering based on a distance measure, resembling shapes might be mistakenly associated.

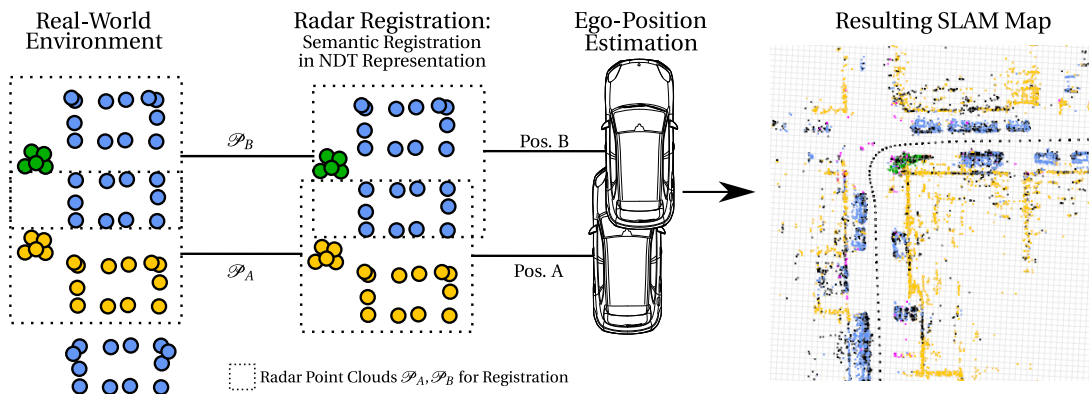


Figure 5.9: Illustration of the effectiveness of the semantic radar registration, yielding a semantically compliant environment map, compared to Figure 5.8. Colors according to the semantic radar classes of Table 3.2.

The NDT registration is discussed with two potential semantic SLAM extensions to respect the semantic radar labels.

5.3.4.1 Weighted Semantic Classes

The NDT registration procedure aims to minimize a target function, consisting of associated normal distributions of the source and target point cloud, see Equation 2.51. Based on the assumption of a rough initial guess and a euclidean distance limitation for associations, practical implementations are defined [137, 92]. Hence, most implementations find the links \mathbf{L}_i of the source scan $G_{i,k}$ with the target scan in the direct neighborhood normal distributions of $\hat{\mathbf{G}}$ of the target scan, resulting in a k-NN normal distribution search:

$$\mathbf{L}_i = \text{k-NN}(G_{i,k}, \hat{\mathbf{G}}) \quad (5.14)$$

With semantic information as additional dimension for the data association, the NDT target function F_{NDT} of Equation 2.51 can be modified. Including only associations with matching semantic label C_i , Equation 5.15 results

$$F_{sem}(\mathbf{R}, \mathbf{t}) = \sum_{G_i \in G} \sum_{\hat{G}_k \in C_n} \tilde{\mathcal{N}}(\mu_{i,k}, \Sigma_{i,k}). \quad (5.15)$$

Equations 5.14-5.15 suggest to utilize the semantic labels to select relevant points in the neighborhood. With a distance based selection, the metric is required to incorporate spatial and label information. With k-NN methods, often attributes are combined by a weighted method. But to define a label-specific weighting scheme is not applicable in general.

In contrast, applying a weighing scheme based on the semantic labels for spatially associated points or point sets, e.g. semantic consistency, can be leveraged. With this approach, data associations are checked for semantic consistency and therewith the pure spatial concept of k-NN based association selection gets improved. Not only checking closest neighbors but nearby detections of the same class enables a semantic association reasoning. As a result, in Equation 5.16 the additional label-weighting $w_{i,k}$ is introduced as $w_{i,k} = 1$ for the case of an equal label:

$$F_{sem,w}(\mathbf{R}, \mathbf{t}) = \sum_{G_i \in G} \sum_{\hat{G}_k \in C_i} w_{i,k} \tilde{\mathcal{N}}(\mu_{i,k}, \Sigma_{i,k}). \quad (5.16)$$

Per cell G_i the ideal semantic label C_{G_i} corresponds to the inlier point cloud sub set \mathcal{P}_{G_i} and is given by the function $\text{SE}_G(\cdot)$,

$$C_{G_i} = SE_G(\mathcal{P}_{G_i}). \quad (5.17)$$

With an approximating classification module, the absolute consistency of points in a cell can not be granted. In addition, labeling of *clutter* as semantic *unknown* = C_u can occur or remain in the point clouds to match even after the spatio-temporal filtering. In general, a cell G_i might contain points p_{radar,G_i} of multiple classes $c \in N_c$, so a majority class is counted, excluding the *unknown* labels C_u .

$$SE_G(\mathcal{P}_{G_i}) = \operatorname{argmax}_{c \in N_c \setminus C_u} |\{p_{radar,G_i,c}\}| \quad (5.18)$$

It is assumed that all close by labels, excluding *unknown* labels C_u , sufficiently describe the semantic label of the environment. Nevertheless, the spatial information of points with *unknown* label C_u are considered for the discretization of the point clouds' normal distribution.

As a result, for two associated cells, the weighting factor $w_{i,j}$ depends now on their relevant majority of semantic labels:

$$w_{i,j}(C_{G_i}, C_{G_j}) = \begin{cases} C_{G_i} & \text{for } C_{G_i} = C_{G_j} \\ 1 - C_{G_i} & \text{for } C_{G_i} \neq C_{G_j} \end{cases} \quad (5.19)$$

Outlook: The weighing factors $w_{i,j}$ are useful for a loop-closure in an area which has undergone changes in the meantime, e.g. a vehicle was moved. Dynamic objects in question can change their location in a scene over time. The weighing factors as defined above indicate which parts of the scene are still both spatially and semantically matching. Low weights indicate a change in the environment and the semantic label of a cell, hence these regions are not relevant for the registration of the changed scene. Further, such cells with a low weighing in the source scan can be deleted in the target scan to compare only the consistent scene structure. If the weight $w_{i,j}$ falls below a threshold of w_{update} , the points of the source point cloud are being deleted:

$$\tilde{\mathbf{P}}_{G_i} = \emptyset. \quad (5.20)$$

The data-association is formulated as Gauss-Newton problem, further references are found in the fundamentals textbook of Jr. and Schnabel [106]. Since associations are made under uncertainty, no analytical solution can be formulated, but the following iterative formulation for the general problem set. Vector p describes the incremental transformation in rotation angles around each axis (α, γ, ψ) and translation components in each coordinate (t_x, t_y, t_z)

$$p = \left[\alpha \quad \gamma \quad \psi \quad t_x \quad t_y \quad t_z \right]^T. \quad (5.21)$$

With the Hessian matrix \mathbf{H} , and gradient g defined for the target function F_{NDT} of Equation 5.15,

$$g_m = \frac{\partial F_{NDT}}{\partial p_m} \quad (5.22)$$

$$\mathbf{H}_{m,n} = \frac{\partial^2 F_{NDT}}{\partial p_m \partial p_n}, \quad (5.23)$$

the linear equation system to solve p can be formulated as

$$\mathbf{H}_{m,n} \Delta p = g_m. \quad (5.24)$$

Incrementally solving this equation yields a solution p^*

$$p^* \leftarrow p \leftarrow p + \Delta p. \quad (5.25)$$

The mathematical deduction of the Hessian matrix and gradient can be found in Peter and Wolfgang [162].

With the introduction of the weighing factor $w_{i,j}$ an additional variable is introduced, yielding into a necessary adaption of the Hessian matrix and gradient formulation.

The summation in both the semantic target function F_{sem} and the weighted target function $F_{sem,w}$ can be split into a sub terms $F_{i,k}$ and $F_{w,i,k}$ respectively.

$$F_{sem,w}(\mathbf{R}, \mathbf{t}) = \sum_{G_i \in G} \sum_{\tilde{G}_k \in C_n} \underbrace{w_{i,k} \tilde{\mathcal{N}}(\mu_{i,k}, \Sigma_{i,k})}_{F_{w,i,k}} \quad (5.26)$$

$$F_{sem,w}(\mathbf{R}, \mathbf{t}) = \sum_{G_i \in G} \sum_{\tilde{G}_k \in C_n} \underbrace{\tilde{\mathcal{N}}(\mu_{i,k}, \Sigma_{i,k})}_{F_{i,k}} \quad (5.27)$$

The derivative of the semantic target function $F_{i,k}$ is given as

$$g_{sem,m} = \frac{\partial F_{sem}}{\partial p_m} = \sum_{G_i \in G} \sum_{\tilde{G}_k \in C_n} \frac{\partial F_{i,k}}{\partial p_m}, \quad (5.28)$$

$$H_{sem,m,n} = \frac{\partial F_{sem}}{\partial p_m \partial p_n} = \sum_{G_i \in G} \sum_{\tilde{G}_k \in C_n} \frac{\partial F_{i,k}}{\partial p_m \partial p_n}. \quad (5.29)$$

As the semantic based label weight $w_{i,k}$ does not depend on the parameter vector p , the derivative of the weighted semantic target function $F_{w,i,k}$ is computed analogously to the Equations 5.28-5.29. Only the weighing factor $w_{i,j}$ needs to be applied on both equation arguments:

$$\frac{\partial F_{sem,w}}{\partial p_m} = w_{i,k} \frac{\partial F_{sem}}{\partial p_m}, \quad (5.30)$$

$$\frac{\partial F_{sem,w}}{\partial p_m \partial p_n} = w_{i,k} \frac{\partial F_{sem}}{\partial p_m \partial p_n}. \quad (5.31)$$

This formulation is applied and compared to an alternative version to incorporate the semantic label in the registration process, explained in the following section.

5.3.4.2 Semantic Separation

Besides the section above, explaining how to emphasize semantically corresponding data associations by a weighting factor, this section introduces an alternative. Instead of matching the spatial distribution of all label points at once, this section elaborates a matching of semantic sub sets per individual semantic label. In order to consider each semantic class as separate point set, the registration can be treated in parallel, for each semantic class independently.

Source point cloud $\tilde{\mathcal{P}}$ and target point cloud \mathcal{P} are split in semantic label point sets according to the point labels. Function $SE(\cdot)$ returns the semantic class per point

$$\tilde{\mathcal{P}}_{\tilde{G}_i, C_k} = \{\tilde{\mathcal{P}}_{G_i} | SE(\tilde{\mathcal{P}}_{G_i}) = C_k\}, \quad (5.32)$$

$$\mathcal{P}_{G_i, C_k} = \{\mathcal{P}_{G_i} | SE(\mathcal{P}_{G_i}) = C_k\}. \quad (5.33)$$

For each semantic grid, the sub-set of points are now also discretized separately as NDT. The resulting parallel channels of each semantic label to be registered yields an adaption of the target function in Equation 5.15 to the separated semantic target function F_s . The summation needs to be adapted to sum all correspondences per class:

$$F_s(\mathbf{R}, \mathbf{t}) = \sum_{k=1}^{|N_c|} \sum_{i \in G_k} \sum_{l \in L_{i,k}} \tilde{N}(\mu_{i,k}, \Sigma_{i,k}) \quad (5.34)$$

Index i, k defines the registration of the i -th normal distribution of the source scan with the k -th semantic normal distribution of the target scan.

With the semantic separation, only neighborhoods of the same semantic class k is searched for corresponding links $\mathbf{L}_{i,k}$

$$\mathbf{L}_{i,k} = \text{k-NN}(G_{i,k}, \tilde{\mathbf{G}}_k). \quad (5.35)$$

With this formulation, the points labeled as *unknown* are not ignored or treated separately, but are treated as any other semantic class.

Unfortunately, based on the radar sensor specification, the uncertainty of points is a function of perception distance. In the test vehicle setup, not only far range detections but also very close detections in sensor proximity yields increased range-uncertainty. Similar to the perception with other sensors, the same scene but from a different perspective points e.g. during a narrow passage and how it looks from a distance, can yield different semantic labels. With different semantic labels the association would not be possible in the semantic separation case.

For the implementation of the Hessian \mathbf{H}_s and gradient g_s , the same Equations 5.29 for \mathbf{H} and Equations 5.28 for g hold, with factor $w_{i,j} = 1$ for the derivatives in Equations 5.31-5.30.

5.3.5 Graph Front-End: Graph Construction

The section before presents the developed methodology to register semantic radar point clouds. In order to relate single relative registrations to a whole scene or environment description, the SLAM problem needs to be defined for the semantic radar point clouds. This section focuses on a graph-SLAM front end, specifically designed to comply with radar specific characteristics. The described front-end builds a SLAM graph, enables the utilization of different odometry sources and describes the loop detection. Together with the SLAM back-end, applying nonlinear optimization with g2o [119], an optimization of the graph-formulation is achieved.

The presented SLAM is designed to be applicable in general urban or sub-urban environments and is free of specific features or model based landmarks. As only input, the radar

characteristic point clouds of two vehicle poses are registered, these registration and the point clouds are saved as graph-nodes. Additionally, the wheel based ego-vehicle motion delivers an a-priori information of the estimated relative motion.

This basic and modular SLAM formulation can easily be extended for further sensor inputs, e.g. visual odometry estimates from camera images, since the only input requirement is to deliver a relative pose estimation and the corresponding covariances.

Figure 5.10 illustrates the graph-SLAM specific front-end modular blocks. First, the pre-filtered radar point cloud is synchronized with the wheel-based odometry pose, to check in the motion gate if the vehicle is moving ($v_{veh} > 0$). Only in motion, the radar scans are synchronized to a vehicle odometry position and constitute a new input point cloud \mathcal{P}_t to the graph-SLAM front-end. During stand-still the received radar point clouds are not accumulated in the SLAM map.

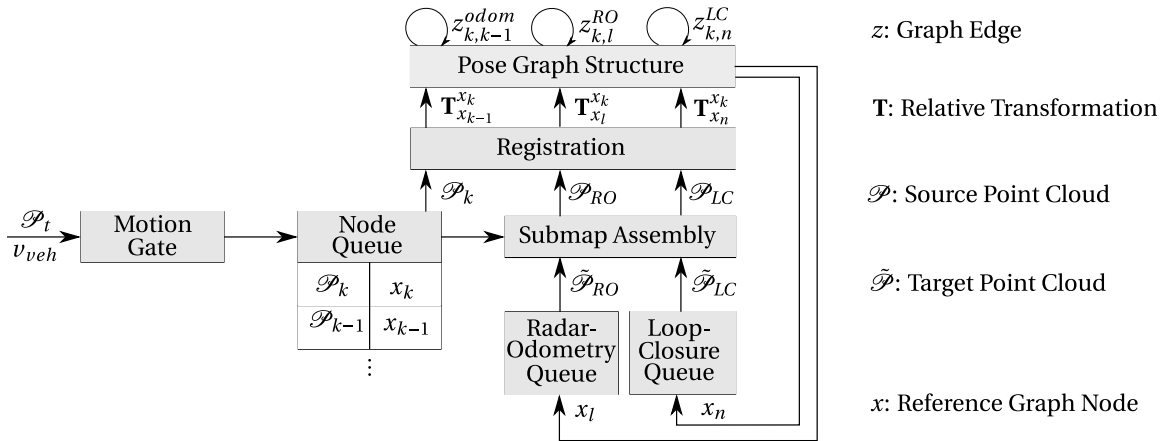


Figure 5.10: Illustration of the pose graph-SLAM front end with different node queues and edges.

If a radar-point cloud is synchronized with a vehicle odometry position, this tuple results in a graph node x_k . The resulting relative transformation $\mathbf{T}_{x_{k-1}}^{x_k}$ between the latest x_{k-1} and the new node x_k , together with the corresponding covariance matrix, describes the graph edge between the two nodes.

Working with queues to express the graph structure dynamically, each new incoming pre-filtered radar point cloud gets saved as new graph node tuple (\mathcal{P}_k, x_k) in the node queue. This total queue describes describes the whole mapped scene as global map. Besides, a subset of nodes is saved in special queues, describing loop-closure sub-map assemblies \mathcal{P}_{LC} and a queue for odometry estimation based on radar maps \mathcal{P}_{RO} . Both of these special queues, contain sub-maps of different sizes which are linked also with relative transformations, formalizes as graph edges z , for both the loop-closure queue and for the radar odometry queue. The sub-maps are registered with the same procedure as single radar scans, but with the

difference of larger point cloud maps. Figure 5.10 displays an exemplary loop closure between node x_n and node x_l , yielding the relative transformation $\mathbf{T}_{x_n}^{x_l}$. The radar odometry registration is analogously displayed by x_k, x_{k-5} and $\mathbf{T}_{x_{k-5}}^{x_k}$

The entity of all edges and nodes describes the constructed graph SLAM formulation. How the edges are defined in this work, specifically between which nodes and sub maps, is a major structural adaption compared to other SLAM structures. Figure 5.11 illustrates the different edge types and their span.

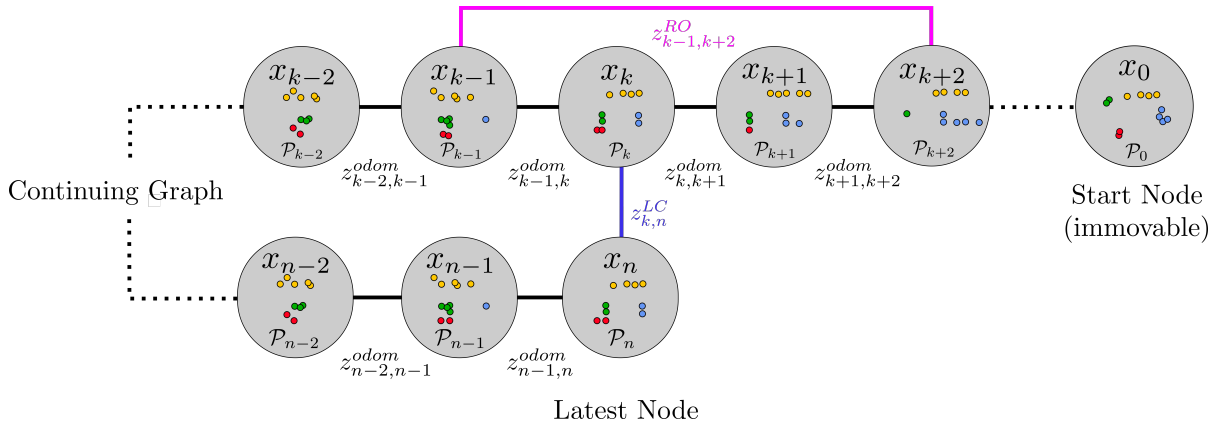


Figure 5.11: Pose graph-SLAM structure to illustrate different edge types: Adjacent wheel-based odometry edges (black), radar odometry, *skip edges* (pink), and far-reaching loop-closure edges (blue). Colors according to the semantic radar classes of Table 3.2. Figure modified from [S14].

Wheel-Odometry (z^{odom}): This subsequent pose chain is considered the SLAM backbone. Sequentially registered nodes are chained with the node-to-node odometry edges to a pose-chain. As the synchronized point clouds and odometry estimates are available to a high frequency (ca. 18-20 Hz), the nodes are defined to have a maximum distance of 0.4 m. The wheel-based odometry estimation yields adequately precise motion estimates in a local context to relate and register the synchronized radar point clouds precisely. The registration for these point clouds yields low translation and rotation components, due to the high sensor frequency and consequently low relative motion between subsequent nodes.

Radar-Odometry (z^{RO}): Alternatively addressed as *skip-edges* or *open-loop edges*, these edges connect non-adjacent nodes with a minimum node distance between the connected nodes. Based on the relevant radar sensing range of circa 50 m, even radar point clouds of non adjacent nodes overlap and can be registered. By the required node distance, these edges span and constrain a larger region to comply with the metric radar point clouds. For sufficient distances, the potential drift from the odometry-estimation is hereby corrected. Since a wheel-based odometry estimation technically accumulates incremental drift errors per integration step, locally correct estimates might contain estimation errors over larger distances. The correction of this incremental drift is realized by the constraint of z^{RO} , which registers not single radar scans, but sub-maps of multiple combined nodes. The more accurate the

registration of the sub-maps, performs the lower a potential drift might be mapped in the graph.

Loop-Closure (z^{LC}): Typically SLAM formulations recognize places which have already been mapped and compare the mapped scene with the current perception of the same scene. If a consistency is found, additional loop-closure edges further reduce the localization error, reducing drift and improving overall SLAM map accuracy to the metric scale of the environment. Therefore, fixed to the latest node in the graph SLAM, its region is searched in a pre-defined region for graph nodes in the loop search radius R_{LC} . If nodes are located inside of R_{LC} , a local sub-map around both node locations are registered to confirm a potential place to be revisited. Each confirmed place revisit yields an additional loop-closure edge between the graph nodes which are registered and match.

Mathematically, the resulting graph SLAM problem minimizes a quality function χ which is formulated by the edges established in the graph construction step. With the before introduced edge types, the modified target function χ^* yields

$$\chi^* = \operatorname{argmin}_X \sum_{k,j} \|z_{i,j}^{odom}\|_{\Omega_{i,j}}^2 + \sum_{k,j} \|z_{m,n}^{RO}\|_{\Omega_{m,n}}^2 + \sum_{k,j} \|z_{r,s}^{LC}\|_{\Omega_{r,s}}^2. \quad (5.36)$$

The different edge types are further described by a covariance, to estimate each edges' stability, similar to a weight or stiffness of the constraint. The following sections discuss for each edge its covariance estimation and translation and rotation definition.

5.3.5.1 Odometry Edges

Major sensitivity of every SLAM problem is the incremental motion or odometry estimation [221, 64]. In this thesis, a standard model for wheel-based motion estimation is applied. Instead of fine-tuning this model to yield SLAM accuracy improvement, the secondary sensitivity of the SLAM problem, to especially tune the SLAM for noisy radar data, is addressed.

For the applied wheel-based bicycle odometry model, see Section 3.3.4 or Werling [223]. Resulting from the motion between two subsequent nodes x_k, x_{k+1} , the relative translation and rotation components are formulated as homogeneous transformations $\mathbf{T}_{V,k}$ and $\mathbf{T}_{V,k+1}$. Hence, the relative transformation between the two nodes, the information which a graph edge contains, is given as

$$\mathbf{T}_{x_k}^{x_{k+1}} = \mathbf{T}_{V,k+1}^{-1} \mathbf{T}_{V,k}. \quad (5.37)$$

Together with this transformation, the SLAM benefits from an uncertainty measure for this constraint. The vehicle which is utilized in this thesis, delivers empirically defined standard deviation measures σ_ν and $\sigma_{\dot{\psi}}$ for the motion model parameters speed ν and yaw rate $\dot{\psi}$. Compliant to the assumptions of the general graph-SLAM formulation [82],

assuming uncorrelated error components, the covariance matrix of the system input equation u can be defined

$$\Sigma_{u,k} = \begin{bmatrix} \sigma_v & 0 \\ 0 & \sigma_{\dot{\psi}} \end{bmatrix}. \quad (5.38)$$

With \hat{x} as system state,

$$\hat{x} = \begin{bmatrix} x & y & \psi_{k+1}^T \end{bmatrix}_{k+1}^T \quad (5.39)$$

the Jacobi matrix of the odometry model $\mathbf{J}_{\hat{x}}, \mathbf{J}_u$ for the state $u = 0$ and $\hat{x} = 0$ is calculated. Hence, the input uncertainty¹ can be approximated from the non-linear system equation by a multidimensional Taylor approximation

$$\Sigma_{\hat{x},k+1} = \mathbf{J}_{\hat{x}} \Sigma_{\hat{x},k} \mathbf{J}_{\hat{x}}^T + \mathbf{J}_u \Sigma_{u,k} \mathbf{J}_u^T. \quad (5.40)$$

Substituting the motion model, Equation 5.40 yields

$$\Sigma_{\hat{x},k+1} = \begin{bmatrix} \Delta t^2 \cdot \sigma_v & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Delta t^2 \cdot \sigma_{\dot{\psi}} \end{bmatrix}. \quad (5.41)$$

The uncertainty estimation is expressed as relative uncertainty for each motion step independently, so only a single step of the motion model from one node x_k to the next node x_{k+1} is considered. Consequently, the matrix $\Sigma_{\hat{x},k}$ is set to $\mathbf{0}$.

As a result of the linearization of the non-linear system in Equation 5.40, the uncertainty of $\sigma_{\dot{\psi}}$ is not anymore capable to directly influence a lateral deviation in the y -coordinate. Hence, measurement certainty for the y -component is suggested. This simplification needs to be reconsidered if the formulation is applied for other use-cases than slow parking maneuvers.

For the current use-case of slow parking $v \leq 15 \frac{km}{h}$, the wheel-odometry signals are available at a rate of $16Hz$. In this case, an uncertainty estimation scheme is derived from a two step consideration. According to the before derived assumptions of a constant yaw angle ψ , after the first propagation step from x_k to x_{k+1} also a second step to x_{k+2} is considered. The input uncertainty is assumed equal over both propagation steps, assuming the yaw angle change neglectable. Hence, for the inoperative state \hat{x}' the linearization yields

$$\hat{x}' = \begin{bmatrix} x_{k+1} & y_{k+1} & 0 \end{bmatrix}^T. \quad (5.42)$$

¹ The input uncertainty was introduced for the 1-d case in Section 2.1.

The linearized systems' attributes are applied for the estimation of a single update step. In order to estimate an uncertainty estimation for only one step from k to $k + 1$, the time step Δt is not substituted for two, but only for one time step $\tilde{\Delta}t = 0.5 \cdot \Delta t$ applies. Hence, the linearized system equation is given as

$$\Sigma_{\hat{x},k+2} = \mathbf{J}'_{\hat{x}} \Sigma'_{\hat{x}',k+1} \mathbf{J}'_{\hat{x}}{}^T + \mathbf{J}_u \Sigma_{u,k} \mathbf{J}_u{}^T. \quad (5.43)$$

Substituting the motion model with the modified step size $\tilde{\Delta}t$, Equation 5.43 yields an explicit formulation of the covariance matrix

$$\Sigma'_{\hat{x},k+1} = \begin{bmatrix} 2\tilde{\Delta}t^2 \cdot \sigma_v & 0 & 0 \\ 0 & \tilde{\Delta}t^4 \cdot v_k^2 & \tilde{\Delta}t^3 \cdot \sigma_{\psi} \cdot v_k \\ 0 & \tilde{\Delta}t^3 \cdot \sigma_{\psi} \cdot v_k & 2\tilde{\Delta}t^2 \cdot \sigma_{\psi} \end{bmatrix}. \quad (5.44)$$

To be applicable in the 3D graph SLAM formulation as edge, the 2D formulation $\Sigma'_{\hat{x},k+1}$ needs to be extended for a 3D formulation $\Sigma''_{\hat{x},k+1}$. Since the applied vehicle motion model is formulated in 2D, the undefined dimensions in z -coordinate, roll angle ϕ and pitch angle θ are substituted by a constant σ_{min} .

$$\Sigma''_{\hat{x},k+1} = \begin{bmatrix} 2\tilde{\Delta}t^2 \cdot \sigma_v & 0 & 0 & 0 & 0 & 0 \\ 0 & \tilde{\Delta}t^4 \cdot v_k^2 & 0 & 0 & 0 & \tilde{\Delta}t^3 \cdot \sigma_{\psi} \cdot v_k \\ 0 & 0 & \sigma_{min} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{min} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{min} & 0 \\ 0 & \tilde{\Delta}t^3 \cdot \sigma_{\psi} \cdot v_k & 0 & 0 & 0 & 2\tilde{\Delta}t^2 \cdot \sigma_{\psi} \end{bmatrix}. \quad (5.45)$$

With $\Sigma'_{\hat{x},k+1}$ defined as 2D formulation (in x, y coordinates), and rotation angle ϕ , the covariance is not defined in an euclidean space. As a result, the incremental optimization if the SLAM problem approximates on a local manifold which approximates an euclidean space, see Section 2.6. To apply the covariance matrix $\Sigma'_{\hat{x},k+1}$ correctly for the SLAM problem on the manifold, the $\Sigma'_{\hat{x},k+1}$ needs to be projected onto the manifold, as described in Grisetti et al. [82]. In this reference work, the transformation \boxplus is defined in detail.

Following the suggested transformation \boxplus of Grisetti et al. [82], an incremental optimization step of the SLAM target function $\tilde{\chi} \in \mathbb{R}^{6 \times 6}$ yields a translation and rotation component in local coordinates. Compare Equation 2.67, with rotation represented as Quaternion.

The 6 degrees of freedom (DOF) are combined in the parameter vector r

$$r = \begin{bmatrix} t_x & t_y & t_z & \alpha & \gamma & \psi \end{bmatrix}^T. \quad (5.46)$$

The rotation components of r can be reformulated as quaternion components, applying the substitution angle $\phi = \sqrt{\alpha^2 + \gamma^2 + \psi^2}$:

$$q(r) = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \alpha \frac{\sin \frac{\phi}{2}}{\phi} \\ \gamma \frac{\sin \frac{\phi}{2}}{\phi} \\ \psi \frac{\sin \frac{\phi}{2}}{\phi} \end{bmatrix} \quad (5.47)$$

With the parameter vector r , the final transformation formulation of the SLAM optimization step $\Delta \tilde{\chi}$ can be formalized as $f_{\Delta \tilde{\chi}}$

$$f_{\Delta \tilde{\chi}}(r) = \begin{bmatrix} t_x \\ t_y \\ t_z \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ q(r) \end{bmatrix} \quad (5.48)$$

The function $f_{\Delta \tilde{\chi}}(r)$ as update function of the incremental SLAM solution can be treated similar to Equation 5.43 to calculate an uncertainty propagation. First, the partial derivative for each component of the parameter vector r is defined

$$\frac{\partial q(r)_i}{\partial r_k} = \begin{cases} r_i^2 \left[\frac{1}{2\phi^2} \cos \frac{\phi}{2} - \frac{1}{\phi^3} \sin \frac{\phi}{2} \right] + \frac{\sin \frac{\phi}{2}}{\phi} & \text{for } i = k \\ r_i r_k \left[\frac{1}{2\phi^2} \cos \frac{\phi}{2} - \frac{1}{\phi^3} \sin \frac{\phi}{2} \right] & \text{else.} \end{cases} \quad (5.49)$$

As final formulation, the resulting covariance matrix $\Delta \Sigma_{\boxplus_{k,k+1}}$ defines the wheel based odometry uncertainty estimation between node k and $k+1$ for the wheel-odometry edge z^{odom} .

$$\Delta \Sigma_{k,i}^{odom} = \Delta \Sigma_{\boxplus_{k,k+1}} = \mathbf{J}_{f_{\Delta \tilde{\chi}}} \Sigma_{\hat{x},k+1}'' \mathbf{J}_{f_{\Delta \tilde{\chi}}}^T \quad (5.50)$$

5.3.5.2 Radar Odometry Skip Edges

To utilize the radar's medium range sensing properties, a significant overlap of point clouds with a relative distance yields well point cloud registration. For this type of edges, the comparison of single nodes is extended to a summary of nodes, constituting a sub-map. A sub-map $\mathcal{P}_{RO,k}$ consists of a set of N_{SM} wheel-odometry edges arranged graph-nodes. Hence, the relative transformation $\mathbf{T}_{x_i}^{x_k}$ based on the wheel odometry, delivers an initial relative position guess. The two sub-maps to be compared are denoted as $\mathcal{P}_{RO,k}$ and $\mathcal{P}_{RO,k+\Delta}$ with Δ as discrete distance measure between nodes and initial translation guess $\mathbf{T}_{x_i}^{x_k}$.

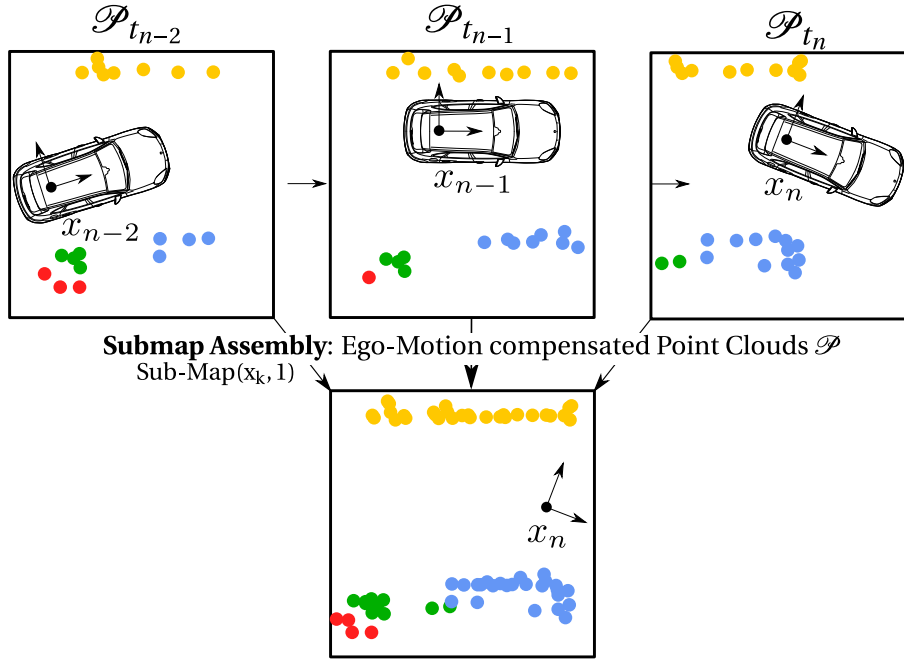


Figure 5.12: Illustration of sub-map assembly of exemplary $k = 1$ adjacent radar scans for node x_k . Colors according to the semantic radar classes of Table 3.2.

Registering the sub-maps is analogously defined as the single point cloud registration and performed with the same registration algorithm, yielding a relative transformation between the sub-maps

$$\mathbf{T}_{RO,x_k}^{x_i} = \text{Registration}(\mathcal{P}_{RO,k}, \mathcal{P}_{RO,k+\Delta}, \mathbf{T}_{x_i}^{x_k}). \quad (5.51)$$

The relative transformation of the sub-maps might vary to the wheel-odometry based transformation estimation. Comparing an assembly of point clouds along the graph arranged nodes, the compared regions gain stability and diminish the potentially registered noise between adjacent point clouds.

Similar to the edge definition of simple odometry edges, an uncertainty measure for this radar odometry edge is derived. As approximation of the covariance matrix, the definition of

the Hessian matrix of the registration function can be applied, according to Bengtsson and Baerveldt [19]

$$\Delta \Sigma_{k,i}^{RO} \approx \frac{F(\mathbf{R}^*, \mathbf{T}^*)}{|\mathbf{G}| - 3} \left[\frac{1}{2} \mathbf{H} \right]_{\mathbf{R}^*, \mathbf{T}^*}^{-1}. \quad (5.52)$$

The node assembly to constitute sub maps and register the sub maps is defined to comply the following requirements:

- **Sensor Range:** Essential to register two corresponding map areas in point cloud representation $\mathcal{P}_{RO,k}, \mathcal{P}_{RO,k+\Delta}$, both point clouds need to overlap sufficiently. Hence, the maximum sensing range R_{max} defines an upper distance threshold for Δ .
- **Δ Distance Selection:** Aiming to correct a potential drift from adjacent node registration based on the wheel-odometry edges, a greater distance Δ between the registered point clouds $\mathcal{P}_{RO,k}, \mathcal{P}_{RO,k+\Delta}$ provides improved registration stability and drift correction potential. This functional idea of the skip-edges defines the lower bound for Δ , requiring the registration precision to exceed the erroneous registration noise of adjacent node registrations.

Illustrated in Figure 5.13, sub-maps are defined around each selected node of the registration pair, constructed from a number of adjacent k_{Map}^{RO} , denoted as function Submap in Algorithm 2. Since the point cloud assembly is defined with the reference node x_i as center node, the adjacent node point clouds are transformed into the reference system. A sub-map is defined as union of the node point cloud sets:

$$\text{Submap}(x_i, k_{Map}^{RO}) = \bigcup_{m=i-\lfloor \frac{1}{2} k_{Map}^{RO} \rfloor}^{i+\lfloor \frac{1}{2} k_{Map}^{RO} \rfloor} \mathbf{T}_{x_i}^{-1} \mathbf{T}_{x_m} \mathcal{P}_{x_m} \quad (5.53)$$

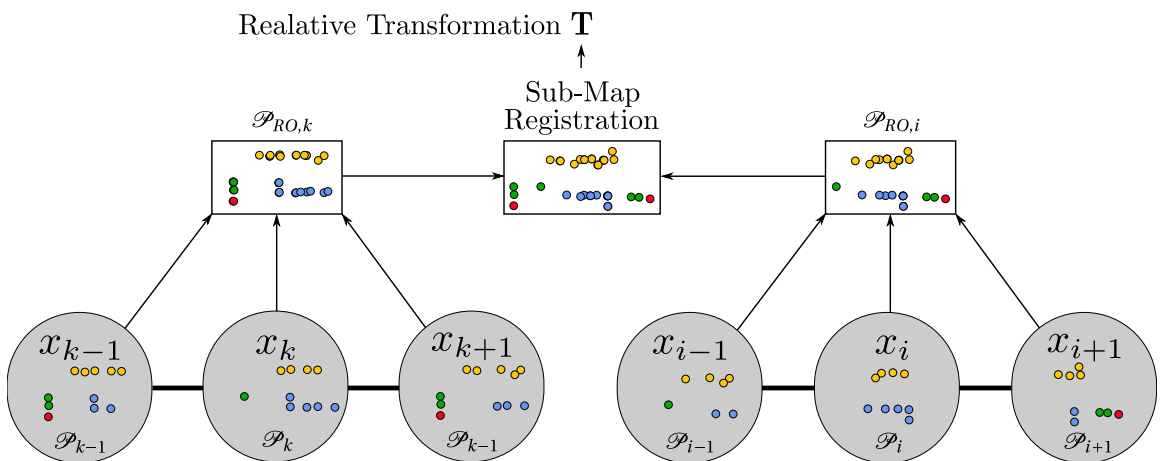


Figure 5.13: Generic illustration of sub map assembly and registration strategy with relative transformation output. Sub-map registration principle applied e.g. for radar odometry edges and for loop-closure sub-map matching. Colors according to the semantic radar classes of Table 3.2.

The selection of the-sub map reference nodes follows Algorithm 2, visualized in Figure 5.13. For node x_k , which must not be a loop-closure node, in a fixed sensing range $r_{\text{SM,max}}$ the farthest node x_i is found. From the constant sensor perception range $R_{\text{max}} > r_{\text{SM,max}}$ a sufficient point cloud overlap is guaranteed. Translating the relative distance into the corresponding node index by f_{index} , the farthest node x_i is found. In iterative steps, the registration is performed for the node pair x_k, x_i , yielding a relative transformation and registration score s_i . The registration score s_i is a distance-based metric to evaluate the registration, comparing the suggested transformed point cloud $\tilde{\mathcal{P}}_{RO,k} = \mathbf{T}_{RO,x_k}^{x_i} \mathcal{P}_{RO,k}$ to the reference $\mathcal{P}_{RO,i}$

$$s_i = \text{Score}(\mathcal{P}_{RO,i}, \tilde{\mathcal{P}}_{RO,k}) = \sum_{j=1}^{|\mathcal{P}_{RO,i}|} \|\mathcal{P}_j - \mathbf{k}\text{-NN}(\mathcal{P}_j, \tilde{\mathcal{P}}_{RO,k})\|. \quad (5.54)$$

The score, or distance measure, s_i represents a metric describing the match of the registered and transformed point clouds. According to Magnusson [137], this metric can be applied for registration quality rating. Hence, if in Algorithm 2 the registration for node x_i was successful, set a radar odometry node, set $k = i$, restart the algorithmic `while` and reduce i until $i_{\text{min}} = n$ at the current vehicle position x_n to find a new odometry edge.

Together with the above derived covariance matrix $\Delta\Sigma_{k,i}^{RO}$, the radar odometry edge $z_{k,i}^{RO}$ is fully defined.

Algorithm 2 Radar odometry node selection for sub map construction.

Input: $k, \text{Graph } X, r_{\text{max}}, s_{\text{min}}$

Output: $\mathbf{T}_{x_k}^{x_i}$

procedure RADAR ODOMETRY NODE SELECTION

$i = k + f_{\text{Node Distance}}(r_{\text{max}})$

$i_{\text{min}} = k + f_{\text{Node Distance}}(s_{\text{min}})$

$\mathcal{P}_{RO,k} = \text{Submap}(x_k)$

while $i \geq i_{\text{min}}$ **do**

$\mathcal{P}_{RO,i} = \text{Submap}(x_i)$

$\mathbf{T}_{RO,x_k}^{x_i} = \text{REG}(\mathcal{P}_{RO,k}, \mathcal{P}_{RO,i}, \mathbf{T}_{x_i}^{x_k})$

$\tilde{\mathcal{P}}_{RO,k} = \mathbf{T}_{RO,x_k}^{x_i} \mathcal{P}_{RO,k}$

$s_i = \text{Score}(\mathcal{P}_{RO,i}, \tilde{\mathcal{P}}_{RO,k})$

if $s_i < s_{\text{min}}$ **then**

return $\mathbf{T}_{RO,x_k}^{x_i}$

else if $i = i - 1$ **then**

return False

end if

end while

end procedure

5.3.5.3 Loop Closure Edges

As third and last edge type, the loop closure edges $z_{k,n}^{LC}$ is defined. This type of edges are reserved to indicate revisited places which are registered also as sub maps, but specifically defining loops of a driven path in the SLAM. This type of edges contains most potential to eradicate accumulated odometry drift and error components [82]. With the advantage to have already mapped a larger region as basis for the regional comparison, the perceptive field for this edge type is increased and radar detection density is reduced. Especially the increased density is important to recognize the already visited and mapped regions. From a larger overlap and larger sub-map regions to be compared, also the registration stability and accuracy benefits and yields superior accuracy compared to single node comparison.

Analogous to the presented procedure to register radar odometry edges, for the loop closure sub-map regions as point cloud assemblies of adjacent nodes is registered. The same registration algorithm is applied with different node distance parameters.

The selection of loop closure candidates originates from the latest SLAM graph node x_n . Neighboring nodes of the SLAM map which are located in a fixed and pre-defined euclidean distance $R_{LC} = 15 m$, are considered as potential loop closure candidates.

$$U = \left\{ x_k \in (X|R_{LC}) > \left\| \mathbf{T}_{x_n}^{-1} \mathbf{T}_{x_k} \right\|_t \right\} \quad (5.55)$$

The formulation $\|\cdot\|_t$ defines the euclidean distance as norm of the translational part of the homogeneous pose \mathbf{T} .

As improvement, alternatively to a fixed distance selection, a dynamically adaptive threshold formulation $R_{LC}(\sigma)$ based on the ego motion uncertainty σ can be applied. The marginalized pose estimation of x_n allows an uncertainty estimation [141]. Drawback of this dynamic formulation is the increasing number of potential candidates U . Without pre-selection, too many registrations remain open to be tested for loop closures during run-time, causing delays and preventing real-time capability. Hence, from the proposed candidates U only a subset in a fixed distance to x_n is tested.

For a potential loop-closure candidate, a sub-map is assembled of $k_{map}^{LC,k}$ nodes

$$\mathcal{P}_{LC,k} = \text{Submap}(x_k, k_{map}^{LC,k}). \quad (5.56)$$

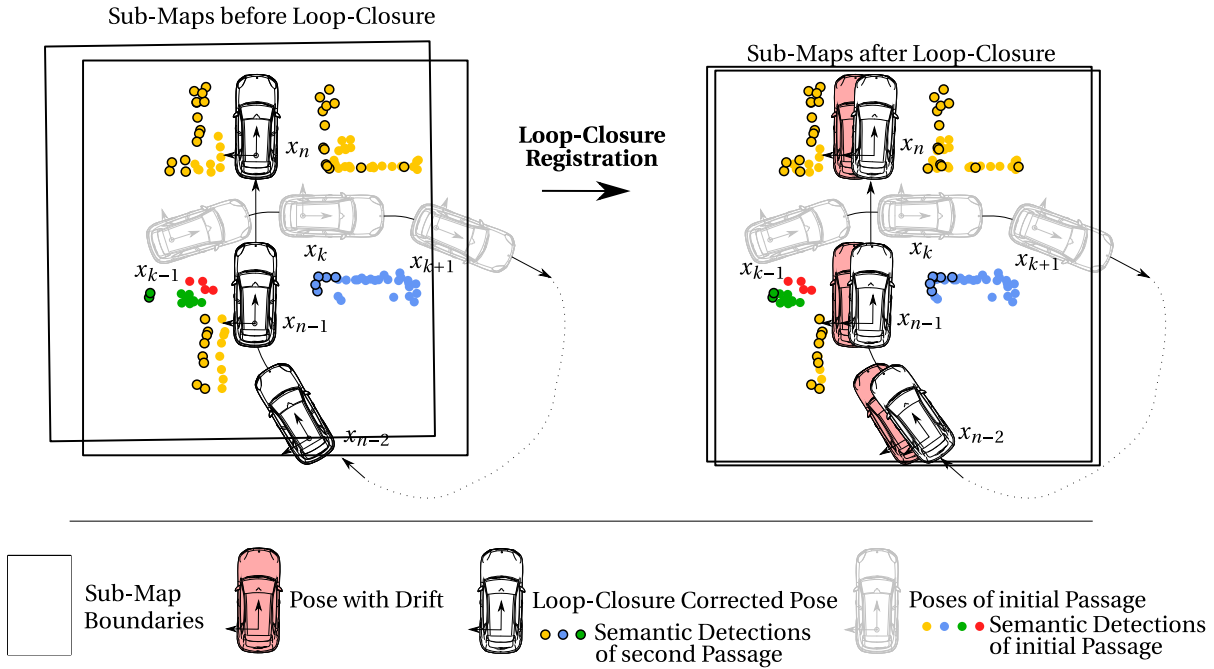


Figure 5.14: Loop-Closure with Sub-Map Assembly for $k_{sub}^{LC,n} = k_{sub}^{LC,k} = 3$. Colors according to the semantic radar classes of Table 3.2.

The corresponding sub-map at the current vehicle position node x_n is defined differently. x_n can not be applied as center node, but only as end node. Neighboring nodes of the latest node x_n are assembled as in Equation 5.57.

$$\mathcal{P}_{LC,n} = \bigcup_{m=n-k_{map}^{LC,n}}^n \mathbf{T}_{x_n}^{-1} \mathbf{T}_{x_m} \mathcal{P}_{x_m}. \quad (5.57)$$

5.4 Experiments

The parts of this section refer each to a method-introducing subsection of Section 5.3. First, the experimental results of the before described functional semantic SLAM components in Sections 5.3.1- 5.3.4 are described on the Sections 5.4.1- 5.4.4. Second, in Section 5.4.5 the semantic radar SLAM functionality is evaluated and tested in different real-world scenarios.

5.4.1 Signal Pre-Processing

The individual sensor raw point clouds are assembled to a unified point cloud and filtered by the pre-filter attributes of Table 5.1. The remaining radar detections lie in the relevant region for the SLAM application.

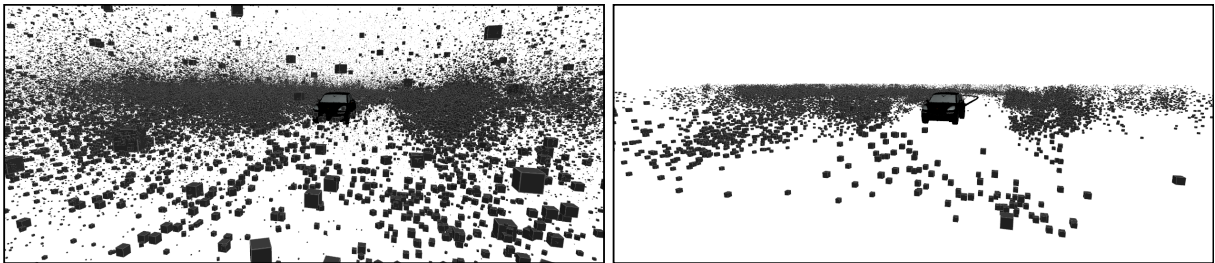


Figure 5.15: A qualitative comparison of the pre-filter effects on the same scene, radar detections accumulated over 25 seconds. Radar detections accumulated without a filter (left) vs. the filtered version (right).

With the reduction of the point cloud to the relevant RoI for the SLAM, mis-associations are reduced. With this step, approximately 34% of the radar point cloud is filtered out.

5.4.2 Semantic Spatio-Temoral Filter

The spatio-temporal filter of Section 5.3.3 follows the pre-filtering to the SLAM-relevant RoI. The effects of the spatio-temporal filter can be measured by its effects on the registration accuracy. Since the filter aims to remove the unstable registration-irrelevant radar detections, the registration accuracy yields a measure to evaluate the filter effectiveness.

Figure 5.16 illustrates the registration error as histogram for three filter variants, the spatial filter only, the spatio-temporal part of the filter, and the full semantic spatio-temporal filter configuration. Based on the translation and rotation error of the registration, the semantic spatio-temporal filter is found to reduce the point cloud to the relevant points the most beneficial. The semantic spatio-temporal filter increases the percentage of especially lower translation errors in the left plot of Figure 5.16, while reducing also the amount of medium translation errors. Considering the rotation errors in the right plot of Figure 5.16, the semantic information improves the registration, achieving a much higher registration below 1° rotation

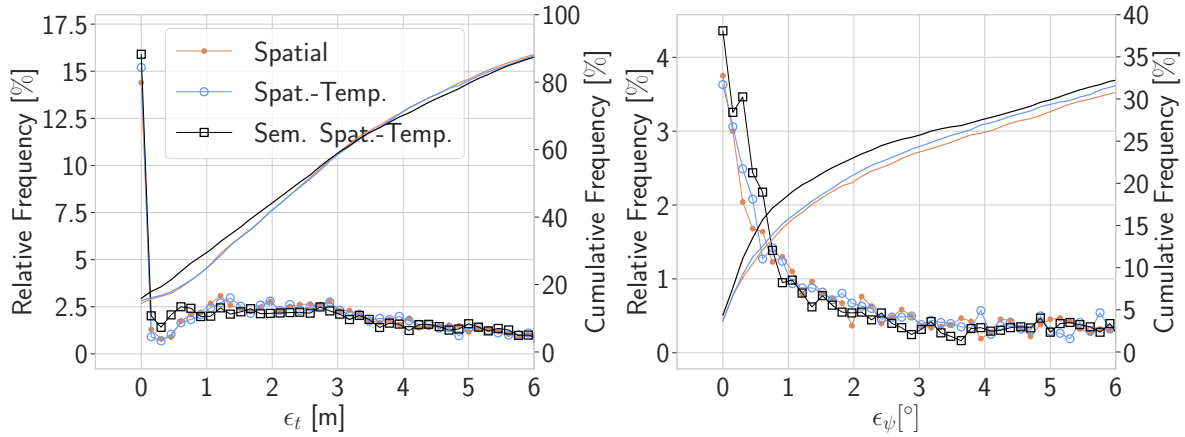


Figure 5.16: Histogram of the relative occurrence of translational ϵ_t (left) and rotatory ϵ_ψ registration error (right) of spatial, spatio-temporal and semantic spatio-temporal filter.

error. The improvement by the suggested semantic spatio-temporal pre-filtering can be found in the higher cumulative frequency curve, especially for the high registration accuracy at low error values. The achieved translation error mainly achieves registration accuracy below 1 m, whereas the rotation error occurs to rise significantly for less than 1° rotation error.

5.4.3 Sub-Map Assembly

As introduced in Section 5.3.5.2, based on a noise averaging sense of assembled single scans to sub-maps, a parametrizable number of radar scans is map-matched in order to robustify the map matching regions.

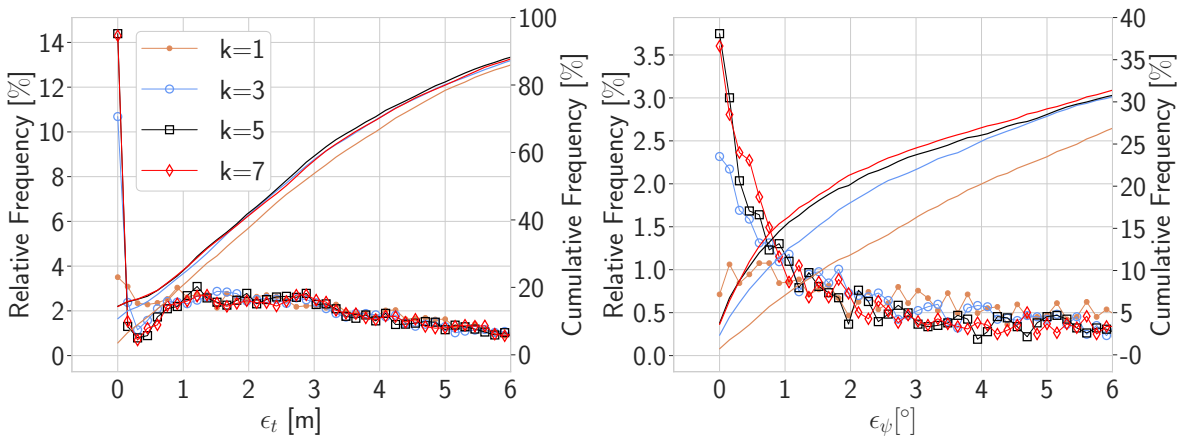


Figure 5.17: Histogram of the relative occurrence of translational ϵ_t (left) and rotational ϵ_ψ registration error (right) for different sub-map configurations k .

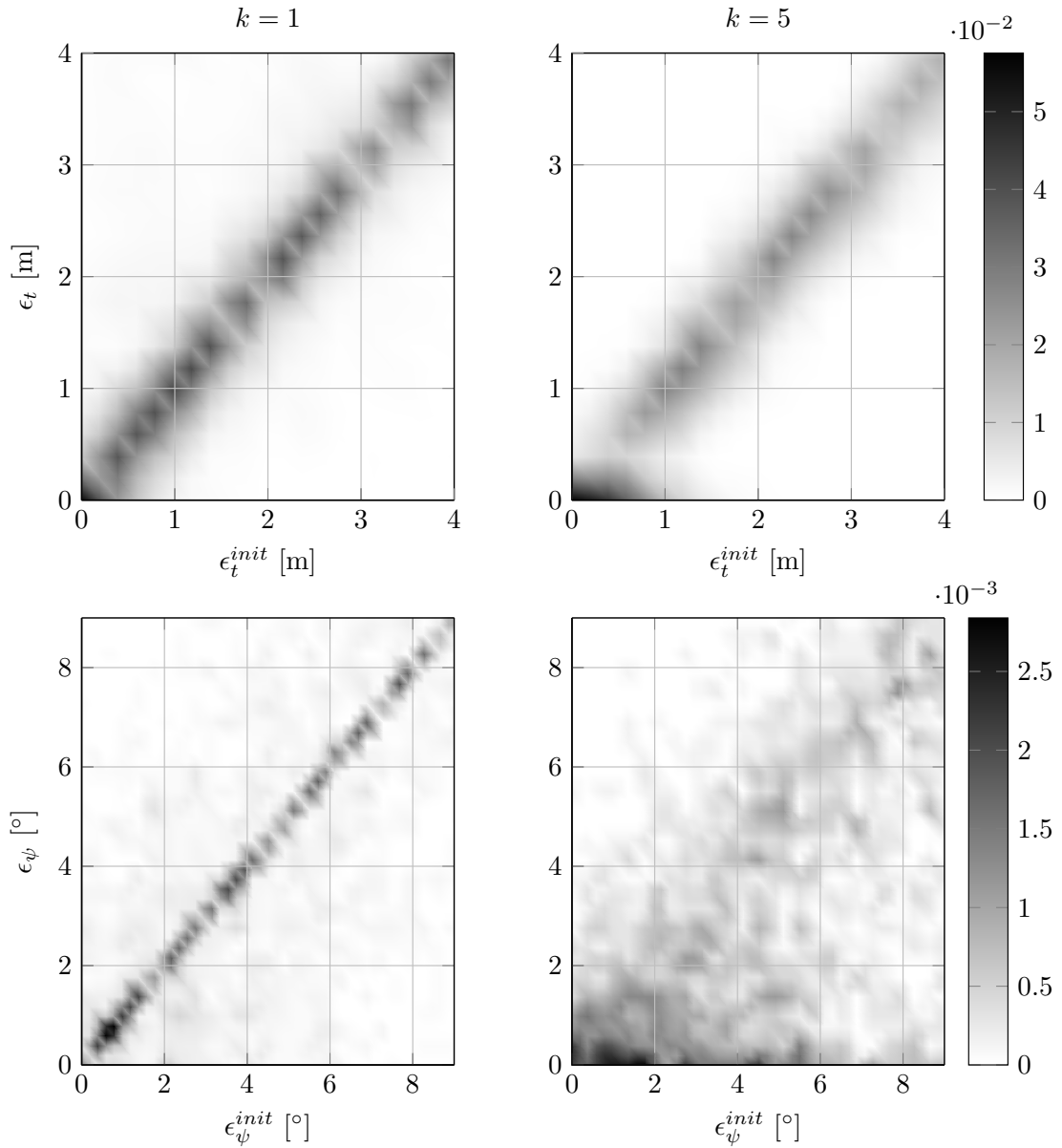


Figure 5.18: Translational ϵ_t and rotational ϵ_ψ registration error over initialization position of initial translational ϵ_{trans}^{init} and initial rotational error ϵ_ψ^{init} for two different sub map configurations, $k = 1$ (left) and $k = 5$ (right). Registration result of [MT5], figure modified.

Comparing to the ground truth registration, the achievable registration accuracy is measured for various sub-map sizes. The illustration of the translational errors ϵ_t and rotational error components ϵ_ψ of a set of registrations in Figure 5.17, indicate larger sub-maps $k \geq 5$ to converge more accurately to lower registration errors. Especially a translation error lower than 1 m and a rotational error below 1° can be achieved.

For the translation error a suitable sub-map size it is found to significantly improve the accuracy of precise registrations, below 0.1 m translation error. Among coarse registration

results >1.0 m of translational error, larger sub-maps decrease the registration result only by a little amount, the major contribution is the increase of the precise registrations.

As indication of the registration robustness, a perturbation based on 4σ variation is tested. Figure 5.18 displays the registration result of 4σ perturbations as comparison of translational error and rotational error at two exemplary sub-map configurations $k = 1$ and $k = 5$.

The registration result heat map in Figure 5.19 displays for the larger sub-map size of $k = 5$ increased convergence stability. This is derived and based on a detailed analysis:

- Remaining values on the diagonal of the heat map represent a registration non-convergence. The initial transformation perturbation yields the same after registration. Hence, no better registration is found.
- A globally lower translation error is achieved. The density of the final error values is significantly reduced from the non-converging diagonal and improved to the range $[0, 1]$. Worth notice is that a registration convergence, e.g. below 0.5 m, is only achieved for the sub-maps with initial translation offset below 2 m.
- A globally lower rotation error is achieved. For the rotational parts, the convergence is significantly improved to a range below 2° , even if a initial perturbation is set to high deviation. As a result, the non-convergence diagonal is definitively reduced.

Besides, the registration result is also rated in Algorithm 2 by a registration score. The mean and scale of this score value s is visualized in Figure 5.19. With the sub-map configuration $k = 5$, it is found that the registration score does not scale with increasing sub-map size k . Hence, a qualitative consideration or rating of a registration can not be assessed or compared based on the registration score metric s . The radar specific variable count of detections deprecates the score measure s as registration accuracy comparison metric.

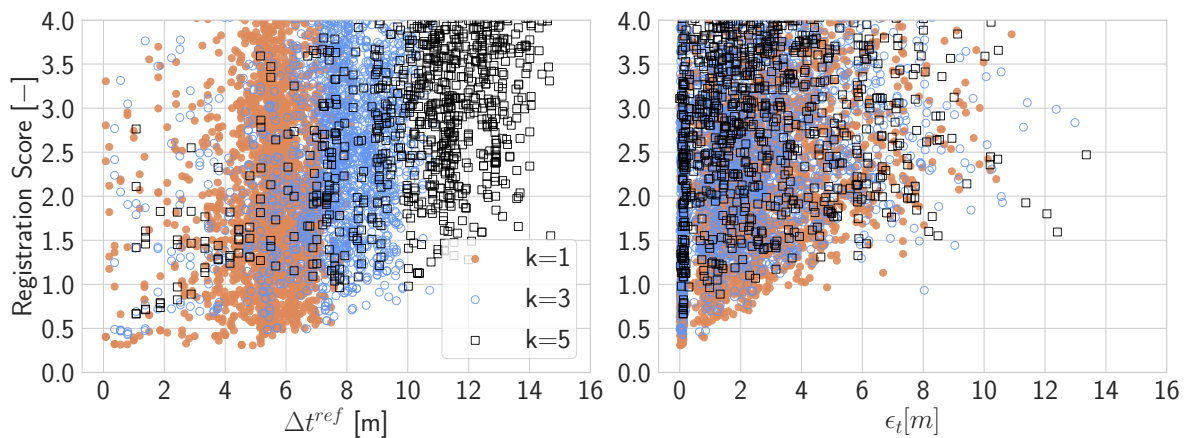


Figure 5.19: Registration score over true translational pose distance Δ_t^{ref} for different sub map configurations $k_{s,d}$ (left), and resulting registration error ϵ_t with sub map configuration $k = 5$.

5.4.4 Semantic Registration

As registration focus of this thesis, the benefit of the NDT-registration, as introduced in Section 5.3.4, is tested different parametrization. Spatially sparse regions cause difficulties for the normal distribution registration. For cells with low occupancy, the sparsity and few detections causes imprecise covariance matrices. The visualization of the normal distribution as input is found in Figure 5.20. Hence, sparse regions are partially not represented as normal distributions and are not respected in the subsequent scan-to-scan or sub-map to sub-map registration. This adds robustness to the semantic normal distributions and precise registration in the semantic radar SLAM concept can be achieved with noisy radar point clouds.

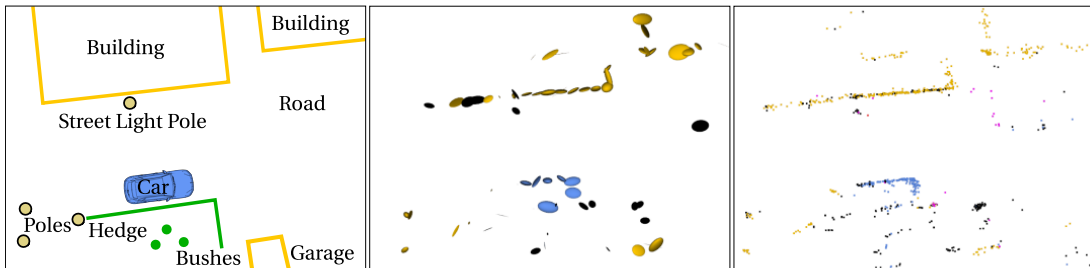


Figure 5.20: Illustration of the covariance of the semantic labeled normal distributions (center) compared to the corresponding point cloud (right). Color coding according to the semantic color convention of Table 3.2. Distribution illustration of [MT5], figure modified.

Depicted as registration error histogram in Figure 5.21, the comparison of three NDT registration algorithms, a regular NDT, the weighted semantic NDT (gSNDT) and the separative semantic NDT (sSNDT) is presented. As expected, both weighed and separated semantic NDT methods (gSNDT&sSNDT) increase the registration precision, yielding a translational error $\epsilon_t < 0.5 m$ and a rotational error $\epsilon_\psi < 1.5^\circ$. With this improved semantic radar registration the basis for a semantic radar SLAM is quantitatively given.

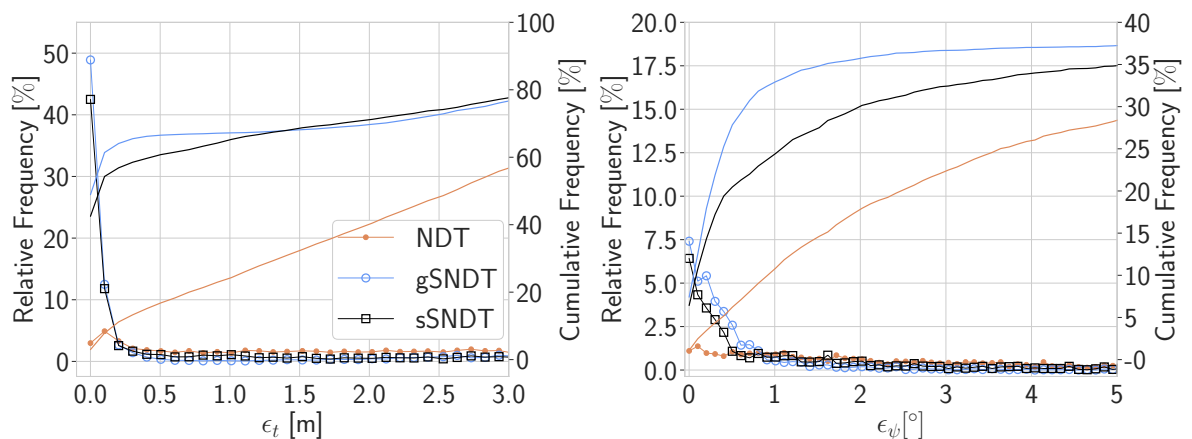


Figure 5.21: Histogram of the occurrence of translational ϵ_t (left) and rotational ϵ_ψ registration error (right) for regular NDT, weighted semantic NDT (gSNDT) and semantically separated NDT (sSNDT) registration.

Besides the improved accuracy, the plots in Figure 5.21 also show the significantly increased robustness of the semantic radar registration. Especially the fraction of non-converging registrations, found on the diagonal of the heat map, is significantly reduced. A large majority of the registrations converge to $\epsilon_t < 0.5 m$ even for initial translational offsets $\epsilon_t^{init} > 1 m - 6 m$. Similar findings apply in same distinctiveness for the registration convergence towards low final rotational errors $\epsilon_\psi < 1.5^\circ$.

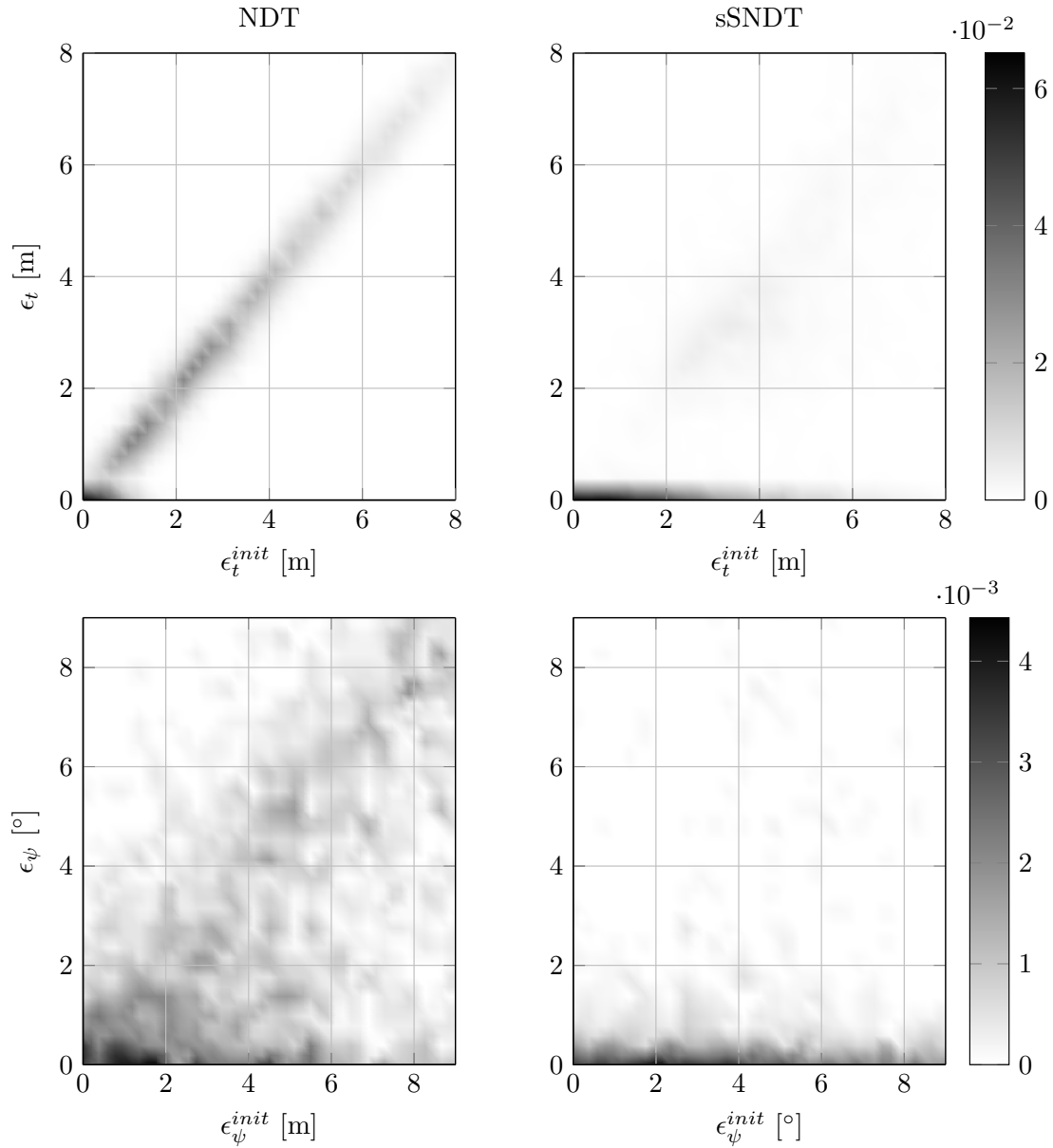


Figure 5.22: Translational ϵ_t and rotational ϵ_ψ registration error over the initial position translational ϵ_{trans}^{init} and rotational error ϵ_ψ^{init} of the initialization position for the registration, for conventional NDT (left) and semantically separating NDT (right). Registration result of [MT5], figure modified.

5.4.5 Semantic Radar SLAM Evaluation

As core focus of this thesis, the benefit of semantic radar labels in combination with the developed semantically separative NDT-registration (sSNDT) as introduced in Section 5.3.4 is experimentally proven in the following. The core of the semantic radar SLAM evaluation is based on the evaluation of achieved radar SLAM accuracy with the semantic information being utilized. Designed as model-free, feature-free and generic solution, the designed semantic radar SLAM aims to be applicable for arbitrary environments, especially for parking scenarios. Table 5.2 gives an overview to compare five different parameterized variants (A)-(E) of the SLAM functionality, on four test scenarios (I)-(IV), see Figure 5.25, to evaluate the semantic radar SLAM performance. The test scenarios of Figure 5.25 are comprised of multiple scenes (e.g. turning, stopping or other driving maneuvers) but consequently result in a scenario-specific parking test, following the scenario definition of Ulbrich et al. [214]. In order to test different environments, different locations serve as test-tracks in which the driven scenarios are tested.

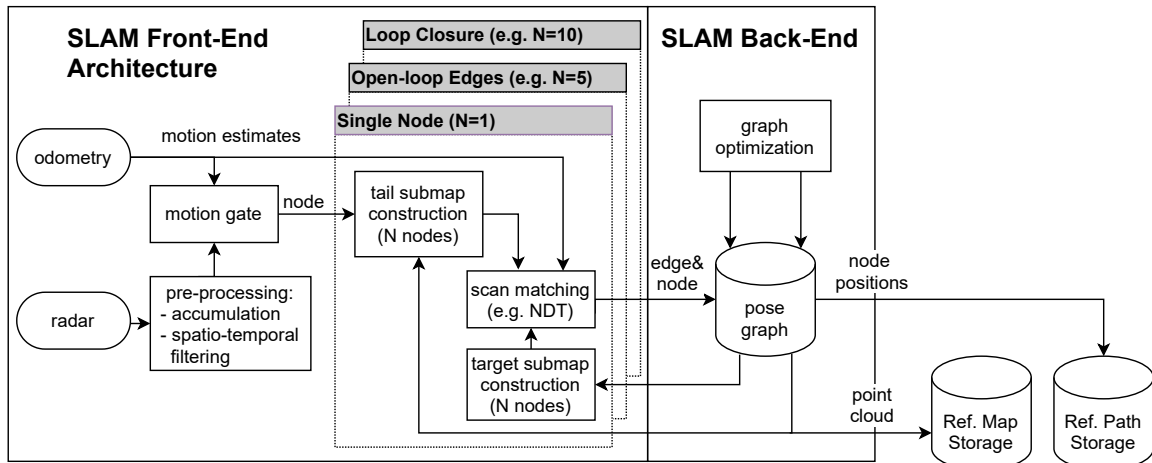


Figure 5.23: Illustration of the SLAM system architecture from perception modules to map storage.

As a result, the evaluation of the semantic radar SLAM accuracy is based on the comparison of the resulting SLAM trajectory with a differential GPS (dGPS) signal of a reference sensor. With the information of a synchronously available reference trajectory and reference position, for SLAM evaluations of the two trajectories, the Umeyamas' matching algorithm [215] is applied. The quantitative evaluation of the absolute pose error (APE) and relative pose error (RPE) yields typical SLAM metrics, reported with mean and maximum values in Table 5.3 and Table 5.4, to be compared.²

² Due to the generic environment applicability, not the environment map itself can be evaluated due to the lack of a reference radar map.

| Parameter | SLAM Variant | | | | |
|-------------------------------|--------------|-------|-----|-----|-----|
| | E | D | C | B | A |
| Scan Accumulation [-] | 4 | 4 | 4 | 4 | 4 |
| Pre-Filtering | SST | SST | ST | ST | GEO |
| Sub map Configuration k [-] | 5 | 5 | 5 | 5 | 5 |
| Registration Method | gSNDT | sSNDT | NDT | NDT | NDT |

Table 5.2: Overview of the evaluated SLAM configurations with different registration methods (NDT,gSNDT: weighted semantic NDT, and sSNDT: semantically separating NDT) and pre-filters (SST: semantic spatio-temporal, ST: spatio-temporal, GEO: spatial).

SLAM Evaluation Scenarios: For the evaluation of the SLAM functionality, four different scenarios are selected of which two are from the same track, but processed with different semantic label quality. One of this equal scenarios is tested with automatically generated semantic radar labels from the automatic labeling in Section 3, while the other scenario is manually corrected to contain semantic ground truth labels. The automatically labeled scenarios contain an additional class **Road** ■, which is an artifact of the automatic labeling. This label is not explicitly considered for the semantic NDT registration, but forms a sub-sort of artifacts and is treated as **Artifact, Unknown** ■. Hence, even in the SLAM map a low number of these labels can be found.

The comparison of Figure 5.24 also matches the findings of the confusion matrix in Figure 4.25, discussed in Section 4.5. Person ■ is mainly confused with vegetation ■, while vehicles ■ are confused with clutter ■.

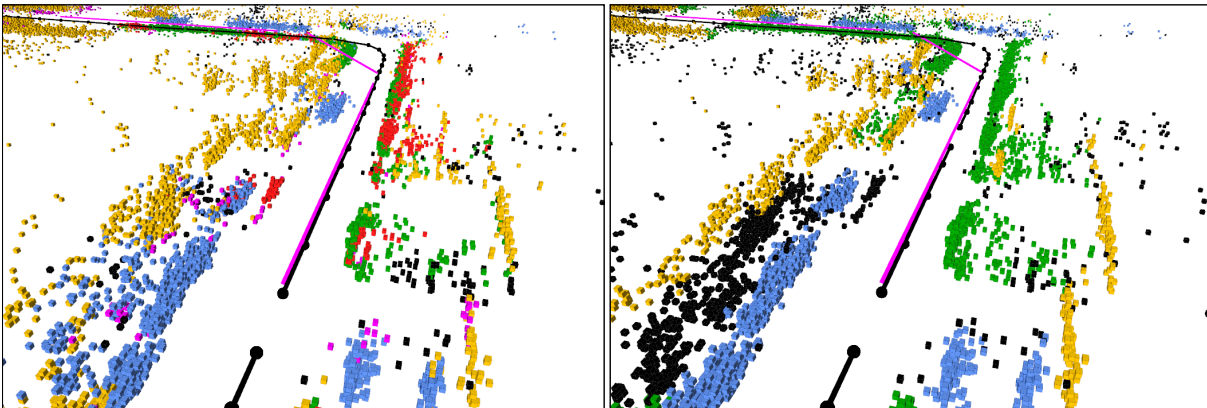


Figure 5.24: Exemplary comparison of the same scene of the scenarios with automatically generated semantic labels (left) versus the same scenario with manually corrected ground-truth semantic labels (right). Colors according to Table 3.2 with additional road ■ and unknown ■.

The test scenarios are driven manually to record all sensor data. For the evaluation of the different SLAM variants, they are applied on a sensor data replay, off-board the vehicle to

guarantee the same input data. With this workflow, the whole SLAM functionality is developed and tested first offline, while the software is integrated in the test vehicle for the real world testing of the automated drive in the next chapter. The test scenarios contain longer parts with turns, loop-closures after a longer *open* part without loop-closure. It is emphasized, not to revisit locations or scenes of the scenario at larger scale, in order to avoid the opportunity to increase accuracy by loop-closures or denser mapping.

Figure 5.25 a)-d) illustrate the resulting semantically segmented radar SLAM maps. The maps illustrate the output of SLAM version (D), with automatically labeled radar data (I-III) and manually corrected semantic ground truth labels (IV).

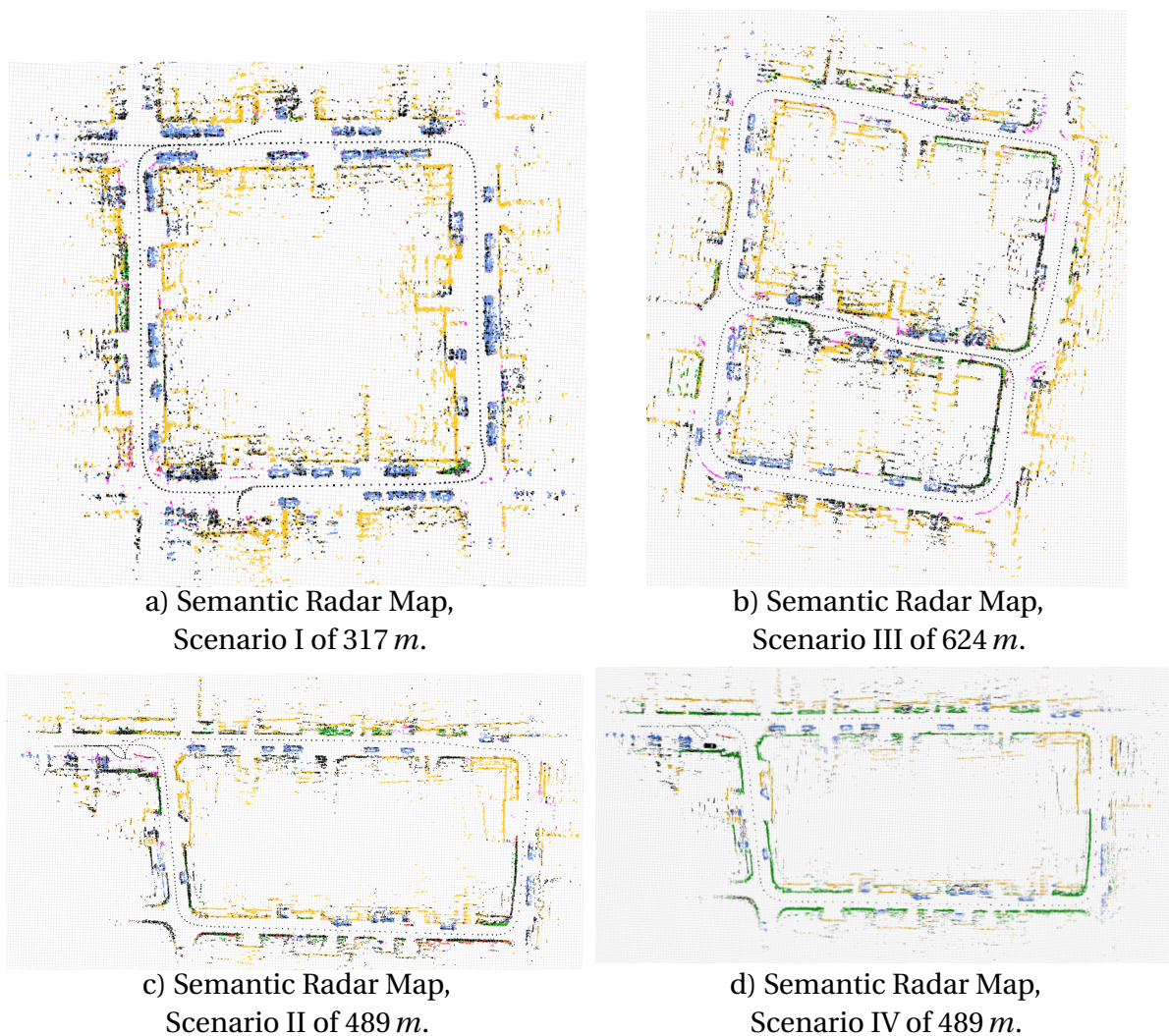


Figure 5.25: Driven ego-trajectory plotted as black dots, resulting radar map of the (sSNDT) SLAM. Semantic coloring according to Table 3.2: Building ■, Vehicle ■, Vegetation ■, Person ■, Pole ■, but with Road ■, and Unknown ■. All plots overtaken from Isele et Al. [SI4].

The evaluated test scenarios yield the following results. APE and RPE error measures are found in the Tables 5.3-5.4. Over all sequences, the achieved relative pose error consis-

| Scenario | (E) | | (D) | | (C) | | (B) | | (A) | |
|----------|------|------|-------------|-------------|------|------|------|------|------|------|
| | max. | mean | max. | mean | max. | mean | max. | mean | max. | mean |
| I | 0.71 | 0.35 | 0.53 | 0.32 | 0.93 | 0.37 | 0.74 | 0.39 | 1.32 | 0.41 |
| II | 1.02 | 0.64 | 0.96 | 0.64 | 2.44 | 1.40 | 2.34 | 1.37 | 2.70 | 1.46 |
| III | 0.71 | 0.42 | 0.66 | 0.40 | 0.78 | 0.42 | 0.77 | 0.42 | 1.12 | 0.63 |
| IV | 1.05 | 0.59 | 0.96 | 0.56 | 2.44 | 1.40 | 2.34 | 1.37 | 2.70 | 1.46 |

Table 5.3: Comparison of the Absolute Pose Error (APE) [m] of the SLAM variant (A-E) in test scenarios I-IV.

| Scenario | (E) | | (D) | | (C) | | (B) | | (A) | |
|----------|------|------|-------------|-------------|------|------|------|------|-------------|-------------|
| | max. | mean | max. | mean | max. | mean | max. | mean | max. | mean |
| I | 0.29 | 0.13 | 0.25 | 0.11 | 0.32 | 0.12 | 0.29 | 0.13 | 0.45 | 0.14 |
| II | 0.56 | 0.1 | 0.52 | 0.1 | 0.45 | 0.1 | 0.46 | 0.1 | 0.23 | 0.09 |
| III | 0.28 | 0.11 | 0.27 | 0.11 | 0.27 | 0.11 | 0.27 | 0.11 | 0.22 | 0.1 |
| IV | 0.56 | 0.1 | 0.52 | 0.1 | 0.45 | 0.1 | 0.46 | 0.1 | 0.23 | 0.09 |

Table 5.4: Comparison of the Relative Pose Error (RPE) [$\frac{m}{10m}$] of the SLAM variant (A-E) in the test scenarios I-IV.

tently yields an approximate error of $0.11 \frac{m}{10m}$. This achievement can be interpreted as local trajectory consistency, consequently yielding also a global radar map consistency as the loop-closure proves.

The maximum and mean value of the RPE shows low variation over the tested SLAM variants. Due to the SLAM construction including the wheel-based odometry estimation with a comparably high covariance, the odometry is most influential on local trajectory scale and the relative pose error. Other graph edge types, e.g. the radar odometry, have comparably lower influence due to their lower defined covariance certainty. Hence trajectory dependent odometry errors occur as influence of RPE differences between the scenarios.

Due to a higher number of loop-closures, both scenarios I, III are consistently more accurate than the difficult scenarios II, IV. The long passage of the mid-section in III and the loop-closure after a full round trip in scenario I allows more loop closures. In both scenarios II and IV, the place is revisited from an other direction, maneuvering is included, but the overlapping map sections of the revisited area are smaller and of difficile vegetation.

SLAM Result Visualization: Based on the error tables, the quantitative error can be evaluated and compared. With the visualization of the mapped scenarios, the radar data associations of the SLAM map can be inspected visually. Figure 5.25 illustrates the whole map to gain an insight into the general scenario and object perception along the track. Especially the effect of the semantic labels supports the consistent association of corresponding parts of the environment. Even if the automated semantic labeling is applied, it is impressively proven by Figure 5.26, that the additional information yields highly beneficial association support. The concept of semantic radar point cloud registration yields the presented semantic radar SLAM maps, which are intuitive to visually interpret, inspect and reuse for other applications. Compare Figure 5.27, how mighty the semantic labels enable an accurate data association to yield a unprecedented opportunity to build semantic radar maps.

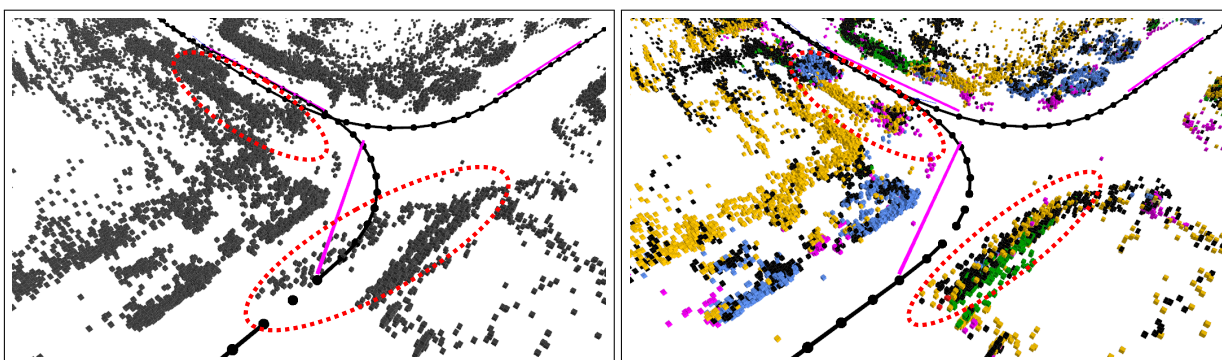


Figure 5.26: Comparison of the spatial radar SLAM (left), utilizing only spatial radar information, and semantically separated (sSNDT) SLAM result (right), both variants display the same revisited part of test scenario III. SLAM nodes visualized as black circles, connected by black odometry edges, loop-closure edges visualized in blue, radar odometry edges are colored in magenta. SLAM map differences highlighted in red ellipses. Colors according the semantic classes of Table 3.2. Figure of Isele et Al. [SI4].

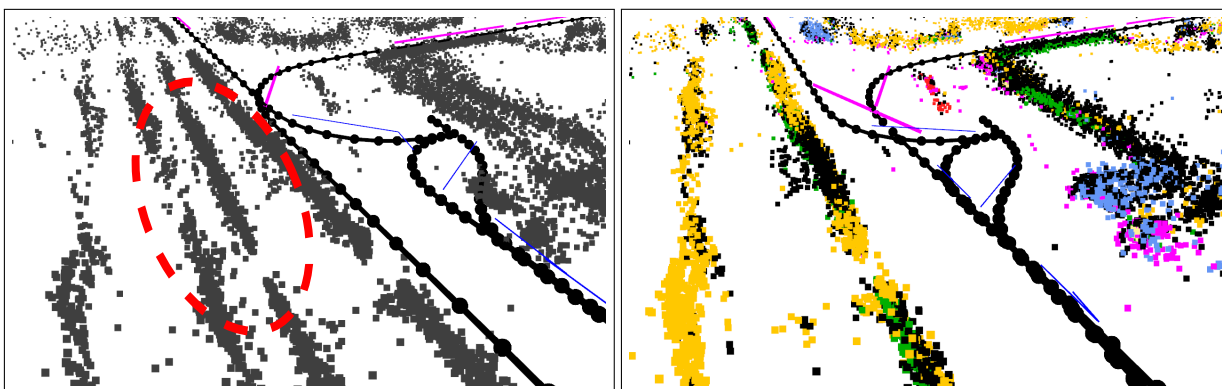


Figure 5.27: Comparison of the spatial radar SLAM (left), utilizing only spatial radar information, and semantically separated (sSNDT) SLAM results (right), both variants display the same revisited part of the test scenario II, including a parking maneuver to illustrate the loop closure over a long drive. SLAM nodes visualized as black circles, connected by black odometry edges, loop-closure edges visualized in blue, radar odometry edges are colored in magenta. SLAM map differences highlighted in red ellipses. Colors according the semantic classes of Table 3.2. Figure of Isele et Al. [SI4].

5.5 Section Conclusion

The capabilities of the presented semantic radar SLAM affirms to the original expectation, that semantic label information substantially improves the SLAM process. Averaged over four real-world test scenarios of comparable complexity and length of 317 m – 624 m , a mean APE of 0.48 m is achieved at an average RPE of 0.11 m for the best SLAM variant.

Answering the *mapping* research questions of Section 1.3, this section contributes four dimensions of improvements from the semantic radar labels and semantic radar point cloud SLAM compared to a classical point cloud SLAM:

A cheap and common automotive radar sensor set can be fused from a multi-sensor setup, even at sub-shell integration as localization and mapping sensor. As the additional semantic information can directly be processed without type conversion or projection, avoiding algorithmic complexity or run-time issues, an intermediary system integration is possible.

Second, the semantically separated NDT (sSNDT) scan-matching is consistently improved to achieve registration convergence in a large range: Translational initial errors of approximately 5 m – 7 m , whereas rotational initial offsets of up to 5° still achieve registration convergence below 0.5 m registration offset. The sSNDT scan-matching performs robustly and yields the similar accuracy when applied with automatically generated semantic labels, containing erroneous semantic labels. The radar SLAM benefits of accurate semantic radar labels but does not necessarily require perfect error-free semantic radar segmentation.

Third, the sSNDT data association enables a radar SLAM map generation, allows loop-closure on complicated tracks and scenarios and under real-world odometry drift over a test loop of approximately 165 m . The parametrization of the NDT grid size 0.2 m is found to be sensitive to the spatio-temporal pre-filter and the de-noising capability of the *RadarNet*. With increasing clutter content in the radar point cloud, the NDT convergence degenerates even with semantic registration.

Fourth, the resulting semantic radar maps of the sSNDT radar SLAM open new opportunities for post-processing in terms of drive-able space estimation, object instance segmentation, or online/ offline map-fusion, map-assembly, or map-updates of large scale radar maps, etc. The realization of this accurate radar mapping, even accumulated and based on the sparse radar sensor data, enables a whole new field of developments to be tested on semantic radar data, which constitutes a new research field.

The presented semantic radar SLAM is the first successful and real-world applicable semantic radar SLAM and is also tested under real-world conditions in a vehicle, independent of a limited data set. The semantic radar environment mapping is subject of multiple patented applications: Parking environment meta-information utilization [Pat1], parking collision detection [Pat2] or scene reconstruction [Pat5].

As Figure 1.4 illustrates, the semantic radar SLAM on system level builds on the already discussed function and feature contributions, see Chapters 3&4, and is the basis of the aimed parking functionality on vehicle level. Section 6 discusses the integral functionality of the compounded system level contributions of the previous sections.

5.6 Section Outlook

SLAM Mapping - Visual Constraints: In order to further robustify the SLAM, future work is open towards the integration of camera images and visual place recognition. Synchronized camera images, preferably of a typical surround-view camera belt of modern vehicles, can be stored with the semantic radar graph SLAM nodes. The visual detection of ego-vehicle motion and orientation from subsequent camera images can be included similarly as radar graph edges with relatively low effort. Including the visual sensors affects the graph SLAM front-end and is expected to add further robustness, especially in resembling environments of repetitive structures.

SLAM Mapping - Optimization Effort: A further optimization of the SLAM system addresses the performance at scale beyond the development compute platform. At each optimization step of the presented graph SLAM, the whole graph is optimized, yielding an increasing compute effort for growing maps. To provide a SLAM solution to be applicable at large scale, the optimization might be constrained to a certain graph region, in order to guarantee a deterministic maximum computational effort.

SLAM Mapping - 3D Path Mapping: Motivated from a multi-story parking garage use-case, the requirement to be able to map and drive over multiple story garages evolves. By now, the radar mapping process does not account for any z-elevation since the radar point cloud registration does not reliably detect elevation changes. Hence, the application is limited to a quasi-flat area. The above proposed SLAM extension to include camera-based visual odometry constraining edges might deliver a robust z-coordinate.

SLAM Mapping - Map Fusion and Updates: Passages of the same scenario result in an increased representation and knowledge of a track and environment. As the visited environment might contain changed or adapted regions along with static or unchanged parts, one could think of a update or fusion of the underlying map or specific map regions to increase the map-precision iteratively. In perspective of a vehicle fleet application, the task to align, maintain the multiple maps, and potentially couple the separate but overlapping graph problems in a back-end. With such a coupling of fleet-based sensor data and graph-representations of the changing environment, future research might address the task of self-updating maps based on a graph-formulation.

6 PRETRAINED AUTOMATED PARKING

In this final section, the before introduced software modules of live semantic radar Segmentation *RadarNet* of Chapter 4 and the semantic Radar SLAM of Chapter 5 are combined to an exemplary use-case of an automated parking functionality, called *trained parking* (TPA). Based on the vision to realize autonomous and automated parking on arbitrary home ground or as general parking pilot, the basic interplay of the developed software modules is empirically tested and performance evaluated under real-world conditions. The achieved semantic radar mapping capacity and re-positioning accuracy of this Proof of Concept (POC) formulates an upper bound realize-able precision of this functional concept.

The section first explains the combined systems' design and interplay, then the experimental results are presented and evaluated. The map content metrics are developed during the Master thesis supervision of Avinash Shankar Bhat [MT6].

6.1 Evaluation and Selection of existing Approaches: Radar-based automated and autonomous Parking Systems

As discussed in the introduction, modern automated parking functionalities in the ADAS space require the passage of an empty parking space, being either framed by nearby objects (e.g. parked cars), or visually indicated by ground parking lines. The parking functionality measures with distance sensors (e.g. ultra-sonic sensors) the ego-position with respect to the free parking space and its objects, then performs the parking maneuver based on this local obstacle perception. In the visual case, the parking lines deliver the framed target position.

Narula et al. [151] is the only work, relying on radar sensors to re-localize in an urban environment. Other works [6, 108, 135, 168, 167] showcase the potential to detect parked cars and free parking spaces in radar data. Works on SLAM or automated driving setups of Table 2.3 are based on HD-maps, LiDAR or other reliable target paths and environment maps [240].

To the authors best knowledge, there is no radar-only based automated or autonomous driving example yet reported in literature. Therefore, this section combines the before discussed *RadarNet* Segmentation together with the semantic radar SLAM, and a radar map relocalization, to realize an actively actuating autonomous parking functionality.

6.2 System Overview

In order to evaluate the practicality of the developed semantic radar SLAM application in real vehicle conditions, a solely radar-based automated parking functionality use-case is built and tested in this section.

Automated Parking: In the first stage, the solely radar-based *trained parking* functionality maps an the environment and records the driven path of a manually driven sample parking maneuver in an arbitrary environment. In the second stage, the automated driving stage, the trained parking functionality re-localizes itself in the radar map and starts an automated drive along the reference path to the manually trained parking position. Both stages involve the developed semantic radar segmentation and semantic radar SLAM as core functionality.

A very general system description of the use-cases' software modules is depicted in Figure 6.1, whereas figure 6.2 is a detailed extension of the generic vehicle setup of Figure 2.15 of Section 2.7. The system in Figure 6.1 processes individual sensor data in four, orange marked exemplary processing steps: The perception creates a meaningful environment interpretation thereof **(A)**, yielding an environment map **(B)**, in which the vehicle is localized. Based on the location comprehension **(C)**, the vehicle plans its further motion and sends drive controls **(D)** to drive in the environment in fully automated operation.

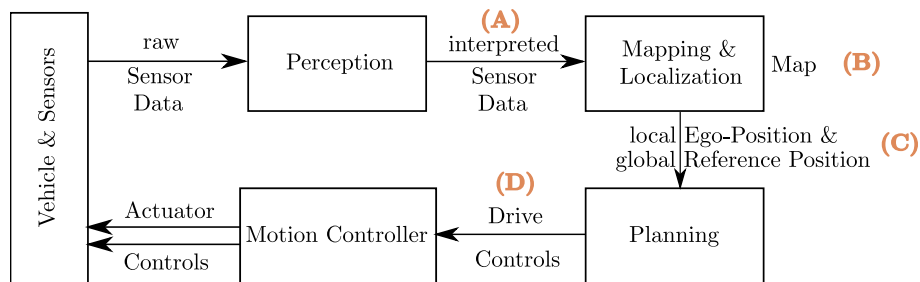


Figure 6.1: Illustration of a general robot control system cycle, from perception modules to vehicle actuators of which the thesis addresses the *perception* and *mapping* specifically for radar.

Perception: The semantic radar segmentation enriches the raw radar signal information content to a new semantic domain. Subsequent filter modules pass-through specific signal ranges to a spatio-temporal filter, reducing further noise in the radar signals.

Mapping: The subsequent semantic radar SLAM makes use of the semantic information dimension to apply a semantic and spatial registration to associate subsequent radar scans to each other. The resulting semantic radar map and reference path of the initial environment passage are depicted as stored entities in the right part of the top block in Figure 6.2.

Instead, of relying on Global Navigation Satellite System (GNSS), for the use-case of *trained parking*, re-localization is performed by map-matching semantic radar maps, analogously applying the semantic NDT registration.

Planning: Having a reference path from the initial SLAM mapping to be followed as target position path, the planner module computes an optimal trajectory to drive from the current ego-position onto the reference path and along.¹ Its input is the current vehicle position based on odometry signals $(x_{\text{ego}}, y_{\text{ego}})$ and the reference path $x_{\text{ref}} = x_0 \dots x_i, y_0, \dots y_i, i \in \mathbb{R}^{1 \times N_{\text{path}}}$ to be followed. Both inputs are given in global coordinates and assume planar motion in (x, y) coordinates.

Motion Control:² The output of the planner module are two actuation requests for lateral motion m_{lat} and longitudinal motion m_{long} , which the motion controller translates into vehicle actuator manipulation.

The perception and planning stack in ROS is connected via a private CAN to a real-time platform on which the Matlab Simulink based controller is running, which again is CAN communicating with the vehicle actuators. According to Figure 6.1, the update cascade follows from slow map-matching updates ($\approx 8 \text{ Hz}$) with a reference path projection update, updating the quick planner ($\approx 15 \text{ Hz}$) conditions of the update to calculating the underlying motion controller outputs at $\approx 30 \text{ Hz}$.

Further details on the system integration, initialization and map-matching are found in the Appendix Section A.

6.3 Design of Experiments

The driving tests of the trained parking functionality is fully automated with no manual interference, except the function initialization and final end-position confirmation. The automated functionality test setup ensures comparable experimental results under changing real-world conditions. For the experiments, a safety driver is required to supervise the system and is only allowed to intervene, if the automated functionality fails during execution due to erroneous re-localization, in cases of potential collisions, or other failures. Furthermore, the system is tested with respect to a changing environment and weather, also including minor natural changing vegetation during the tested time and different initialization positions on different test days.

¹ It is assumed that the drive-space is free, without interference of blocking obstacles, persons, of other traffic.

² The deployed trajectory planner is designed by Lukas Köhrer from Forschungszentrum Informatik FZI ³. As part of joint supervision of the Master thesis of Fabian Bischoff [MT4], the planner and the actuation controller are integrated in the test vehicle and treated in this work as given modules.

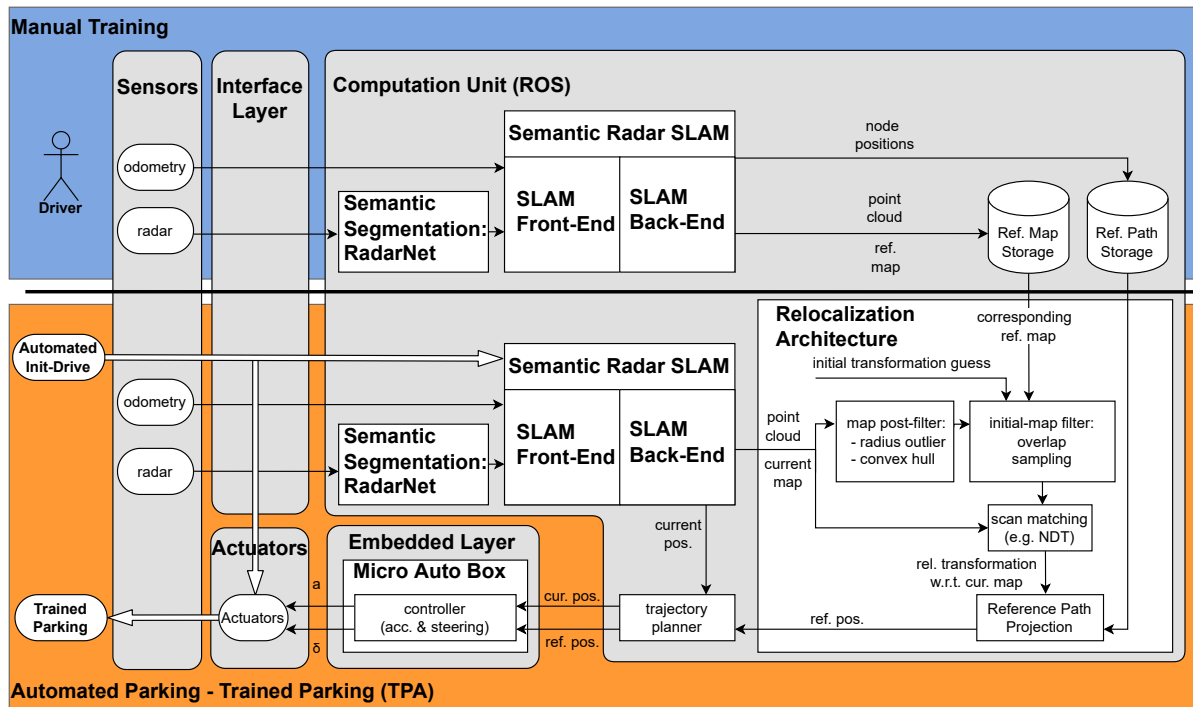


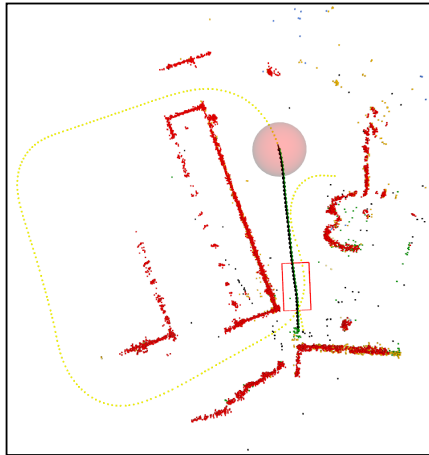
Figure 6.2: Illustration of the overall system architecture from perception modules to vehicle actuators as more detailed variant of the generic system Figure 2.15. The blue top section depicts the system operation in manual training mode, whereas the bottom orange section illustrates the system operation in automated parking mode. Detail information of the computation modules are found in Chapter 4 for the semantic radar segmentation and the semantic radar SLAM is discussed in Chapter 5.

Reference Map and Path: Per test environment a single initially mapped or parking maneuver serves as reference map and reference path for all subsequent executions and evaluations of the automated trained parking runs.

Fully automated Function: First, the test vehicle is set to the start position, the systems are started and the initialization drive is manually triggered. The vehicle automatically starts an initialization drive into the red square of Figure 6.3 and stops. From the position inside the red rectangle, the relocalization module performs the initial registration and displays the map registration guess. The safety driver confirms the re-localization and reference path projection to start the autonomous parking functionality.

Visual Supervision and Confirmation: During the execution of the autonomous parking, the safety driver inspects a surveillance registration projection, see Figure 6.3. In this registration surveillance screen, the updating re-localization and map-matching is shown in top-view perspective.

Registration: SLAM Map + Reference Map



SLAM Map

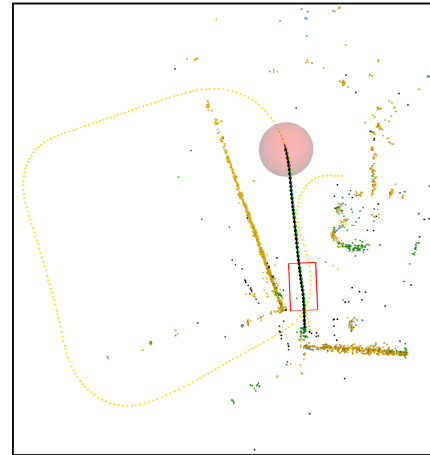


Figure 6.3: Registration surveillance (left) with red reference map and live built semantic SLAM-map (right) of test scenario A and B, see Section 6.4 and Figures 6.5- 6.7. Projected reference path in yellow, driven SLAM path in black with a 7 m red loop-closure search range. The red rectangle shows the start position of the automated drive, equivalent to the end-position of the init-drive.

In a second, surveillance camera projection of Figure 6.4, the reference path is projected into a front-facing camera image, including the projected reference map and the current SLAM map detections. The radar target positions can be checked in vehicle perspective during the drive for potential collisions.

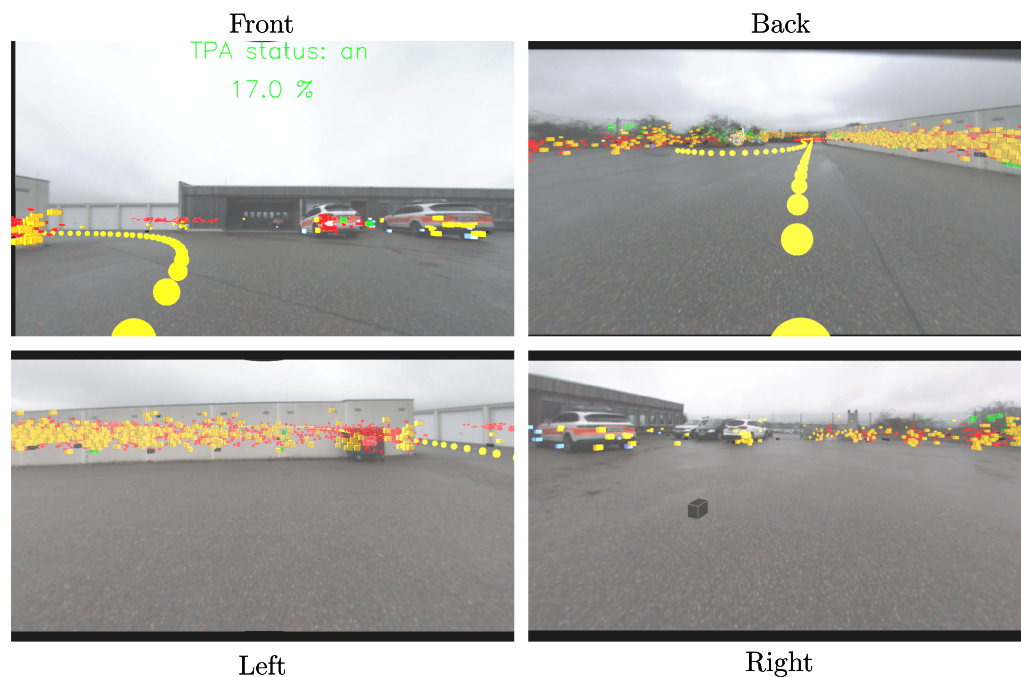


Figure 6.4: Camera perspective of surround view cameras with projected reference map (red) and live semantic radar SLAM-map of test scenario B, compare Figure 6.7. Projected reference path in yellow, red boxes are radar detections of the reference map, compare Figure 6.3 (left) in overlay to the live semantic radar SLAM-map as colored boxes according to the color code in Table 3.2.

6.4 Test Scenarios

Tested under different weather conditions, the test scenarios are recorded on three different autumn days to test in rain, with fog and during sunshine. Four different test scenarios are initially mapped and consequently driven automatically to evaluate the system' precision. The test sequences are depicted as scene images or in top-view perspective in Figures 6.5, 6.7, 6.9, 6.11.

The first three of the scenarios A-C are open-sky scenarios, which allow differential GPS reference position to be recorded and evaluated for the accuracy of the automated parking. For the fourth scenario D, the multi-story car park blocks the differential GPS data to be recorded. Based on the d-GPS, a trajectory-based evaluation of the semantic radar SLAM system positioning accuracy can be performed.

Including minor environment changes, the effect of changing maps is compared by a second measure based on semantic voxel distance of a map to map comparison measure. With this measure, the robust radar segmentation and map stability is quantified to check the semantic and structural semantic radar map consistency besides the changing objects.

The first two scenarios to be tested include urban scenarios with garages and buildings. Both start at the same location, surpassing partly the same environment but following different path ends. Sequence A, shown in Figure 6.5 contains a potential loop-closure. Test sequence B, shown in Figure 6.7, is a longer open-loop drive around a block of garages with flanking vegetation, cars and a wooden shelter building.

As third scenario C a public parking lot with large proportion of vegetation such as hedges, bushes and with a loop-closure ensures a vegetation-framed test scenario with changing parking lot occupancy, see Figure 6.9.

As last test scenario D, a multi-story parking garage is entered from outside, following a path through parked cars to an end-position. This final test scenario is chosen to test the robustness against map changes. Including a larger count of changing vehicles, the moving and static vehicles provide significant environment changes. As drawback, the multi-story building blocks the differential GPS reference sensors. Consequently, no absolute positioning accuracy can be measured for this test scenario and track.

Each of the test scenarios is presented in detail in the following.

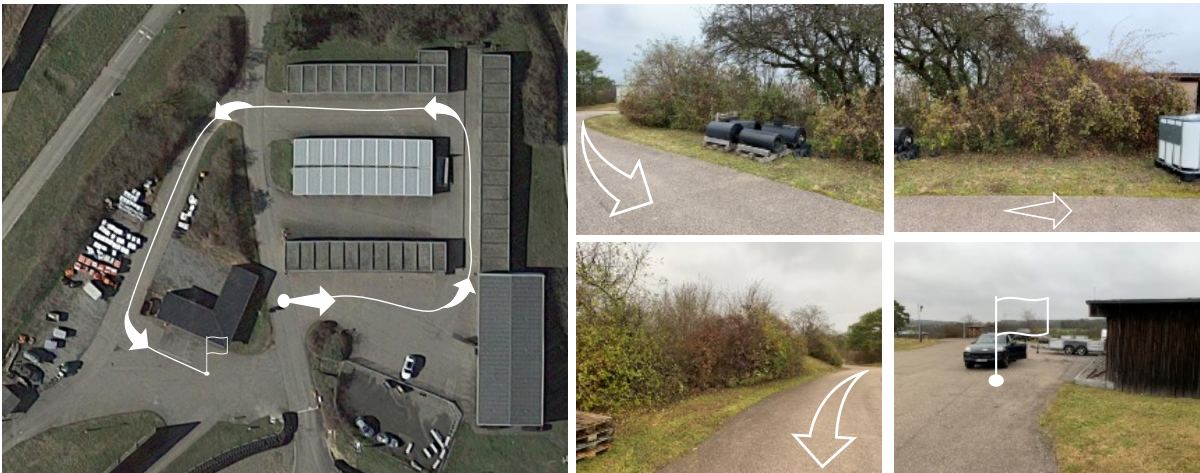


Figure 6.5: TPA scenario A as satellite image [79] and scene images around garages as open loop drive.

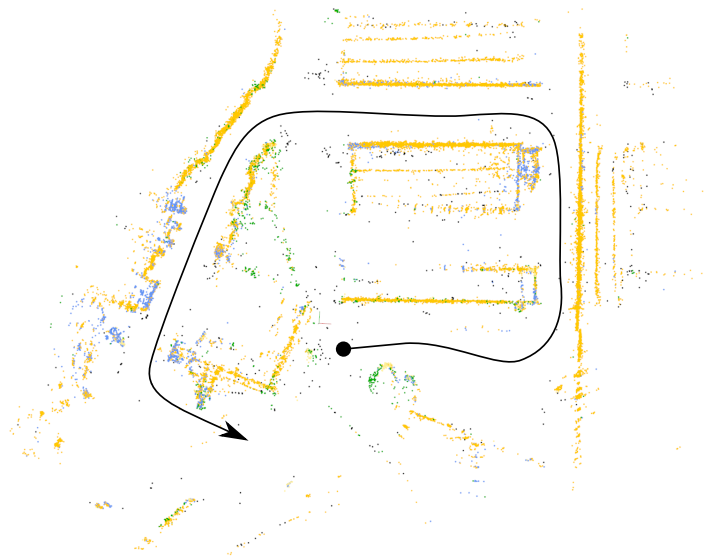


Figure 6.6: Illustration of the test scenario A as resulting semantic radar SLAM map (see Section 5.4.5), with sSNDT registration of Section 5.3.4 and Section 5.3.4.2, applied on the live inference of *RadarNet* of Section 4.5. Colors according the semantic classes in Table 3.2.

The open-loop runs through the garage block, not reversing directly to the start area, but passing through a vegetation⁴ covered hump-framed path around a wooden building. Framed by arbitrary objects, the latter of this path is semantically identified as vehicles or buildings.

⁴ From the top-view perspective of the radar map, it is not visible that the potential mis-classification of vegetation (left top-corner) is interpreted misleadingly, as the underlying humps or hillocks are mainly sensed. Hence, the visual top-view inspections does not comply with the expectation to find vegetation-classification in the map.



Figure 6.7: TPA scenario B as satellite image [79] and scene images with loop-closure.

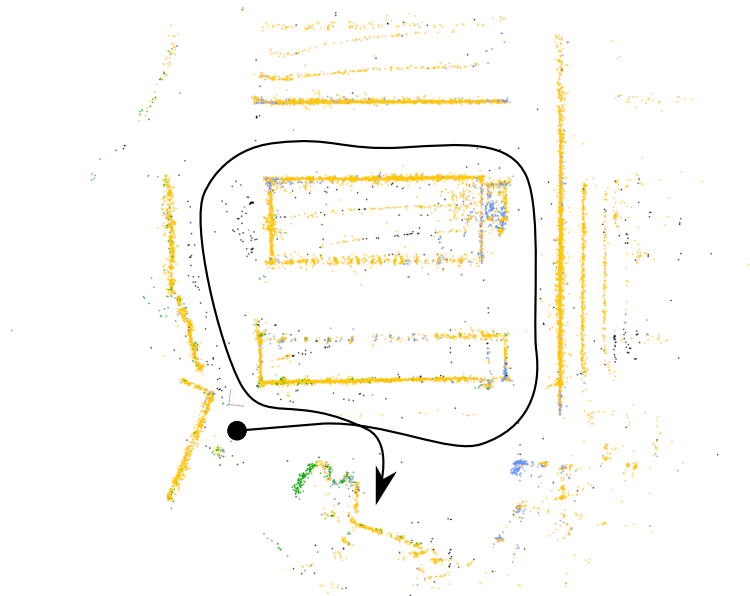


Figure 6.8: Illustration of the test scenario B as resulting semantic radar SLAM map (see Section 5.4.5), with sSNDT registration of Section 5.3.4 and Section 5.3.4.2, applied on the live inference of *RadarNet* of Section 4.5. Colors according the semantic classes in Table 3.2.

The loop-closure trajectory leads the vehicle a second time through the starting area, allowing a SLAM loop-closure to occur, before parking around a vegetation-isle in front of a metal wire fence.



Figure 6.9: TPA scenario C as satellite image [78] and scene images of the public parking lot with loop closure.

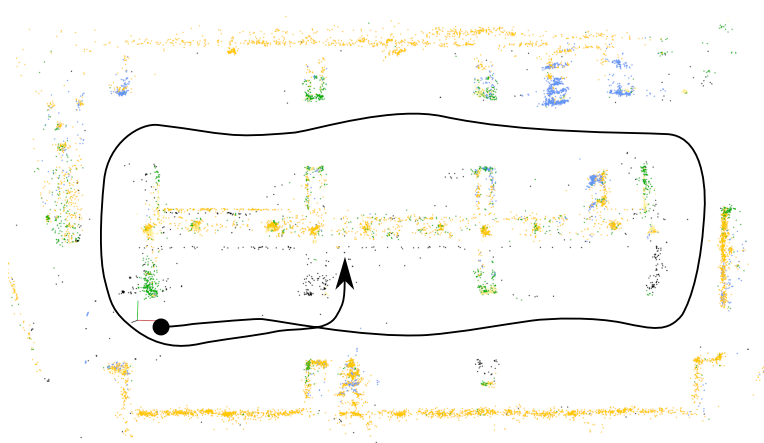


Figure 6.10: Illustration of the test scenario C as resulting semantic radar SLAM map (see Section 5.4.5), with sSNDT registration of Section 5.3.4 and Section 5.3.4.2, applied on the live inference of *RadarNet* of Section 4.5. Colors according the semantic classes in Table 3.2.

As third test scenario, a public parking lot is chosen, depicted in Figure 6.9. In this scenario, the reference path is framed by vegetation (dense hedges and bushes are supposed to reflect radar randomly), containing flower beds and parked cars perpendicular to the driving path. The final parking position is located besides a flower bed. This scenario includes a loop closure in order to increase the traveled distance and to check if the semantically-assisted loop-closure is achieved with a large map proportion of vegetation samples. The reference

radar map is shown in Figure 6.10.⁵ From the scene photos, the non-planarity of the scenario can be seen. The parking arms are connected by a ramp, which is separated by a little trench in between. Traffic lights and trunks of trees can be found as circular *pole* clusters in the radar map.



Figure 6.11: TPA scenario D as scene images of the car park at floor level as open loop drive.

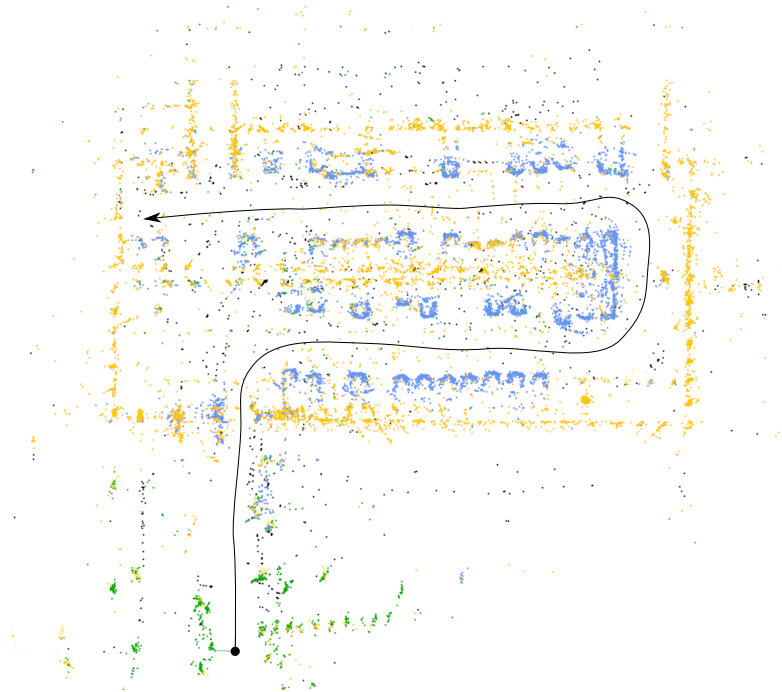


Figure 6.12: Illustration of the test scenario D as resulting semantic radar SLAM map (see Section 5.4.5), with sSNDT registration of Section 5.3.4 and Section 5.3.4.2, applied on the live inference of *RadarNet* of Section 4.5. Colors according the semantic classes in Table 3.2.

⁵ The map structure in the lower part and on the right hand side of the map represent reflections of dense vegetation hedges, but these are consistently semantically interpreted as *building*. The map structure at the top of the image, describes a stone-wall and is correctly classified as *building*.

The fourth scenario completes the potential parking scenarios with a multi-story car park scenario as a difficult radar scenario due to multi-path and clutter reflections of the steel construction. The scenario is illustrated in Figure 6.11 and the corresponding radar map is found in Figure 6.12. Starting outside, the trajectory covers an open-loop drive on the floor level, following a longer passage in an alley of parked vehicles and a parallel return. This scenario is chosen as example of significant environment changes by a changing count of parked vehicles. In the map evaluation later in this section, the static environment map is compared after exclusion of the potentially dynamic *vehicle* radar detections. Also *vegetation* and circular *pole* clusters are perceived and correctly classified at the start of the reference path.

6.5 Results of automated Parking Experiments

In the context of automated parking, the evaluation measures specifically emphasize the positioning accuracy. Besides the realization of a generally applicable radar-based automated parking functionality, the main evaluation focus is not to measure driving performance KPIs or driving comfort criteria for passengers. Of special interest is the positioning accuracy during the automated drive with respect to a reference path and the end-position accuracy in order to measure the semantic radar SLAM. Both accuracies describe the capabilities a solely radar-based autonomous parking functionality is able to realize.

As reference data, the vehicle position is recorded with on-board differential GPS sensors. Similar to the SLAM mapping comparison in Section 5, the wheel odometry and the SLAM trajectory can be compared with the differential GPS position by their time-stamp. This is referred to *intra-drive* measures. The odometry drift versus SLAM trajectory can be measured against the synchronized time-stamp position of the d-GPS signals of the same drive.

Table 6.1: Automated parking test drives (A-D) with test details (X.1 /X.2 /X.3) driven at 19./ 22./ 25.11.2021 different length, environmental content and respective Δz of 0.82 m, 0.79 m, 1.3 m, and 0.0 m.

| Test Scenario | Positioning Accuracy [cm] | | | | Length [m] | | # Map Points [-] | Environmental Content [%] | | | | | |
|---------------|---------------------------|----------|------------|---------|------------|---|------------------|---------------------------|---------|----------|------------|-------|--------|
| | End Position Dist | mean APE | median APE | std APE | Length | $\varnothing v \left[\frac{km}{h} \right]$ | | Artifact | Vehicle | Building | Vegetation | Pole | Person |
| A - Mapping | - | - | - | - | 182.12 | 13.7 | 29383 | 1.82 | 14.85 | 74.63 | 4.04 | 4.66 | 0.00 |
| 4xDrive A.1 | 17.25 | 26.04 | 22.21 | 18.8 | 183.44 | 3.51 | 28792 | 1.88 | 14.89 | 74.25 | 4.32 | 4.65 | 0.01 |
| 4xDrive A.2 | 11.88 | 19.71 | 13.25 | 21.73 | 182.23 | 3.74 | 28615 | 2.25 | 17.20 | 71.46 | 4.18 | 4.91 | 0.00 |
| 3xDrive A.3 | 15.89 | 18.66 | 14.33 | 20.57 | 184.25 | 3.53 | 26569 | 2.10 | 15.62 | 72.1 | 3.91 | 6.27 | 0.00 |
| B - Mapping | - | - | - | - | 165.66 | 13.5 | 21993 | 2.26 | 10.37 | 78.89 | 3.31 | 5.12 | 0.05 |
| 4xDrive B.1 | 34.36 | 36.55 | 31.78 | 25.78 | 167.79 | 3.78 | 24286 | 2.73 | 9.95 | 78.60 | 4.34 | 4.7 | 0.04 |
| 3xDrive B.2 | 47.47 | 28.75 | 21.72 | 23.7 | 164.73 | 4.02 | 19163 | 3.00 | 11.04 | 76.68 | 3.73 | 5.53 | 0.02 |
| 4xDrive B.3 | 20.21 | 32.34 | 20.89 | 32.82 | 167.73 | 3.8 | 26468 | 2.19 | 9.07 | 79.38 | 4.51 | 4.78 | 0.07 |
| C - Mapping | - | - | - | - | 168.14 | 14.2 | 13774 | 4.69 | 14.12 | 57.72 | 12.86 | 10.61 | 0.0 |
| 4xDrive C.1 | 18.2 | 32.71 | 26.83 | 22.95 | 168.25 | 3.90 | 14319 | 5.04 | 15.0 | 52.53 | 16.63 | 10.72 | 0.08 |
| 4xDrive C.2 | 26.18 | 29.71 | 27.48 | 20.15 | 167.65 | 3.71 | 13987 | 5.88 | 16.95 | 48.54 | 15.68 | 12.84 | 0.11 |
| 5xDrive C.3 | 28.45 | 34.97 | 33.6 | 22.5 | 168.64 | 3.71 | 14623 | 4.29 | 27.39 | 42.42 | 15.48 | 10.41 | 0.01 |
| AVERAGE | 24.43 | 28.83 | 23.5 | 23.22 | 172.75 | 3.73 | | | | | | | |

Different from the evaluation of the SLAM mapping, for the trained parking evaluation a comparison of multiple different driving experiments of the same autonomous parking maneuver is necessary. Referred to as *inter-drive* comparison, the comparison between different runs of the parking is addressed. The first manually driven maneuver serves as reference to be compared to subsequent automatically driven parking maneuvers.

Since the automated function does not replay driver inputs or drive specifics, e.g. ego-velocity during the mapping process, but performs an own planning and velocity actuation, the driving performance is not comparable between manual and automated parking. Also the maneuver time can not be compared due to a limited vehicle speed of 5 km/h . In this case, the accuracy of multiple automated drives is measured against the differential GPS coordinates of the manual reference drive. The trained parking follows a projected reference path, so the d-GPS position of the automated drive can be measured against the reference d-GPS path.

Comparing both d-GPS tracks from mapping and the automated driving delivers an upper limit estimation of the achievable accuracy. The resulting positioning system error $\epsilon_{\text{pos, System}}$ results from the combination of all system modules which are combined to the overall system

$$\epsilon_{\text{pos, System}} = \epsilon_{\text{pos, SLAM}} + \epsilon_{\text{pos, Map-Matching}} + \epsilon_{\text{pos, Planner}} \quad (6.1)$$

Since not all trained parking function modules, e.g. the planner and actuation module, are optimized for the overall system evaluation but still contribute to the positioning error terms, the achieved system error delivers an upper bound of the possible system accuracy. The path planner and motion controller are tested in open-loop without re-localization to achieve nearly ground truth quality and contribute low induced error component.

6.5.1 Vehicle Positioning Accuracy

The evaluation is based on the comparison of reference d-GPS tracks with the d-GPS tracks of the automatically driven trajectory of independent autonomous parking maneuvers. First, the general position deviation of the automated parking maneuvers is depicted in the top-view plots of Figure 6.13.

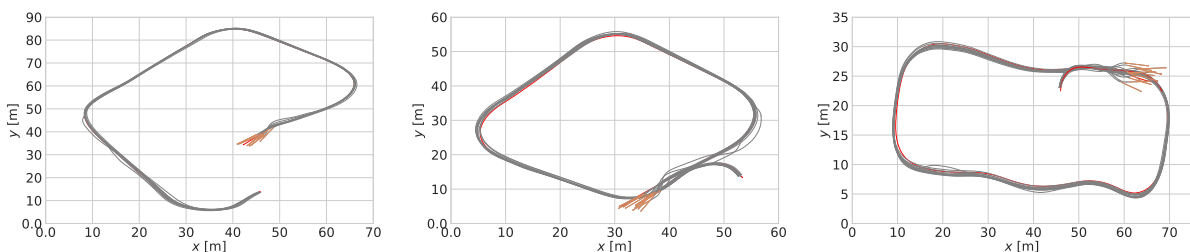


Figure 6.13: Top-view trajectories of the initialization (orange) from different start positions and the full automated drive of the test scenario A (left), scenario B (middle) and car park scenario C (right).

Starting from different start positions and vehicle orientation, the plots show the success to autonomously plan a path onto and to automatically follow the reference path to the final parking position. As the APE measure describes the metric deviation of the ego-vehicle to the reference path, a minimum deviation is aimed to be achieved. Starting from an initialization position, the automated maneuver manages to realize maximum position error of 1.5 *m* as rare peak deviation in the APE measures of Figure 6.14. The driving functionality reduces this peak-error and realizes an average position deviation below 0.5 *m* near 0.25 *m*, as depicted in the course of the APE in Figures 6.14- 6.17 and summarized as Average in Table 6.1.

The individual drives in Figure 6.14 reveal isolated APE peaks in some runs, throughout the automated maneuver. This de-positioning peaks result from a temporary deviation of the regular map-matching, resulting in a temporary erroneous-projection of the reference path to follow.

Considering Equation 6.1, the remaining source of error can be found in the SLAM term $\epsilon_{\text{pos, SLAM}}$. With a projected reference path from map-matching, the automated function follows potential projection errors. Hence, if this path to follow is projected with slight position or orientation deviation to the current environment, the resulting d-GPS position of the vehicle following this projected path deviates from the reference path coordinates - the APE increases. Inspecting Figure 6.14, before and after corner turns of the reference path, the realized APE tends to peak. This finding indicates a weakness of the large map-matching, converging to a map-registration which potentially includes slight orientation errors.

The peaking APE errors mainly occur close to turns and are reduced subsequently again on straighter parts of the trajectory. Considering the perceived environment during the automated drive, the mapping of turns add significant structural information to the structure of the maps. The map-registration architecture in Figure A.10 includes an area-based sub-sampling of the reference map before registration with the current map. Especially before turns scenes of the scenarios, the current perceived environment structure is not fully recognized, compare situations close before and after turns in Figure A.13 and Figure A.14. As a result, the sub-sampled reference map contains a complete environment structure, while the current map contains less structure, or only very sparse structures around corners. The effect of sensor sparsity and aspect ratio of the scenario also increases this effect. Only a fraction of the radar sensors *see* already around the corner, based on their FoV.

Being a distance based optimization, the map-registration fits the two map-point clouds and converges including a remaining orientation error. This potential orientation error is responsible for the re-projection of the reference path and a peaking APE, which is corrected after the turn yielding a reduction of the APE.

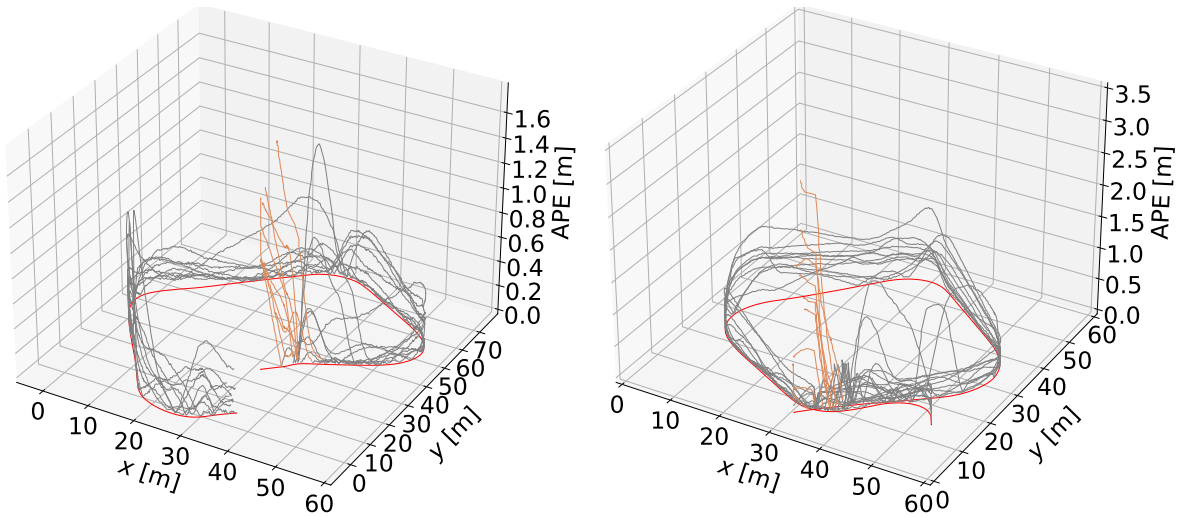


Figure 6.14: Illustration of the test scenario A (left) and test scenario B (right) with reference d-GPS paths (red), initialization drives (orange), and APE error (grey) of the independent automated drives. The APE error is displayed to check the amount of error along the reference path, not the real deviation coordinates of the automated maneuver.

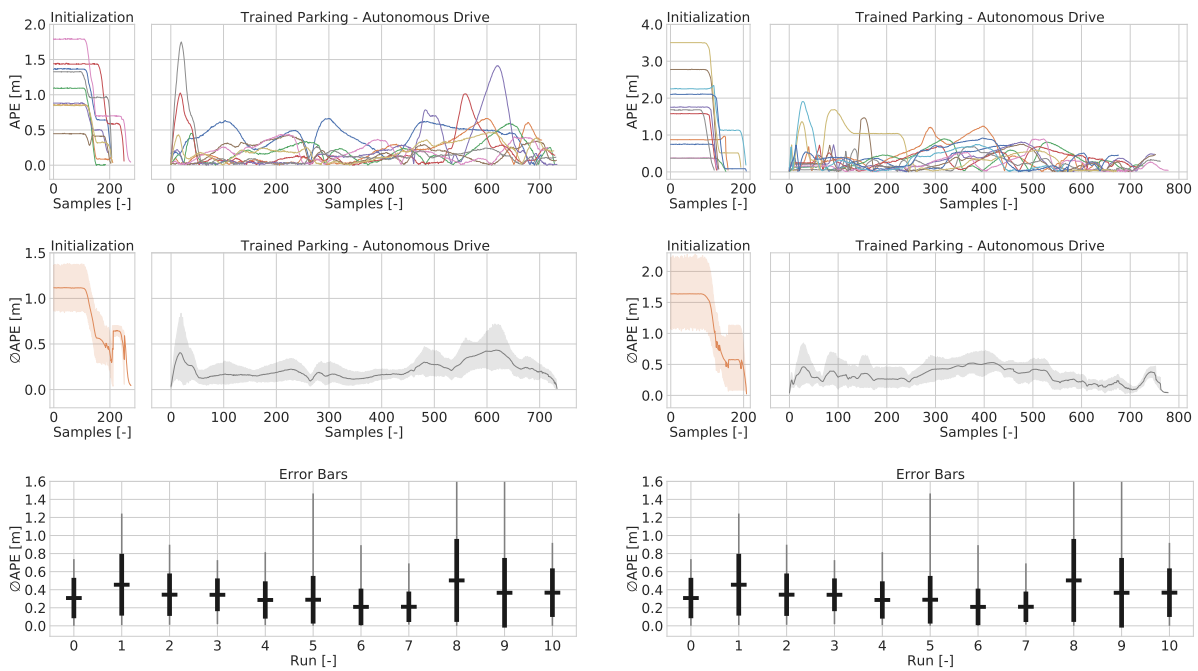


Figure 6.15: Illustration of the individual APE to the closest d-GPS reference of all automated drives (top) versus the averaged APE error (middle) and the error bar plots of independent automated drives (bottom) of test scenario A (left) and test scenario B (right).

Finding the disadvantages of the environment mapping in corner regions, the same corner regions are responsible to define the environment structure. After the passage of a turn, perceiving the turn with all sensors during the passage, the map-matching quality increases due to the significant structure of the environment corner area. This effect helps to reduce peaking accumulated errors of before mis-registered maps, yielding a more accurate reference path projection to follow after turns. This trade-off is responsible for the pre-turn APE peaks and the subsequent reduction of APE after the turns. The box-plots reveal the average APE value of single runs to be comparable over multiple runs, but the APE peaks to occur independently from run to run.

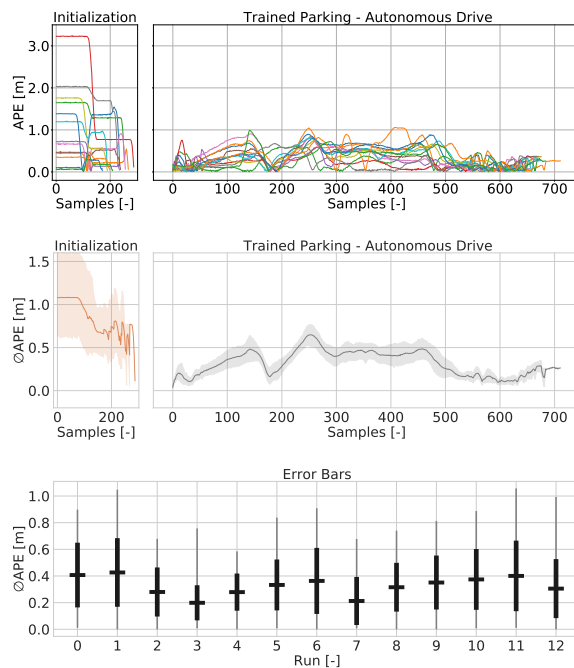


Figure 6.16: Illustration of the individual APE to the closest d-GPS reference of all automated drives (top) versus the averaged APE error (center) of the test scenario C. Error bar plots of 13 independent individual automated drives of the scenario C (bottom)

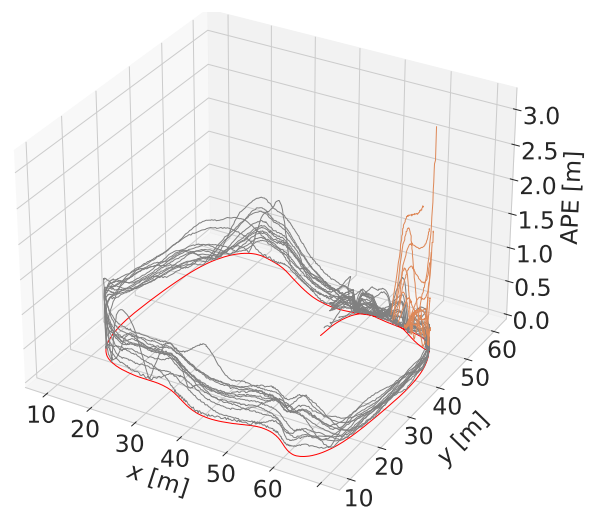


Figure 6.17: Illustration of the test scenario C with reference d-GPS paths (red), initialization drive (orange), and realized APE error (grey) as z-coordinate of the independent automated drives in two perspectives. The APE error is displayed to check the amount of error along the reference path, not the real deviation coordinates of the automated maneuver.

Initialization Position: The test maneuver is initialized at different starting positions but similar vehicle orientation. In order to test a realistic initialization, the automated parking maneuver is initialized at approximately the same starting position with orientation variation. See Figure 6.18, showing the straight initialization drive (orange) from different starting orientation with respect to the original reference (red). The realized automated parking maneuver path is depicted as gray path.

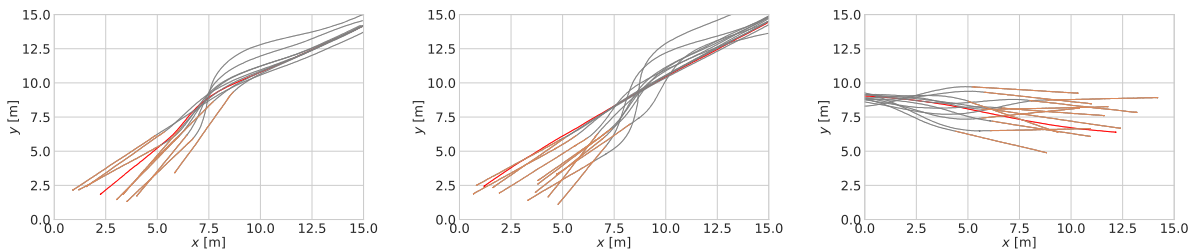


Figure 6.18: Illustration of the initialization and start position of the automated initialization drive of the test scenario A (left), scenario B (middle) and car park scenario C (right).

Final Park Position: The achieved end-position accuracy of the individual autonomous parking maneuvers is found in Figure 6.19, at the end of the APE curves. Over the tested scenarios, the average lateral end-position error compared to the red colored ground-truth reference trajectory in Figure 6.19 yields $\varnothing 0.15\text{cm}$ for test scenario A, $\varnothing 34.45\text{cm}$ for test scenario B and $\varnothing 24.28\text{cm}$ for test scenario C. Due to a structural planner deviation considering the parking maneuver end-position, the comparison of the lateral APE is chosen to be considered for the end position evaluation. The orientation error of the end position vanishes. The resulting orientation error of the test scenario C in Figure 6.19 results from the planner-dependent early stopping before the reference trajectory ends, explained in the following.

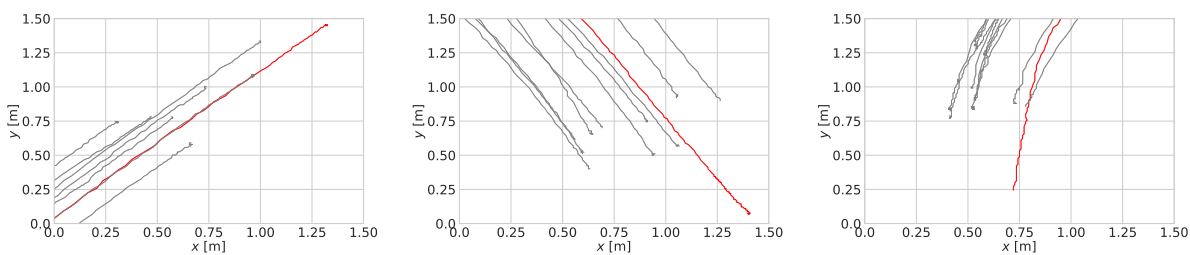


Figure 6.19: Illustration of the final position of the automated drive of the test scenario A (left), scenario B (middle) and car park scenario C (right). The red path represents the manual reference dGPS path, whereas the grey paths are from each different test drives.

Planner Dependency: Equation 6.1 of the overall positioning error contains the term $\epsilon_{\text{pos, Planner}}$ to describe both error terms of the planner as well as a re-localization error. The planner-dependent early stopping before reaching the reference trajectory end yields from a non-optimal parametrization of a lateral safety margin at the reference paths end. Figure 6.19 illustrates this safety margin as distance between the red reference path to follow and the grey realized paths, stopping consistently before reaching the end of the reference path. For the end position, this planner policy prevents to follow to the total end of the reference path. The parametrization results in an artificial stop within $\approx 64\text{cm}$ to hold a lateral safety margin. As a result, the last remaining $\approx 64\text{cm}$ of the reference path are not driven, derived from ideal simulation conditions to re-simulate the planner. To compare the resulting end position, this planner-induced lateral default distance is compensated for the automated parking maneuver error in Table 6.1. During the drive, the planner influence is neglectable, Figure 6.13 illustrates the global path following potential along the path.

Map-Registration Dependency: During initialization and subsequent relocalization only a smaller current radar map is perceived but map-matched with a much larger reference map. Resulting from growing current maps with increasing map details, the map-registration is found to be non-sensitive to rotational errors. Hence, based on this map-registration the reference path projection can shift or rotate, and therewith vehicle follow an erroneous reference path. Especially the rotational map-registration updates cause path deviations - visible as significant orientation changes in Figure 6.18 or similar path variations along the path in Figure 6.13. Depending on the planner and actuator parameter application, the vehicle is following a comparably drastic planner policy to laterally drive onto to the projected reference path. This *agressive* tuning depends on the planner parametrization and is found to result in slight overshoots of the ego-vehicle position with respect to the reference position. Consequently, especially for long and curvy automated parking maneuvers, especially the rotational map-registration accuracy is found to be critical.

6.5.2 Map Reproducibility Evaluation

Besides the position accuracy evaluation of d-GPS tracks, the resulting semantic radar environment maps can be evaluated to rate the mapping quality of the test environments as second semantic radar SLAM evaluation method. The following section describes an offline map-evaluation process to evaluate the radar mapping reliability and reproducibility of the static environment over independent runs. Since no ground-truth semantic radar map is available for general test scenarios, the semantic radar maps of the automated drives of all sequences (scenarios A - D) are compared to their corresponding reference map from the manual drive.

The radar detections of the semantic classes *person* and *vehicle* detections are excluded from the map comparison, in order to concentrate on the *static* stable environment and not measure dynamic objects. Hence, the remaining semantic radar SLAM map content for the following evaluation is reduced to *building*, *pole* and *vegetation* detections.

Evaluation Procedure: Since the semantic radar maps are model-free and do not contain any featured landmarks, the comparison of mapped parts of the global map is based on associated radar points of compared maps. A grid discretization is applied to calculate the distance per cell-centroid in order to avoid the map comparison metric to depend on the number of radar detections in associated grid areas. The association includes both, the spatial matching of radar detections and semantic correspondence of associated map-cells.

Figure 6.20 depicts the map pre-processing as *Static Map Extraction* step as first of the process steps. The semantic classes are neglected in this visualization and in Figure 6.21. Also, a radius outlier filter (minimum 5 neighbors in a range of 0.5 m) is applied to compare only significant structural radar map-regions.

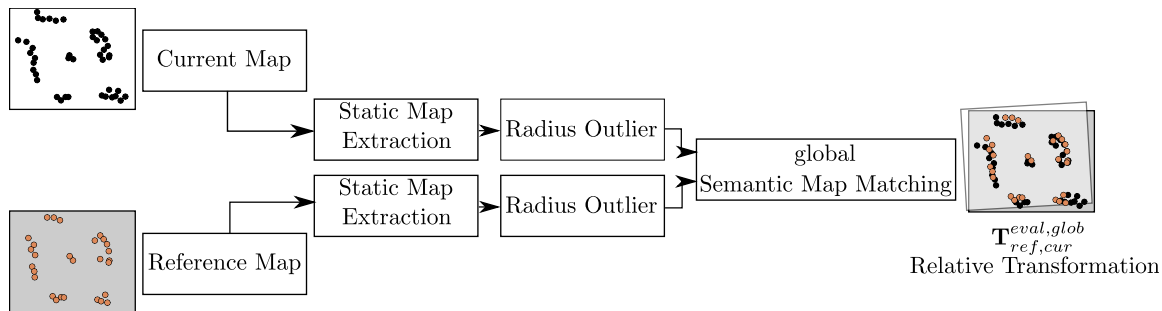


Figure 6.20: Architecture of the offline map evaluation to derive metrics describing the radar mapping quality.

After the map pre-processing, a first global matching is applied to achieve a general global map alignment. Given the global map registration, the overlapping map regions not necessarily match precisely as the non-overlapping parts of Figure 6.21 show. It is found in the nature of scattered radar detections, to yield especially rotational errors for map margins of widespread maps.

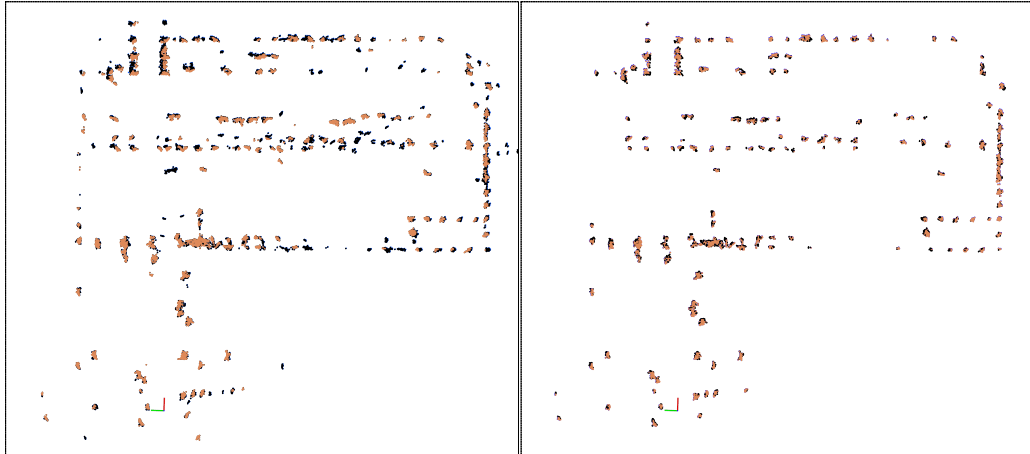


Figure 6.21: Illustration of global matching (left) of reference map (orange) with current map (black) compared to the local matching result (right) for test scenario D. Both maps are filtered by the radius outlier filter after registration. Figure overtaken of [MT6].

As mitigation of the global map-registration step, a divide and conquer approach helps to separate the global map into local tiles and compare these. Based on the first global alignment, both reference and current map are rasterized in local tiles of same size $[30\ m \times 30\ m]$ and registered again as second local tile matching. Figure 6.21 shows exemplary results of the local tile matching. Figure 6.22 illustrates a process chart, Figure 6.23 shows a map example to a local tile map. This local tiles registration refines the global-matching, resulting in a better map-overlay. With the local overlap, the map consistency regions and completeness regions can be quantified.

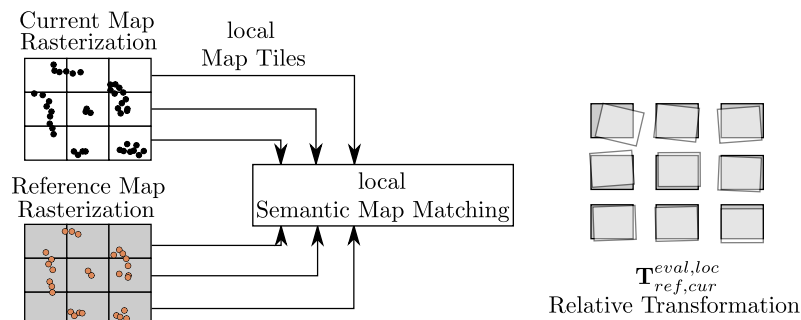


Figure 6.22: Illustration of local semantic matching for a precise map comparison.

The resulting *static* well registered points, a map *fusion* yields a possibility to improve map density and map completeness by overlying static and registered points to a combined map.

Map Evaluation Metrics: Comparing the distribution of the radar points as evaluation measure does neither indicate a mis-matching *content* of the mapped environment, nor emphasizes robust and significant map structures. Similarly impractical is the distance-based map registration fitness-score of a matching algorithm to compare a point cloud map

content or the representation of structures. Defined as a distance measure between reference and source points, the dependency of specifically associated point pairs and the point-count dependency is structural. A difference in the number of points or a higher count of false associations with outliers can drastically effect the fitness-score measure.

In contrast, a L2-distance norm as measure of spatially and semantically corresponding voxels is applied, avoiding any point-count or feature dependencies. Instead of relying on point-based distances, the 3D map-tiles are further discretized to smaller 3D entities of 50 cm voxel cubes. Corresponding to Equation 5.7, assuming a maximum range of 40 m the radar detection in each tile, the 2σ standard deviation yields a lateral deviation of 0.28 m , which is doubled to include also scattered detections along environment shapes robustly. Assuming the relevant environment shape to be captured in a 0.5 m cubic volume of scattered radar detections, the goal is to capture shapes of the environment objects in a voxel volume. A finer voxel size yields uninterpretable environment fragments, yielding potentially higher resolution of smaller environment details, but are less robust to associate between maps. Voxels are considered as empty or non-occupied with equal or less than 3 radar detections. Hence, in the map comparison, the general environment shape comparison is prioritized over a fine-grained detail comparison. Figure 6.23 visualizes the exemplary discretization scheme in of map *tiles* and *voxel grid*.

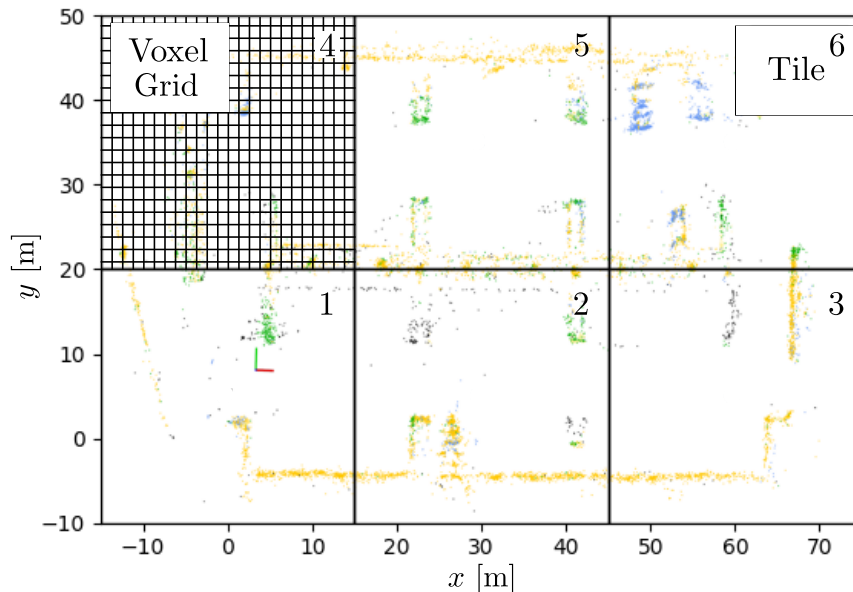


Figure 6.23: Exemplary illustration of the 30 m quadratic map *tiles* and map *grid cells* of test scenario D. The voxel grid of 0.5 m cubes is not displayed to scale.

Per voxel, the inlying radar points are summarized to a cell abstraction level: Each map voxel receives the majority vote of the radar detections' semantic labels y_{sem} as voxel label. All

points $p(x, y, z)|_{y_{sem, majority}}$ of the majority class in each voxel are summarized to a representative cluster position $p_{cluster, i}(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ per voxel i .

$$p_{cluster, i}(\tilde{x}, \tilde{y}, \tilde{z}) = \frac{1}{N_{y_{sem, majority}}} \sum_{i=1}^{N_{y_{sem, majority}}} (p_i(x_i, y_i, z_i)|_{y_{sem, majority}}) \quad (6.2)$$

For the general map comparison, the registration process yields overlying maps. To report a measure of matching percentage of the associated map voxels, the count of overlying, in both maps occupied (non-empty) voxels of the coinciding semantic majority label is reported.

$$c_{\text{voxel static}} = \frac{\text{\#spatio-semantic associated, occupied voxels excluding } vehicle \text{ and } person}{\text{\#all occupied voxels excluding } vehicle \text{ and } person} \quad (6.3)$$

The count $c_{\text{voxel static}}$ of corresponding voxels is assumed to yield a description of regions with coinciding *map content* of the captured environment structures of the two compared maps.

Based on the spatio-semantic voxel association, this subset of $N_v \in \mathbb{R}$ associated voxels are considered to compute the distance metric $d_{\text{voxel static}}$ between the point cluster center position $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$ of two associated voxels $i = 1, \dots, N_v$. This metric measures how similar the perceived map regions in the associated voxels are.

$$d_{\text{voxel static}, i} = \left\| p_{\text{cluster, map}, i}(\tilde{x}_{\text{map}, i}, \tilde{y}_{\text{map}, i}, \tilde{z}_{\text{map}, i}), p_{\text{cluster, ref}, i}(\tilde{x}_{\text{ref}, i}, \tilde{y}_{\text{ref}, i}, \tilde{z}_{\text{ref}, i}) \right\| \quad (6.4)$$

$$\emptyset d_{\text{voxel static}} = \frac{1}{N_v} \sum_{i=1}^{N_v} d_{\text{voxel static}, i} \quad (6.5)$$

By reporting the average mean distance $\emptyset d_{\text{voxel static}}$ of point clusters in voxels, the dependency of the total point-count of the two compared environment regions is avoided. It gets irrelevant by how many specific radar reflections the map content or the environment structure is represented.⁶ As point-count indifferent measure, the comparison is also free of specific landmark shapes but indicates differences of the mapped environment point cloud represented by the local reflection clusters.

Two identical maps would yield a 100% association percentage at a distance measure of 0.0cm, compared to which the probabilistic nature of radar sampling can be compared. This KPIs deliver two straight-forward interpretable metrics describing the fraction of matching map content, and to which metric accuracy two maps correspond.

Map Evaluation Results: The box plots of the average map metrics per test scenario in Figure 6.24 can be compared to the visualization of matching map regions in Figure 6.25 and the original semantic radar maps in Figure 6.26.

⁶ The radius outlier filter in Figure 6.20 removes noise and other single detections to compare only the plausible and relevant environment mapping.

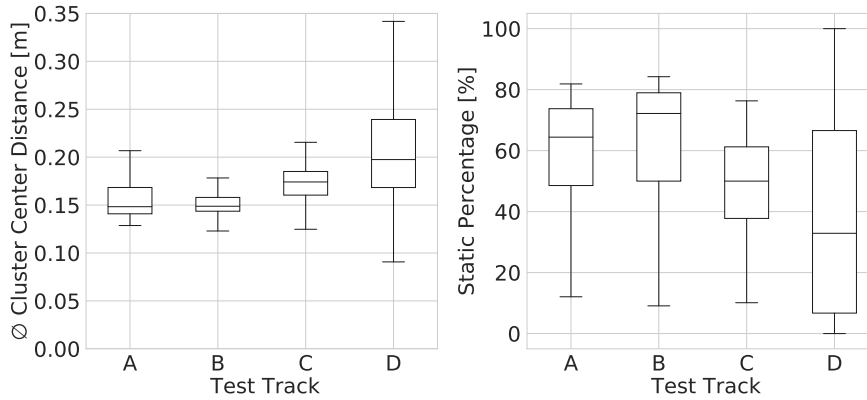


Figure 6.24: Averaged map evaluation metrics of all scenarios A-D averaged metrics over independent autonomous parking runs.

Summarizing over 43 runs (11 for scenario A, 11 for scenario B, 13 for scenario C, and 8 for scenario D), the automated drive evaluation in Table 6.1, as well as the map evaluation in Table 6.2, and Figure 6.24 reveals a general reproducibility of a maximum distance error of $16,25\text{ cm}$ and reliable environment mapping over the test sequences A-C and throughout the passed time and changing environment between independent test drives. The maximum map distance error of $16,25\text{ cm}$ yields as average value of the measured cluster center distances in the left plot of Figure 6.24.

The right plot of Figure 6.24 illustrates the measured static percentage of the semantic radar SLAM map, for the reduced static detections of *building*, *pole* and *vegetation*. Except for the car park scenario D, the right plot of Figure 6.24 shows matching percentages beyond 50% to a peak value of 72%, as measure of the spatio-semantic coinciding map-content, especially with respect to a consistent semantic label of the radar detections (*building*, *pole* and *vegetation*). Test scenario D plays a special role as most difficult test environment, since the parked cars in the car park are fully re-arranged and exchanged, resulting in a significantly differing environment and differing occlusions of the static environment. Without the semantic radar labels, it would be impossible to separate relevant static structures (also including the correctly labeled radar multi-path detections behind cars), from the potentially moving e.g. *vehicle* radar detections - still less a re-localization would be possible based on non-seperable, non-interpretable the point cloud data.

In contrast, the remaining static map content of $\approx 30\%$ of the semantic radar maps still delivers a robust data-basis to still achieve and realize the same parking accuracy as the other test drives⁷. As a result of the majority of the scenario being constituted by *vehicles*, compare Figure 6.26, the robustly sensed static environment serves as sufficient map-content to realize

⁷ The car park scenario could not be measured with d-GPS due to d-GPS signal non-availability in the multi-story car-park, but was successfully executed and visually inspected.

the autonomous parking maneuver. The percentage seems low, but only the overlap with reference map voxels are counted. Newly occupied cells in the current map, e.g. due to other aspect conditions, reduce the metric, even if this structure is relevant and static. The robustness of the positioning accuracy of the autonomous parking which is achieved with this level of spatio-semantic coinciding map-content.

This semantically and spatially matching map voxel distances yields an average mapping precision of 16.3 *cm*, for a matching average map content of $\approx 56\%$ over the test scenarios A-D. All matching regions, similar to e.g. a landmark matching comparison, are averaged over a whole map to this average deviation of 16.3 *cm*.

Table 6.2: SLAM map evaluation of the averaged test scenario metrics. Test scenario A-D with different length and environmental content.

| Test Scenario | Positioning Accuracy | | | | Path Metrics | | | | Map Points per Path [$\frac{\text{points}}{\text{m}}$] | Environmental Content | | | | | |
|---------------|-------------------------------------|-----------------------------------|-------------------------------|--------------------------------|------------------------------------|----------------|--|------------------|--|-----------------------|-------------|--------------|----------------|----------|------------|
| | \emptyset Start Position Dist [m] | \emptyset End Position Dist [m] | \emptyset Static Points [%] | \emptyset Dist of Center [m] | \emptyset Driven Path Length [m] | Δz [m] | $\emptyset v_{\text{autonomous}}$ [$\frac{\text{km}}{\text{h}}$] | # Map Points [-] | | Artifact [%] | Vehicle [%] | Building [%] | Vegetation [%] | Pole [%] | Person [%] |
| A | 1.14 | 0.1501 | 64 | 0.1487 | 183.33 | 0.82 | 3.59 | 152.69 | 27992 | 2.08 | 15.9 | 72.6 | 4.14 | 5.28 | 5.28 |
| B | 1.63 | 0.3401 | 73 | 0.1493 | 166.75 | 0.79 | 3.83 | 139.76 | 23305 | 2.64 | 9.9 | 78.22 | 4.19 | 5.0 | 5.0 |
| C | 1.04 | 0.2428 | 53 | 0.1625 | 168.18 | 1.3 | 3.77 | 85.09 | 14309 | 5.07 | 19.79 | 47.83 | 15.93 | 11.32 | 11.32 |
| D | - | - | 34 | 0.1986 | 135.53 | 0.0 | 4.0 | 158.72 | 26468 | 3.66 | 46.98 | 35.7 | 5.98 | 7.66 | 7.66 |

A visual inspection of the map evaluation KPIs is possible in Figure 6.25, indicating the specific regions of the map with respect to the initial reference map. Matching map grid cells are highlighted in green, newly occupied map grid cells are high-lighted in orange color.

It is found that mainly vegetation structures in test scenario C and car park building structure (of a metal fence) for test scenario D is highlighted in orange, showing a difference to the reference map. The total spatially and semantically coinciding static voxel percentage closely beyond $\approx 50\%$ in scenario C specifically results from the vegetation framing. The vegetation (hedge) scatters radar reflections causing non-corresponding semantic labels or spatial shapes. In contrast, the structurally distinct reflections of trees and light-poles at the *y* center of the plot, are consistently found as corresponding (green marked) voxels. This scenario is specifically depicted in order to illustrate an example of the lowest achieved spatially and semantically coinciding static voxel percentage. The other test scenario contain less vegetation,

which induces less reflection scatter and additionally yields increased semantic accordance closely to 80%, see Figure 6.24.

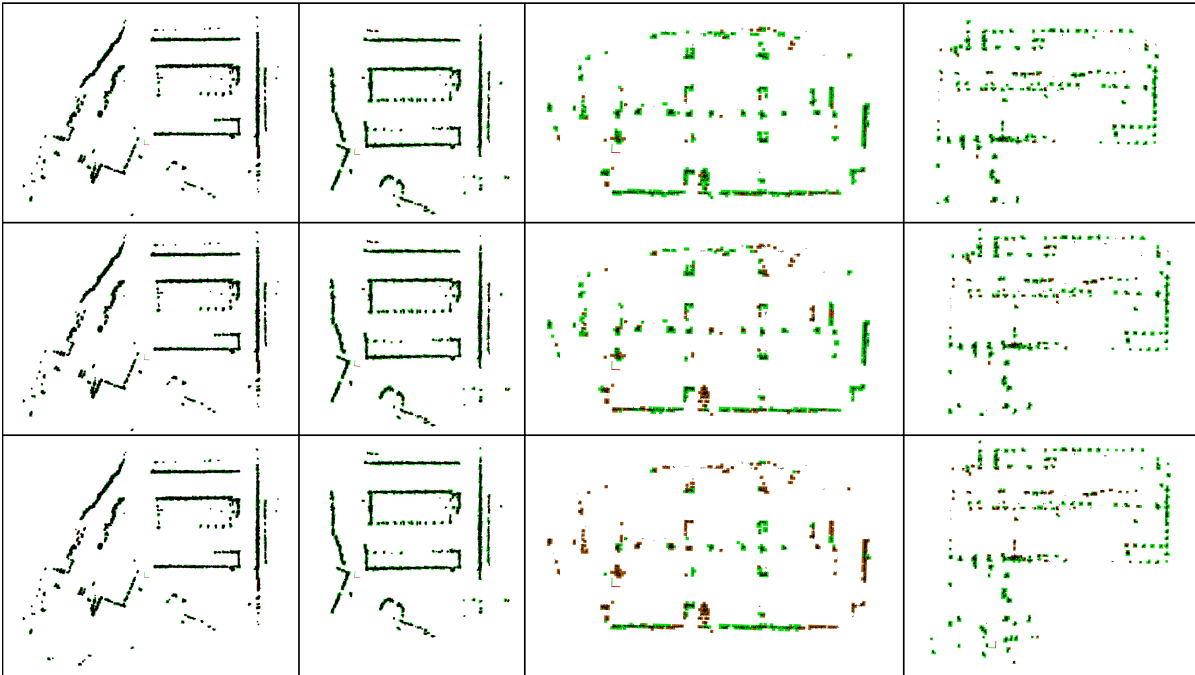


Figure 6.25: Illustration of matching cells (green) and newly occupied regions (orange) of three independent maps of automated parking runs (vertically) of the four test scenarios A-D (left to right).

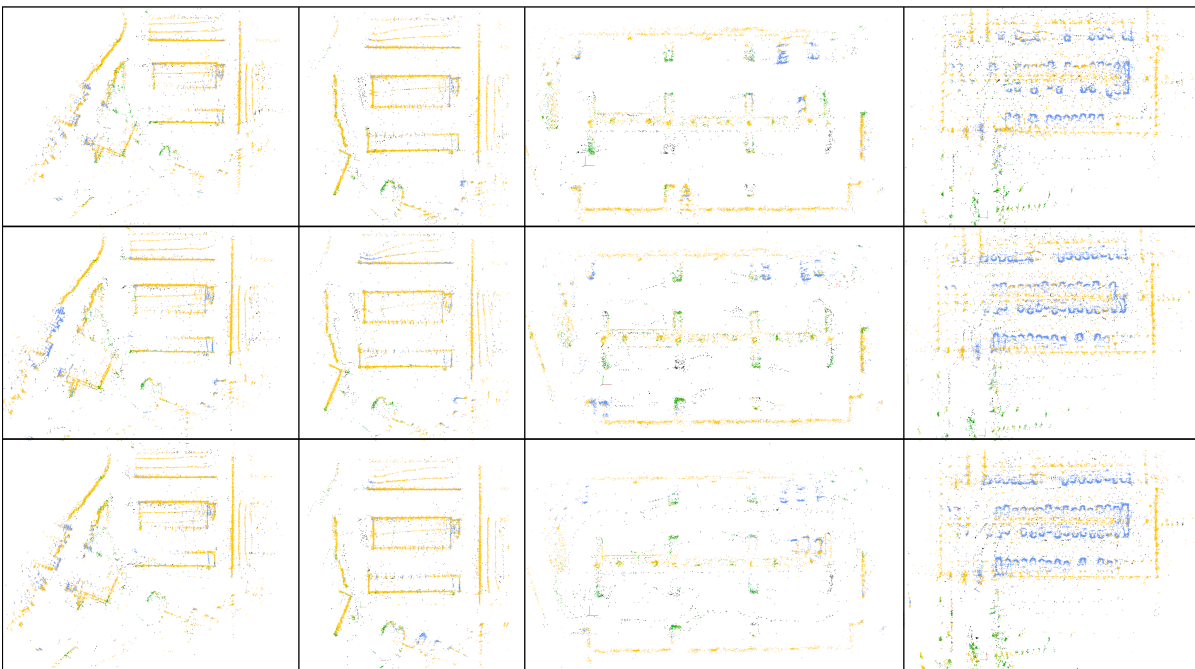


Figure 6.26: Three different autonomous drive semantic radar maps (vertically) of the four test scenarios A-D (left to right).⁸

Especially for the car park in scenario D, the removed vehicle detections and direct line-of-sight occlusion of the static environment structure behind parked cars reduce the coinciding static map structure significantly. Compare the remaining map structure (black) in Figure 6.25 with the original map, including original vehicle detections (blue) in Figure 6.26. As a result, the percentage of the remaining map content is lower, compared to the other test scenarios A-C.

For a general visual comparison of the semantic radar maps, the exemplary SLAM maps in Figure 6.26 can be inspected. The 3D inspection of the radar maps allows further details to be grasped: Poles, trees or other vertical structures are well-represented. And also the 3D vehicle shape or shape of building structures (wall-planes, etc.) can be detected by the 3D radar map.

Semantic Map Classification: The presented automated parking maneuver tests are performed over the course of late fall into winter. Due to the time frame of the thesis, summer conditions could not be tested as closed loop autonomous parking run. The changes of the environment maps are displayed in Figure 6.26, to compare exemplary semantic segmentation and semantic mapping on different days.

It can be found from Figure 6.26, that the semantic segmentation yields stable and consistent labels for the same map regions. The resulting radar maps of the environment are consistently use-able and also interchangeable as reference map, no mapping degradation based on environmental changes are measured. Even weather conditions of mist, rain or darkness in the dusk, are included in the test runs, but do not influence the semantic classification and map.

Consistent in the segmentation, the semantic labels nevertheless reveal optimization potential. Achieving a well determined de-noising capability to filter non-relevant noise from the map, the confusion matrix of the applied segmentation network, see Figure 4.25, reveals a mis-classification of *vegetation*, *person* and partly *vehicle* that can be found in the semantic radar maps accordingly. Especially test scenario C, consisting of dense hedges as environment framing structure, significantly shows the classification confusion of *vegetation*. The dense hedges can be found in Figure 6.9 are mis-classified as *building*, instead of *vegetation*. Nevertheless, the data association of the SLAM mapping and re-localization map-matching is not deprecated from this mis-classification, since the mis-classification seems systematic. But for a semantic environment evaluation of the scenario, or further processings including e.g. to remove *vegetation* from the reliable map structure, the mis-classified *vegetation* causes difficulties.

⁸ TPA test scenario dates (top to bottom), scenario A recorded at 19.11.2021, 22.11.2021 and 25.22.2021, scenario B recorded at 19.11.2021, 22.11.2021 and 25.22.2021, scenario C recorded at 19.11.2021, 22.11.2021 and 25.22.2021, and scenario D recorded at 19.11.2021, 22.11.2021 and 22.11.2021.

Changing Environment: Different weather conditions (day, night, rainy, and foggy) are met during the tested autonomous parking maneuvers, not causing any direct perceptible difficulties or any notable system degradation, in mapping or in reference path following accuracy. As expected, the radar sensor and semantic radar segmentation is unconditionally available at the same mapping and classification performance.

The evaluation over a sequence of days includes also drastic changes of the potentially dynamic *vehicle* objects. The car park sequence D illustrates the systems robustness in case of the significantly changing environment by fully re-arranged vehicles. Comparable environment changes are found during the tests on test scenario C. A vivid change of occupied and free parking lots with bypassing pedestrians is tested. Especially cars or trailers have been re-parked or re-moved, dynamic objects such as passing vehicles or crossing pedestrians have been met and cause a changing environment structure. The semantic radar SLAM enables to associate the relevant environment structure and enables a robust data association.

The vegetation changes only marginally during the testing time of Figure 6.26. Summer conditions of the same scenario are not tested.

For a detailed analysis of specific map regions, especially to fuse the information of multiple maps, the regional map KPIs per map tile are relevant, instead of the global map average. With this regional KPIs, a significant deviation of the average, especially of the matching voxel percentage indicates occlusions or other environment changes and map-fusion can be performed per tile or per semantic detail of map-tiles.

6.6 Section Conclusion

This section gives an evaluation of the integral parking functionality, compounded from different radar research contributions, broken down into system level contributions of Figure 1.4 and discussed in the previous Chapters 3-5. With this final section, the functional chain is discussed and presented as semantic radar-based autonomous parking functionality.

Tested with an arbitrary start position with average distance of $\varnothing 1.27m$ to the reference start position at a similar orientation, the radar-only trained parking system is initialized and the parking maneuver performed without further human interference.

Based only on the initially driven path and pre-recorded semantic radar SLAM map, the radar-only based assistance function achieves over an average passage of $172.75 m$ an median APE of $23.5 cm$ to the reference path, at a standard deviation of $23.22 cm$. The end parking position is reached with an mean lateral deviation of $24.43 cm$ to the reference path.⁹ With this finding, the applicability of a radar-based parking functionality experimentally and measurably answers the *system* research questions of Section 1.3.

Assuming a free space to drive along the initially mapped path, the presented system achieves unprecedented positioning accuracy, while relying only on the radar sensor set. The tested vehicle setup yields a positioning accuracy to be applicable for further development of a trained parking functionality under typical real-world parking conditions.

The evaluated closed-loop tests yields an upper bound accuracy evaluation serving as benchmark for further optimization. The positioning error is affected by all integrated module' errors, including the radar semantic segmentation CNN, the semantic Radar SLAM, the semantic radar map-matching and re-localization, plus the path planner, motion controller and actuator imprecision.

Based on the comparably low median positioning error of $23.5 cm$ of 43 autonomous parking maneuvers, the realized first semantic radar based autonomous parking functionality for an tested path length of $135 m - 209 m$ establishes a new application of radar sensors in point cloud processing and advanced driver assistance systems. The robust sensing modalities, low sensor-set integration effort and low sensor cost, allows to further process the development of automotive assistance functions similar to the presented semantic radar SLAM and autonomous parking basis.

Recent comparable localization on automotive radar in urban scenarios of Narula et al. [151] are outperformed with the presented method by 50%, considering the localization precision. The presented work also enriches the radar environment map applicability by a whole new semantic dimension towards radar map-interpretation, instance segmentation or map fusion.

⁹ A systematic planner-related error stops the vehicle $\approx 64 cm$ earlier to the reference path ends.

Based on the similarity of the semantic radar maps of different passages, the map-fusion enables accurate map-updates and map-complements of e.g. occlusions or environment changes. Also, the open question of semantic-reasoning can be tackled, to speed-up perceptive conclusions from the semantic map.

The application range of the presented system ranges from semantic radar SLAM based localization [Pat7] in an automated or autonomous driving context [Pat8], to a driver-feedback parameterized vehicle actuation policy [Pat6] or as augmented reality application [Pat9]

6.7 Section Outlook

Automated Parking - Collision Avoidance: As additional features for the realization of an automated parking, collision avoidance modules are necessary. To cope with potentially blocking objects and dynamic obstacles, the presented functionality should be enriched with an obstacle detection breaking mode. By the integration either of camera-based modules, in combination with radar object detection, an object or obstacle could be respected in the planner module.

Either avoiding a collision by re-planning the trajectory to drive in the range of a maximum deviation distance, the vehicle can be directed around a detected object. Applicable rather to static objects, dynamic objects such as pedestrians or bicycles or other passing vehicles, an emergency braking module offers higher safety.

Automated Parking - Enrichment with Parking Lot Lines: Derived from the use-case of automated parking in public or private parking lots, the final orientation, lateral or longitudinal offset could potentially be adapted by a fusion with parking line detection. Given the assumption of valid and trustworthy line detection based on the surround view cameras, the planning and vehicle positioning towards the end-position of the automated drive could be improved.

As humans, the parking performance of a parked car is (unconsciously but) naturally evaluated by a comparison of the framing parking lines as reference target position. An other common measure is to evaluate the orientation and distance with respect to neighbor vehicles. In order to optimize the end-position in such an environment with parking lines and potentially other ground markings, the planner and perception need a common mode to adapt the final pose to a situation adaptive optimum position and orientation. As such, TPA is capable to navigate the vehicle to its target position along a given path, but apart from the human quasi-ideal training data (initial mapping drive), the current scenario environment might change slightly and yield to an other optimum end-position.

SLAM Mapping - Surface Reconstruction: Based on the dense radar detections accumulated by the SLAM map, as post-processing a surface reconstruction is tested briefly. The advantage of enclosed volumes could be used to improve the semantic labels on objects, e.g. by a removal of overlapping labels. The semantically isolated point-set of vehicles in a parking garage is applied for an exemplary 3D reconstruction in Figure 6.27. The vehicle surfaces are generated by a point to mesh reconstruction, as proposed by Open3D of Zhou et al. [251]: First, the semantic radar detections of a single class is isolated, a radius outlier filter is applied (minimum 6 neighbors in a 0.2 m range), then a mesh is created with the alpha shape approach ($\alpha = 0.43$) of Edelsbrunner et al. [60]. Allowing object segmentation as map post-processing, the processing of bounding box detection and instance clustering can be transferred to the semantic radar maps, or class-specific sub-sets.



Figure 6.27: Illustration of an exemplary surface reconstruction of the isolated vehicle detections in a car park from two different semantic radar SLAM maps. The bumper shells are prominently reconstructable for the passed vehicle rows.

7 THESIS CONCLUSION

Starting out from the ultimate goal to develop a novel purely radar based ADAS functionality, to be integrated and tested in a real-world vehicle, this thesis presents the novel theoretical foundation of semantic radar signal processing, the radar-based semantic environment modeling, to a full vehicle actuation for autonomous parking.

As enabler, an automated semantic radar labeling pipeline [SI1, SI2], a novel semantic radar data set of the static environment is built. As new benchmark of direct semantic radar segmentation of especially the static environment, the proposed semantic segmentation architecture *RadarNet* achieves 28.97% mIoU on six classes (Discrimination of *clutter* to de-noise the radar data, and to determine *buildings, vehicles, vegetation, persons* and *poles*) [SI3].

This segmentation model is applied to extend a SLAM formulation to specifically comply with radar data, outperforming other radar localization methods and enabling a new dimension of *semantic radar SLAM* perception [SI4]. A specially radar-adapted Graph-SLAM front-end assembles the spatio-temporally pre-filtered semantic radar detection point clouds to yield the novelty of a consistent metric semantic radar map.

Showcasing the full potential of the proposed segmentation and SLAM, this thesis evaluates the benefits of a complete signal processing development from raw sensor perception to full autonomous vehicle control. Integrated into a real-world vehicle with a given trajectory planner and controller, a solely radar-based autonomous parking functionality is developed, built on-top of the theoretical segmentation and SLAM findings. Outperforming other radar-map localization results by large margin of $\approx 50\%$, the real-world implemented autonomous parking maneuver achieves a map consistency of $\approx 56\%$ along a $\varnothing 165\text{ m}$ long mapping path in a changing environment at a total map deviation of $\varnothing 0.163\text{ m}$. The autonomous radar-only parking maneuver over an average length of $\varnothing 135\text{ m}$ yields an average of $\varnothing 0.23\text{ m}$ lateral distance of a reference trajectory. To the authors best knowledge, a comparable radar-based holistic function concept for a full autonomous parking has not been published yet.

For the research questions of this thesis, the findings are presented in the specific sections, as well as subject specific-outlooks. The main contribution can be defined as the novel real-time capable radar segmentation network *RadarNet*, the semantic radar SLAM *SeRaLoc*, and the combined application in a autonomous, radar-only based *trained parking functionality* as novel advanced radar ADAS. With the empirical testing of the *trained parking functionality*, its real-world applicability is evaluated.

7.1 Thesis Outlook

The findings of a radar-only based potential driving function suggest, that advanced semantic radar processing based on learning approaches will yield a new generation of semantic radar perception to potentially substitute or support other sensors in the future.

The lack of publicly available large-scale automotive radar data sets still is a major hurdle to advance the potential of learning based radar processing. But, the proposed automated radar labeling framework *SeRaLF* attempts to provide a cross-sensor solution to generate large-scale labeled data sets of radar data. In addition, the presented *RadarNet* semantic segmentation directly facilitates the direct generation of larger data sets, which can be refined by the *SeRaLF* framework. The research field of radar segmentation is expected to gain more attention.

The experimental automated parking functionality showcases a glimpse towards the future of radar-based applications in the automotive context. It remains open to research, how systems of currently dominant camera-systems can be boosted by a radar fusion, by optimized sensor set designs [Pat10] or in company with improved sensors in general, e.g. distributed camera lenses [Pat11]. Exemplary, the camera blindness-compensating radar is proven in this thesis to be capable already to be applicable for vehicle localization and automated driving even without a HD-map as reference. With future radar sensors delivering a denser and more accurate radar point cloud, the potential of this sensor type is expected to play an essential role for further driver assistance systems directly or as redundant fallback in case of visual sensor blindness.

A APPENDIX

A.1 Alternative Segmentation Approaches

As extension to the discussed approaches of semantic segmentation in Section 4.2, existing radar segmentation approaches also include and build on sensor fusion. Fusion approaches of multiple sensors and potentially different pre-trained expert networks, introduce system complexity, the necessity of a specific sensor set and mostly require large memory and a special data. Hence, these fusion approaches are not further discussed in this thesis but denoted briefly as existing works:

Choi and Kim [44] fuses raw radar data with the depth information of a RGB-D camera and try to validate the a-priori known motion ground-truth data of objects. Given the dynamic objects' trajectories, the authors prove that the object motion can be applied to separate object instances. These findings are limited to an offline data set. Steyer et al. [206] suggest the fusion of radar point clouds with dense LiDAR point clouds to estimate object bounding boxes. Similarly, Nobis et al. [153] suggest neural sensor fusion of radar point cloud and camera image automatically by a CNN classifier to distinguish 7 classes of VRUs. A similar network structure is proposed for RSS-Net [110], fusing a separate semantically segmented camera image with the radar point cloud, to classify from a set of seven classes (pole-like, bike-like, vegetation, construction, pedestrian, vehicle and empty). Meyer and Kuschik [143] propose the similar fusion, but detect 3D bounding boxes of vehicles. The proposed CNN intakes the 3D spatial information of the radar point cloud with magnitude and an rgb image. From both data sources, feature extractors are combined to region proposals and fused with fully connected layers.

A.2 Density Plots per Semantic Class

In the following plots, the original data distribution is displayed in a range of $[0, 50]$ m for the classes *building*, *vehicle*, *vegetation*, *person* and *pole*. For the summary over all classes in Figure A.1, and the *clutter/ noise* class, Figure A.7 shows the range of $[0, 100]$ m .

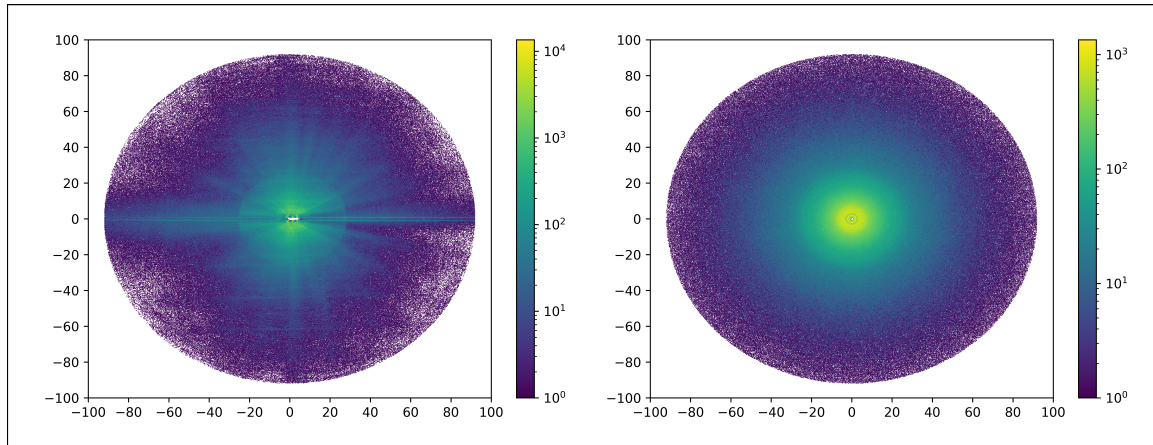


Figure A.1: Original spatial distribution of all radar point classes (left) compared to augmented by rotation and coordinate flipping (right).

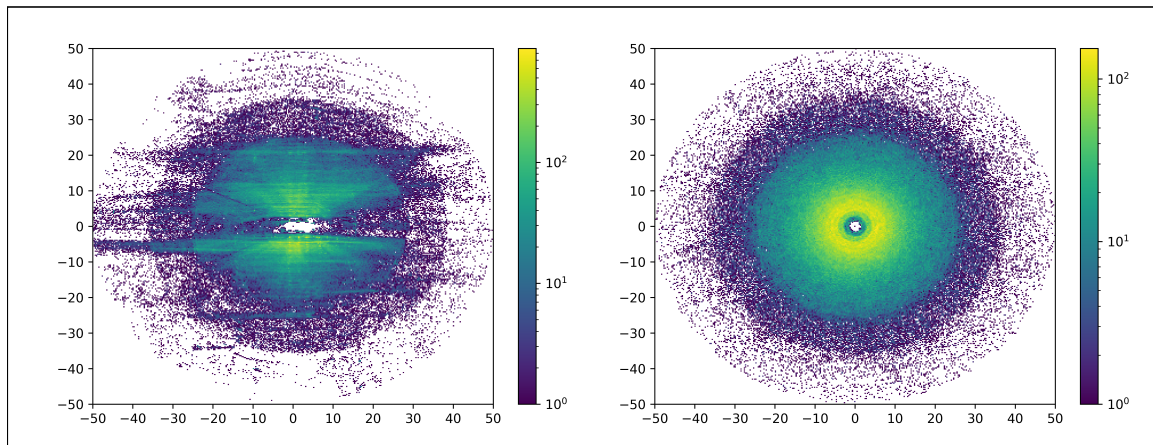


Figure A.2: Original spatial distribution of *building* radar point class (left) compared to augmented by rotation and coordinate flipping (right).

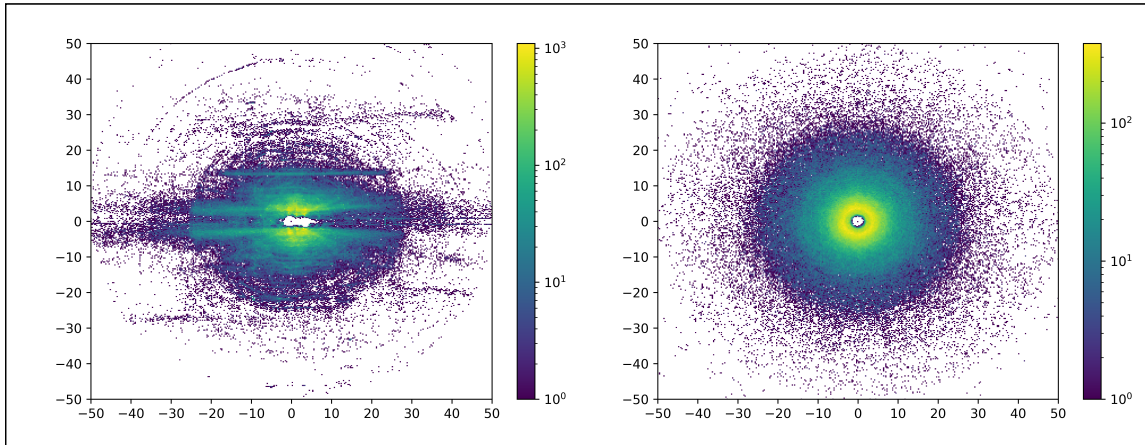


Figure A.3: Original spatial distribution of *vehicle* radar point class (left) compared to augmented by rotation and coordinate flipping (right).

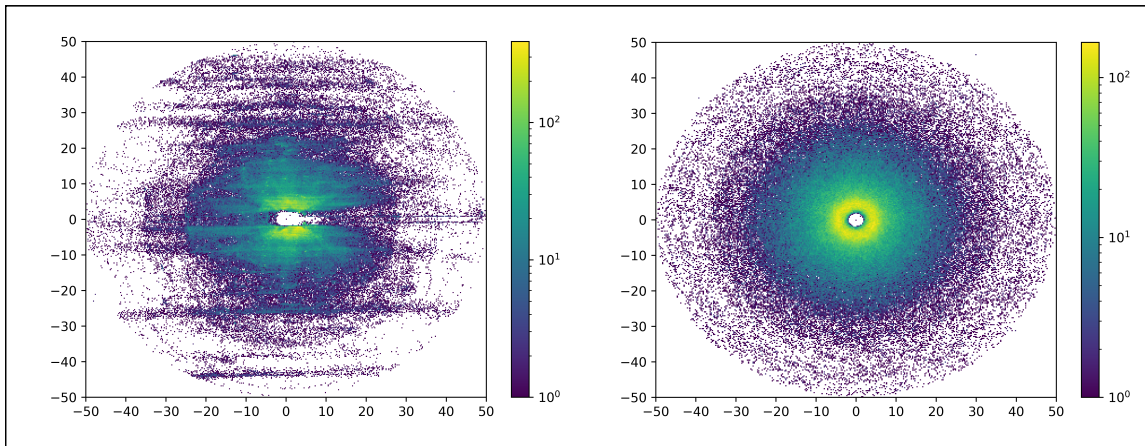


Figure A.4: Original spatial distribution of *vegetation* radar point class (left) compared to augmented by rotation and coordinate flipping (right).

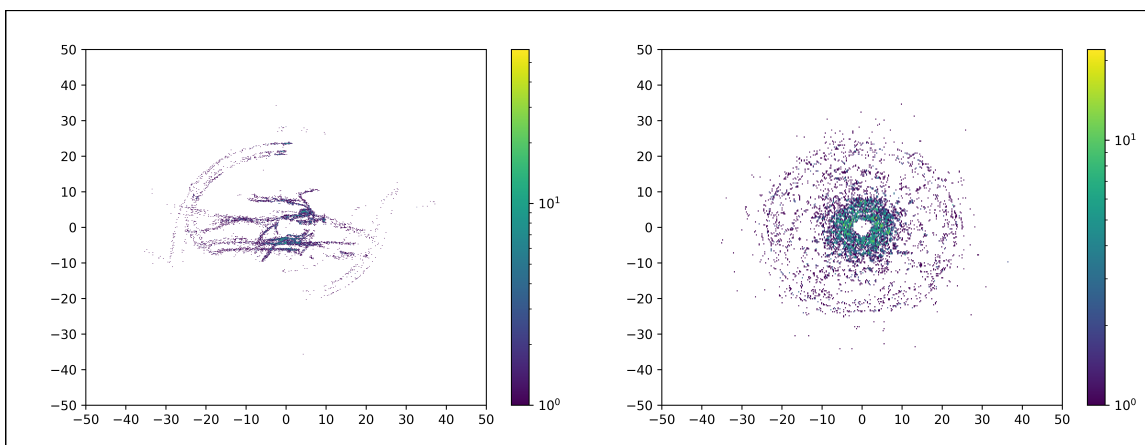


Figure A.5: Original spatial distribution of *person* radar point class (left) compared to augmented by rotation and coordinate flipping (right).

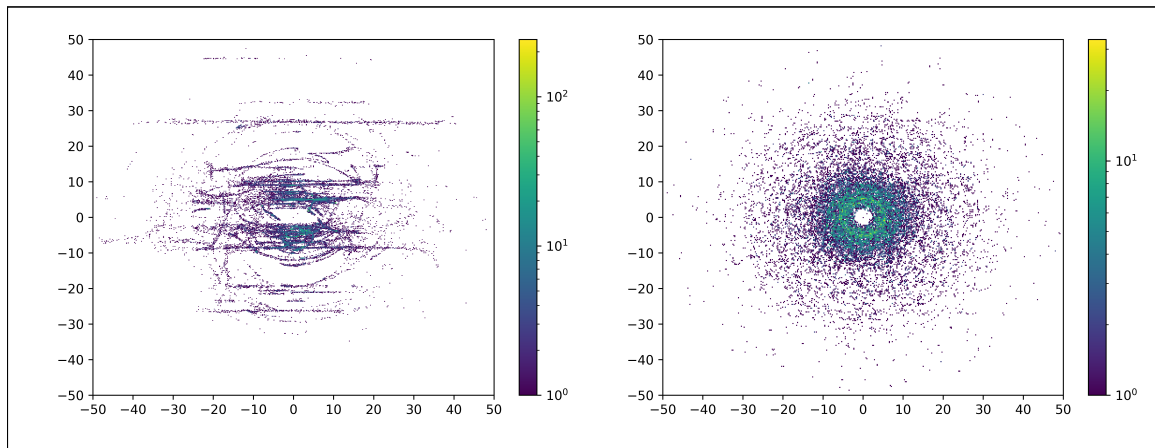


Figure A.6: Original spatial distribution of *pole* radar point class (left) compared to augmented by rotation and coordinate flipping (right).

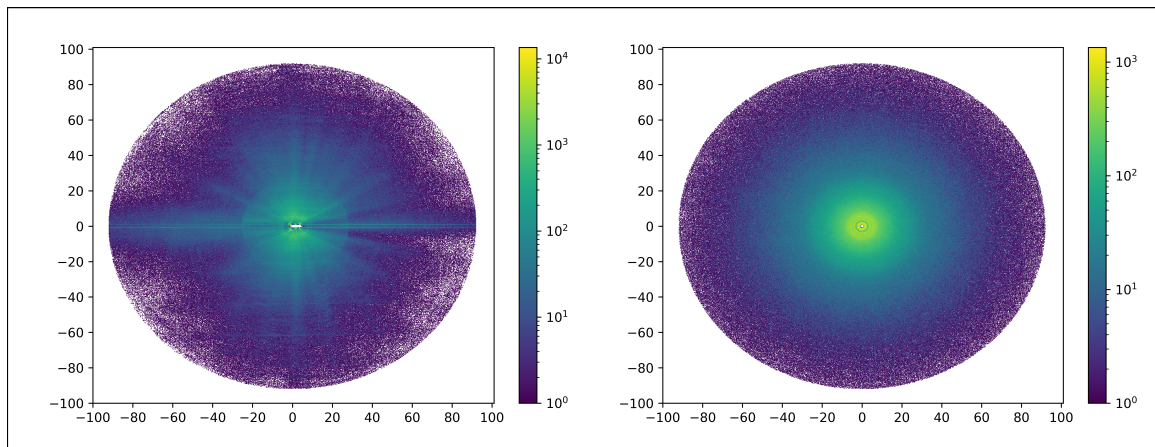


Figure A.7: Original spatial distribution of *clutter/ noise* radar point class (left) compared to augmented by rotation and coordinate flipping (right).

A.3 Semantic Radar SLAM Map Postprocessing

The final semantic radar maps of the radar SLAM still include scattered radar detections or *outliers* and the maps are not checked to contain consistent labels per local structure. In perspective to process the maps for further applications, e.g. for a free space estimation, a map fusion, or for semantic instance segmentation, the maps benefit from a post-processing.

Label Morphing: *Clutter* detections, as "container" for semantically labeled clutter can still be contained in the SLAM map \mathbf{M}_{SLAM} , if the detections pass the pre-filtering and the temporal filter of the SLAM pipeline. Semantically labeled as *clutter*, but still contained in the map \mathbf{M}_{SLAM} , these points can be assumed to be semantically labeled incorrectly by the radar semantic segmentation CNN. Hence, an online label morphing algorithm is implemented, to process a running window of SLAM nodes, similar to a graph-tail lagging running window sub-map, while the SLAM nodes at the latest graph-tail are unchanged. The label-morphing updates only the semantic labeling of SLAM mapped initially labeled *clutter* according to the semantic labels of the local neighborhood. The window of the considered SLAM Map nodes starts in a node offset of $d_{x,\min} = 15$ from the latest SLAM node to the farthest $d_{x,\max} = 30$ SLAM node.

All the semantic radar SLAM maps shown in this thesis are post-processed with the Algorithm 3.

Algorithm 3 Applied SLAM label morphing in a lagging window of 15 SLAM nodes.

Require: $\mathcal{P}_{\text{SLAM,radar}}(t)$, \mathbf{M}_{SLAM} , $d_{x,\max}$, $d_{x,\min}$
for $C(p_{\text{SLAM},i}) = \text{unknown}$ **do**
 $C(p_{\text{SLAM},i}) \leftarrow \text{K-NN}((\mathbf{M}_{\text{SLAM}}|d_{x,\min}, d_{x,\max}), p_{\text{SLAM},i})$
end for

A.4 Trained Parking System Integration

For the closed-loop real-world testing of the automated parking functionality, the introduced modules of this thesis are integrated and tested in a vehicle: The 3D semantic radar segmentation *RadarNet*, the *semantic radar SLAM* for mapping, a *semantic radar relocalization*, together with a *trajectory planning* and *vehicle actuation* module.

To perform the closed-loop automated parking tests, the following software modules are additionally implemented.

1. **Re-localization Approach:** To find the position of the ego-vehicle in a reference map, this module compares map extracts and finds the best match. Starting from an initial position guess at $(x_{\text{init}} = 5, y_{\text{init}} = 0)$ map matching is performed in a constant rate of every $N_{\text{map, update}}$ map update cycles.
2. **Vehicle Interface:** A real-time controller platform is integrated as ROS to CAN interface, realizing the communication between CarPC \leftrightarrow embedded platform \leftrightarrow vehicle.
3. **Vehicle Actuation Pipeline:** A motion planner module finds the path to follow the reference path, which is translated by a controller to vehicle actuation signals (acceleration a and steering angle δ).

From the list above, the relocalization registration re-uses the semantic radar registration of Section 5.3.4, but applied in larger map regions.

The second module, as interface between the compute platform and the real-time controller platform, is specifically implemented for the vehicle setup.

The remaining third module of planner and actuation, are not part of this thesis work and treated as given modules. The trajectory planner is designed by Lukas Köhrer from Forschungszentrum Informatik FZI ¹. As part of joint supervision of the Master thesis of Fabian Bischoff [MT4], the planner and the actuation controller are integrated in the test vehicle and treated in this work as given modules.

¹ www.fzi.de

A.5 Trained Parking Initialization Drive

To start the trained parking functionality, a short *initialization drive* is necessary to map the current environment and register with a reference map. This initialization process is described in the following.

Due to the sparsity of radar point clouds, it is found fragile to match single radar point clouds directly onto a map. Improved localization robustness is reached when a small environment map is recorded and compared to the reference radar map of the scenario.

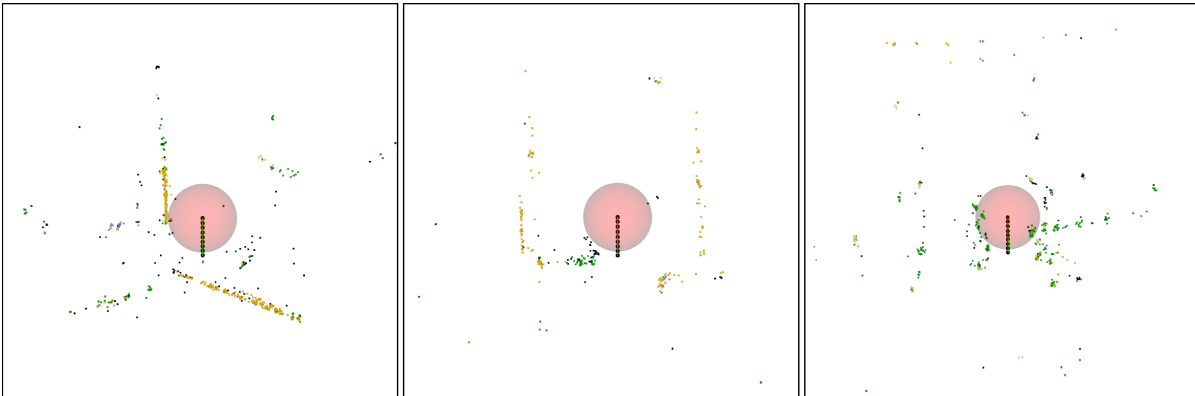


Figure A.8: Illustration of the automated initialization drive resulting semantic radar SLAM map for the test scenarios A and B (left), scenario C (center) and car park scenario D (right).

Hence, the automated parking function requires an automated drive to generate this initial environment map. As most simple automated drive, a straight path of 5 m is driven automatically after the parking function initialization to generate a current environment map. Based on this environment map, the map matching with the reference map can be computed.

From a functional point of view, the initialization position of the automated parking functionality is required to be positioned in close proximity to the original start position of the reference mapping. Since a manually driven position will never match the exact initial start position, the system needs to cope with variable start positions. The assumption is formulated, that the starting position of the autonomous parking can be located in a search range of 10 meters from the reference start point ($x_{\text{ref, start}} = 0, y_{\text{ref, start}} = 0$), with the vehicle orientation pointing towards the reference driving direction.

After the initialization drive, the re-localization assumes a position guess ($x_{\text{init}} = 5.0, y_{\text{init}} = 0.0$) and registers the current map so the reference map. This initialization drive does not

respect the reference driving path in any way, but only needs to generate a first environment map for the map-matching.²

The test drives are initialized from an arbitrary initialization position, in 5 meters range around the reference mapping start, pointing in a similar driving direction. The semantic radar maps in Figure A.8 illustrate the initial maps, resulting from the automated initialization drive. The 5 m straight initialization drive is visible by the driven path, depicted as black graph-SLAM nodes, with the current semantic radar environment map. This sparse current initialization radar map is map-matched with the reference map of the scenario to register the current vehicle pose in the reference map, as discussed in the next section.

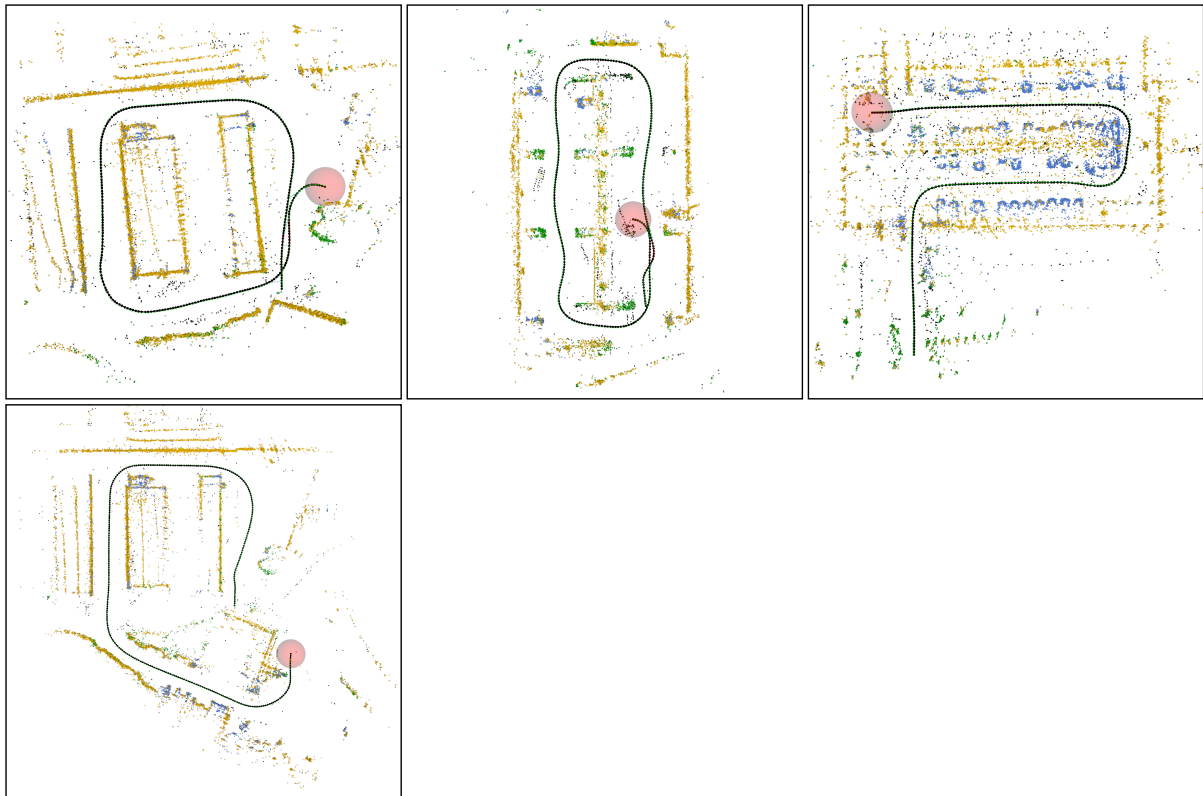


Figure A.9: Illustration of the total semantic radar SLAM map for the test scenarios A and B (left), scenario C (center) and car park scenario D (right).

² A more elaborate vehicle integration scheme could eradicate this initial drive, e.g. by implementing a continuous initialization environment mapping as background functionality, allowing to immediately supply a current map.

A.6 Trained Parking Map-Matching

Not only once for the initialization of the trained parking functionality, but recurrently with a growing current map, the two maps of the current perception and the initially mapped reference map are registered to find their relative transformation $\mathbf{T}_{ref,cur}$. The relative transformation $\mathbf{T}_{ref,cur}$ between the maps is used to project the initially mapped reference map into the current map coordinate system.

The interplay of current map matching for the map registration to the reference path projection procedure is illustrated in Figure A.10.

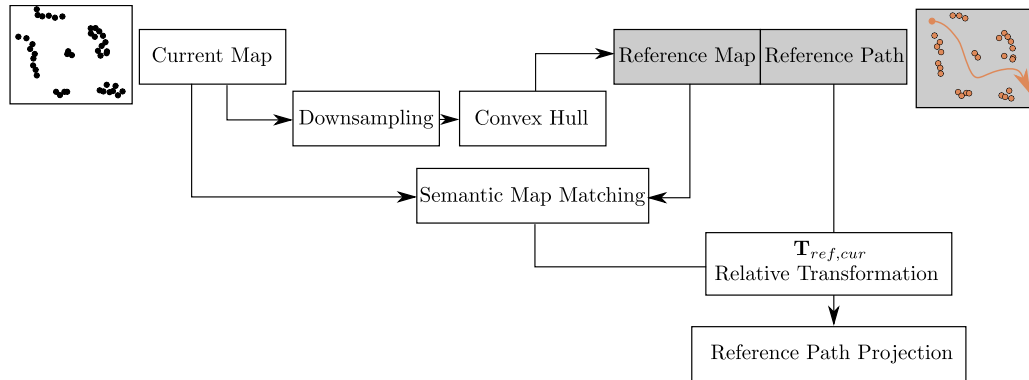


Figure A.10: Illustration of the automated re-localization process, showing the submap extraction of the reference map based on the convex hull of the current map, submap alignment and relative transformation.

The reference map from a prior training passage is therefore denoted as global initial reference map $\mathbf{M}_{glob,init}$. The current SLAM map, now created at revisiting the scenario and driving the automated parking maneuver, is referred to as current map \mathbf{M}_{curr} .

Based on the initialization position assumption (x_{init}, y_{init}) , the size of the current map \mathbf{M}_{curr} is down-sampled and serves to calculate the bounding convex hull of the map area size. Based on the initial relative transformation guess, the initial reference map $\mathbf{M}_{glob,init}$ is masked to a sub-map \mathbf{SM}_{curr} of the same size as the convex hull of \mathbf{M}_{curr} . Figure A.11 illustrates an exemplary bounding convex hull and the corresponding extracted area to extract from the reference map around the initial guess.



Figure A.11: Illustration of the automated initialization drive resulting semantic radar SLAM map (left), with respect to the reference map (right) of the test scenario B. The sub-sampled and convex-hull extract is depicted as dashed line.

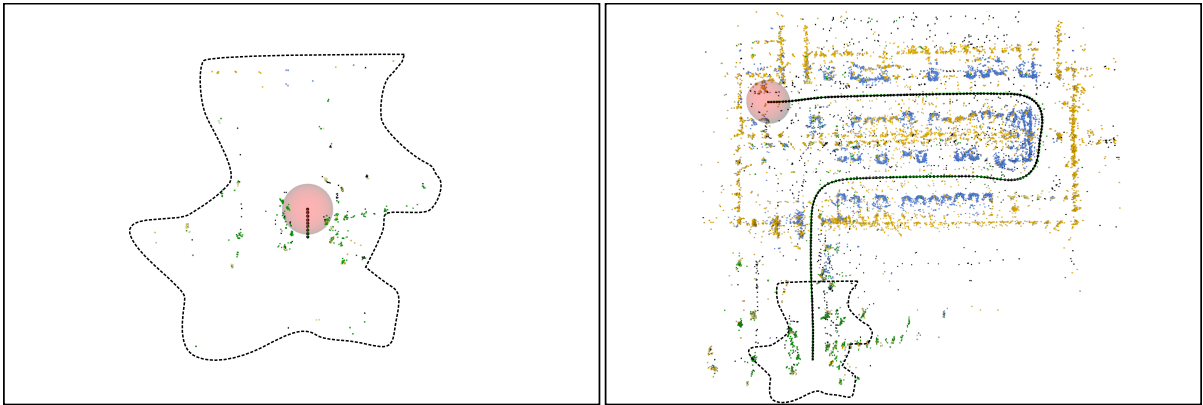


Figure A.12: Illustration of the automated initialization drive resulting semantic radar SLAM map (left), with respect to the reference map (right) of the test scenario D. The sub-sampled and convex-hull extract is depicted as dashed line.

Analogously to the semantic scan-matching in Section 5.3.4, the sub-map point clouds are compared with respect to their spatial and semantic attributes by a semantic NDT matching.

From the registration of both point clouds, the relative transformation $\mathbf{T}_{\text{ref, cur}}$ between both point clouds is given as rotation R_{map} and translation T_{map} and can be used to project the maps in overlay.

In order to avoid potential local registration minima and misaligned maps, the initial parameter guess is varied (translation and rotation) and tested in parallel.³

³ Real-time requirements are not necessary for the calculation step of re-localization. The vehicle is considered to be at standstill during this function call.

Hence, the relative rotation R_{map} and translation T_{map} of the map registration describes the spatial relation of both maps. M_{curr} and $M_{\text{glob, init}}$ - both maps are generally constructed with $(0, 0, 0)$ as their coordinate system origin.

Reference Path from Map-Matching: Besides the best matching overlay of the maps, the projection of the initially driven SLAM reference path into the current SLAM map coordinate system can be performed

$$\begin{bmatrix} x_{\text{ref}} \\ y_{\text{ref}} \\ z_{\text{ref}} \end{bmatrix}_{\text{cur}} = \mathbf{T}_{\text{ref, cur}} \cdot \mathbf{T}_{\text{ref}}. \quad (\text{A.1})$$

Together with the current ego-position coordinates, the projected reference path points $(x_{\text{ref}}, y_{\text{ref}})$ are sent to the planner module.

Live Reference Path Updates: Based on the small initial sub-maps extract, the initial map-matching and registration is responsible to converge to a first correct transformation $\mathbf{T}_{\text{ref, cur}}$. Performing the map-matching not only one initial time at system start, but in a frequent update rate, it is possible to take the growing current map into account and increase the size of the registered sub-maps. By increasing the current sub-map size according to the automated driven path and environment map, larger map parts are registered, which yields higher map-matching accuracy, map matching robustness and stable registration results. Increasing the map-matches yield robust and more accurate results, increasing also the accuracy of the reference path projection to follow.

An example of a growing current map during an automated parking maneuver can be found in Figure A.13-A.14.

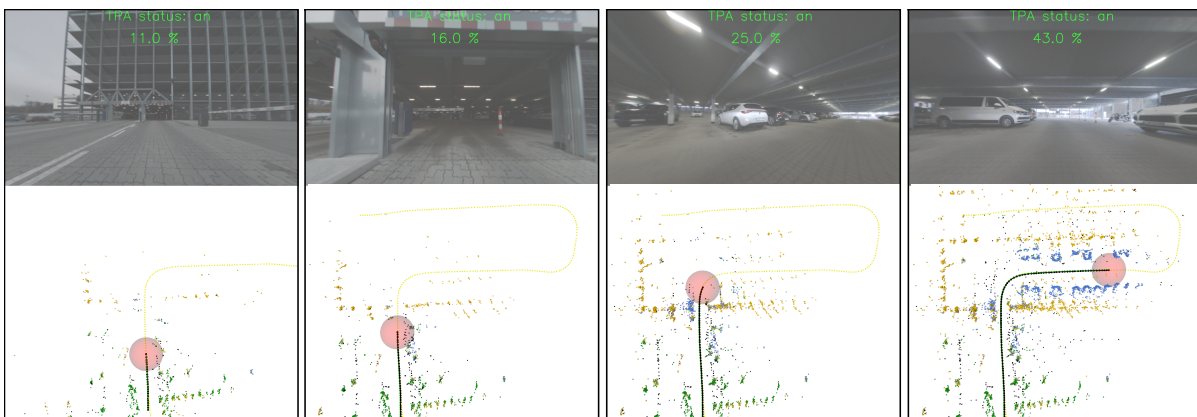


Figure A.13: Different TPA maps and camera images at 44s, 50s, 60s and 80s of an automated parking maneuver. Top-View front facing camera images (top), with synchronous semantic radar map, projected reference path (yellow) and driven path (black) in the bottom image. The radar segmentation is depicted by the color scheme.

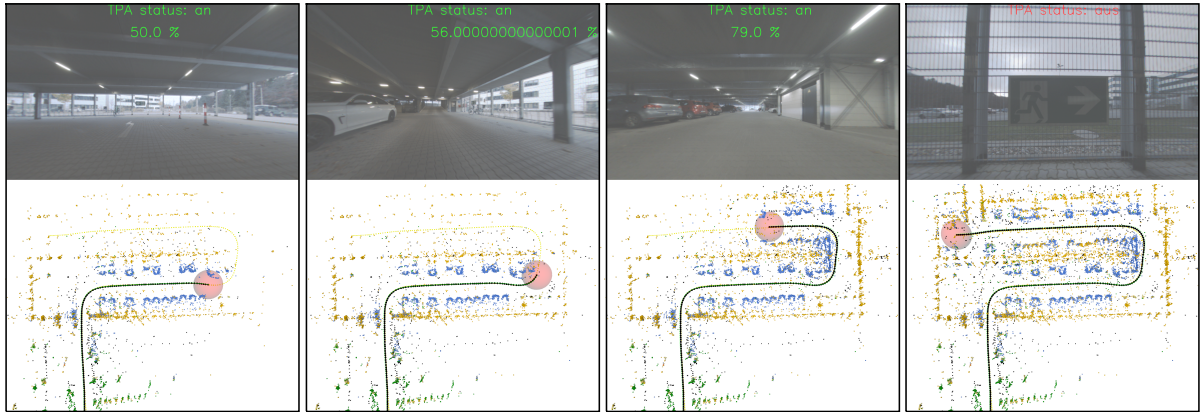


Figure A.14: Different TPA maps and camera images at 88s, 95s, 125s and at the end-position of an automated parking maneuver. Top-View front facing camera images (top), with synchronous semantic radar map, projected reference path (yellow) and driven path (black) in the bottom image. The radar segmentation is depicted by the color scheme.












The map-matching is performed in a regular update rate to benefit from the increasing local perception knowledge in form of the current map. Consequently, with every map-registration update, a new relative map-transformation between current map and reference map is available. Hence, with the same update rate, the planner module needs to reload its input variable of the reference-path projection and ego-vehicle position.

LIST OF FIGURES

| | | |
|------|---|----|
| 1.1 | Satellite images [78, 79] of TPA test scenarios of Section 6.4 as illustration of the aimed use-case. | 4 |
| 1.2 | Illustration of different registration algorithms for radar point cloud association in the same environment. Radar point map colored in z-coordinate, registered poses from red to green nodes. | 5 |
| 1.3 | Consecutive chapter structure of the thesis. | 7 |
| 1.4 | Illustration of the parking functionality structure in a system level context. | 7 |
| 2.1 | Image of an exemplary 77 GHz radar sensor, applied in the automotive context. Courtesy of Hella GmbH & Co.KGaA [114]. | 9 |
| 2.2 | Exemplary block diagram of a typical automotive FMCW 77 GHz radar sensor. | 11 |
| 2.3 | Exemplary linear modulation of an typical automotive FMCW 77 GHz radar emitted signal (black) and reflected echo (blue dotted). Illustration according to Patole et al. [160]. | 12 |
| 2.4 | Exemplary multi-path reflection of radar detections, in horizontal view (left) and top-view (right). Illustration according to Holder et al. [91]. | 15 |
| 2.5 | Exemplary sensor coordinate systems S_{xx}^4 of the applied radar sensor assembly with respect to the central rear-axle vehicle coordinate system $S_{base\ link}$. The mounting position and the resulting sensor view angle is exemplary displayed in this figure. | 17 |
| 2.6 | Coordinate system relation. Parental coordinates on top of child coordinate systems, in the top-down order <i>World</i> → <i>Global</i> → <i>Vehicle</i> → <i>Sensor</i> as consecutive coordinate system transforms. | 18 |
| 2.7 | Exemplary illustration of a single artificial neural neuron. | 22 |
| 2.8 | Exemplary illustration of a ANN as fully connected MLP (left), besides a Convolutional Neural Network (CNN) (right). | 23 |
| 2.9 | Exemplary illustration of a classical image classification CNN. | 24 |
| 2.10 | Illustration of a 3x3 kernel-based convolution with stride length 2 (left), and a 3x3 Max-Pooling operation (right) on the same image. | 25 |
| 2.11 | Exemplary illustration of the error $e_{i,j}$ based on the measurement $z_{i,j}$ relating node x_i to the node x_j . Illustration based on Grisetti et al. [82]. | 36 |
| 2.12 | Sensor setup illustration in top view perspective with exemplary radar sensor FoV. | 39 |
| 2.13 | Radar sensor uncertainty in the mid-range [30 m, 55 m] for center based detections $\phi = 0.5\phi_{max}$ | 41 |










| | | |
|------|--|----|
| 2.14 | Radar sensor uncertainty in the near-range $[0\text{ m}, 25\text{ m}]$ for a strong reflector of $RCS = 10\text{ m}^2$, at the FoV margin $\phi = 0^\circ$ and a center detection $\phi = 0.5\phi_{max}$ | 41 |
| 2.15 | Test vehicle setup as block diagram of the different hardware devices. Signal flow from sensors to actuators shown as arrows. | 42 |
| 3.1 | Radar Sensors of different radar sensor sets: Oxford data set [15] (left), nuScenes sensor set [30](center), and the NLOS sensor set [187](right). | 45 |
| 3.2 | Labeling concept for the proposed automated semantic radar labeling based on reference LiDAR point clouds and camera images with subsequent data preparation steps for machine learning. | 50 |
| 3.3 | Illustration of the top-view camera image processing steps. Raw fish-eye images (left), rectified (center) and perspective transformed image (right), with exemplary frame (red) of the effective image region. Camera images from [MT2], figure modified. | 56 |
| 3.4 | Illustration of rectified top-view camera images and exemplary KITTI Benchmark[71] images, applied for depth prediction with a selection of mono-camera depth estimation CNNs [120, 75, 217, 237]. Estimated relative depth difference in the images are decoded as brightness. Results of [MT2], figure modified. | 58 |
| 3.5 | Illustration of 3 accumulated, subsequent 360° radar point cloud assemblies (3 sensing cycles), covering approximately 150 ms and a $45\text{m} \times 45\text{m}$ grid: Top-view perspective (left), side view (middle) and horizontal view (left) are colored according to the points' z -coordinate $\in [-0.5\text{ m}, 5\text{ m}]$. Please note the interpretation difficulty of even accumulated radar scans. | 60 |
| 3.6 | Illustration of approximately 100 accumulated clouds on a $45\text{m} \times 45\text{m}$ grid. Colors according to z -coordinate $\in [-0.5, 5]$: Top-view (left), side-view (middle), and horizontal-view (right). | 60 |
| 3.7 | Exemplary LiDAR and camera blind spot regions from roof mounted position and rectification. | 63 |
| 3.8 | Exemplary radar reliability factor γ_s over the azimuth angle φ with affected sensor regions (right). | 64 |
| 3.9 | Exemplary LiDAR semantic segmentation pipeline. Data conversion to a range-image and inferred semantic classes on the range image. | 66 |
| 3.10 | Data set format inspired by Behley et al. [18], with extension for radar specific channels <i>Signal Power</i> , <i>Signal to Noise Ratio</i> and <i>Doppler Velocity</i> in additional binary files per channel. | 70 |
| 3.11 | Exemplary sample of trees (green) and poles (yellow), depicted as dense grey LiDAR scan assembly (left) and manually corrected sparse radar point cloud (right) without noise. | 72 |
| 3.12 | Exemplary sample of a hedge (green), poles (yellow), and cars (blue), depicted as dense grey LiDAR scan assembly (left) and manually corrected sparse radar point cloud (right) without noise. | 72 |

| | | |
|------|--|-----|
| 3.13 | Exemplary sample of a hedge (green) and wall structure (yellow), depicted as dense grey LiDAR scan assembly (left) and manually corrected sparse radar point cloud (right) without noise. | 72 |
| 3.14 | Exemplary data set sequences 00-07 with grey LiDAR reference points, of Isele et Al. [SI1]. All <i>noise</i> radar points (red) and <i>plausible</i> radar points (blue) in the top figures (a), while the bottom images (b) show the same sequence, showing exclusively the remaining <i>plausible</i> radar detections. | 74 |
| 3.15 | Exemplary scenes with resulting semantic radar labeling of <i>plausible</i> radar detections, illustrated without noise, class-wise colors according to Table 3.2. Figure of Isele et Al. [SI3]. | 75 |
| 3.16 | Exemplary scenes of the data set with ego-vehicle and its park-in motion as solid (park-out as dotted) grey line in typical environment structures. Black blocks represent buildings, black circles represent poles, white vehicles are either parked or their path is indicated as white arrow. | 75 |
| 3.17 | Data set label distribution as pie chart, visualizing semantic radar classes in the colors according to Table 3.2, clutter excluded. | 77 |
| 3.18 | Heatmap of spatial occurrence count of all radar detections in a 90 m radius around the vehicle independent of their semantic class. | 78 |
| 3.19 | Heatmaps of the spatial occurrence of radar detections per semantic class in a 50 m radius: <i>Clutter, building, vehicle, vegetation, pole</i> and <i>person</i> | 79 |
| 3.20 | Confusion matrix of automated semantic labeling vs. manually corrected semantic labels. | 81 |
| 3.21 | SeRaLF Results of Isele et Al. [SI2]: IoU over distance to the sensor. | 83 |
| 4.1 | Illustration of segmented LiDAR point clouds of street scenes, assembled by a SLAM approach from the KITTI Vision Benchmark [71], published as the semanticKITTI data set of Behley et al. [18] with class-wise colors of the semanticKITTI data set. | 87 |
| 4.2 | Semantic map building with 360 deg. automotive radar SLAM; mirror camera images (left/ right) for visual scene inspection. Figure of Isele et Al. [SI4], color encoding according to Table 3.2. | 89 |
| 4.3 | Exemplary conversion example of the same 3D point cloud in top view with spatial attributes (top left) or with semantic attributes (top right), to the corresponding spatial 2D range images (top landscape image) and semantic 2D range image (bottom landscape image.) The 3D spatial point cloud and the projected 2D range image depicts the range as rgb value. The semantic 3D point cloud and the 2D semantic range image color the semantic class as rgb-channel according to Table 3.2. | 94 |
| 4.4 | Evaluated models architectural network comparison: PolarNet[248], Cylinder3D [250], and ASAP-Net [31]. The authors of these works apply a PointNet-inspired feature extractor, mainly implemented as fully-connected MLP layers. | 102 |
| 4.5 | Illustration of the PointNet architectural principle, illustration from Qi et al. [169]. | 104 |

| | | |
|------|--|-----|
| 4.6 | Illustration of the PointNet++ architectural principle of hierarchical feature extraction for 2D points, illustration from Qi et al. [170]. | 105 |
| 4.7 | Illustration of a Cartesian grid discretization and a polar grid discretization of an exemplary point cloud $\mathcal{P}_{\text{radar}}$. Compared to the static cell area of an cartesian grid, the polar grid yields area-wise increasing regions with increasing radius. Hence, less polar grid cells remain empty compared to a regular cartesian grid. | 106 |
| 4.8 | Illustration of the PolarNet architectural principle, illustration of Zhang et al. [248]. The depicted MLP block is inspired from the PointNet [169] architecture but simplified to a fully connected MLP. The ring convolutions are applied in a not depicted U-Net [176] CNN backbone with skip-connections. | 106 |
| 4.9 | Illustration of the Cylinder3D architectural principle as presented by Zhou et al. [250]. The 3D input discretization is processed with sparse 3D convolutions in a 3D U-Net backbone, followed by a Dimension-Decomposition based Context Modeling (DDCM) block and a segmentation head. | 108 |
| 4.10 | Illustration of the ASAP-Net architecture from original publication of Cao et al. [31]. The concatenation of multiple single-scan feature maps is realized by the ASAP block. With a PointNet feature extractor and backbone, the network segments the point cloud with improved spatio-temporal correlation. | 109 |
| 4.11 | Illustration of a the Cartesian grid discretization (left) versus a polar grid discretization (right). The selection of a Cartesian resolution is non-trivial, whereas the polar discretization matches the radar point distribution of the exemplary radar point cloud better. Figure of Isele et Al. [SI3] | 111 |
| 4.12 | Histogram of Signal Power P_{sig} (left), Signal to Noise Ratio SNR (center), and Doppler Velocity v_D (right). | 112 |
| 4.13 | Illustration of the confusion matrix for the best PolarNet variant. | 115 |
| 4.14 | Illustration of the RadarNet architecture: Additional radar feature channels and reduced output dimensionality. | 119 |
| 4.15 | Illustration of the PolarNet variant with additional radar feature channels. Blue: 19 class, 6D PolarNet variant from architecture comparison. | 121 |
| 4.16 | Illustration of semantic class balance in the data set sequences including <i>clutter</i>  , visualized as pie charts. Color encoding: <i>building</i>  , <i>vehicle</i>  , <i>vegetation</i>  , <i>person</i>  , and <i>pole</i>  | 123 |
| 4.17 | Illustration of data set and distribution of radar semantic classes without clutter, visualized as pie charts with class-specific counts per label. Color encoding: <i>building</i>  , <i>vehicle</i>  , <i>vegetation</i>  , <i>person</i>  , and <i>pole</i>  | 123 |
| 4.18 | Illustration of data augmentation result for the total of all spatial locations. Original heat map (left) compared to the spatial distribution with data augmentation (right). | 124 |

| | | |
|------|--|-----|
| 4.19 | Training loss (left) versus validation loss (right) for every possible cross-validation sequence variant from independent training runs for 50 epochs with same network configuration. | 129 |
| 4.20 | Validation mIoU (left) versus validation accuracy (right) for every possible cross-validation sequence variant from independent training runs for 50 epochs with same network configuration. | 130 |
| 4.21 | Overall validation mIoU (classes: <i>Artifact, Building, Vehicle, Vegetation, Pole, Person</i>) of the best RadarNet variant. These four independently trained model runs are compared. | 131 |
| 4.22 | Training loss and validation loss of best RadarNet variants of four independent models (left) in comparison to the averaged loss components (right): Robustly decreasing validation Lovasz-Loss (top) versus increasing Cross-Entropy validation loss (bottom). | 132 |
| 4.23 | IoU of classes <i>clutter</i> (left), <i>vehicle</i> (center), and class <i>building</i> (right). | 133 |
| 4.24 | IoU of classes <i>vegetation</i> (left), <i>person</i> (center), and class <i>pole</i> (right). | 133 |
| 4.25 | Illustration of the <i>RadarNet</i> confusion matrix at 52 epochs (left) and at 180 epochs (right). | 135 |
| 4.26 | Illustration of the IoU per class over range for the same <i>RadarNet</i> model but at alternative training epochs evaluated in 2m range steps for $R \in [2, 50]$. The early epoch model (trained for 52 epochs) is depicted on top, with the same model, but trained for additional 128 epochs (180 epochs in total), in the bottom plot. | 136 |
| 4.27 | Top-View Illustration of the RadarNet deployment on the test data set: Raw radar point clouds (left), the semantic predictions of the best performing <i>RadarNet</i> (center) variant, compared to semantically labeled ground-truth radar detections (right). | 138 |
| 5.1 | Spatio-temporal filter principle with binary label $\hat{y}_{\text{filter},j}$ to determine static, stable radar detections and filter instable detections. | 152 |
| 5.2 | 2D cell discretization of the 3D radar detection points with processing steps to the calculation of the detection probability $p_d(G_{i,k}, t)$. Colors according to the semantic radar classes of Table 3.2. | 153 |
| 5.4 | Position uncertainty illustration of the lateral \hat{d}_y and radial \hat{d}_r uncertainty components. Exemplary plot of a point detection, based on the sensor error of Figures 2.13 and 2.14. | 154 |
| 5.3 | Illustration of the perception probability $p_d(G_{i,k})$ per cell-wise accumulated signal power $s_{G_{i,k}}$, plotted for an exemplary discretized filter cell $G_{i,k}$, with $s_{\min} = 40$, $s_{\max} = 220$ | 154 |
| 5.5 | Probability of Signal Power s intensity (blue) and accumulated curve (orange) of all detections of a typical recorded scene. s_{\min} at $p_{\text{cum}} = 5\%$, s_{\max} at $p_{\text{cum}} = 90\%$ cumulative frequency. Figure extended from [MT5]. | 155 |
| 5.6 | Temporal activation model depicted as modular blocks: Input-, Delay- and Dynamics-Component as illustrated in [MT5]. | 156 |

| | | |
|------|---|-----|
| 5.7 | Comparison of translational and rotational error of commonly applied point cloud registration procedures (ICP [133], GICP [196], NDT [137], and RPM [235]) for an exemplary radar point cloud registration. | 158 |
| 5.8 | Illustration of the classical spatial radar point cloud registration and exemplary results. | 159 |
| 5.9 | Illustration of the effectiveness of the semantic radar registration, yielding a semantically compliant environment map, compared to Figure 5.8. Colors according to the semantic radar classes of Table 3.2. | 159 |
| 5.10 | Illustration of the pose graph-SLAM front end with different node queues and edges. | 165 |
| 5.11 | Pose graph-SLAM structure to illustrate different edge types: Adjacent wheel-based odometry edges(black), radar odometry, <i>skip edges</i> (pink), and far-reaching loop-closure edges (blue). Colors according to the semantic radar classes of Table 3.2. Figure modified from [SI4]. | 166 |
| 5.12 | Illustration of sub-map assembly of exemplary $k = 1$ adjacent radar scans for node x_k . Colors according to the semantic radar classes of Table 3.2. | 171 |
| 5.13 | Generic illustration of sub map assembly and registration strategy with relative transformation output. Sub-map registration principle applied e.g. for radar odometry edges and for loop-closure sub-map matching. Colors according to the semantic radar classes of Table 3.2. | 172 |
| 5.14 | Loop-Closure with Sub-Map Assembly for $k_{sub}^{LC,n} = k_{sub}^{LC,k} = 3$. Colors according to the semantic radar classes of Table 3.2. | 175 |
| 5.15 | A qualitative comparison of the pre-filter effects on the same scene, radar detections accumulated over 25 seconds. Radar detections accumulated without a filter (left) vs. the filtered version (right). | 176 |
| 5.16 | Histogram of the relative occurrence of translational ϵ_t (left) and rotatory ϵ_ψ registration error (right) of spatial, spatio-temporal and semantic spatio-temporal filter. | 177 |
| 5.17 | Histogram of the relative occurrence of translational ϵ_t (left) and rotational ϵ_ψ registration error (right) for different sub-map configurations k | 177 |
| 5.18 | Translational ϵ_t and rotational ϵ_ψ registration error over initialization position of initial translational ϵ_{trans}^{init} and initial rotational error ϵ_ψ^{init} for two different sub map configurations, $k = 1$ (left) and $k = 5$ (right). Registration result of [MT5], figure modified. | 178 |
| 5.19 | Registration score over true translational pose distance Δ_t^{ref} for different sub map configurations $k_{s,d}$ (left), and resulting registration error ϵ_t with sub map configuration $k = 5$ | 179 |
| 5.20 | Illustration of the covariance of the semantic labeled normal distributions (center) compared to the corresponding point cloud (right). Color coding according to the semantic color convention of Table 3.2. Distribution illustration of [MT5], figure modified. | 180 |

| | | |
|------|---|-----|
| 5.21 | Histogram of the occurrence of translational ϵ_t (left) and rotatational ϵ_ψ registration error (right) for regular NDT, weighted semantic NDT (gSNDT) and semantically separated NDT (sSNDT) registration. | 180 |
| 5.22 | Translational ϵ_t and rotatational ϵ_ψ registration error over the initial position translational ϵ_{trans}^{init} and rotatational error ϵ_ψ^{init} of the initialization position for the registration, for conventional NDT (left) and semantically separating NDT (right). Registration result of [MT5], figure modified. | 181 |
| 5.23 | Illustration of the SLAM system architecture from perception modules to map storage. | 182 |
| 5.24 | Exemplary comparison of the same scene of the scenarios with automatically generated semantic labels (left) versus the same scenario with manually corrected ground-truth semantic labels (right). Colors according to Table 3.2 with additional road  and unknown  | 183 |
| 5.25 | Driven ego-trajectory plotted as black dots, resulting radar map of the (sSNDT) SLAM. Semantic coloring according to Table 3.2: Building  , Vehicle  , Vegetation  , Person  , Pole  , but with Road  , and Unknown  . All plots overtaken from Isele et Al. [SI4]. | 184 |
| 5.26 | Comparison of the spatial radar SLAM (left), utilizing only spatial radar information, and semantically separated (sSNDT) SLAM result (right), both variants display the same revisited part of test scenario III. SLAM nodes visualized as black circles, connected by black odometry edges, loop-closure edges visualized in blue, radar odometry edges are colored in magenta. SLAM map differences highlighted in red ellipses. Colors according the semantic classes of Table 3.2. Figure of Isele et Al. [SI4]. | 186 |
| 5.27 | Comparison of the spatial radar SLAM (left), utilizing only spatial radar information, and semantically separated (sSNDT) SLAM results (right), both variants display the same revisited part of the test scenario II, including a parking maneuver to illustrate the loop closure over a long drive. SLAM nodes visualized as black circles, connected by black odometry edges, loop-closure edges visualized in blue, radar odometry edges are colored in magenta. SLAM map differences highlighted in red ellipses. Colors according the semantic classes of Table 3.2. Figure of Isele et Al. [SI4]. | 186 |
| 6.1 | Illustration of a general robot control system cycle, from perception modules to vehicle actuators of which the thesis addresses the <i>perception</i> and <i>mapping</i> specifically for radar. | 190 |
| 6.2 | Illustration of the overall system architecture from perception modules to vehicle actuators as more detailed variant of te generic system Figure 2.15. The blue top section depicts the system operation in manual training mode, whereas the bottom orange section illustrates the system operation in automated parking mode. Detail information of the computation modules are found in Chapter 4 for the semantic radar segmentation and the semantic radar SLAM is discussed in Chapter 5. | 192 |

| | | |
|------|---|-----|
| 6.3 | Registration surveillance (left) with red reference map and live built semantic SLAM-map (right) of test scenario A and B, see Section 6.4 and Figures 6.5- 6.7. Projected reference path in yellow, driven SLAM path in black with a 7 m red loop-closure search range. The red rectangle shows the start position of the automated drive, equivalent to the end-position of the init-drive. | 193 |
| 6.4 | Camera perspective of surround view cameras with projected reference map (red) and live semantic radar SLAM-map of test scenario B, compare Figure 6.7. Projected reference path in yellow, red boxes are radar detections of the reference map, compare Figure 6.3 (left) in overlay to the live semantic radar SLAM-map as colored boxes according to the color code in Table 3.2. | 193 |
| 6.5 | TPA scenario A as satellite image [79] and scene images around garages as open loop drive. | 195 |
| 6.6 | Illustration of the test scenario A as resulting semantic radar SLAM map (see Section 5.4.5), with sSNDT registration of Section 5.3.4 and Section 5.3.4.2, applied on the live inference of <i>RadarNet</i> of Section 4.5. Colors according the semantic classes in Table 3.2. | 195 |
| 6.7 | TPA scenario B as satellite image [79] and scene images with loop-closure. | 196 |
| 6.8 | Illustration of the test scenario B as resulting semantic radar SLAM map (see Section 5.4.5), with sSNDT registration of Section 5.3.4 and Section 5.3.4.2, applied on the live inference of <i>RadarNet</i> of Section 4.5. Colors according the semantic classes in Table 3.2. | 196 |
| 6.9 | TPA scenario C as satellite image [78] and scene images of the public parking lot with loop closure. | 197 |
| 6.10 | Illustration of the test scenario C as resulting semantic radar SLAM map (see Section 5.4.5), with sSNDT registration of Section 5.3.4 and Section 5.3.4.2, applied on the live inference of <i>RadarNet</i> of Section 4.5. Colors according the semantic classes in Table 3.2. | 197 |
| 6.11 | TPA scenario D as scene images of the car park at floor level as open loop drive. | 198 |
| 6.12 | Illustration of the test scenario D as resulting semantic radar SLAM map (see Section 5.4.5), with sSNDT registration of Section 5.3.4 and Section 5.3.4.2, applied on the live inference of <i>RadarNet</i> of Section 4.5. Colors according the semantic classes in Table 3.2. | 198 |
| 6.13 | Top-view trajectories of the initialization (orange) from different start positions and the full automated drive of the test scenario A (left), scenario B (middle) and car park scenario C (right). | 201 |
| 6.14 | Illustration of the test scenario A (left) and test scenario B (right) with reference d-GPS paths (red), initialization drives (orange), and APE error (grey) of the independent automated drives. The APE error is displayed to check the amount of error along the reference path, not the real deviation coordinates of the automated maneuver. | 203 |

| | | |
|------|---|-----|
| 6.15 | Illustration of the individual APE to the closest d-GPS reference of all automated drives (top) versus the averaged APE error (middle) and the error bar plots of independent automated drives (bottom) of test scenario A (left) and test scenario B (right). | 203 |
| 6.16 | Illustration of the individual APE to the closest d-GPS reference of all automated drives (top) versus the averaged APE error (center) of the test scenario C. Error bar plots of 13 independent individual automated drives of the scenario C (bottom) | 204 |
| 6.17 | Illustration of the test scenario C with reference d-GPS paths (red), initialization drive (orange), and realized APE error (grey) as z-coordinate of the independent automated drives in two perspectives. The APE error is displayed to check the amount of error along the reference path, not the real deviation coordinates of the automated maneuver. | 204 |
| 6.18 | Illustration of the initialization and start position of the automated initialization drive of the test scenario A (left), scenario B (middle) and car park scenario C (right). | 205 |
| 6.19 | Illustration of the final position of the automated drive of the test scenario A (left), scenario B (middle) and car park scenario C (right). The red path represents the manual reference dGPS path, whereas the grey paths are from each different test drives. | 205 |
| 6.20 | Architecture of the offline map evaluation to derive metrics describing the radar mapping quality. | 207 |
| 6.21 | Illustration of global matching (left) of reference map (orange) with current map (black) compared to the local matching result (left) the test scenario D. Both maps are filtered by the radius outlier filter after registration. Figure overtaken of [MT6]. | 208 |
| 6.22 | Illustration of local semantic matching for a precise map comparison. | 208 |
| 6.23 | Exemplary illustration of the 30 <i>m</i> quadratic map <i>tiles</i> and map <i>grid cells</i> of test scenario D. The voxel grid of 0.5 <i>m</i> cubes is not displayed to scale. | 209 |
| 6.24 | Averaged map evaluation metrics of all scenarios A-D averaged metrics over independent autonomous parking runs. | 211 |
| 6.25 | Illustration of matching cells (green) and newly occupied regions (orange) of three independent maps of automated parking runs (vertical) of the four test scenarios A-D (left to right). | 213 |
| 6.26 | Three different autonomous drive semantic radar maps (vertically) of the four test scenarios A-D (left to right). ⁵ | 213 |
| 6.27 | Illustration of an exemplary surface reconstruction of the isolated vehicle detections in a car park from a two different semantic radar SLAM maps. The bumper shells are prominently reconstructable for the passaged vehicle rows. | 218 |
| A.1 | Original spatial distribution of all radar point classes (left) compared to augmented by rotation and coordinate flipping (right). | 222 |

| | | |
|------|---|-----|
| A.2 | Original spatial distribution of <i>building</i> radar point class (left) compared to augmented by rotation and coordinate flipping (right). | 222 |
| A.3 | Original spatial distribution of <i>vehicle</i> radar point class (left) compared to augmented by rotation and coordinate flipping (right). | 223 |
| A.4 | Original spatial distribution of <i>vegetation</i> radar point class (left) compared to augmented by rotation and coordinate flipping (right). | 223 |
| A.5 | Original spatial distribution of <i>person</i> radar point class (left) compared to augmented by rotation and coordinate flipping (right). | 223 |
| A.6 | Original spatial distribution of <i>pole</i> radar point class (left) compared to augmented by rotation and coordinate flipping (right). | 224 |
| A.7 | Original spatial distribution of <i>clutter/ noise</i> radar point class (left) compared to augmented by rotation and coordinate flipping (right). | 224 |
| A.8 | Illustration of the automated initialization drive resulting semantic radar SLAM map for the test scenarios A and B (left), scenario C (center) and car park scenario D (right). | 227 |
| A.9 | Illustration of the total semantic radar SLAM map for the test scenarios A and B (left), scenario C (center) and car park scenario D (right). | 228 |
| A.10 | Illustration of the automated re-localization process, showing the submap extraction of the reference map based on the convex hull of the current map, submap alignment and relative transformation. | 229 |
| A.11 | Illustration of the automated initialization drive resulting semantic radar SLAM map (left), with respect to the reference map (right) of the test scenario B. The sub-sampled and convex-hull extract is depicted as dashed line. | 230 |
| A.12 | Illustration of the automated initialization drive resulting semantic radar SLAM map (left), with respect to the reference map (right) of the test scenario D. The sub-sampled and convex-hull extract is depicted as dashed line. | 230 |
| A.13 | Different TPA maps and camera images at 44s, 50s, 60s and 80s of an automated parking maneuver. Top-View front facing camera images (top), with synchronous semantic radar map, projected reference path (yellow) and driven path (black) in the bottom image. The radar segmentation is depicted by the color scheme. | 231 |
| A.14 | Different TPA maps and camera images at 88s, 95s, 125s and at the end-position of an automated parking maneuver. Top-View front facing camera images (top), with synchronous semantic radar map, projected reference path (yellow) and driven path (black) in the bottom image. The radar segmentation is depicted by the color scheme. | 232 |

LIST OF TABLES

| | | |
|-----|---|-----|
| 2.1 | Exemplary activation function formulations with the Softmax formulation. | 23 |
| 2.2 | Exemplary confusion matrix evaluation of a classifier result. Illustration in reference to class 1 to denote the nomenclature of positive and negative samples. Predictions of class 2 and class 3 are mis-classifications with respect to class 1, so negative. | 32 |
| 2.3 | On board sensor set of automated vehicles according to Yurtsever et al. [240]. | 39 |
| 2.4 | Sensor set details. | 40 |
| 3.1 | Tabular overview of public radar data sets according to the RadarScenes publication [194]. The authors compare radar data sets for machine learning purposes with special focus for dynamic objects, referred as road users. Column <i>scenario variations</i> are considered as weather, traffic, or road types variation. <i>Sequential data</i> describes if temporally subsequent radar scans are available. Our data set contains as single data set point-wise labels for environment detections. | 44 |
| 3.2 | Applied consolidation of the 22 SemanticKITTI [18] classes to six radar applicable classes. | 46 |
| 3.3 | Parameter set applied for the labeling pipeline. | 65 |
| 3.4 | Data set overview of semantically labeled radar data. | 76 |
| 3.5 | Data set details for the 00-10 sub-set, as in Isele et Al. [SI1]. | 77 |
| 3.6 | Tested plausibility label data set of $M = 11$ sequences with bold marking of min. and max. metric scores. | 80 |
| 3.7 | Dataset evaluation of the automated semantic labeling, according to Isele et Al. [SI2]. | 82 |
| 4.1 | Evaluation of different semantic segmentation approaches, rating the applicability to perform on sparse radar point clouds. | 99 |
| 4.2 | Overview of public semantic segmentation networks on the SemanticKITTI leaderboard with public implementation. Selection based on performance on SemanticKITTI [18] and nuScenes [30] data. | 102 |
| 4.3 | Data set overview of for the architecture test of LiDAR segmentation networks on radar data. Sequences 00-10 used for training, while Sequence 05 exclusively applies as test sequence for evaluation. | 110 |

| | | |
|-----|---|-----|
| 4.4 | Ablation study over test IoU [%] of the three classes <i>clutter</i> , <i>building</i> and <i>vehicle</i> . The other classes' IoU are not included in the reported average IoU since they often remain unrecognized at 0.0% ⁶ and do not occur in the exclusive test sequence 05. | 114 |
| 4.5 | Architecture comparison, rating from positive (++) to negative (- -), with weighted wIoU: Considering only <i>clutter</i> , <i>building</i> , and <i>vehicles</i> | 118 |
| 4.6 | Model study of PolarNet with variations in discretization, features and network architecture. | 121 |
| 4.7 | Repetition of Table 3.4, now as data set split overview of training data for <i>RadarNet</i> training, validation (light grey marked) and testing (dark grey). | 122 |
| 4.8 | PolarNet architecture radar variants as comparison of different Loss-functions for semantic segmentation: Network variant 4D PolarNet with radar channel signal power P_{sig} and grid size [200, 200, 32], trained on Sequences 00-10. | 126 |
| 4.9 | Proposed class weights w_i per class C to balance rare classes in a reduced data set for architecture tests (sequences 00-10) and for the <i>RadarNet</i> training on the full data set (sequences 00-18). | 127 |
| 5.1 | Rule based pre-filter parameters. | 151 |
| 5.2 | Overview of the evaluated SLAM configurations with different registration methods (NDT,gSNDT: weighted semantic NDT, and sSNDT: semantically separating NDT) and pre-filters (SST: semantic spatio-temporal, ST: spatio-temporal, GEO: spatial. | 183 |
| 5.3 | Comparison of the Absolute Pose Error (APE) [m] of the SLAM variant (A-E) in test scenarios I-IV. | 185 |
| 5.4 | Comparison of the Relative Pose Error (RPE) [$\frac{m}{10m}$] of the SLAM variant (A-E) in the test scenarios I-IV. | 185 |
| 6.1 | Automated parking test drives (A-D) with test details (X.1 /X.2 /X.3) driven at 19./ 22./ 25.11.2021 different length, environmental content and respective Δz of 0.82 m, 0.79 m, 1.3 m, and 0.0 m. | 200 |
| 6.2 | SLAM map evaluation of the averaged test scenario metrics. Test scenario A-D with different length and environmental content. | 212 |

ACRONYMS AND SYMBOLS

Acronyms

| | |
|---------------|---|
| 2D | 2-Dimensional |
| 3D | 3-Dimensional |
| 4D | 4-Dimensional |
| ACC | Adaptive Cruise Control |
| ADC | Analog-to-Digital Converter |
| ANN | Artificial Neural Network |
| APE | Absolute Positioning Error |
| ATE | Attentive Temporal Embedding |
| AVP | Automated Valet Parking |
| BASD | Binary Annular Statistics Descriptor |
| BN | Batch Normalization |
| BRIEF | Binary Robust Independent Elementary Features |
| CBO | Cumulative Binary Occupancy |
| CFAR | Constant False Alarm Rate |
| CNN | Convolutional Neural Network |
| Conv. | Concolution |
| CSM | Correlative Scan Matching |
| DBSCAN | Density-Based Spatial Clustering of Applications with Noise |
| DDCM | Dimension-Decomposition Context Modeling |
| dGPS | Differential Global Positioning System |
| DOF | Degrees of Freedom |
| DSC | Dice Similarity Coefficient |

| | |
|-----------------|---|
| FAST | Features from Accelerated Segment Test |
| FFT | Fast Fourier Transform |
| FMCW | Frequency Modulated Continuous Wave |
| FN | False Negative |
| FP | False Positive |
| GEO | spatial |
| GLARE | Geometric Landmark Relations |
| GMPF | Grid Mapping Particle Filter |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| GPU | Graphics Processing Unit |
| gSNDT | weighted Semantically Separated Normal Distribution Transform |
| GUI | Graphical User Interface |
| HAD | Highly Automated Driving |
| HD – Map | High-Definition Map |
| ICP | Iterative Closest Point |
| IoU | Intersection Over Union |
| mIoU | mean Intersection Over Union |
| IPA | Intelligent Park Assist |
| k – NN | K-Nearest Neighbors |
| KIT | Karlsruher Institute for Technology |
| LiDAR | Light Detection and Ranging |
| LSTM | Long Short-Term Memory |
| MIMO | Multiple Input Multiple Output |
| MSG | Multi-Scale Grouping |
| NDT | Normal Distribution Transform |
| ORB | Oriented FAST and rotated BRIEF |
| PCA | Principal Component Analysis |

| | |
|---------------|--|
| POC | Proof of Concept |
| PPF | Point Pair Features |
| Radar | Radiowave Detection and Ranging |
| RANSAC | Random Sample Consensus |
| RCS | Radar Cross Section |
| ReLU | Rectified Linear Unit |
| ROI | Region of Interest |
| RPA | Remote Park Assist |
| RPE | Relative Positioning Error |
| SAR | Synthetic Aperture Radar |
| SfM | Structure from Motion |
| SGD | Stochastic Gradient Descent |
| SLAM | Simultaneous Localization and Mapping |
| SNR | Signal Noise Ratio |
| sSNDT | semantically Separated Normal Distribution Transform |
| SST | Semantic Spatio-Temporal |
| ST | Spatio-Temporal |
| SVD | Singular Value Decomposition |
| TLS | Truncated Least Squares |
| TN | True Negative |
| TP | True Positive |
| TPA | Trained Parking Assistant |
| UAV | Unmanned Aerial Vehicle |
| VRU | Vulnerable Road User |

Mathematical Definitions symbols

| | |
|-------------------------------|--|
| i, j, k, l | Indices |
| $\ \mathbf{q} - \mathbf{p}\ $ | Distance Norm of two points \mathbf{q}, \mathbf{p} |

| | |
|--------------------------------------|--|
| e^x | Exponential function of x |
| $\frac{\partial(\cdot)}{\partial x}$ | Partial derivative of (\cdot) with respect to e.g. x |
| $\nabla(\cdot)$ | Gradient of a Function (\cdot) |
| $p(X Z)$ | Conditional probability of X , given Z |
| ϕ, ψ | Different Angles |
| β, ϵ | Model Parameters |
| H | Hessian Matrix |
| J | Jacobi Matrix |

Radar Measurements Definitions

| | |
|--|--|
| t | Timestamp of measurements |
| $\mathcal{P}_{\text{radar}}, \mathcal{P}_{\text{lidar}}$ | Point cloud of a Radar / LiDAR point cloud containing the cartesian coordinates |
| $\mathcal{P}_r, \mathcal{P}_l$ | Point cloud of a Radar / LiDAR point cloud containing the cartesian coordinates |
| $\mathcal{P}_{r, V_{ref}}, \mathcal{P}_{r, S_i}$ | Point cloud of a Radar expressed in V_{ref} / Sensor S_i coordinates |
| p | Point vector of a radar point cloud containing the cartesian coordinates |
| X | Point attribute vector of a radar point cloud containing additional radar attributes |
| p | Single point of a radar point cloud containing the cartesian coordinates |
| (x, y, z) | Cartesian coordinates of a point p |
| SNR | Signal Noise Ratio |
| P_{sig} | Signal Power |
| v_D | Doppler Velocity |
| R | Distance to the sensor |
| ϕ | Azimuthal Angle of a point |
| P_e, P_r | Emitted Power, Received Power |
| G_t | Antenna Gain Factor |
| σ | RCS: Radar Cross Section |

| | |
|------------------------------|---|
| A | Amplitude |
| ϕ_0 | (zero) Phase Angle |
| $f(t)$ | Modulated Frequency |
| f_b | Beat Frequency |
| B | Bandwidth |
| T_m | Modulation Time |
| T_c | Chirp Time |
| $\Phi(t)$ | Instantaneous Phase |
| τ | Temporal Shift of the Instantaneous Phase |
| A_e, A_{TX}, A_{RX} | Antenna Area: Emitting Antenna, Transceiving Antenna, Receiving Antenna |
| d_{arr} | Distance of Antenna Arrays |
| $\Delta\Phi$ | Antenna Phase Shift |
| $w(\mathbf{p}_{r,i,t})$ | Radar Plausibility Score |
| $w_{lm}(\mathbf{p}_{r,i,t})$ | LiDAR Matching Radar Plausibility Score |
| $w_{cm}(\mathbf{p}_{r,i,t})$ | Camera Matching Radar Plausibility Score |
| $w_{tr}(\mathbf{p}_{r,i,t})$ | Spatio-Temporal Tracking Radar Plausibility Score |
| $y(\mathbf{p}_{r,i,t})$ | Binary Radar Plausibility Label |
| β_{cm} | Scalar Camera Matching Parameter |
| β_{lm} | Scalar LiDAR Matching Parameter |
| β_{tr} | Scalar Tracking Decay Parameter |
| K | Neighbor Points in KNN-Matching Algorithm |
| d | Distance between Radar and closest co-located Lidar Point |
| σ | Measurement Uncertainty |
| s_{lidar} | LiDAR Scaling Factor |
| n_b | Number of subsequent or preceding radar scans |
| α_{lidar} | Horizontal Lidar FOV Opening Angle |
| $V_{bs, lidar}$ | LiDAR Blind Spot Volume |

| | |
|--|--|
| $V_{\text{bs, rec, cam}}$ | Camera Blind Spot Volume after Rectification and Perspective Transformation |
| $\gamma_s(\phi)$ | Radar Reliability over the azimuthal detection angle |
| α | Weighting parameter in the plausibility selection |
| $\mathcal{I}_{\text{lidar}}$ | Range Image of a LiDAR Point Cloud |
| $y_{\text{sem, radar}}(\mathbf{p}_{r,i,t})$ | Semantic Radar Label |
| $y_{\text{sem, LiDAR}}(\mathbf{p}_{r,i,t})$ | Semantic Lidar Label |
| $y_{\text{sem, camera}}(\mathbf{p}_{r,i,t})$ | Semantic Camera Label |
| L_i | Semantic Label of Grid Cell Z_i |
| $w_{i,l}$ | Semantic Label Weight of Grid Cell Z_i |
| $SE_G(\cdot)$ | Semantic Label Selection per cell |
| ϕ, θ, ψ | Roll, Pitch, and Yaw Angles |
| \mathbf{s} | Radar Sensor Index $\mathbf{s} \in [FL, FR, ML, MR, BL, BR]$ |
| \mathbf{T} | Homogeneous transformation matrix for a transformation of a point p to a different coordinate system |
| $\mathbf{T}_{x_k}^{x_{k+1}}$ | Homogeneous Relative transformation matrix from position x_k to the position x_{k+1} |
| $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ | Rotation matrix for a rotation according to the subscript axis |
| T | Translation Vector |
| Q, \mathcal{K} | Different point sets |

Machine Learning Definitions

| | |
|--------------------|---|
| \mathbf{x} | Input tensor of multiple artificial neurons (1D case) |
| \mathbf{X} | Input tensor of multiple artificial neurons (2D case) |
| $\hat{\mathbf{y}}$ | Estimated output tensor of multiple artificial neurons |
| \mathbf{y} | Ground-Truth output tensor of multiple artificial neurons |
| x | Input vector of an artificial neuron (1D case) |
| a | Weight vector of an artificial neuron (1D case) |
| a_0 | Bias encoding of an artificial neuron (1D case) |

| | |
|-------------------------|--|
| A | Filter Mask of the weighted inputs of an artificial neuron (2D case) |
| z | Weighted sum of the inputs of an artificial neuron |
| Z | Feature map of the weighted inputs of an artificial neuron (2D case) |
| $f(z)$ | Activation function of an artificial neurons |
| $\sigma(\mathbf{y})$ | Softmax function of the output tensor of multiple artificial neurons |
| q | Number of training samples used for training |
| \mathcal{L} | Discrete Set of labels or classes used for prediction |
| N_c, C | Number of classes used for prediction |
| η | Learning rate |
| L | Loss function |
| w_i | Class i specific Loss function weights |
| ∇L | Back Propagated Loss Gradient |
| H | Height of a feature map or tensor |
| W | Width of a feature map or tensor |
| C | Number of channel of a feature map or tensor |
| p_a | Padding width of a convolution layer |
| s_a | Stride of a convolution layer |
| $\bar{\mathbf{x}}$ | Zero Mean Normalization of Tensor \mathbf{x} |
| $\sigma_{\mathbf{x}}^2$ | Standard Deviation of Tensor \mathbf{x} |
| \mathbf{x}_{norm} | Normalized Tensor \mathbf{x} |
| C_i | The point samples classified as class i |

Point Cloud Registration Definitions

| | |
|------------|---|
| p | Point in cartesian coordinates |
| R | Rotation Matrix |
| T | Translation Vector |
| N | Noise, e.g. Gaussian Noise per point |
| $F(\cdot)$ | Registration formulation, quadratic distance minimization formulation |

| | |
|-------------------------|---|
| $(\cdot)'$ | Analytic solution |
| $(\cdot)^*$ | Optimum solution |
| Z | 2D grid discretization of the source point cloud |
| \bar{Z} | 2D grid discretization of the target point cloud |
| $\ \mathcal{P}_{Z_i}\ $ | Count of points of the point cloud \mathcal{P} in grid cell Z_i |
| μ_{Z_i} | Mean coordinate of all points located in grid cell Z_i |
| Σ_{Z_i} | Covariance matrix of all points located in grid cell Z_i |
| \mathcal{N} | Normal Distribution of a point set |

Graph SLAM Formulation Definitions

| | |
|-----------------|---|
| \ominus | Pose distance operator between two nodes (positions) |
| x | Node Positions |
| X | Trajectory consisting of a set of node positions x |
| z | Measurement |
| Z | Measurement set consisting of a set of measurements z |
| z^{odom} | Odometry Graph Edge |
| z^{RO} | Radar Odometry Graph Edge |
| z^{LC} | Loop-Closure Graph Edge |
| χ | SLAM Target Function |
| Σ | Covariance Matrix |
| Ω | Information Matrix |
| $e_{i,j}$ | Positioning error between node x_i and node x_j |
| e_k | Absolute positioning error at node x_k with measurement z_k |
| $(\bar{\cdot})$ | Taylor expansion linearization point |
| $J_{i,j}$ | Jacobi Matrix e.g. of the error $e_{i,j}$ |
| c, b, H | Substitution variables |
| R | Rotation Matrix |
| \boxplus | Manifold Operator |

| | |
|---------------------------------------|--|
| t | Translation vector |
| q | Rotation Quaternion |
| G | Total Grid Discretization of a Point Cloud |
| $G_{i,k}$ | Grid Discretized Cell i of semantic Layer k |
| $w_{i,k}$ | Semantic Label Weight of Grid Cell i of semantic Layer k |
| \hat{y}_{filter} | Binary spatio-temporal filter qualifier |
| $p_d(G_i)$ | Detection propability per grid cell G_i |
| $p_a(G_i)$ | Activation propability per grid cell G_i |
| $p_p(G_i)$ | Perception propability per grid cell G_i |
| s_{G_i} | Summarized Signal Power or Radar points in a Grid Cell |
| \hat{d}_y | Lateral Measurement Uncertainty |
| \hat{d}_r | Radial Measurement Uncertainty |
| K^d | Temporal Decay Factor of the Activation Probability |
| $K^{d,s}$ | Dynamic Temporal Decay Factor of the Activation Probability |
| s_{min}, s_{max} | Detection Propability Threshold |
| $p_{p,th}$ | Perception Probability Threshold |
| L_{i,k} | Links or Correspondences between Source and Target Point Cloud |
| $\epsilon_{\text{pos, System}}$ | Total System Positioning Error |
| $\epsilon_{\text{pos, SLAM}}$ | SLAM-related Mapping and Registration Error Component |
| $\epsilon_{\text{pos, Map-Matching}}$ | Map-Matching related Registration Error Component |
| $\epsilon_{\text{pos, Planner}}$ | Planner-related Positioning Error Component |

Autonomous Parking Definitions

| | |
|-----------------------|--|
| $T_{\text{ref, cur}}$ | Transformation Matrix of the Reference Path into the current Odometry Coordinate System |
| M | Semantic Radar Point Cloud Map |
| SM | Semantic Radar Point Cloud Sub-Map |
| R_{map} | Relative Rotation between Reference Map \mathbf{M}_{ref} and Current Map \mathbf{M}_{curr} |

| | |
|-------------------------------------|---|
| T_{map} | Relative Translation between Reference Map \mathbf{M}_{ref} and Current Map \mathbf{M}_{curr} |
| $(x_{\text{ego}}, y_{\text{ego}})$ | Vehicle Position in the current Odometry Coordinate System |
| $(x_{\text{ref}}, y_{\text{ref}})$ | Reference Path Target Position in the current Odometry Coordinate System |
| N_{Path} | Number of Reference Path Positions |
| $y_{\text{sem, majority}}$ | Semantic Majority Class of a Radar Map Voxel |
| $p_{\text{cluster}, i}$ | Point Cluster of a Radar Map Voxel i |
| $c_{\text{voxel static}}$ | Count of static, occupied Voxels of a Map Comparison |
| $d_{\text{voxel static}, i, j}$ | Distance of two Point Clusters of a Radar Map Voxel i and j |
| $\emptyset d_{\text{voxel static}}$ | Distance Average of all associated static Point Clusters of a Map Comparison |

LIST OF PUBLICATIONS

Conference Contributions

- [SI1] Simon T. Isele, Marcel Schilling, Fabian E. Klein, Sascha Saralajew, and J. Marius Zöllner, “Radar artifact labeling framework (RALF): Method for plausible radar detections in datasets,” in *VEHITS*, Dec 2021.
- [SI2] Simon T. Isele, Marcel Schilling, Fabian E. Klein, and J. Marius Zöllner, “Annotating automotive radar efficiently: Semantic radar labeling framework (SeRaLF),” in *Neural Information Processing Systems (NiPS), Workshop: Machine Learning for Autonomous Driving*, Apr 2020.
- [SI3] Simon T. Isele, Fabian Klein, Mathis Brosowsky, and J. Marius Zöllner, “Learning semantics on radar point-clouds,” in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 810–817, Jul 2021.
- [SI4] Simon T. Isele, Fabian Haas-Fickinger, and J. Marius Zöllner, “Seraloc: Slam on semantically annotated radar point-clouds,” in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 2917–2924, Sep 2021.

Co-authored Publications

- [Co1] Philip Elspas, Yannick Klose, Simon T. Isele, Johannes Bach, and Eric Sax, “Time series segmentation for driving scenario detection with fully convolutional networks,” in *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS 2021, Online Streaming, April 28-30, 2021* (Karsten Berns, Markus Helfert, and Oleg Gusikhin, eds.), pp. 56–64, SCITEPRESS, 2021.
- [Co2] Lars Ohnemus, Lukas Ewecker, Ebubekir Asan, Stefan Roos, Simon T. Isele, Jakob Ketterer, Leopold Müller, and Sascha Saralajew, “Provident vehicle detection at night: The PVDN dataset,” *CoRR*, vol. abs/2012.15376, 2020.

- [Co3] Sascha Saralajew, Lars Ohnemus, Lukas Ewecker, Ebubekir Asan, Simon T. Isele, and Stefan Roos, “A dataset for provident vehicle detection at night,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pp. 9750–9757, IEEE, 2021.
- [Co4] Mathis Brosowsky, Florian Keck, Jakob Ketterer, Simon T. Isele, Daniel Slieter, and Johann Marius Zöllner, “Safe deep reinforcement learning for adaptive cruise control by imposing state-specific safe sets,” in *IEEE Intelligent Vehicles Symposium, IV 2021, Nagoya, Japan, July 11-17, 2021*, pp. 488–495, IEEE, 2021.

Supervised Theses

- [MT1] Nico Schlucke, “Lokalisierung durch Datenverarbeitung von Radarsignalen in einem beliebigen Umfeld für zukünftige Parkassistentenfunktionen.” Master Thesis, 2021.
- [MT2] Marcel Schilling, “Automatisierte Annotation von Radarpunktwolken zur Optimierung von automatisierten Parkanwendungen mittels Deep Learning.” Master Thesis, 2020.
- [MT3] Fabian E. Klein, “Fast and accurate semantic segmentation of radar data for simultaneous localization and mapping.” Master Thesis, 2020.
- [MT4] Fabian Bischof, “Experimentelle Evaluation von Trajektorienfolgeregelungen für hochautomatisierte Fahrzeuge.” Master Thesis, 2021.
- [MT5] Fabian Haas-Fickinger, “Entwicklung einer effizienten, semantischen Methodik zur radargestützten Lokalisierung automatisierter Fahrzeuge.” Master Thesis, 2021.
- [MT6] Avinash Shankar-Bhat, “Relocalization and map fusion using semantic radar data.” Master Thesis, 2022.
- [MT7] Daniel Rotärmel, “Optimierung der Semantischen Segmentierung von Radar Punktwolken im Kontext Pose-Graph SLAM.” Master Thesis, 2022.

Filed Patents

- [Pat1] Simon Tobias Isele and Sascha Saralajew, “Device and method for navigating a vehicle,” May 2020. active and published patent: DE, US, CN, KR, DE102020112559A1.
- [Pat2] Simon Tobias Isele and Sascha Saralajew, “Method and device for warning of damage to an electric vehicle,” Jun 2020. active and published patent: DE, DE102020117431A1.

-
- [Pat3] Simon Tobias Isele and Marcel Peter Schilling, "Method and system for the automatic labeling of radar data," Sep 2020. active and published patent: DE, US, CN, KR, DE102020123920B3.
- [Pat4] Simon Tobias Isele and Marcel Peter Schilling, "Labelingverfahren," Nov 2020. patent pending: DE, 10 2020 131 779.4.
- [Pat5] Simon Tobias Isele, Stefan Roos, Marcel Peter Schilling, and Björn Bentz, "Method for the three-dimensional reconstruction of a scene in front of a vehicle," Feb 2021. active and published patent: DE, DE102021102818B3.
- [Pat6] Simon Tobias Isele and Al., "Kraftfahrzeug mit subjektiv angepasster Dynamik eines Fahrbeeinflussungssystems, System aufweisend," Feb 2021. patent pending: DE, 10 2021 106 226.8.
- [Pat7] Simon Tobias Isele and Al., "Verfahren, System und Computerprogrammprodukt zur automatisierten Lokalisierung eines Fahrzeugs," Jun 2021. patent pending: DE, CN, US 10 2021 119 124.6.
- [Pat8] Simon Tobias Isele and Al., "Querführung mittels Radardaten II," Mar 2022. patent pending: DE, 10 2022 107 283.5.
- [Pat9] Simon Tobias Isele and Al., "RoboAR," Mar 2022. patent pending: DE, 10 2022 105 155.2.
- [Pat10] Simon Tobias Isele and Al., "Karosserie follows Sensorset," Mar 2022. patent pending: DE, 10 2022 107 276.2.
- [Pat11] Simon Tobias Isele and Al., "Reduktion von Einzelkameras mittels verteilter Linsen und zentralem Bildsensor," May 2022. patent pending: DE, 10 2022 111 485.6.

BIBLIOGRAPHY

- [1] Design and implementation of an fmcw radar signal processing module for automotive applications.
- [2] Apollo Auto, 2022. URL <https://github.com/ApolloAuto/apollo>.
- [3] Continental AG 2022. Automated Valet Parking, 2022. URL <https://www.continental-automotive.com/DE/Passenger-Cars/Autonomous-Mobility/Functions/Low-Speed-Maneuvering/Overall-Parking-Story/VALET-PARKING>.
- [4] Pratik Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust map optimization using dynamic covariance scaling. In *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*, pages 62–69. IEEE, 2013. doi: 10.1109/ICRA.2013.6630557. URL <https://doi.org/10.1109/ICRA.2013.6630557>.
- [5] Hamed Habibi Aghdam and Elnaz Jahani Heravi. *Guide to Convolutional Neural Networks*. Springer, 2017. ISBN 978-3-319-57550-6.
- [6] Tokihiko Akita and Seiichi Mita. Accurate Parking Scene Reconstruction using High-Resolution Millimeter-Wave Radar. In *23rd IEEE International Conference on Intelligent Transportation Systems, ITSC 2020, Rhodes, Greece, September 20-23, 2020*, pages 1–6. IEEE, 2020. doi: 10.1109/ITSC45102.2020.9294210. URL <https://doi.org/10.1109/ITSC45102.2020.9294210>.
- [7] Roberto Aldera, Daniele De Martini, Matthew Gadd, and Paul Newman. Fast Radar Motion Estimation with a Learnt Focus of Attention using Weak Supervision. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 1190–1196. IEEE, 2019. doi: 10.1109/ICRA.2019.8794014. URL <https://doi.org/10.1109/ICRA.2019.8794014>.
- [8] Roberto Aldera, Danile De Martini, Matthew Gadd, and Paul Newman. What Could Go Wrong? Introspective Radar Odometry in Challenging Environments. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2835–2842, 2019. doi: 10.1109/ITSC.2019.8917111.

- [9] Stephen Alland, Wayne E. Stark, Murtaza Ali, and Manju V. Hegde. Interference in Automotive Radar Systems: Characteristics, Mitigation Techniques, and current and future Research. *IEEE Signal Process. Mag.*, 36(5):45–59, 2019. doi: 10.1109/MSP.2019.2908214. URL <https://doi.org/10.1109/MSP.2019.2908214>.
- [10] Robert Andersen. Modern Methods for Robust Regression. *Sage Series: Quantitative Applications in the Social Sciences*, 152, 2008. doi: 10.4135/9781412985109. URL <https://dx.doi.org/10.4135/9781412985109>.
- [11] Tilo Arens, Frank Hettlich, Christian Karpfinger, Ulrich Kockelkorn, et al. *Mathematik*. Springer Spektrum, 3rd edition edition, 2015. ISBN 9783642449192.
- [12] Kai Oliver Arras. An Introduction To Error Propagation: Derivation, Meaning and Examples. Technical report, Eidgenössische Technische Hochschule Lousanne, 1998. URL <http://srl.informatik.uni-freiburg.de/papers/arrasTR98.pdf>.
- [13] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987. doi: 10.1109/TPAMI.1987.4767965.
- [14] Dan Barnes and Ingmar Posner. Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 9484–9490. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9196835. URL <https://doi.org/10.1109/ICRA40945.2020.9196835>.
- [15] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 6433–6438. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9196884. URL <https://doi.org/10.1109/ICRA40945.2020.9196884>.
- [16] Torsten Becker, Richard Herrmann, Viktor Sandor, Dominik Schäfer, and Ulrich Wellisch. *Stochastische Risikomodellierung und statistische Methoden*. Springer Spektrum, 1st edition edition, 2020. ISBN 978-3-662-49406-6. doi: 10.1007/978-3-662-49407-3.
- [17] Jens Behley and Cyrill Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018. doi: 10.15607/RSS.2018.XIV.016. URL <http://www.roboticsproceedings.org/rss14/p16.html>.
- [18] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding

- of LiDAR Sequences. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9296–9306. IEEE, 2019. doi: 10.1109/ICCV.2019.00939. URL <https://doi.org/10.1109/ICCV.2019.00939>.
- [19] Ola Bengtsson and Albert-Jan Baerfeldt. Robot Localization based on Scan-Matching - Estimating the Covariance Matrix for the IDC Algorithm. *Robotics Auton. Syst.*, 44(1): 29–40, 2003. doi: 10.1016/S0921-8890(03)00008-3. URL [https://doi.org/10.1016/S0921-8890\(03\)00008-3](https://doi.org/10.1016/S0921-8890(03)00008-3).
- [20] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. doi: 10.1109/CVPR.2018.00464.
- [21] Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition edition, 1999. ISBN 1-886529-00-0.
- [22] Jonathan M. Blackledge. *Digital Signal Processing: Mathematical and Computational Methods, Software Development and Applications*. Horwood Publishing, 2nd edition edition, 2006. ISBN 1-904275-26-5.
- [23] Sean L. Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J. Pappas. Probabilistic data association for semantic SLAM. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 1722–1729. IEEE, 2017. doi: 10.1109/ICRA.2017.7989203. URL <https://doi.org/10.1109/ICRA.2017.7989203>.
- [24] Sean L. Bowman, Kostas Daniilidis, and George J. Pappas. Robust and Efficient Semantic SLAM with Semantic Keypoints. 2020.
- [25] Max Bramer. *Principles of Data Mining*. Springer, 4th edition edition, 2020. ISBN 978-1-4471-7493-6.
- [26] Philip R. Brevington and D. Keith Robinson. *Principles of Data Mining*. McGraw Hill, 3rd edition edition, 2003. ISBN 0-07-247227-8.
- [27] Alberto Broggi, Michele Buzzoni, Stefano Debattisti, Paolo Grisleri, et al. Extensive Tests of Autonomous Driving Technologies. *IEEE Trans. Intell. Transp. Syst.*, 14(3):1403–1415, 2013. doi: 10.1109/TITS.2013.2262331. URL <https://doi.org/10.1109/TITS.2013.2262331>.
- [28] Markus Buhren and Yang Bin. Simulation of Automotive Radar Target Lists using a Novel Approach of Object Representation. In *2006 IEEE Intelligent Vehicles Symposium*, pages 314–319, 2006. doi: 10.1109/IVS.2006.1689647.

- [29] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robotics*, 32(6):1309–1332, 2016. doi: 10.1109/TRO.2016.2624754. URL <https://doi.org/10.1109/TRO.2016.2624754>.
- [30] 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. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11618–11628. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01164. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html.
- [31] Hanwen Cao, Yongyi Lu, Bo Pang, Cewu Lu, Alan L. Yuille, and Gongshen Liu. ASAP-Net: Attention and Structure Aware Point Cloud Sequence Segmentation. In *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020. URL <https://www.bmvc2020-conference.com/assets/papers/0129.pdf>.
- [32] Sarah H. Cen and Paul Newman. Precise Ego-Motion Estimation with Millimeter-Wave Radar Under Diverse and Challenging Conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6045–6052, 2018. doi: 10.1109/ICRA.2018.8460687.
- [33] Sarah H. Cen and Paul Newman. Radar-only Ego-Motion Estimation in difficult Settings via Graph Matching. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 298–304, 2019. doi: 10.1109/ICRA.2019.8793990.
- [34] Alessandro Cennamo, Florian Kästner, and Anton Kummert. A Neural Network Based System for Efficient Semantic Segmentation of Radar Point Clouds. *Neural Process. Lett.*, 53(5):3217–3235, 2021. doi: 10.1007/s11063-021-10544-4. URL <https://doi.org/10.1007/s11063-021-10544-4>.
- [35] Ping Chao, Chao-Yang Kao, Yu-Shan Ruan, Chien-Hsiang Huang, and Youn-Long Lin. HarDNet: A Low Memory Traffic Network. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3551–3560. IEEE, 2019. doi: 10.1109/ICCV.2019.00365. URL <https://doi.org/10.1109/ICCV.2019.00365>.
- [36] Yawgeng Chau, Hsiao-Lun Wang, and Hong-Han Chau. Parking Monitoring Based on FMCW Radar Imaging with Deep Transfer Learning. In *4th IEEE International*

- Conference on Knowledge Innovation and Invention, ICKII 2021, Taichung, Taiwan, July 23-25, 2021*, pages 151–154. IEEE, 2021. doi: 10.1109/ICKII51822.2021.9574723. URL <https://doi.org/10.1109/ICKII51822.2021.9574723>.
- [37] Chaofan Chen, Shengsheng Qian, Quan Fang, and Changsheng Xu. HAPGN: Hierarchical Attentive Pooling Graph Network for Point Cloud Segmentation. *IEEE Trans. Multim.*, 23:2335–2346, 2021. doi: 10.1109/TMM.2020.3009499. URL <https://doi.org/10.1109/TMM.2020.3009499>.
- [38] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, and Cyrill Stachniss. OverlapNet: Loop Closing for LiDAR-based SLAM. In *Robotics: Science and Systems XVI, Virtual Event / Corvallis, Oregon, USA, July 12-16, 2020*, 2020. doi: 10.15607/RSS.2020.XVI.009. URL <https://doi.org/10.15607/RSS.2020.XVI.009>.
- [39] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguère, Jens Behley, and Cyrill Stachniss. SuMa++: Efficient LiDAR-based Semantic SLAM. *CoRR*, abs/2105.11320, 2021. URL <https://arxiv.org/abs/2105.11320>.
- [40] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Jens Behley, and Cyrill Stachniss. OverlapNet: a siamese network for computing LiDAR scan similarity with applications to loop closing and localization. *Auton. Robots*, 46(1):61–81, 2022. doi: 10.1007/s10514-021-09999-0. URL <https://doi.org/10.1007/s10514-021-09999-0>.
- [41] Yang Chen and Gérard G. Medioni. Object modeling by registration of multiple range images. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9-11 April 1991*, pages 2724–2729. IEEE Computer Society, 1991. doi: 10.1109/ROBOT.1991.132043. URL <https://doi.org/10.1109/ROBOT.1991.132043>.
- [42] Yuwei Chen, Jian Tang, Changhui Jiang, Lingli Zhu, Matti Lehtomäki, Harri Kaartinen, Risto Kaijaluoto, Yiwu Wang, Juha Hyypä, Hannu Hyypä, Hui Zhou, Ling Pei, and Ruizhi Chen. The Accuracy Comparison of Three Simultaneous Localization and Mapping (SLAM)-Based Indoor Mapping Technologies. *Sensors*, 18(10):3228, 2018. doi: 10.3390/s18103228. URL <https://doi.org/10.3390/s18103228>.
- [43] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. (AF)2-S3Net: Attentive Feature Fusion With Adaptive Feature Selection for Sparse Semantic Segmentation Network. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12547–12556. Computer Vision Foundation / IEEE, 2021. URL <https://openaccess.thecvf.com/content/CVPR2021/html/Ch>

- eng_AF2-S3Net_Attentive_Feature_Fusion_With_Adaptive_Feature_Selection_for_Sparse_CVPR_2021_paper.html.
- [44] Ji Dong Choi and Min Young Kim. A Sensor Fusion System with Thermal Infrared Camera and LiDAR for Autonomous Vehicles and Deep Learning based Object Detection. *ICT Express*, 2022. ISSN 2405-9595. doi: <https://doi.org/10.1016/j.icte.2021.12.016>. URL <https://www.sciencedirect.com/science/article/pii/S2405959521001818>.
- [45] Francois Chollet. *Deep Learning with Python*. Manning, 2017. ISBN 9781617294433.
- [46] Dieter Conrads, W. Plaßmann, E. Döring, D. Schulz, et al. *Handbuch Elektrotechnik: Grundlagen und Anwendungen für Elektrotechniker*. Vieweg Praxis. Vieweg+Teubner Verlag, 2009. ISBN 9783834892454.
- [47] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.350. URL <https://doi.org/10.1109/CVPR.2016.350>.
- [48] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of LiDAR Point Clouds. In *Advances in Visual Computing - 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5-7, 2020, Proceedings, Part II*, volume 12510 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2020. doi: 10.1007/978-3-030-64559-5_16. URL https://doi.org/10.1007/978-3-030-64559-5_16.
- [49] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-Balanced Loss Based on Effective Number of Samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9268–9277. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00949. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Cui_Class-Balanced_Loss_Based_on_Effective_Number_of_Samples_CVPR_2019_paper.html.
- [50] Andreas Danzer, Thomas Griebel, Martin Bach, and Klaus Dietmayer. 2D Car Detection in Radar Data with PointNets. In *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, October 27-30, 2019*, pages 61–66. IEEE, 2019. doi: 10.1109/ITSC.2019.8917000. URL <https://doi.org/10.1109/ITSC.2019.8917000>.

- [51] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 195–205. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00028. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Deng_PPFNet_Global_Context_CVPR_2018_paper.html.
- [52] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPF-FoldNet: Unsupervised Learning of Rotation Invariant 3D Local Descriptors. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, volume 11209 of *Lecture Notes in Computer Science*, pages 620–638. Springer, 2018. doi: 10.1007/978-3-030-01228-1_37. URL https://doi.org/10.1007/978-3-030-01228-1_37.
- [53] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global Context Aware Local Features for Robust 3D Point Matching. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 195–205. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00028. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Deng_PPFNet_Global_Context_CVPR_2018_paper.html.
- [54] Lee Raymond Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26:297–302, 1945.
- [55] Fabian Diewald, Jens Klappstein, Frederik Sarholz, Jürgen Dickmann, and Klaus Dietmayer. Radar-interference-based Bridge Identification for Collision Avoidance Systems. In *IEEE Intelligent Vehicles Symposium (IV), 2011, Baden-Baden, Germany, June 5-9, 2011*, pages 113–118. IEEE, 2011. doi: 10.1109/IVS.2011.5940422. URL <https://doi.org/10.1109/IVS.2011.5940422>.
- [56] Kevin Doherty. *Robust non-Gaussian Semantic Simultaneous Localization and Mapping*. PhD thesis, Massachusetts Institute of Technology, 01 2019.
- [57] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.
- [58] Maria Dreher, Emec Ercelik, Timo Bänziger, and Alois C. Knoll. Radar-based 2D Car Detection Using Deep Neural Networks. In *23rd IEEE International Conference on Intelligent Transportation Systems, ITSC 2020, Rhodes, Greece, September 20-23, 2020*, pages 1–8. IEEE, 2020. doi: 10.1109/ITSC45102.2020.9294546. URL <https://doi.org/10.1109/ITSC45102.2020.9294546>.

- [59] Liya Duan, Renxia Wu, Xin Wang, Yinghe Liu, Yongjie Ma, and Hui Fang. ORB-SLAM based ICP and Optical Flow Combination Registration Algorithm in 3D Dense Reconstruction. In *Proceedings of the 2nd International Conference on Intelligent Manufacturing and Materials - Volume 1: ICIMM*, pages 537–541. INSTICC, SciTePress, 2018. ISBN 978-989-758-345-2. doi: 10.5220/0007535205370541.
- [60] Herbert Edelsbrunner, David G. Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inf. Theory*, 29(4):551–558, 1983. doi: 10.1109/TIT.1983.1056714. URL <https://doi.org/10.1109/TIT.1983.1056714>.
- [61] Christian Eggenberger. Automated Valet Parking a Cinderella or an emerging Princess, 11 2021.
- [62] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11131 of *Lecture Notes in Computer Science*, pages 395–409. Springer, 2018. doi: 10.1007/978-3-030-11015-4_29. URL https://doi.org/10.1007/978-3-030-11015-4_29.
- [63] Florian Engels, Philipp Heidenreich, Abdelhak M. Zoubir, Friedrich K. Jondral, and Markus Wintermantel. Advances in Automotive Radar: A framework on computationally efficient high-resolution frequency estimation. *IEEE Signal Process. Mag.*, 34(2): 36–46, 2017. doi: 10.1109/MSP.2016.2637700. URL <https://doi.org/10.1109/MSP.2016.2637700>.
- [64] Máté Fazekas, Péter Gáspár, and Balázs Németh. Calibration and Improvement of an Odometry Model with Dynamic Wheel and Lateral Dynamics Integration. *Sensors*, 21(2):337, 2021. doi: 10.3390/s21020337. URL <https://doi.org/10.3390/s21020337>.
- [65] Di Feng, Christian Haase-Schutz, Lars Rosenbaum, Heinz Hertlein, Claudius Gläser, Fabian Timm, w. Wiesbeck, and Klaus Dietmayer. Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–20, 02 2020. doi: 10.1109/TITS.2020.2972974.
- [66] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395, 1981. doi: 10.1145/358669.358692. URL <http://doi.acm.org/10.1145/358669.358692>.
- [67] Takahiko Furuya, Xu Hang, Ryutarou Ohbuchi, and Jinliang Yao. Convolution on Rotation-Invariant and Multi-Scale Feature Graph for 3D Point Set Segmentation.

- IEEE Access*, 8:140250–140260, 2020. doi: 10.1109/ACCESS.2020.3012613. URL <https://doi.org/10.1109/ACCESS.2020.3012613>.
- [68] Matthew Gadd, Daniele De Martini, and Paul Newman. Look Around You: Sequence-based Radar Place Recognition with Learned Rotational Invariance. In *IEEE/ION Position, Location and Navigation Symposium, PLANS 2020, Portland, OR, USA, April 20-23, 2020*, pages 270–276. IEEE, 2020. doi: 10.1109/PLANS46316.2020.9109951. URL <https://doi.org/10.1109/PLANS46316.2020.9109951>.
- [69] Yarín Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org, 2016. URL <http://proceedings.mlr.press/v48/gal16.html>.
- [70] Jonah Gamba. *Radar Signal Processing for Autonomous Driving*. Springer Nature Singapore Pte Ltd., 1st edition edition, 2020. ISBN 978-981-13-9193-4. doi: <https://doi.org/10.1007/978-981-13-9193-4>.
- [71] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [72] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. DropBlock: A Regularization Method for convolutional Networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10750–10760, 2018. URL <https://proceedings.neurips.cc/paper/2018/hash/7edcfb2d8f6a659ef4cd1e6c9b6d7079-Abstract.html>.
- [73] Robert Bosch GmbH. Parken neu erleben: Mit vernetzten und automatisierten Parklösungen, 2022. URL <https://www.bosch-mobility-solutions.com/de/mobility-themen/vernetztes-und-automatisiertes-parken/>.
- [74] Robert Bosch GmbH. Weltpremiere: Bosch und Daimler erhalten Zulassung für fahrerloses Parken ohne menschliche Überwachung, 2022. URL <https://www.bosch-mobility-solutions.com/de/mobility-themen/vernetztes-und-automatisiertes-parken/>.
- [75] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging Into Self-Supervised Monocular Depth Estimation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3827–3837. IEEE, 2019. doi: 10.1109/ICCV.2019.00393. URL <https://doi.org/10.1109/ICCV.2019.00393>.

- [76] Zan Gojcic, Caifa Zhou, Jan D. Wegner, and Andreas Wieser. The Perfect Match: 3D Point Cloud Matching With Smoothed Densities. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5545–5554. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00569. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Gojcic_The_Perfect_Match_3D_Point_Cloud_Matching_With_Smoothed_Densities_CVPR_2019_paper.html.
- [77] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [78] Google. Google map of Heimsheim, 2022. URL <https://www.google.com/maps/@48.8104573,8.8588816,289a,35y,274.13h/data=!3m1!1e3!5m1!1e4>.
- [79] Google. Google map of Weissach, 2022. URL <https://www.google.com/maps/@48.8456305,8.9060049,288a,35y,306.18h/data=!3m1!1e3!5m1!1e4>.
- [80] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation With Submanifold Sparse Convolutional Networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9224–9232. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00961. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Graham_3D_Semantic_Segmentation_CVPR_2018_paper.html.
- [81] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. Robotics*, 23(1):34–46, 2007. doi: 10.1109/TRO.2006.889486. URL <https://doi.org/10.1109/TRO.2006.889486>.
- [82] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A Tutorial on Graph-Based SLAM. *IEEE Intell. Transp. Syst. Mag.*, 2(4):31–43, 2010. doi: 10.1109/MITS.2010.939925. URL <https://doi.org/10.1109/MITS.2010.939925>.
- [83] Erico Guizzo. How google’s self-driving car works, 2011. URL <https://spectrum.ieee.org/how-google-self-driving-car-works>.
- [84] Yulan Guo, Ferdous A. Sohel, Mohammed Bennamoun, Jianwei Wan, and Min Lu. RoPS: A local Feature Descriptor for 3D Rigid Objects based on Rotational Projection Statistics. In *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, pages 1–6, 2013. doi: 10.1109/ICCSPA.2013.6487310.
- [85] Andre-Marcel Hellmund, Sascha Wirges, Ömer Sahin Tas, Claudio Bandera, and Niels Ole Salscheider. Robot Operating System: A modular software framework for

- automated driving. In *19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016, Rio de Janeiro, Brazil, November 1-4, 2016*, pages 1564–1570. IEEE, 2016. doi: 10.1109/ITSC.2016.7795766. URL <https://doi.org/10.1109/ITSC.2016.7795766>.
- [86] Christoph Hertzberg. *A Framework for Sparse, Non-Linear Least Square Problems on Manifolds*. PhD thesis, University of Bremen, 11 2008.
- [87] Marian Himstedt, Jan Frost, Sven Hellbach, Hans-Joachim Böhme, and Erik Maehle. Large scale place recognition in 2D LIDAR scans using Geometrical Landmark Relations. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 5030–5035. IEEE, 2014. doi: 10.1109/IROS.2014.6943277. URL <https://doi.org/10.1109/IROS.2014.6943277>.
- [88] Yaoshiang Ho and Samuel Wookey. The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling. *IEEE Access*, 8:4806–4813, 2020. doi: 10.1109/ACCESS.2019.2962617. URL <https://doi.org/10.1109/ACCESS.2019.2962617>.
- [89] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [90] Martin Holder, Sven Hellwig, and Hermann Winner. Real-Time Pose Graph SLSM based on Radar. In *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*, pages 1145–1151. IEEE, 2019. doi: 10.1109/IVS.2019.8813841. URL <https://doi.org/10.1109/IVS.2019.8813841>.
- [91] Martin F. Holder, Clemens Linnhoff, Philipp Rosenberger, Christoph Popp, and Prof. Dr. Hermann Winner. Modeling and Simulation of Radar Sensor Artifacts for Virtual Testing of Autonomous Driving. In *9. Tagung Automatisiertes Fahren*, München, 2019. Lehrstuhl für Fahrzeugtechnik mit TÜV SÜD Akademie.
- [92] Hyunki Hong and B. H. Lee. Dynamic Scaling Factors of Covariances for Accurate 3D Normal Distributions Transform Registration. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*, pages 1190–1196. IEEE, 2018. doi: 10.1109/IROS.2018.8593839. URL <https://doi.org/10.1109/IROS.2018.8593839>.
- [93] Ziyang Hong, Yvan R. Petillot, and Sen Wang. RadarSLAM: Radar based Large-Scale SLAM in All Weathers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, October 24, 2020 - January 24, 2021*, pages 5164–5170. IEEE, 2020. doi: 10.1109/IROS45743.2020.9341287. URL <https://doi.org/10.1109/IROS45743.2020.9341287>.

- [94] Jiang Hu, Xin Liu, Zaiwen Wenn, and Ya xiang Yuan. A Brief Introduction to Manifold Optimization. *Journal of the Operations Research Society of China*, 8:199–248, 2020. doi: <https://doi.org/10.1007/s40305-020-00295-9>.
- [95] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11105–11114. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01112. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Hu_RandLA-Net_Efficient_Semantic_Segmentation_of_Large-Scale_Point_Clouds_CVPR_2020_paper.html.
- [96] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-Guided Graph Neural Networks for Referring 3D Instance Segmentation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 1610–1618. AAAI Press, 2021. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16253>.
- [97] Zhongling Huang, Corneliu Octavian Dumitru, Zongxu Pan, Bin Lei, and Mihai Datcu. Classification of Large-Scale High-Resolution SAR Images With Deep Transfer Learning. *IEEE Geosci. Remote. Sens. Lett.*, 18(1):107–111, 2021. doi: 10.1109/LGRS.2020.2965558. URL <https://doi.org/10.1109/LGRS.2020.2965558>.
- [98] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. doi: 10.1214/aoms/1177703732. URL <https://doi.org/10.1214/aoms/1177703732>.
- [99] Philipp Hügler, Fabian Roos, Markus ScharTEL, Martin Geiger, and Christian Waldschmidt. Radar Taking Off: New Capabilities for UAVs. *IEEE Microwave Magazine*, 19(7):43–53, 2018. doi: 10.1109/MMM.2018.2862558.
- [100] Philipp Hügler, Timo Grebner, Christina Knill, and Christian Waldschmidt. UAV-Borne 2-D and 3-D Radar-Based Grid Mapping. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. doi: 10.1109/LGRS.2020.3025109.
- [101] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015. URL <http://proceedings.mlr.press/v37/ioffe15.html>.

- [102] ISO Central Secretary. Road vehicles — Vehicle dynamics and road-holding ability — Vocabulary. Standard ISO 8855:2011, International Organization for Standardization, Geneva, CH, 2013-11.
- [103] Paul Jaccard. Etude de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:547–579, 01 1901. doi: 10.5169/seals-266450.
- [104] Shruti Jadon. A Survey of Loss Functions for Semantic Segmentation. In *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2020, Viña del Mar, Chile, October 27-29, 2020*, pages 1–7. IEEE, 2020. doi: 10.1109/CIBCB48159.2020.9277638. URL <https://doi.org/10.1109/CIBCB48159.2020.9277638>.
- [105] Weipeng Jing, Wenjun Zhang, Linhui Li, Donglin Di, Guangsheng Chen, and Jian Wang. AGNet: An Attention-Based Graph Network for Point Cloud Classification and Segmentation. *Remote. Sens.*, 14(4):1036, 2022. doi: 10.3390/rs14041036. URL <https://doi.org/10.3390/rs14041036>.
- [106] J.E. Dennis Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 2nd edition edition, 1996. ISBN 0-89871-364-1.
- [107] Lu Jun, Lui Wei, Dong Donglai, and Shao Qiang. Point Cloud Registration Algorithm based on NDT with variable Size Voxel. In *2015 34th Chinese Control Conference (CCC)*, pages 3707–3712, 2015. doi: 10.1109/ChiCC.2015.7260213.
- [108] Ho Gi Jung, Young Ha Cho, Pal Joo Yoon, and Jaihie Kim. Scanning Laser Radar-Based Target Position Designation for Parking Aid System. *IEEE Trans. Intell. Transp. Syst.*, 9(3):406–424, 2008. doi: 10.1109/TITS.2008.922980. URL <https://doi.org/10.1109/TITS.2008.922980>.
- [109] Yao-Chiang Kan, Kuan-Tzu Chen, Hsueh-Chun Lin, and Junghsi Lee. A Parking Monitoring System Using FMCW Radars. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2021, Tokyo, Japan, December 14-17, 2021*, pages 1931–1934. IEEE, 2021. URL <https://ieeexplore.ieee.org/document/9689577>.
- [110] Prannay Kaul, Daniele De Martini, Matthew Gadd, and Paul Newman. RSS-Net: Weakly-Supervised Multi-Class Semantic Segmentation with FMCW Radar. In *IEEE Intelligent Vehicles Symposium, IV 2020, Las Vegas, NV, USA, October 19 - November 13, 2020*, pages 431–436. IEEE, 2020. doi: 10.1109/IV47402.2020.9304674. URL <https://doi.org/10.1109/IV47402.2020.9304674>.

- [111] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5574–5584, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html>.
- [112] Tobias Kessler, Julian Bernhard, Martin Buechel, Klemens Esterle, et al. Bridging the Gap between Open Source Software and Vehicle Hardware for Autonomous Driving. In *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*, pages 1612–1619. IEEE, 2019. doi: 10.1109/IVS.2019.8813784. URL <https://doi.org/10.1109/IVS.2019.8813784>.
- [113] Nikhil Ketkar. *Deep Learning with Python - A Hands-on Introduction*. Apress, 2017. ISBN 978-1-4842-2766-4.
- [114] Hella GmbH Co. KGaA. Der 77 GHz Radar Sensor: Fahrerassistenzsystem für On und Off-Highway Anwendungen, 2022. URL <https://www.hella.com/soe/de/Elektrik-und-Elektronik-Fahrerassistenzsysteme-Radarsensoren-77-GHz-2351/>.
- [115] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. MulRan: Multimodal Range Dataset for Urban Place Recognition. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 6246–6253. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9197298. URL <https://doi.org/10.1109/ICRA40945.2020.9197298>.
- [116] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6980>.
- [117] Roman Klokov and Victor S. Lempitsky. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 863–872. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.99. URL <https://doi.org/10.1109/ICCV.2017.99>.
- [118] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012. URL <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>.

- [119] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G^2o : A general Framework for Graph Optimization. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pages 3607–3613. IEEE, 2011. doi: 10.1109/ICRA.2011.5979949. URL <https://doi.org/10.1109/ICRA.2011.5979949>.
- [120] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From Big to Small: Multi-Scale Local Planar Guidance for Monocular Depth Estimation. *CoRR*, abs/1907.10326, 2019. URL <http://arxiv.org/abs/1907.10326>.
- [121] Nadav Levanon and Eli Mozeson. *Radar Signals*. John Wiley & Sons, Inc., 1st edition edition, 2004. ISBN 0-471-47378-2.
- [122] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, et al. Towards fully autonomous driving: Systems and algorithms. In *IEEE Intelligent Vehicles Symposium (IV), 2011, Baden-Baden, Germany, June 5-9, 2011*, pages 163–168. IEEE, 2011. doi: 10.1109/IVS.2011.5940562. URL <https://doi.org/10.1109/IVS.2011.5940562>.
- [123] Jian Li and Petre Stoica. *MIMO radar signal processing*. John Wiley & Sons, Inc., 1st edition edition, 2009. ISBN 978-0-470-17898-0.
- [124] Kaicheng Li, Biruk K. Habtemariam, Ratnasingham Tharmarasa, Michel Pelletier, and Thia Kirubarajan. Multitarget Tracking with Doppler Ambiguity. *IEEE Trans. Aerosp. Electron. Syst.*, 49(4):2640–2656, 2013. doi: 10.1109/TAES.2013.6621842. URL <https://doi.org/10.1109/TAES.2013.6621842>.
- [125] Yang Li, Yutong Liu, Yanping Wang, Yun Lin, and Wenjie Shen. The Millimeter-Wave Radar SLAM Assisted by the RCS Feature of the Target and IMU. *Sensors*, 20(18):5421–5429, 2020. doi: 10.3390/s20185421. URL <https://doi.org/10.3390/s20185421>.
- [126] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution On X-Transformed Points. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 828–838, 2018. URL <https://proceedings.neurips.cc/paper/2018/hash/f5f8590cd58a54e94377e6ae2eded4d9-Abstract.html>.
- [127] Ying Li, Lingfei Ma, Zilong Zhong, Dongpu Cao, and Jonathan Li. TGNet: Geometric Graph CNN on 3-D Point Cloud Segmentation. *IEEE Trans. Geosci. Remote. Sens.*, 58(5): 3588–3600, 2020. doi: 10.1109/TGRS.2019.2958517. URL <https://doi.org/10.1109/TGRS.2019.2958517>.

- [128] Jianan Liu, Weiyi Xiong, Liping Bai, Yuxuan Xia, and Bing Zhu. Deep Instance Segmentation with High-Resolution Automotive Radar. *CoRR*, abs/2110.01775, 2021. URL <https://arxiv.org/abs/2110.01775>.
- [129] Jakob Lombacher, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Potential of radar for static object classification using deep learning methods. In *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility ICMIM*, pages 1–4, 2016. doi: 10.1109/ICMIM.2016.7533931.
- [130] Jakob Lombacher, Kilian Laudt, Markus Hahn, Jürgen Dickmann, and Christian Wöhler. Semantic Radar Grids. In *IEEE Intelligent Vehicles Symposium, IV 2017, Los Angeles, CA, USA, June 11-14, 2017*, pages 1170–1175. IEEE, 2017. doi: 10.1109/IVS.2017.7995871. URL <https://doi.org/10.1109/IVS.2017.7995871>.
- [131] Maurice W. Long and R. A. Llamas. CA-CFAR Performance with linear, square-law, and fourth-power Detectors. *2011 IEEE RadarCon (RADAR)*, pages 350–355, 2011.
- [132] Antonio Loquercio, Mattia Segù, and Davide Scaramuzza. A General Framework for Uncertainty Estimation in Deep Learning. *IEEE Robotics Autom. Lett.*, 5(2):3153–3160, 2020. doi: 10.1109/LRA.2020.2974682. URL <https://doi.org/10.1109/LRA.2020.2974682>.
- [133] Kok-Lim Low. Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. *Department of Computer Science, University of North Carolina at Chapel Hill*, 04(TR04-004), 2004.
- [134] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deep-ICP: An End-to-End Deep Neural Network for 3D Point Cloud Registration. *CoRR*, abs/1905.04153, 2019. URL <http://arxiv.org/abs/1905.04153>.
- [135] Qi Luo, Romesh Saigal, Robert C. Hampshire, and Xinyi Wu. A Statistical Method for Parking Spaces Occupancy Detection via Automotive Radars. In *85th IEEE Vehicular Technology Conference, VTC Spring 2017, Sydney, Australia, June 4-7, 2017*, pages 1–5. IEEE, 2017. doi: 10.1109/VTCSpring.2017.8108418. URL <https://doi.org/10.1109/VTCSpring.2017.8108418>.
- [136] Michael Lutz and Monsij Biswal. Supervised Noise Reduction for Clustering on Automotive 4D Radar. In *IEEE Symposium Series on Computational Intelligence, SSCI 2021, Orlando, FL, USA, December 5-7, 2021*, pages 1–7. IEEE, 2021. doi: 10.1109/SSCI50451.2021.9659953. URL <https://doi.org/10.1109/SSCI50451.2021.9659953>.
- [137] Martin Magnusson. *The Three-Dimensional Normal-Distributions Transform - an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, Örebro University, 12 2009.

- [138] Martin Magnusson, Henrik Andreasson, Andreas Nüchter, and Achim J. Lilienthal. Automatic appearance-based loop detection from three-dimensional laser data using the normal distributions transform. *J. Field Robotics*, 26(11-12):892–914, 2009. doi: 10.1002/rob.20314. URL <https://doi.org/10.1002/rob.20314>.
- [139] Martin Magnusson, Andreas Nüchter, Christopher Lörken, Achim J. Lilienthal, and Joachim Hertzberg. Evaluation of 3D Registration Reliability and Speed - A Comparison of ICP and NDT. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*, pages 3907–3912. IEEE, 2009. doi: 10.1109/ROBOT.2009.5152538. URL <https://doi.org/10.1109/ROBOT.2009.5152538>.
- [140] Jan Willem Marck, Ali Mohamoud, Eriv van Houwen, and Rob van Heijster. Indoor Radar SLAM: A Radar Application for Vision and GPS denied Environments. In *2013 European Radar Conference*, pages 471–474, 2013.
- [141] Mladen Mazuran, Wolfram Burgard, and Gian Diego Tipaldi. Nonlinear Factor Recovery for Long-Term SLAM, journal = Int. J. Robotics Res., volume = 35, number = 1-3, pages = 50–72, year = 2016, url = <https://doi.org/10.1177/0278364915581629>, doi = 10.1177/0278364915581629, timestamp = Thu, 17 Sep 2020 12:02:34 +0200, biburl = <https://dblp.org/rec/journals/ijrr/MazuranBT16.bib>, bibsource = dblp computer science bibliography, <https://dblp.org>.
- [142] Michael Meyer and Georg Kuschik. Automotive Radar Dataset for Deep Learning Based 3D Object Detection. In *2019 16th European Radar Conference (EuRAD)*, pages 129–132, 2019.
- [143] Michael Meyer and Georg Kuschik. Deep Learning Based 3D Object Detection for Automotive Radar and Camera. *2019 16th European Radar Conference (EuRAD)*, pages 133–136, 2019.
- [144] Michael Meyer, Georg Kuschik, and Sven Tomforde. Graph Convolutional Networks for 3D Object Detection on Radar Data. In *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*, pages 3053–3062. IEEE, 2021. doi: 10.1109/ICCVW54120.2021.00340. URL <https://doi.org/10.1109/ICCVW54120.2021.00340>.
- [145] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*, pages 4213–4220. IEEE, 2019. doi: 10.1109/IROS40897.2019.8967762. URL <https://doi.org/10.1109/IROS40897.2019.8967762>.
- [146] Andrew Moore. Efficient Memory-based Learning for Robot Control. Technical report, Carnegie Mellon University, Pittsburgh, PA, November 1990.

- [147] Mohammadreza Mostajabi, Ching Ming Wang, Darsh Ranjan, and Gilbert Hsyu. High Resolution Radar Dataset for Semi-Supervised Learning of Dynamic Objects. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pages 450–457. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPRW50498.2020.00058. URL https://openaccess.thecvf.com/content_CVPRW_2020/html/w6/Mostajabi_High-Resolution_Radar_Dataset_for_Semi-Supervised_Learning_of_Dynamic_Objects_CVPRW_2020_paper.html.
- [148] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *CoRR*, abs/1610.06475, 2016. URL <http://arxiv.org/abs/1610.06475>.
- [149] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015. doi: 10.1109/TRO.2015.2463671. URL <https://doi.org/10.1109/TRO.2015.2463671>.
- [150] Filip Nálepa, Michal Batko, and Pavel Zezula. Continuous Time-Dependent KNN Join by Binary Sketches. In *Proceedings of the 22nd International Database Engineering & Applications Symposium, IDEAS 2018*, pages 64–73, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450365277. doi: 10.1145/3216122.3216159. URL <https://doi.org/10.1145/3216122.3216159>.
- [151] Lakshay Narula, Peter Anthony Iannucci, and Todd E. Humphreys. Automotive-Radar-Based 50-cm Urban Positioning. In *IEEE/ION Position, Location and Navigation Symposium, PLANS 2020, Portland, OR, USA, April 20-23, 2020*, pages 856–867. IEEE, 2020. doi: 10.1109/PLANS46316.2020.9109917. URL <https://doi.org/10.1109/PLANS46316.2020.9109917>.
- [152] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. QuadricSLAM: Dual Quadrics From Object Detections as Landmarks in Object-Oriented SLAM. *IEEE Robotics Autom. Lett.*, 4(1):1–8, 2019. doi: 10.1109/LRA.2018.2866205. URL <https://doi.org/10.1109/LRA.2018.2866205>.
- [153] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection. In *2019 Sensor Data Fusion: Trends, Solutions, Applications, SDF 2019, Bonn, Germany, October 15-17, 2019*, pages 1–7. IEEE, 2019. doi: 10.1109/SDF.2019.8916629. URL <https://doi.org/10.1109/SDF.2019.8916629>.

- [154] Edwin Olson. M3RSM: Many-to-Many Multi-Resolution Scan Matching. In *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, pages 5815–5821. IEEE, 2015. doi: 10.1109/ICRA.2015.7140013. URL <https://doi.org/10.1109/ICRA.2015.7140013>.
- [155] Eva Ortlieb. Entwicklung eines Qualitätsmaßes für Landmarken. Technical report, Universität Hannover, 2006. URL <http://www.ikg.uni-hannover.de/index.php?id=276>.
- [156] Arthur Ouaknine, Alasdair Newson, Julien Rebut, Florence Tupin, and Patrick Pérez. CARRADA Dataset: Camera and Automotive Radar with Range- Angle- Doppler Annotations. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pages 5068–5075. IEEE, 2020. doi: 10.1109/ICPR48806.2021.9413181. URL <https://doi.org/10.1109/ICPR48806.2021.9413181>.
- [157] Arthur Ouaknine, Alasdair Newson, Patrick Pérez, Florence Tupin, and Julien Rebut. Multi-View Radar Semantic Segmentation. *CoRR*, abs/2103.16214, 2021. URL <https://arxiv.org/abs/2103.16214>.
- [158] Yeong Sang Park, Young-Sik Shin, and Ayoung Kim. PhaRaO: Direct Radar Odometry using Phase Correlation. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 2617–2623. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9197231. URL <https://doi.org/10.1109/ICRA40945.2020.9197231>.
- [159] Kanil Patel, Kilian Rambach, Tristan Visentin, Daniel Rusev, Michael Pfeiffer, and Bin Yang. Deep Learning-based Object Classification on Automotive Radar Spectra. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–6, 07 2019. doi: 10.1109/RADAR.2019.8835775.
- [160] Sujeet Patole, Murat Torlak, Dan Wang, and Murtaza Ali. Automotive Radars: A review of signal processing techniques. *IEEE Signal Process. Mag.*, 34(2):22–35, 2017. doi: 10.1109/MSP.2016.2628914. URL <https://doi.org/10.1109/MSP.2016.2628914>.
- [161] Karl Pearson. Note on Regression and Inheritance in the Case of two Parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895. doi: <https://doi.org/10.1098/rspl.1895.0041>.
- [162] Biber Peter and Straßer Wolfgang. The Normal Distributions Transform: A New Approach to Laser Scan Matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 3, pages 2743–2748, 11 2003. ISBN 0-7803-7860-1. doi: 10.1109/IROS.2003.1249285.

- [163] Florian Piewak. *LiDAR-based Semantic Labeling: Automotive 3D Scene Understanding*. PhD thesis, Karlsruhe Institute of Technology, Germany, 2020. URL <https://nbn-resolving.org/urn:nbn:de:101:1-2020062404584222571324>.
- [164] Florian Piewak. *LiDAR-based Semantic Labeling: Automotive 3D Scene Understanding*. PhD thesis, Karlsruhe Institute of Technology, Germany, 2020. URL <https://nbn-resolving.org/urn:nbn:de:101:1-2020062404584222571324>.
- [165] Florian Piewak, Peter Pinggera, and Marius Zöllner. Analyzing the Cross-Sensor Portability of Neural Network Architectures for LiDAR-based Semantic Labeling. In *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, October 27-30, 2019*, pages 3419–3426. IEEE, 2019. doi: 10.1109/ITSC.2019.8917412. URL <https://doi.org/10.1109/ITSC.2019.8917412>.
- [166] Vasyl Pihur, Susmita Datta, and Somnath Datta. Weighted Rank Aggregation of Cluster Validation Measures: A Monte Carlo Cross-Entropy Approach. *Bioinform.*, 23(13):1607–1615, 2007. doi: 10.1093/bioinformatics/btm158. URL <https://doi.org/10.1093/bioinformatics/btm158>.
- [167] Robert Prophet, Marcel Hoffmann, Martin Vossiek, Li Gang, and Christian Sturm. Parking Space Detection from a Radar-based Target List. In *2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 91–94, 2017. doi: 10.1109/ICMIM.2017.7918864.
- [168] Robert Prophet, Gang Li, Christian Sturm, and Martin Vossiek. Semantic Segmentation on Automotive Radar Maps. In *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*, pages 756–763. IEEE, 2019. doi: 10.1109/IVS.2019.8813808. URL <https://doi.org/10.1109/IVS.2019.8813808>.
- [169] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.16. URL <https://doi.org/10.1109/CVPR.2017.16>.
- [170] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5099–5108, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/d8bf84be3800d12f74d8b05e9b89836f-Abstract.html>.

- [171] Yunbo Rao, Menghan Zhang, Zhanglin Cheng, Junmin Xue, Jiansu Pu, and Zairong Wang. Semantic Point Cloud Segmentation Using Fast Deep Neural Network and DCRF. *Sensors*, 21(8):2731, 2021. doi: 10.3390/s21082731. URL <https://doi.org/10.3390/s21082731>.
- [172] Matthias Rapp, Klaus C. J. Dietmayer, Markus Hahn, Frank Schuster, Jakob Lombacher, and Jürgen Dickmann. FSCD and BASD: Robust Landmark Detection and Description on Radar-based Grids. *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, pages 1–4, 2016.
- [173] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *CoRR*, abs/1804.02767, 2018. URL <http://arxiv.org/abs/1804.02767>.
- [174] Marc A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill, 1st edition edition, 2005. ISBN 9780071444743.
- [175] Philip R. Bevington; D. Keith Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill, New York, 2002. ISBN 0-07-91124-9. doi: <https://doi.org/10.1063/1.4823194>.
- [176] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28. URL https://doi.org/10.1007/978-3-319-24574-4_28.
- [177] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Representations by Back-Propagating Errors. *Nature*, 323:533–536, 1986.
- [178] Radu Rusu and Steve Cousins. 3D is here: Point cloud library (PCL). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4, 05 2011. doi: 10.1109/ICRA.2011.5980567.
- [179] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*, pages 3212–3217. IEEE, 2009. doi: 10.1109/ROBOT.2009.5152473. URL <https://doi.org/10.1109/ROBOT.2009.5152473>.
- [180] Stefan Saftescu, Matthew Gadd, Daniele De Martini, Dan Barnes, and Paul Newman. Kidnapped Radar: Topological Radar Localisation using Rotationally-Invariant Metric Learning. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 4358–4364. IEEE, 2020. doi: 10.

- 1109/ICRA40945.2020.9196682. URL <https://doi.org/10.1109/ICRA40945.2020.9196682>.
- [181] Samuele Salti, Federico Tombari, and Luigi di Stefano. SHOT: Unique Signatures of Histograms for Surface and Texture Description. *Comput. Vis. Image Underst.*, 125: 251–264, 2014. doi: 10.1016/j.cviu.2014.04.011. URL <https://doi.org/10.1016/j.cviu.2014.04.011>.
- [182] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A Toolbox for Easily Calibrating Omnidirectional Cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006, October 9-15, 2006, Beijing, China*, pages 5695–5701. IEEE, 2006. doi: 10.1109/IROS.2006.282372. URL <https://doi.org/10.1109/IROS.2006.282372>.
- [183] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Trans. Neural Networks*, 20(1): 61–80, 2009. doi: 10.1109/TNN.2008.2005605. URL <https://doi.org/10.1109/TNN.2008.2005605>.
- [184] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Radar-based Feature Design and Multiclass Classification for Road User Recognition. In *2018 IEEE Intelligent Vehicles Symposium, IV 2018, Changshu, Suzhou, China, June 26-30, 2018*, pages 779–786. IEEE, 2018. doi: 10.1109/IVS.2018.8500607. URL <https://doi.org/10.1109/IVS.2018.8500607>.
- [185] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Radar-based Road User Classification and Novelty Detection with Recurrent Neural Network Ensembles. In *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*, pages 722–729. IEEE, 2019. doi: 10.1109/IVS.2019.8813773. URL <https://doi.org/10.1109/IVS.2019.8813773>.
- [186] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. A Multi-Stage Clustering Framework for Automotive Radar Data. In *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, October 27-30, 2019*, pages 2060–2067. IEEE, 2019. doi: 10.1109/ITSC.2019.8916873. URL <https://doi.org/10.1109/ITSC.2019.8916873>.
- [187] 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. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2065–2074. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00214. URL

https://openaccess.thecvf.com/content_CVPR_2020/html/Scheiner_Seeing_Around_Street_Corners_Non-Line-of-Sight_Detection_and_Tracking_In-the-Wild_Using_CVPR_2020_paper.html.

- [188] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2):214–226, 2007. doi: <https://doi.org/10.1111/j.1467-8659.2007.01016.x>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01016.x>.
- [189] Markus Schoen, Markus Horn, Markus Hahn, and Juergen Dickmann. Real-Time Radar SLAM. Technical report, Universität Ulm, 2016.
- [190] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4104–4113. IEEE Computer Society, 2016. doi: [10.1109/CVPR.2016.445](https://doi.org/10.1109/CVPR.2016.445). URL <https://doi.org/10.1109/CVPR.2016.445>.
- [191] Cornelia Schulz and Andreas Zell. Real-Time Graph-Based SLAM with Occupancy Normal Distributions Transforms. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 3106–3111. IEEE, 2020. doi: [10.1109/ICRA40945.2020.9197325](https://doi.org/10.1109/ICRA40945.2020.9197325). URL <https://doi.org/10.1109/ICRA40945.2020.9197325>.
- [192] Ole Schumann. *Machine learning applied to radar data: Classification and semantic instance segmentation of moving road users*. PhD thesis, Universität Dortmund, 2021. URL <http://dx.doi.org/10.17877/DE290R-22034>.
- [193] Ole Schumann, Jakob Lombacher, Markus Hahn, Christian Wöhler, and Jürgen Dickmann. Scene Understanding With Automotive Radar. *IEEE Trans. Intell. Veh.*, 5(2): 188–203, 2020. doi: [10.1109/TIV.2019.2955853](https://doi.org/10.1109/TIV.2019.2955853). URL <https://doi.org/10.1109/TIV.2019.2955853>.
- [194] 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. In *24th IEEE International Conference on Information Fusion, FUSION 2021, Sun City, South Africa, November 1-4, 2021*, pages 1–8. IEEE, 2021. URL <https://ieeexplore.ieee.org/document/9627037>.
- [195] Frank Schuster, Christoph G. Keller, Matthias Rapp, Martin Haueis, and Cristobal Curio. Landmark based Radar SLAM using Graph Optimization. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2559–2564, 2016. doi: [10.1109/ITSC.2016.7795967](https://doi.org/10.1109/ITSC.2016.7795967).

- [196] Aleksandr Segal, Dirk Hähnel, and Sebastian Thrun. Generalized-ICP. In *Robotics: Science and Systems V, University of Washington, Seattle, USA, June 28 - July 1, 2009*. The MIT Press, 2009. doi: 10.15607/RSS.2009.V.021. URL <http://www.roboticsproceedings.org/rss05/p21.html>.
- [197] Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2006. ISBN 0-471-47378-2.
- [198] Bahareh Shakibajahromi, Saeed Shayestehmanesh, Daniel Schwartz, and Ali Shokoufandeh. HyNet: 3D Segmentation Using Hybrid Graph Networks. In *International Conference on 3D Vision, 3DV 2021, London, United Kingdom, December 1-3, 2021*, pages 805–814. IEEE, 2021. doi: 10.1109/3DV53792.2021.00089. URL <https://doi.org/10.1109/3DV53792.2021.00089>.
- [199] Claude E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, 1949. doi: 10.1109/JRPROC.1949.232969.
- [200] Marcel Sheeny, Andrew Wallace, and Sen Wang. 300 GHz Radar Object Recognition based on Deep Neural Networks and Transfer Learning. *IET Radar, Sonar & Navigation*, 14, 05 2020. doi: 10.1049/iet-rsn.2019.0601.
- [201] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew M. Wallace. RADIATE: A Radar Dataset for Automotive Perception in Bad Weather. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pages 1–7. IEEE, 2021. doi: 10.1109/ICRA48506.2021.9562089. URL <https://doi.org/10.1109/ICRA48506.2021.9562089>.
- [202] Merrill I. Skolnik. *Introduction to Radar Systems*. McGraw-Hill, 2nd edition edition, 1981. ISBN 0-07-057909-1.
- [203] Heather Somerville, Paul Lienert, and Alexandria Sage. Uber’s use of fewer safety sensors prompts questions after arizona crash, 2016. URL <https://www.reuters.com/article/us-uber-selfdriving-sensors-insight-idUSKBN1H337Q>.
- [204] Charles Spearman. The Proof and Measurement of Association between two Things. *The American Journal of Psychology*, 100(3/4):441–471, 1987.
- [205] Cyrill Stachniss. *Robotic Mapping and Exploration*, volume 55. Springer, Berlin, 01 2009. ISBN 978-3-642-01096-5. doi: 10.1007/978-3-642-01097-2.
- [206] Sascha Steyer, Christian Lenk, Dominik Kellner, Georg Tanzmeister, and Dirk Wollherr. Grid-Based Object Tracking With Nonlinear Dynamic State and Shape Estimation. *IEEE Trans. Intell. Transp. Syst.*, 21(7):2874–2893, 2020. doi: 10.1109/TITS.2019.2921248. URL <https://doi.org/10.1109/TITS.2019.2921248>.

- [207] Suleyman Suleymanov. Design and Implementation of an FMCW Radar Signal Processing Module for Automotive Applications, August 2016. URL <http://essay.utwente.nl/70986/>.
- [208] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII*, volume 12373 of *Lecture Notes in Computer Science*, pages 685–702. Springer, 2020. doi: 10.1007/978-3-030-58604-1_41. URL https://doi.org/10.1007/978-3-030-58604-1_41.
- [209] Wenming Tang and Guoping Qiu. Dense Graph Convolutional Neural Networks on 3D Meshes for 3D Object Segmentation and Classification. *Image Vis. Comput.*, 114: 104265, 2021. doi: 10.1016/j.imavis.2021.104265. URL <https://doi.org/10.1016/j.imavis.2021.104265>.
- [210] INC. TESLA. MODEL Y OWNER’S MANUAL: Software Version 2022.12, 2022. URL https://www.tesla.com/ownersmanual/modely/is_is/GUID-6B9A1AEA-579C-400E-A7A6-E4916BCD5DED.html.
- [211] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6410–6419. IEEE, 2019. doi: 10.1109/ICCV.2019.00651. URL <https://doi.org/10.1109/ICCV.2019.00651>.
- [212] Sebastian Thrun and Michael Montemerlo. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006. doi: 10.1177/0278364906065387. URL <https://doi.org/10.1177/0278364906065387>.
- [213] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. volume 5, pages 253–271, 1998. doi: 10.1023/A:1008806205438. URL <https://doi.org/10.1023/A:1008806205438>.
- [214] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren. In *Fahrerassistenzworkshop Walting*, volume 10, 09 2015.
- [215] Shinji Umeyama. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380, 1991. doi: 10.1109/34.88573. URL <https://doi.org/10.1109/34.88573>.

- [216] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher R. Baker, Robert Bittner, et al. Autonomous Driving in Urban Environments: Boss and the Urban Challenge. In *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, George Air Force Base, Victorville, California, USA, volume 56 of *Springer Tracts in Advanced Robotics*, pages 1–59. Springer, 2009. doi: 10.1007/978-3-642-03991-1_1. URL https://doi.org/10.1007/978-3-642-03991-1_1.
- [217] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. SfM-Net: Learning of Structure and Motion from Video. *CoRR*, abs/1704.07804, 2017. URL <http://arxiv.org/abs/1704.07804>.
- [218] Han Wang, Chen Wang, and Lihua Xie. Lightweight 3-D Localization and Mapping for Solid-State LiDAR. *IEEE Robotics Autom. Lett.*, 6(2):1801–1807, 2021. doi: 10.1109/LRA.2021.3060392. URL <https://doi.org/10.1109/LRA.2021.3060392>.
- [219] 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 J. Sel. Top. Signal Process.*, 15(4):954–967, 2021. doi: 10.1109/JSTSP.2021.3058895. URL <https://doi.org/10.1109/JSTSP.2021.3058895>.
- [220] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.*, 38(5):146:1–146:12, 2019. doi: 10.1145/3326362. URL <https://doi.org/10.1145/3326362>.
- [221] Tuopu Wen, Zhongyang Xiao, Benny Wijaya, Kun Jiang, Mengmeng Yang, and Diange Yang. High Precision Vehicle Localization based on Tightly-coupled Visual Odometry and Vector HD Map. In *IEEE Intelligent Vehicles Symposium, IV 2020, Las Vegas, NV, USA, October 19 - November 13, 2020*, pages 672–679. IEEE, 2020. doi: 10.1109/IV47402.2020.9304659. URL <https://doi.org/10.1109/IV47402.2020.9304659>.
- [222] Klaudius Werber, Matthias Rapp, Jens Klappstein, Markus Hahn, Jürgen Dickmann, Klaus Dietmayer, and Christian Waldschmidt. Automotive Radar Gridmap Representations. In *2015 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility ICMIM*, pages 1–4, 2015. doi: 10.1109/ICMIM.2015.7117922.
- [223] Moritz Werling. *Optimale aktive Fahreingriffe: für Sicherheits- und Komfortsysteme in Fahrzeugen*. De Gruyter Oldenbourg, 2017. ISBN 9783110531923. doi: doi:10.1515/9783110531923. URL <https://doi.org/10.1515/9783110531923>.
- [224] Volker Winkler. Range Doppler detection for automotive FMCW radars. In *2007 European Microwave Conference*, pages 1445–1448, 2007. doi: 10.1109/EUMC.2007.4405477.

- [225] Hermann Winner, editor. *Handbuch Fahrerassistenzsysteme : Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort ; mit 45 Tabellen*. Vieweg + Teubner, Wiesbaden, 2., korrigierte aufl. edition, 2012. ISBN 3834814571; 9783834814579.
- [226] Christian Wöhler, Ole Schumann, Markus Hahn, and Jürgen Dickmann. Comparison of Random Forest and Long Short-Term Memory Network Performances in Classification Tasks using Radar. In *Sensor Data Fusion: Trends, Solutions, Applications, SDF 2017, Bonn, Germany, October 10-12, 2017*, pages 1–6. IEEE, 2017. doi: 10.1109/SDF.2017.8126350. URL <https://doi.org/10.1109/SDF.2017.8126350>.
- [227] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 1887–1893. IEEE, 2018. doi: 10.1109/ICRA.2018.8462926. URL <https://doi.org/10.1109/ICRA.2018.8462926>.
- [228] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 4376–4382. IEEE, 2019. doi: 10.1109/ICRA.2019.8793495. URL <https://doi.org/10.1109/ICRA.2019.8793495>.
- [229] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVIII, volume 12373 of Lecture Notes in Computer Science*, pages 1–19. Springer, 2020. doi: 10.1007/978-3-030-58604-1_1. URL https://doi.org/10.1007/978-3-030-58604-1_1.
- [230] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse Single Sweep LiDAR Point Cloud Segmentation via Learning Contextual Shape Priors from Scene Completion. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 3101–3109. AAAI Press, 2021. URL <https://ojs.aaai.org/index.php/AAAI/article/view/16419>.
- [231] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse Single Sweep LiDAR Point Cloud Segmentation via Learning Contextual Shape Priors from Scene Completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3101–3109, 2021.

- [232] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. Robotics*, 37(2):314–333, 2021. doi: 10.1109/TRO.2020.3033695. URL <https://doi.org/10.1109/TRO.2020.3033695>.
- [233] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(11):2241–2254, 2016. doi: 10.1109/TPAMI.2015.2513405. URL <https://doi.org/10.1109/TPAMI.2015.2513405>.
- [234] Hichame Yessou, Gencer Sumbul, and Begüm Demir. A Comparative Study of Deep Learning Loss Functions for Multi-Label Remote Sensing Image Classification. In *IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2020, Waikoloa, HI, USA, September 26 - October 2, 2020*, pages 1349–1352. IEEE, 2020. doi: 10.1109/IGARSS39084.2020.9323583. URL <https://doi.org/10.1109/IGARSS39084.2020.9323583>.
- [235] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust Point Matching Using Learned Features. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11821–11830. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01184. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Yew_RPM-Net_Robust_Point_Matching_Using_Learned_Features_CVPR_2020_paper.html.
- [236] Ma Yi-De, Liu Qing, and Qian Zhi-Bai. Automated Image Segmentation using improved PCNN Model based on Cross-Entropy. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, pages 743–746, 2004. doi: 10.1109/ISIMP.2004.1434171.
- [237] Wei Yin, Yifan Liu, and Chunhua Shen. Virtual Normal: Enforcing Geometric Constraints for Accurate and Robust Depth Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.
- [238] Senthil Kumar Yogamani, Christian Witt, Hazem Rashed, Sanjaya Nayak, Saquib Mansoor, Pdraig Varley, Xavier Perrotton, Derek O’Dea, Patrick Pérez, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Sumanth Chennupati, Michal Uricár, Stefan Milz, Martin Simon, and Karl Amende. WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9307–9317. IEEE, 2019. doi: 10.1109/ICCV.2019.00940. URL <https://doi.org/10.1109/ICCV.2019.00940>.
- [239] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Fei Qiao. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In *2018 IEEE/RSJ International*

- Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*, pages 1168–1174. IEEE, 2018. doi: 10.1109/IROS.2018.8593691. URL <https://doi.org/10.1109/IROS.2018.8593691>.
- [240] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access*, 8: 58443–58469, 2020. doi: 10.1109/ACCESS.2020.2983149. URL <https://doi.org/10.1109/ACCESS.2020.2983149>.
- [241] Anestis Zaganidis, Martin Magnusson, Tom Duckett, and Grzegorz Cielniak. Semantic-assisted 3D Normal Distributions Transform for Scan Registration in Environments with limited Structure. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 4064–4069. IEEE, 2017. doi: 10.1109/IROS.2017.8206262. URL <https://doi.org/10.1109/IROS.2017.8206262>.
- [242] Anestis Zaganidis, Alexandros Zerntev, Tom Duckett, and Grzegorz Cielniak. Semantically Assisted Loop Closure in SLAM Using NDT Histograms. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*, pages 4562–4568. IEEE, 2019. doi: 10.1109/IROS40897.2019.8968140. URL <https://doi.org/10.1109/IROS40897.2019.8968140>.
- [243] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833. Springer, 2014. doi: 10.1007/978-3-319-10590-1_53. URL https://doi.org/10.1007/978-3-319-10590-1_53.
- [244] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 199–208, 2017. doi: 10.1109/CVPR.2017.29.
- [245] Wei Zeng and Theo Gevers. 3DContextNet: K-d Tree Guided Hierarchical Learning of Point Clouds Using Local and Global Contextual Cues. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11131 of *Lecture Notes in Computer Science*, pages 314–330. Springer, 2018. doi: 10.1007/978-3-030-11015-4_24. URL https://doi.org/10.1007/978-3-030-11015-4_24.
- [246] Ji Zhang and Sanjiv Singh. LOAM: Lidar Odometry and Mapping in real-time. *Robotics: Science and Systems Conference RSS*, pages 109–111, 01 2014.

- [247] Jinming Zhang, Xiangyun Hu, and Hengming Dai. A Graph-Voxel Joint Convolution Neural Network for ALS Point Cloud Segmentation. *IEEE Access*, 8:139781–139791, 2020. doi: 10.1109/ACCESS.2020.3013293. URL <https://doi.org/10.1109/ACCESS.2020.3013293>.
- [248] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. PolarNet: An Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9598–9607, 2020. doi: 10.1109/CVPR42600.2020.00962.
- [249] Bufan Zhao, Xianghong Hua, Kegen Yu, Xiaoxing He, Weixing Xue, Qiqi Li, Hanwen Qi, Lujie Zou, and Cheng Li. An Inverse Node Graph-Based Method for the Urban Scene Segmentation of 3D Point Clouds. *Remote. Sens.*, 13(15):3021, 2021. doi: 10.3390/rs13153021. URL <https://doi.org/10.3390/rs13153021>.
- [250] Hui Zhou, Xinge Zhu, Xiao Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylinder3D: An Effective 3D Framework for Driving-scene LiDAR Semantic Segmentation. *CoRR*, abs/2008.01550, 2020. URL <https://arxiv.org/abs/2008.01550>.
- [251] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A Modern Library for 3D Data Processing. *ArXiv*, abs/1801.09847, 2018.
- [252] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4490–4499. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00472. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhou_VoxelNet_End-to-End_Learning_CVPR_2018_paper.html.
- [253] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, et al. Making Bertha Drive - An Autonomous Journey on a Historic Route. *IEEE Intell. Transp. Syst. Mag.*, 6(2):8–20, 2014. doi: 10.1109/MITS.2014.2306552. URL <https://doi.org/10.1109/MITS.2014.2306552>.