UNIVERSITÄT BONN

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation

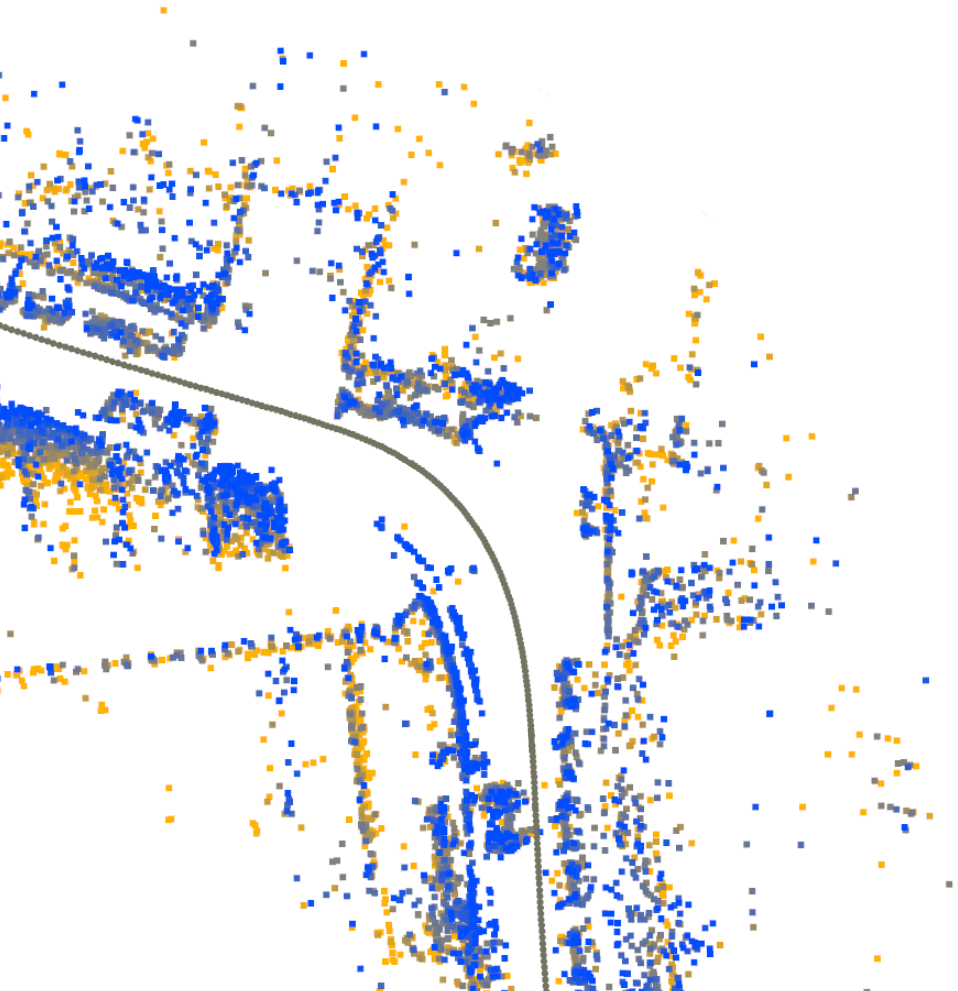
Localization and Mapping for Autonomous Vehicles Using Radar

von

Daniel Casado Herráez

aus

Ávila, Spain



Bonn 2026

Referent:
Prof. Dr. Cyrill Stachniss, University of Bonn, Germany

Korreferent:
Prof. Dr. Javier Civera, University of Zaragoza, Spain

Tag der mündlichen Prüfung: 29. April 2026

Angefertigt mit Genehmigung der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät der Universität Bonn

Zusammenfassung

Menschliches Versagen ist nach wie vor die häufigste Ursache für tödliche Verkehrsunfälle [46]. Durch die Minimierung menschlicher Eingriffe im Straßenverkehr sollen autonome Fahrzeuge Verkehrsunfälle reduzieren und ein sichereres Transportmittel bieten. Um sich zuverlässig in einer Umgebung zu bewegen, müssen autonome Fahrzeuge ihre genaue Position kennen. Ohne genaue Lokalisierung sind die Trajektorienplanung und Hindernisvermeidung nicht möglich, was schwerwiegende Folgen haben kann.

Obwohl die Lokalisierung in der Regel mithilfe globaler Navigationssatellitensysteme erfolgt, ist die Verfügbarkeit von verlässlicher Satellitenkommunikation in Gebieten mit Wolkenkratzern sowie in Tunneln und Parkhäusern eingeschränkt. Daher müssen autonome Fahrzeuge in der Lage sein, ihre eigene Position und Ausrichtung ausschließlich anhand von Bordsensoren und zuvor aufgezeichneten Karten der Umgebung zu ermitteln. Die Lokalisierung und Kartierung erfolgt in der Regel mit Kameras oder LiDAR-Sensoren. Kameras werden jedoch durch schlechte Lichtverhältnisse und widrige Wetterbedingungen beeinträchtigt, was die Lokalisierung erheblich verschlechtert. Die Leistungsfähigkeit von LiDAR-Sensoren hängt ebenfalls von Wetterbedingungen ab und wird negativ von starkem Regen, Nebel und Schnee beeinflusst. Radarsensoren, die bereits heute in Serienfahrzeugen verfügbar sind, sind unempfindlich gegenüber veränderten Lichtverhältnissen und Wetterbedingungen. Darüber hinaus liefern Radarsensoren auch Geschwindigkeitsinformationen und den Radarquerschnitt der erfassten Objekte, die zur Verbesserung der Lokalisierung und Kartierung genutzt werden können. Der Nachteil ist jedoch, dass Radarsensoren eine Punktwolke ausgeben, die im Vergleich zu LiDAR-Daten spärlich und verrauscht ist, was zu Herausforderungen bei der sensorbasierten Posen-schätzung führt. Obwohl es bereits einige Arbeiten zur Lokalisierung und Kartierung auf Basis von Punktwolken gibt, stoßen diese Methoden bei starkem Rauschen und der geringen Anzahl von Punkten in Radarpunktwolken an ihre Grenzen. Darüber hinaus nutzen sie nicht die spezifischen Informationen der Radarsensoren, wie die gemessenen Dopplergeschwindigkeit und Radarquerschnitte von Objekten, aus.

In dieser Arbeit erweitern wir die Möglichkeiten der Radarlokalisierung und Kartierung für autonome Fahrzeuge, indem wir neuartige und wirkungsvolle Techniken einführen, die speziell für Radarsensoren entwickelt wurden. Wir beginnen mit der Schätzung der Position des Fahrzeugs und erstellen im Laufe der Zeit eine Karte der Umgebung. Um dies zu erreichen, entwickeln wir neuartige Algorithmen, welche die spärliche Beschaffenheit von Radarpunktwolken und die damit verbundenen Dopplergeschwindigkeiten nutzen und zu genaueren Positionsschätzungen führen. Aufgrund der geringen vertikalen Auflösung von Radarsensoren stellen wir fest, dass sich die Lokalisierungsgenauigkeit bei Höhenänderungen verschlechtert. Um diese Einschränkung zu bewältigen, schlagen wir einen Ansatz vor, der die Radareigenschaften nutzt, um Merkmale der Straße zu extrahieren und so die Genauigkeit bei Neigungsänderungen zu verbessern. Wir verbessern die Genauigkeit der Position weiter, indem wir eine zusätzliche inertiale Messeinheit in das System integrieren. Diese Strategien erzielen zwar kurzfristig eine hohe Genauigkeit, jedoch können sich Fehler über lange Strecken akkumulieren, was zu Inkonsistenzen in großflächigen Karten führt. Daher entwickeln wir ein radarspezifisches Modul, das eine Ortserkennung durchführt, um bereits zuvor besuchte Orte zu identifizieren. Wir integrieren diese Informationen in ein neuartiges radar-inertiales System zur simultanen Lokalisierung und Kartierung, wodurch eine genaue Posen-Schätzung erreicht wird und konsistente Karten erstellt werden. Um verschiedene Bereiche des befahrbaren Gebiets zu unterschiedlichen Zeitpunkten aufzuzeichnen und zu kombinieren, schlagen wir ein Multi-Session-Kartierungssystem vor, das mehrere Karten zusammenführt. Unser Ansatz berücksichtigt dabei auch zeitliche Veränderungen in der Umgebung, wie geparkte Autos und Neubauten. Dies ist entscheidend für die Aufrechterhaltung einer langfristigen, genauen Darstellung der Umgebung. Darüber hinaus ist unsere Methode in der Lage, sich innerhalb zuvor aufgezeichneter Radarkarten mit minimalem kumulativem Fehler genau zu lokalisieren.

Abschließend lässt sich festhalten, dass unsere Ansätze zu wesentlichen Fortschritten und der Verbesserung des Standes der Technik für die Lokalisierung und Kartierung mittels Radarsensoren beitragen. Darüber hinaus stellen wir eine der ersten Methoden zur Ortserkennung vor, die speziell auf Radarsensoren für autonome Fahrzeuge zugeschnitten ist und auf verschiedene Datensätze übertragen werden kann. Außerdem stellen wir eine der ersten Techniken zur langfristigen Kartenerstellung, Kartenwartung und Lokalisierung vor. Alle in dieser Arbeit vorgestellten Ansätze wurden anhand öffentlicher Datensätze evaluiert und in begutachteten Konferenzbeiträgen und Zeitschriftenartikeln veröffentlicht. Einige unserer Methoden wurden auch als Open-Source-Software veröffentlicht, wodurch insgesamt die Möglichkeit zur sicheren Lokalisierung und Kartierung für autonome Fahrzeuge unter Verwendung von Radar verbessert wird.

Abstract

Human error remains the leading cause of road fatalities [46]. By minimizing human intervention behind the wheel, autonomous vehicles aim to reduce traffic accidents and provide a safer means of transportation. These vehicles must know their precise location to navigate reliably within an environment. Without accurate localization, trajectory planning and obstacle avoidance systems will have an incorrect estimate of the vehicle's state, potentially leading to a collision.

Although localization is commonly achieved using global navigation satellite systems, satellite visibility is limited in areas with tall skyscrapers and indoor settings such as tunnels and parking garages. Therefore, autonomous cars also need the ability to estimate their own position and orientation solely leveraging local sensing and previously recorded maps of the environment, without relying on satellite availability. Onboard localization and mapping are typically achieved using cameras or LiDAR sensors. Cameras, however, are affected by low light and adverse weather conditions, and the performance of LiDARs degrades in challenging weather scenarios such as heavy rain, fog, and snow. On the contrary, radar sensors are resilient to environmental conditions and are already integrated into consumer vehicles today. Additionally, automotive radars also provide velocity information and the radar cross section of the measured targets, which can be leveraged to enhance localization performance. As a downside, these sensors produce a sparse and noisy point cloud compared to LiDAR data, resulting in challenges for sensor-based pose estimation. Although some works on radar localization and mapping already exist, these methods face limitations at high noise levels and with the limited number of points present in radar scans. Moreover, they do not exploit the specific properties of radar sensing.

In this thesis, we push the boundaries of radar localization and mapping for autonomous vehicles by introducing novel and impactful techniques specifically designed for automotive radar sensors. We begin by estimating the pose of the vehicle and creating a map of the environment over time. To achieve this, we propose novel algorithms that exploit the sparsity of automotive radar scans and their associated Doppler velocities, yielding accurate pose estimates. Due to the

low vertical resolution of automotive radars, we observe that localization performance degrades when there are changes in elevation. To address this limitation, we propose an approach that exploits the radar properties to extract road features, enhancing accuracy during slope variations. We further improve pose estimation performance by introducing an additional inertial measurement unit into the system. While these strategies achieve high short-term accuracy, errors can accumulate over long trajectories, leading to inconsistencies in large-scale maps. Therefore, we develop a radar-specific module that performs place recognition to identify previously visited locations. We integrate this information within a novel radar-inertial simultaneous localization and mapping system, achieving accurate online pose estimation and producing consistent maps. To record and combine different regions of the drivable area at various points in time, we propose a multi-session mapping system that merges multiple maps. Our proposed approach also accounts for temporal changes in the environment, such as parked cars and new constructions. This is crucial for maintaining a long-term, accurate representation of the environment. Moreover, our technique accurately localizes within previously recorded radar maps with minimal accumulated error.

Our proposed estimation approaches achieve state-of-the-art results on automotive radar data. Additionally, we present one of the first place recognition methods explicitly tailored for automotive radar sensors, capable of generalizing to different datasets. We also present one of the first long-term map construction and localization techniques for automotive radars. All of our approaches have been evaluated on public datasets and have been published in peer-reviewed conferences and journals. Some of our methods have also been open-sourced, collectively enhancing the capabilities of safe localization and mapping for autonomous vehicles using radar.

Acknowledgements

The past three years have been an exciting journey researching the latest techniques for radar localization and mapping for autonomous vehicles. However, no big feat can be accomplished alone. This journey would not have been possible without many people who have supported me along the way. First and foremost, I want to thank Cyrill Stachniss and Michael Heidingsfeld for believing in me from the very beginning of my PhD and giving me the confidence to achieve my goals.

I am sincerely grateful to Cyrill Stachniss for his precious advice and support. Despite being an external industry PhD, I always felt he was there whenever I needed help, assisting me with correcting papers and discussing research directions and topics. I am also thankful for his constant drive for excellence, which has inspired me to strive for perfection. I feel truly fortunate, since it is thanks to his YouTube videos that I became interested in SLAM, and completing a PhD under his supervision is a great honor for me.

I also want to express my immense gratitude to Michael Heidingsfeld as my industry supervisor and friend. I felt his support from our very first call, and he has helped me grow professionally and personally. With his guidance, he paved my path through the corporate world and helped me navigate my research topic effectively. Moreover, moving from bigger cities to Mönsheim has been a challenge, and I feel deeply grateful for his help during my ups and downs.

I also need to thank my colleague, friend, and Radar Bro, Matthias Zeller, for his inspiration and for making me feel at home. His motivation to pursue higher goals has served as a role model, encouraging me to learn and improve my results over the last three years. I am also profoundly thankful for his support during the challenging times of completing a PhD in industry, always helping me to stay positive and pursue my goals. And for all the good times we spent together. I would not have made it without him. Thank you.

During my time at CARIAD, I also had the privilege of meeting the most talented and motivated individuals. First, my students, who supported me in pushing my ideas forward and taught me how to improve as a supervisor. Particularly, Le Chang, Franz Kaschner, Abhay Balasaheb Jadhav, Vijay Mistry,

Umair Khawaja, Muhammad Kaleem Ullah, and Francesco Maglie, for their commitment and best efforts during their time with me. Second, all the talent that has surrounded me during the PhD program has been invaluable to me. This includes Mareike Grund, Lorenz Ott, Fabian Konstantinidis, Christian Witte, Yin Wu, Gasser Elazab, Christoph Huemmer, Johana Eiting, Florian Jaumann, Sharang Kaul, Shubham Poolari, Muhammet Uslu, Tihomir Rohlinger, Nina Burdorf, Jonathan Stief, Liangyu Zhong, Jiarong Wei, and my friends and travel buddies Xavier Timoneda i Comas, Francesco Maldonato, Antoine Foricher, and Arslan Tariq. Also, I would like to extend my thanks to our managers at CARIAD who have supported the PhD program.

I also owe my deepest gratitude to my long-distance labmates at Stachnisslab, who have made my journey easier and more enjoyable. In particular, I would like to thank Jens Behley for his invaluable mentorship regarding new topics and for his detailed feedback on my research. To Nacho Vizzo, for his advice and optimism during the start of my PhD. To Louis Wiesmann for his cool visualizations. To Yue Pan, for being a source of inspiration. To Tiziano Guadagnino, Benedikt Mersch, and Meher Malladi, for their expertise and support on SLAM. And to Rodrigo Marcuzzi, Lucas Nunes, Saurabh Gupta, Thomas Läbe, Yue (Linn) Chong, Elias Marks, Federico Magistri, Olga Vysotska, and Matteo Sodano, with whom it has been a privilege to work. I would also like to express my appreciation to Igor Bogoslavskyi for his invaluable career advice. And to Dong Wang, from the University of Würzburg, for all our insightful discussions and collaborations.

Outside the academic and industry contexts, however, others have also supported me and believed in me since the beginning of my PhD. Thank you to Maria, Belen, Philipp, Íñigo, Angel, Jorge, Gloria, and Isa for making me feel that I am not alone in Germany. I would like to express my long-lasting appreciation to Edu for his constant support, motivation, and joy, and to all my other wonderful friends who make my life more fulfilling: Galis, Luis, Richi, Raquel, Jota, Tuli, Gaspy, Maite, Mazin, Patorisia San, and Davito San.

Also, thank you to Meli, my constant source of support during the last year of my PhD. This was the most stressful part of the journey, and your words always encouraged me to be the best version of myself.

Finally, I am eternally grateful to my family, who have been the foundation of all of my achievements. To my parents, Teresa and Seve, and to my sister Cristina, for their unconditional love, support, and guidance, always encouraging me to aim high and give my best. Nothing that I will ever achieve will have been possible without you. Finally, my heartfelt thanks to my grandparents, la Yaya, for praying for me no matter where I am in the world, and el Pane, for planting in me the seed of curiosity that has grown to bring me where I am today.

Contents

Acronyms

APE	absolute pose error
ATE	absolute translation error
ARE	absolute rotation error
FFT	fast Fourier transform
FMCW	frequency modulated continuous wave
GNSS	global navigation satellite system
IMU	inertial measurement unit
LiDAR	light detection and ranging
MLP	multi layer perceptron
RADAR	radio detection and ranging
RANSAC	random sample consensus
RCS	radar cross section
ROS	robot operating system
RPE	relative pose error
RTE	relative translation error
RRE	relative rotation error
SLAM	simultaneous localization and mapping

Mathematical Notation

To improve readability, we collect here the most common notations that will be used in this thesis. Further notation specific to a particular chapter will be defined within that chapter.

General Notation

a	A scalar, used for indexing and parameters
A	A scalar, used for denoting dimensions
$\mathbf{a}$	A column vector
$\mathbf{A}$	A matrix, used for denoting generic matrices
T	A matrix, used for denoting homogeneous transformations in $SE(3)$
R	A matrix, used for denoting rotations in $SO(3)$
$\|\cdot\|$	The $L2$ norm of a vector
$\mathcal{S}$	A set
$\|\mathcal{S}\|$	The cardinality of a set

Geometry Notation

$\{F\}$	A three-dimensional coordinate frame
${}^{F}\mathbf{a}$	A vector $\mathbf{a}$ expressed in $\{F\}$
${}^{F}\boldsymbol{v}_{A}$	The linear velocity of the origin of frame $\{A\}$ expressed in $\{F\}$
${}^{F}\boldsymbol{\omega}_{A}$	The angular velocity of the origin of frame $\{A\}$ expressed in $\{F\}$
${}^{F}\boldsymbol{a}_{A}$	The linear acceleration of the origin of frame $\{A\}$ expressed in $\{F\}$
$SO(3)$	The special orthogonal Lie group used to represent 3D rotations
$\mathfrak{so}(3)$	The Lie algebra associated with $SO(3)$
$SE(3)$	The special Euclidean Lie group used to represent 3D transformations
$\mathfrak{se}(3)$	The Lie algebra associated with $SE(3)$
$(\cdot)^{\wedge}$	The skew-symmetric operator, used to map a vector within $\mathbb{R}^3$ to its matrix form in $\mathfrak{so}(3)$ or a vector within $\mathbb{R}^6$ to its matrix form in $\mathfrak{se}(3)$

R_B^A The $SO(3)$ rotation from the origin of frame $\{A\}$ to the origin of frame $\{B\}$

T_B^A The $SE(3)$ homogeneous transformation from the origin of frame $\{A\}$ to the origin of frame $\{B\}$

${}^F\mathbf{t}_B^A$ The translation within $\mathbb{R}^3$ from the origin of frame $\{A\}$ to the origin of frame $\{B\}$ expressed in $\{F\}$

Chapter 1

Introduction

In 2024 alone, around 19,800 people were killed in traffic accidents in the European Union [46]. This tragic figure highlights the human toll paid on European roads each year. While there is a wide variety of causes of these traffic fatalities, the vast majority are attributable to human factors, including distractions during driving, speeding, and fatigue [186].

Autonomous vehicles have the transformative potential to mitigate human error by minimizing human intervention while driving, which can lead to an increase in safety and a reduction in road fatalities. To achieve that, autonomous cars must operate safely and reliably in all possible conditions. This requires the ability to perceive the environment and accurately estimate their own position and orientation. Similar to how humans localize in a scene, an autonomous vehicle must know its precise location relative to its surroundings. Without it, the vehicle's predicted motion may be incorrect, potentially causing it to leave the road or to collide with pedestrians and other vehicles.

Although localization can be achieved using global navigation satellite systems (GNSS), this requires satellite visibility, which can degrade in environments with tall skyscrapers, dense foliage, tunnels, and indoor parking lots. As a result, an autonomous vehicle should not rely solely on global positioning techniques during operation, calling for pose estimation methods that leverage local sensor measurements. The problem of estimating the pose of a vehicle in an unknown environment without external signals is referred to as simultaneous localization and mapping, or in short, SLAM. It provides an estimate of the vehicle's trajectory and a map of its surroundings. This map can be exploited in the future to improve the accuracy of a localization system, and thus, it should contain reliable information that can be used for pose estimation.

In addition to a reliable pose estimation mechanism for safe autonomous driving, it is crucial to have a robust sensor setup, which must be carefully selected to cover all weather conditions, ranging from sunny days to heavy snowfall.

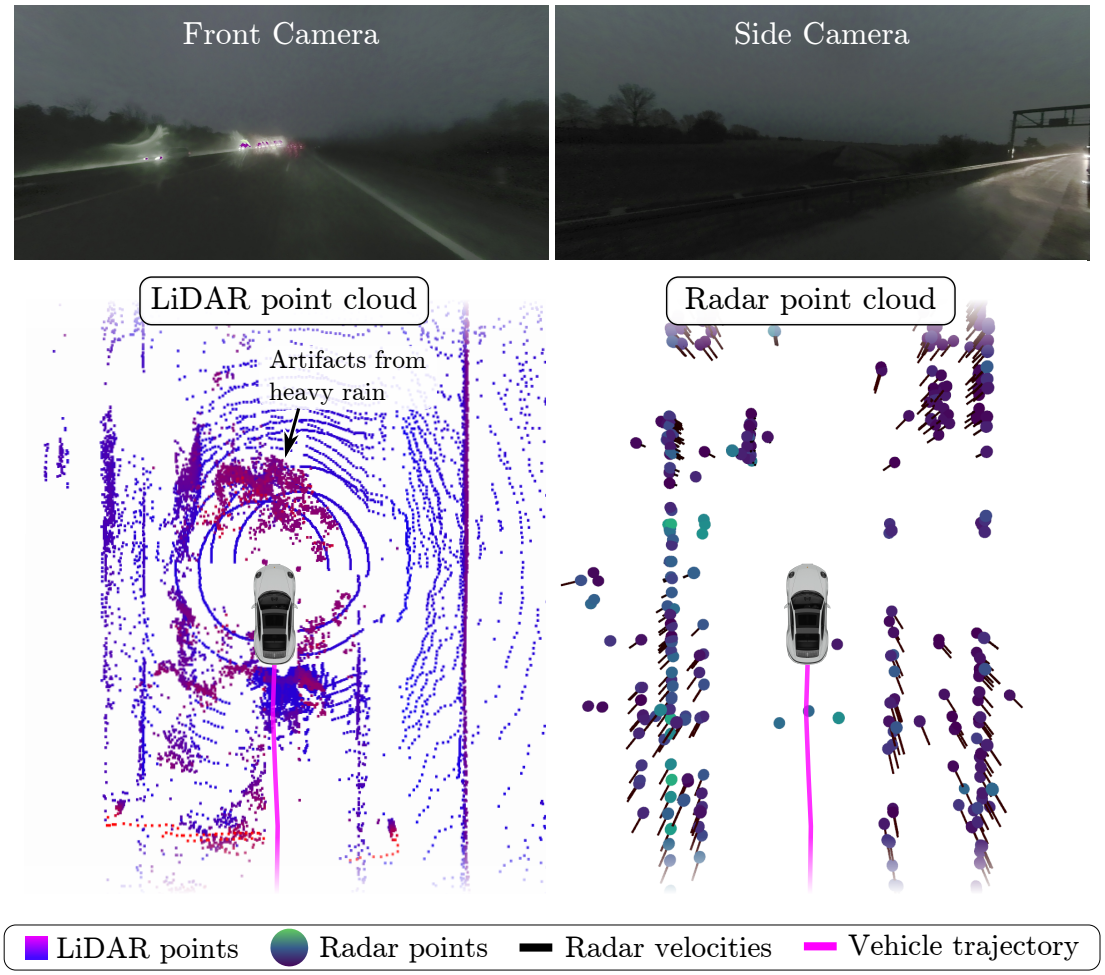


Figure 1.1: Visualization of sensor data and the car trajectory in a typical highway setting under adverse weather. Camera images are dark and blurry, with other cars barely visible. LiDAR artifacts are also present due to the laser reflections with water drops. Radar, however, remains unaffected by the adverse weather but produces a sparse and noisy point cloud.

Diverse weather scenarios, however, have different effects on each sensor. While some autonomous driving solutions [80] have focused on camera-centric techniques, this provides low levels of redundancy within the system and faces limitations in low-light and bad-weather conditions. In addition, individual cameras do not provide depth information, and stereo cameras are bulky to mount around the vehicle. To circumvent this problem, others [138] have also integrated light detection and ranging sensors (LiDARs) into their vehicles, which, despite their high measurement accuracy, are expensive to manufacture. Furthermore, LiDARs are not robust to bad weather, resulting in camera-LiDAR systems not being resilient to such scenarios. This is illustrated in the scenario from the truckScenes dataset [50] in Figure 1.1, where reduced visibility and heavy rain noticeably degrade camera and LiDAR sensor measurements. Automotive radars, on the contrary, offer high resilience to bad weather and low-lighting scenarios,

with low prices and compactness of the sensor. Radars can serve as a complementary sensor to cameras to cover the full spectrum of driving scenarios. Nevertheless, to fully leverage their potential, it is essential to address the inherent noise and sparsity of their output point cloud, which underscores the necessity for radar-specific methods in pose estimation and mapping for autonomous vehicles.

In this thesis, we focus on the problem of localization and mapping leveraging automotive radar sensors, which are already present in consumer vehicles today. Our contributions are threefold: (i) we push the capabilities of automotive radars to perform pose estimation of an autonomous vehicle; (ii) we construct radar maps of the environment that can be leveraged for navigation when a place is being revisited; and (iii) we merge and maintain maps of coinciding regions over longer periods of time, handling temporal changes from moving objects and varying environments. All the methods presented in this thesis have been published in peer-reviewed international conferences and journals, and some have been made available as open-source software.

The thesis is organized as follows. In Chapter 2, we introduce the basic techniques supporting the understanding of this work, including the working principles of automotive radars, a brief summary of relative body motion, an outline of the least-squares problem, and how it is useful for solving optimization problems in SLAM. We also present the main characteristics of SLAM systems and how they can be evaluated. In Chapter 3, we review existing work relevant to the problem of pose estimation and mapping addressed in this thesis. In Chapter 4 and Chapter 5, we focus on the challenge of relative pose estimation, where the ego-motion of the vehicle is estimated within consecutive radar measurements. In Chapter 6, we address the problem of loop closure detection and place recognition, where a pre-recorded map database is employed to identify revisited locations. In Chapter 7, we combine the learnings from our previous chapters and integrate an inertial measurement unit into a system that performs large-scale pose estimation and mapping. Then, Chapter 8 introduces a long-term mapping and localization pipeline that combines multiple maps created at different times, identifies temporal changes between them, and exploits this information to perform accurate pose estimation. Finally, Chapter 9 summarizes the contents of this thesis and presents possible directions of future research within the field of automotive radar localization and mapping.

1.1 Main Contributions

This thesis provides contributions across key areas of radar localization and mapping, including pose estimation, map construction, and map processing. These contributions have been validated in real driving scenarios and demonstrate prac-

tical relevance within the automotive radar domain. The following section summarizes the main achievements of this thesis and highlights how they extend the state of the art in autonomous driving and robotics.

The first contribution, introduced in Chapter 4, is related to online pose estimation and mapping solely relying on radars [26]. We develop a state-of-the-art approach for online radar pose estimation and map construction that leverages the Doppler velocities provided by radar sensors. We also handle the sparsity and noise from the radar output to construct accurate radar maps. We demonstrate on real data that our approach successfully leverages the additional information provided by automotive radars to improve pose estimation performance and achieves LiDAR-level pose estimation accuracy.

The second contribution of this thesis is done in close collaboration with Franz Kaschner, and is presented in Chapter 5. It addresses the vertical drift produced by the limited elevation resolution of automotive radars [25]. Our approach leverages the ground plane information to improve the ego-motion estimation in the vertical axis, achieving state-of-the-art performance comparable to that of LiDAR-based approaches.

Our third contribution, introduced in Chapter 6, is in the context of radar place recognition [24]. Our method identifies coinciding locations from sparse and noisy radar scans, achieving state-of-the-art results on real-world driving data. While existing techniques rely on scan aggregation and image-based projection to mitigate radar sparsity, our approach is designed specifically to target automotive radar data. In addition, our method achieves a remarkably low-dimensional scan descriptor with minimal storage requirements. Our place recognition approach further enables radar loop closure detection and multi-session mapping with automotive radar sensors.

Our fourth contribution, presented in Chapter 7, combines our learnings of radar-only odometry and place recognition into a full SLAM system to correct accumulated drift [28]. Moreover, while our previous contributions focused on fully exploiting the capabilities of automotive radars for pose estimation, in this chapter, we add an additional inertial measurement unit to further enhance the accuracy of our pose estimates. Additionally, we focus on having a modular architecture with independent odometry, loop detection, and global pose optimization that achieves accurate large-scale trajectory estimates. Our approach achieves state-of-the-art results on radar-inertial SLAM, and is open-source, facilitating further research in the field.

Our fifth contribution, introduced in Chapter 8, presents the first framework to date that performs multi-session map alignment, map maintenance, and localization, within automotive radar maps [27]. Our multi-session mapping approach is able to successfully align trajectories recorded at different times, and

our map maintenance procedure identifies changes in the environment associated with moveable objects or noise. Furthermore, we develop an approach to localize within these maintained maps, resulting in improvements in the pose estimation performance compared to existing LiDAR localization approaches.

In sum, this thesis presents five contributions associated with localization and mapping for autonomous vehicles using automotive radars that have also been published in five papers. Each of them is targeted at specific components of a SLAM pipeline, where the autonomous vehicle estimates its own pose, constructs a map of the environment, and localizes within the map over long periods of time. All methods presented in this work are tested on radar data collected using automotive radar sensors, and are designed to be leveraged within real-world autonomous driving scenarios.

1.2 Publications

Parts of this thesis have been published in the following peer-reviewed conference and journal articles:

- D. Casado Herraez, M. Zeller, L. Chang, I. Vizzo, M. Heidingsfeld, and C. Stachniss. Radar-Only Odometry and Mapping for Autonomous Vehicles. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024. DOI: 10.1109/ICRA57147.2024.10610311

- D. Casado Herraez, L. Chang, M. Zeller, L. Wiesmann, J. Behley, M. Heidingsfeld, and C. Stachniss. SPR: Single-Scan Radar Place Recognition. *IEEE Robotics and Automation Letters (RA-L)*, 9(10):9079–9086, 2024. DOI: 10.1109/LRA.2024.3426369

- D. Casado Herraez, F. Kaschner, M. Zeller, D. Muhle, J. Behley, M. Heidingsfeld, D. Cremers, and C. Stachniss. Ground-Aware Automotive Radar Odometry. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2025. DOI: 10.1109/ICRA55743.2025.11128161

- D. Casado Herraez, M. Zeller, D. Wang, J. Behley, M. Heidingsfeld, and C. Stachniss. RaI-SLAM: Radar-Inertial SLAM for Autonomous Vehicles. *IEEE Robotics and Automation Letters (RA-L)*, 10(6):5257–5264, 2025. DOI: 10.1109/LRA.2025.3557296

- D. Casado Herraez, M. Zeller, D. Wang, J. Behley, M. Heidingsfeld, and C. Stachniss. Multi-Session Mapping and Long-Term Localization for Autonomous Vehicles Using Radar. *IEEE Robotics and Automation Letters (RA-L)*, 11(3):2522–2529, 2025. DOI: 10.1109/LRA.2026.3653332

In addition, this thesis has contributed to projects on automotive radar perception and SLAM published in the following peer-reviewed conference and journal articles:

- M. Zeller, D. Casado Herraez, J. Behley, M. Heidingsfeld, and C. Stachniss. Radar Tracker: Moving Instance Tracking in Sparse and Noisy Radar Point Clouds. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024. DOI: 10.1109/ICRA57147.2024.10610198

- M. Zeller, D. Casado Herraez, B. Ayan, J. Behley, M. Heidingsfeld, and C. Stachniss. SemRaFiner: Panoptic Segmentation in Sparse and Noisy Radar Point Clouds. *IEEE Robotics and Automation Letters (RA-L)*, 10(2):923–930, 2024. DOI: 10.1109/LRA.2024.3502058

- L. Schwarzer, M. Zeller, D. Casado Herraez, D. Simon, M. Heidingsfeld, and C. Stachniss. Self-Supervised Moving Object Segmentation of Sparse and Noisy Radar Point Clouds. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2025. DOI: 10.1109/ITSC60802.2025.11423257

- D. Wang, H. Haag, D. Casado Herraez, S. May, C. Stachniss, and A. Nuechter. Doppler-SLAM: Doppler-Aided Radar-Inertial and LiDAR-Inertial Simultaneous Localization and Mapping. *IEEE Robotics and Automation Letters (RA-L)*, 10(9):9438–9445, 2025. DOI: 10.1109/LRA.2025.3595071

Chapter 2

Basic Techniques

This chapter introduces basic techniques and methodologies used in this thesis. We first give an overview of the working principle of automotive radar sensors and how they can be leveraged for autonomous driving. We then provide a summary of relative body motion and how this can be leveraged given radar velocity measurements. This is followed by an outline of the least squares problem formulation and Lie theory as the basis for many of the methodologies introduced in this thesis. We continue the fundamentals by deriving the formulation of the simultaneous localization and mapping (SLAM) problem, associating it with the least-squares formulation of factor graphs, and describing the details of each individual component. We conclude the chapter by presenting the SLAM evaluation metrics that will be used in the following chapters of the thesis.

2.1 Automotive Sensors

Having a reliable and robust sensor set is essential for safe autonomous driving in all scenarios, including rain, snow, fog, low light, and conditions with no satellite visibility. Currently, the most used sensors for perceiving the surroundings of an autonomous vehicle are cameras and LiDARs. Cameras are popular due to their low cost, ease of packaging within a vehicle, and the high amount of visual information they provide about a scene. Research in camera perception has been a significant focus in autonomous driving [60], as it provides information about our world in a similar way to how we see it. However, since the output of monocular cameras is a projection onto an image, one of the main limitations of monocular camera systems is the lack of depth information. Several works have approached this problem with depth estimation techniques [38, 47], but this remains a prediction coming from 2D image information subject to scale ambiguity.

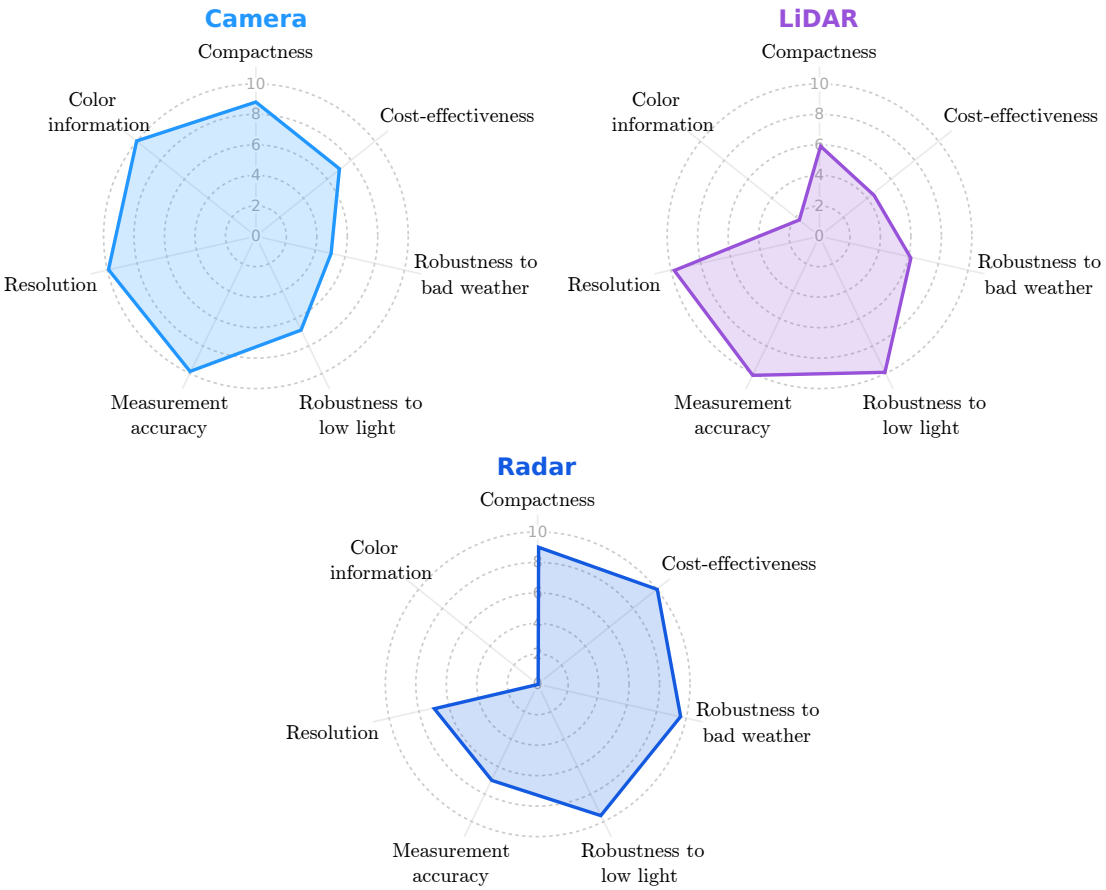


Figure 2.1: Comparison of sensors used in autonomous vehicles inspired by Chan et al. [31]. Although cameras provide high-resolution, texturized images, they are affected by low light and bad weather conditions. LiDARs are robust to low light, but remain expensive alternatives hard to pack into consumer vehicles. Radars, however, are compact, affordable, and resilient to environmental conditions, but present challenges in resolution and accuracy of the point clouds.

Stereo cameras address this issue, but they are less compact to integrate within a vehicle. They are also affected by lighting and bad weather conditions [115], and face challenges while driving in the dark, heavy rain, fog, and snow. LiDARs capture depth information by emitting light pulses and measuring the time delay between the emission and the reception of the reflected signal. The sensor operates at a frequency between the microwave and infrared spectrum, producing a 3D point cloud that contains spatial information not present in 2D images. LiDAR sensors that operate with a rotating mechanical component are called 'spinning LiDARs' while those that have no moving parts are called 'solid-state LiDARs'. Due to the high accuracy of the output point cloud, this sensor modality has received significant attention in autonomous vehicle perception, including mapping [99, 113, 116, 123, 141, 172, 192], place recognition [48, 93, 171, 179, 185, 189], or scene understanding [35, 119, 131]. The main disadvantages of LiDARs are that they come at a high cost, making them unsuitable for mass production, and, sim-

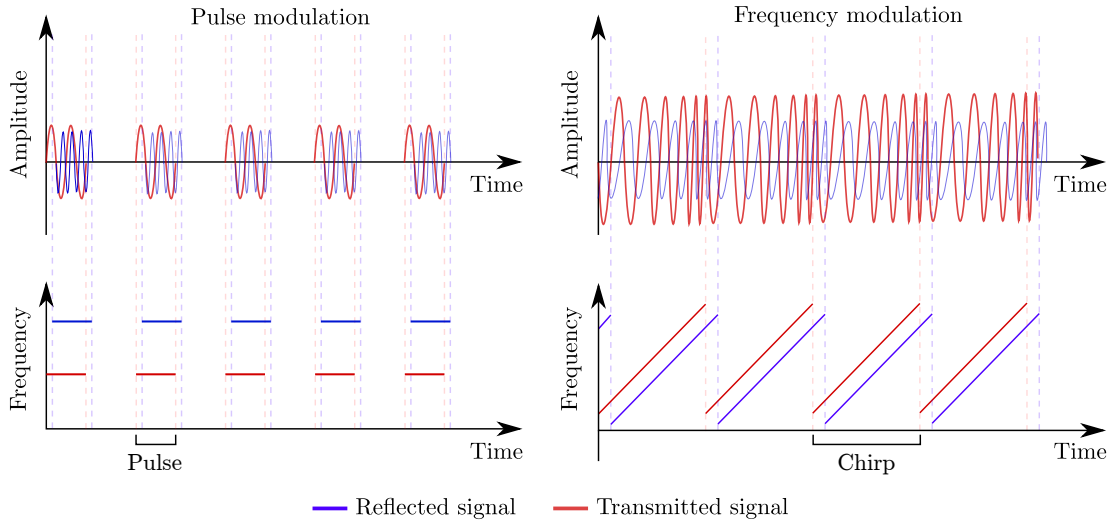


Figure 2.2: Comparison of pulse modulation (left) and frequency modulation (right) of radar sensors. While pulse modulation generates periodic pulses of a specific frequency, frequency modulation generates consecutive chirps of varying frequency over time. A variation in the received frequency indicates a velocity in the measured target due to the Doppler effect.

ilar to cameras, their performance degrades under bad weather [194]. Compared to cameras and LiDARs, radars provide depth information, are affordable, and are not affected by bad weather and low-lighting conditions. Moreover, they are already integrated in end-user vehicles today. They overcome the main challenges of cameras by actively measuring depth in the scene, and are more compact and affordable than LiDARs, as shown in Figure 2.1. Furthermore, they provide additional information, including the relative radial velocity of the measured target as well as its radar cross section. However, their noisy and sparse output presents a different set of challenges when it comes to pose estimation and mapping.

2.1.1 Radar Working Principles

Radar (radio detection and ranging) sensors operate on the radio frequency spectrum. This difference in the spectrum between LiDARs and radars is the key to their advantages and limitations [205]. The LiDAR's shorter wavelength enables a high angular resolution, but the radar's longer wavelength facilitates penetration through heavy rain, fog, and snow. This difference in the frequency range also results in a different form of interaction of the waves with the environment. The longer radio wavelengths are minimally affected when penetrating through small water droplets, while short visible light and infrared wavelengths scatter when interacting with these particles. Furthermore, the cost of automotive radars is lower (below 100€) than the price of LiDARs (usually above 500€).

Based on their working principle, radars can be categorized into two categories illustrated in Figure 2.2, pulse and frequency modulation. Pulse modulated

radars emit a series of short-duration pulses with high energy. They can obtain the range of the target by measuring the time delay between the transmitted and received pulse, and the Doppler velocity by measuring the change in frequency between the transmitted and received pulse. However, these sensors require that the signal is sufficiently powerful such that the return energy of the pulses can be measured by the receiver. As a result, the sensor becomes larger and more expensive. This limitation is not present in frequency modulated radars, making them suitable alternatives to integrate within end-user vehicles. Moreover, as we will be discussing in this chapter, signal processing techniques enable obtaining additional information from frequency modulated radars, including range, Doppler velocity, azimuth, and elevation angles.

2.1.2 Frequency Modulated Continuous Wave Radars

Frequency modulated continuous wave (FMCW) radar sensors operate with electromagnetic wavelengths between 1 and 10 mm, corresponding to frequencies between 30 and 300 GHz. In automotive applications, these sensors use standardized frequencies between 76 and 81 GHz, which enables compact antenna designs.

Frequency modulated radars contain a voltage-controlled oscillator that produces a frequency-modulated wave. The shape of this wave in automotive radars commonly follows a sawtooth pattern with the frequency increasing linearly over a chirp's duration, as illustrated in Figure 2.2. This signal is transmitted over the environment, and the resulting reflections are collected in the receiver antenna. As it will be described in this section, by comparing the transmitted and received signals, it is possible to obtain both the range and the Doppler velocity of the measured targets. Additionally, combining multiple transmitter and receiver antennas enables the retrieval of the azimuth angle of the measured target.

2.1.3 The Radar Equation

The radar equation summarizes the working principle of radar sensors. Although it is highly relevant for designing and manufacturing radar sensors, it serves as a useful introduction to understand how the radar measurements are collected and how different factors affect sensing performance. The radar equation is given as

$$P_r = P_t \left(\frac{G_t}{4\pi R_t^2}\right)\left(\frac{G_r}{4\pi R_r^2}\right)\left(\frac{\lambda^2\sigma}{4\pi}\right)\left(\frac{1}{L}\right) = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R_t^2 R_r^2 L}. \tag{2.1}$$

This formula quantifies the relationship between the received power P_r, and the transmitted power P_t. Higher transmitting power can increase the range and detection capabilities, trading off power consumption. The received power depends on several factors affecting the transmitted signal.

The main influencing factors include the transmitter antenna gain G_t and the receiver antenna gain G_r. These variables indicate how effectively the emitting antenna directs radiated power into a specific direction compared to an isotropic antenna, and how well the antenna can focus the energy coming from a specific direction into the radar receiver compared to an isotropic antenna. While higher antenna gains indicate higher directionality, lower gains provide a wider angle of transmitting and receiving radio signals.

Another influencing factor on the received power is the wavelength of the radar signal $\lambda = \frac{c}{f}$, which depends on the speed of light c and the carrier signal frequency f. The wavelength determines the spatial resolution of the radar, with smaller wavelengths making it possible to extract finer details.

The target's reflectivity also has an impact on the received power, and it is known as the radar cross section (RCS), σ. It depends on the size, material, angle of incidence, and frequency of the signal. For instance, the RCS associated with a car is different from the RCS corresponding to a person [198]. Note, however, that the RCS determines how much energy is reflected back from the target, but it does not depend on the target's range.

Additional signal degradation comes from losses due to hardware inefficiencies, propagation effects, and signal processing, denoted as L. Moreover, it can be observed how the first and second terms of the radar equation represent the transmitter and receiver gain spreading loss over spheres of radius R_t and R_r, known as two-way propagation loss. Since most automotive radar systems employ the same receiving and transmitting antenna, which results in $R = R_t = R_r$, and $G = G_t = G_r$, Equation (2.1) simplifies to

$$P_r = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 L}. \tag{2.2}$$

The radar equation contains the fundamental principles of radar operational design; however, it does not describe how useful information can be extracted from radar signals. In the following, we describe the signal processing techniques of how radars measure the environment and output a point cloud that can be used for autonomous driving functionalities.

2.1.4 Range-Doppler Information From FMCW Radars

To obtain the range-Doppler information from the radar signals, it is necessary to understand the difference between the received frequency (f_R), the beat frequency (f_B), the Doppler frequency (f_D), and the relationship between them [106].

First, due to the time delay between the transmitted and received signals, the received signal has a frequency shift of f_R. This is dependent on the range of the target R and parameters from the sawtooth modulation including the signal

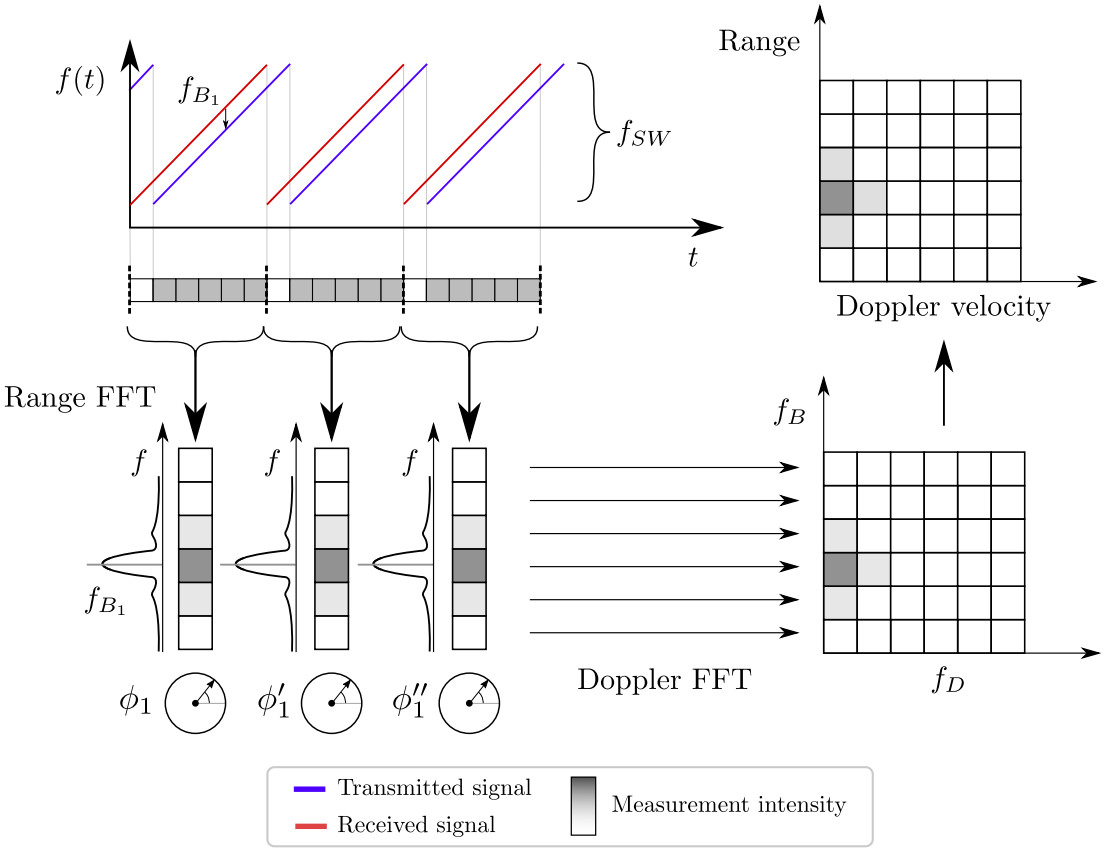


Figure 2.3: Extraction of the range-Doppler values for a **single static target**. The received signal is shifted in time, and the beat frequency f_{B_1} and phases $\phi_1, \phi_1', \phi_1''$ of each individual chirp can be extracted with the range FFT. By computing the Doppler FFT over multiple chirps, the Doppler frequency shift of the signal is retrieved based on the change in phase across chips. Note that for a static target, no phase change occurs. The result can be converted to range-Doppler measurements following Equations (2.7) and (2.8).

bandwidth f_{SW} and the chirp duration T_{chirp}, such that

$$f_R = \frac{f_{SW}}{T_{\text{chirp}}} \frac{2R}{c}. \tag{2.3}$$

The factor $\frac{f_{SW}}{T_{\text{chirp}}}$ represents the chirp-rate, which is the rate at which the frequency changes over time, and $\frac{2R}{c}$ represents the round-trip time, with c being the speed of light.

The beat frequency f_B represents the difference in frequency between the transmitted signal $f_t(t)$ and the received signal $f_r(t)$. Subsequently, when measuring a static target, as in Figure 2.3, the beat frequency is only influenced by the measured target's range, resulting in

$$f_B = f_t(t) - f_r(t) = f_R. \tag{2.4}$$

Under the assumption that the target's range and velocity remain constant within a single chirp, a fast Fourier transform (FFT) known as "range FFT"

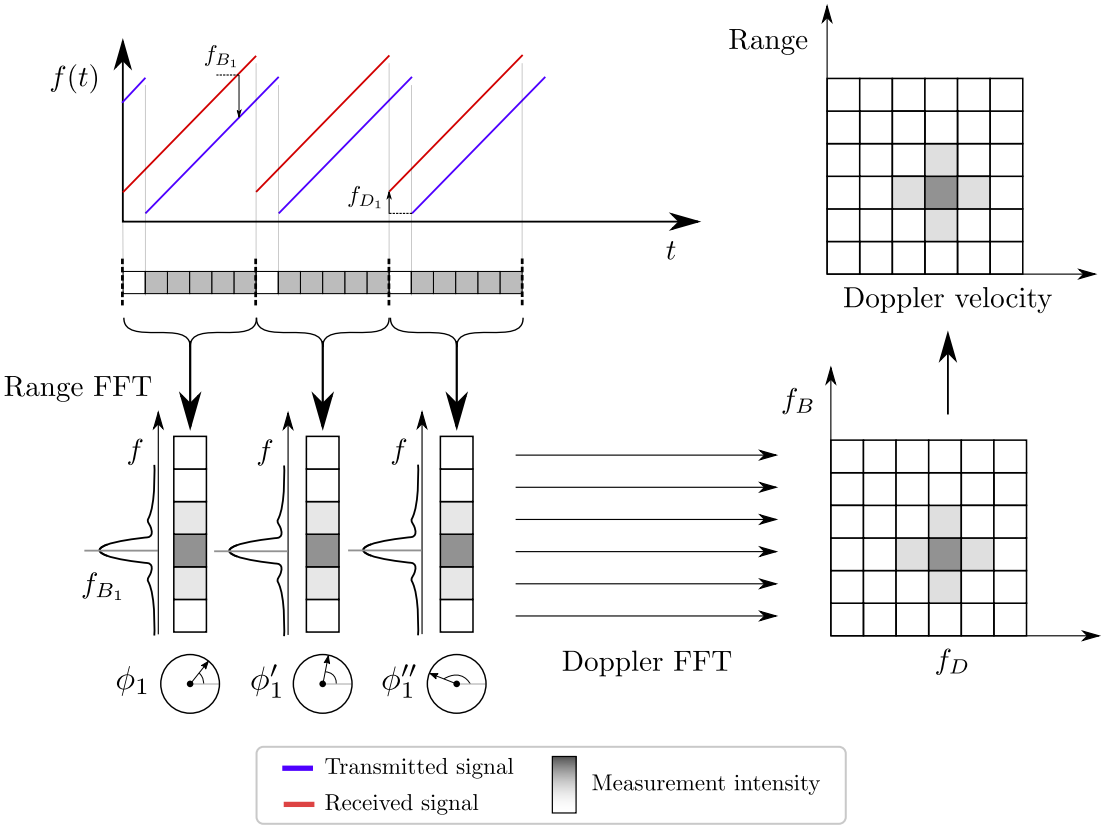


Figure 2.4: Extraction of the range-Doppler values for a **single dynamic target**. The received signal is shifted in time, and the beat frequency f_{B_1} and phases $\phi_1, \phi_1', \phi_1''$ of each individual chirp can be extracted with the range FFT. By computing the Doppler FFT over multiple chirps, the Doppler frequency shift of the signal is retrieved based on the change in phase across chips. The result can be converted to range-Doppler measurements following Equations (2.7) and (2.8).

or "fast time FFT" measures the difference between the transmitted and received signals and outputs a complex number whose magnitude determines the beat frequency f_B, while the angle defines the phase of the received signal ϕ. If there is a single target, like in Figure 2.3 and Figure 2.4, a single beat frequency value can be extracted, corresponding to the measured target. However, if there are multiple targets, like in Figure 2.5 and Figure 2.6, the FFT will return multiple peaks and their corresponding phases, each corresponding to one measured target.

In the case of dynamic targets, Figure 2.4 and Figure 2.6, the measured point is moving closer or farther away from the sensor. In the sawtooth modulation pattern typically used in FMCW radars, this introduces a change in phase between the individual chirps, leading to a Doppler frequency shift defined as

$$f_D = -2\frac{v_r}{\lambda}, \tag{2.5}$$

which is affected by the target's radial velocity with respect to the vehicle v_r and the signal wavelength λ. This change in frequency caused by a target's velocity is commonly known as the Doppler effect. It is important to note that this frequency

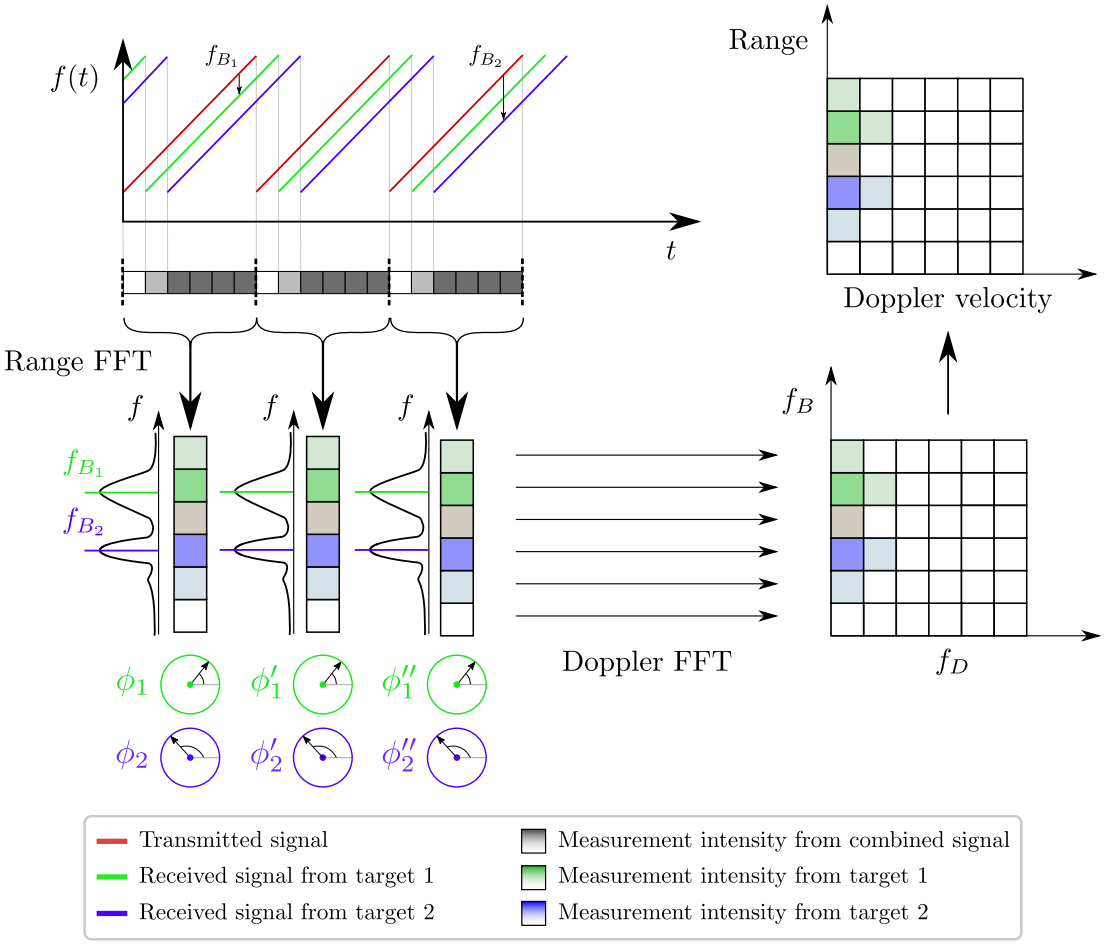


Figure 2.5: Extraction of the range-Doppler values for **multiple static targets**. The beat frequencies of the reflected signals from both targets result in two peaks from the range FFT. Since the phases remain constant, no Doppler velocity is computed from the Doppler FFT.

shift within a single chirp is generally small, minimally affecting the extraction of the beat frequency from the range FFT. This effect on the beat frequency from Equation (2.4) is defined by

$$f_B = f_R + f_D = \frac{f_{SW}}{T_{\text{chirp}}} \frac{2R}{c} - 2\frac{v_r}{\lambda}. \tag{2.6}$$

Once a certain number of chirps have been received, it is possible to determine the Doppler frequency shift caused by the moving target by analyzing the change in phase of several individual chirps. This is done by applying an FFT over a set of results from multiple range FFTs. This procedure is known as the "Doppler FFT", or "slow time FFT". The result is a 2D grid where the columns represent the Doppler frequency shift f_D and the rows represent the beat frequency f_B ranges.

After computing f_B and f_D, it is possible to move from the beat-Doppler frequency grid to the range-velocity measurement grid with the equations derived

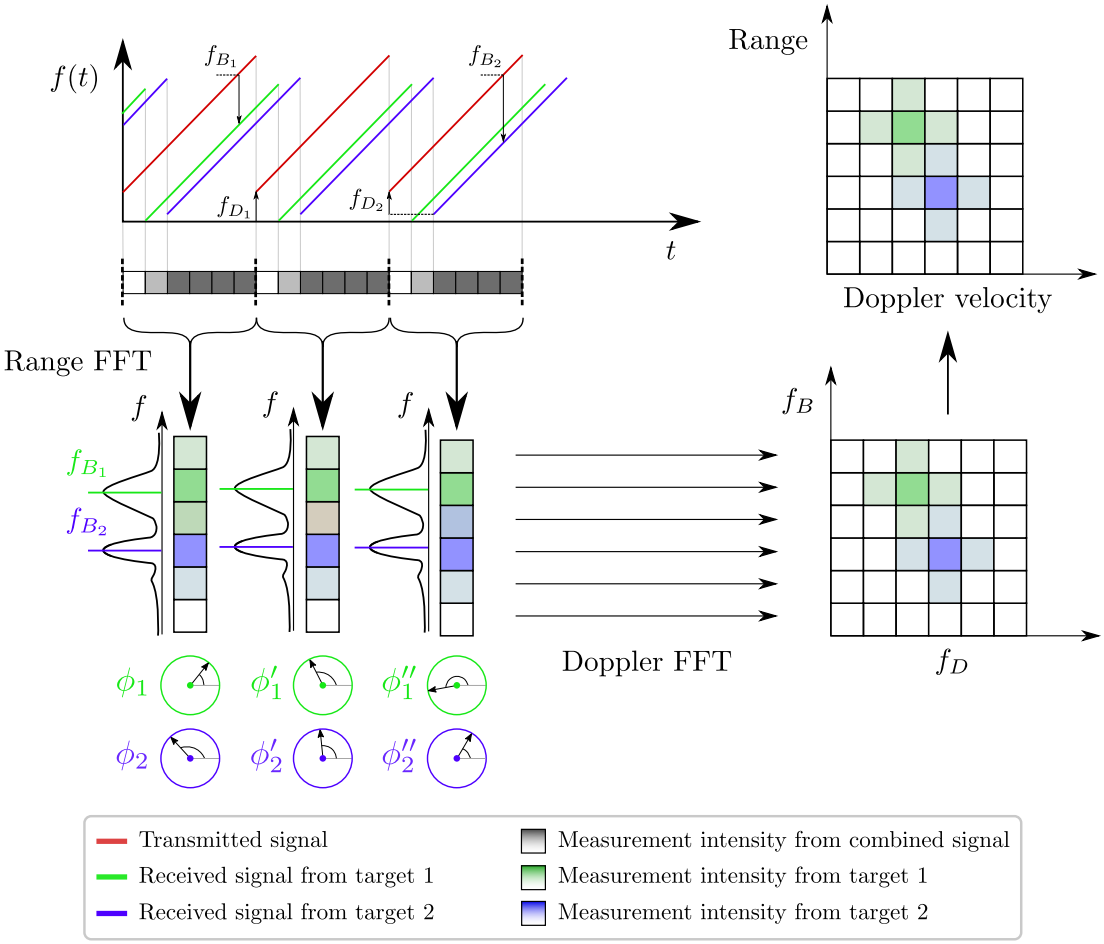


Figure 2.6: Extraction of the range-Doppler values for **multiple dynamic targets**. The received signals are shifted in time and frequency, leading to phase variations from the range FFT. These phase variations result in Doppler velocities measured by the Doppler FFT.

from Equation (2.5) and Equation (2.6) following

$$R = (f_B - f_D)\frac{T_{chirp}}{f_{SW}}\frac{c}{2}, \tag{2.7}$$

$$v_r = -\frac{\lambda}{2} f_D. \tag{2.8}$$

2.1.5 Azimuth Angle Information From FMCW Radars

The previous section explained how to obtain range and velocity information from an FMCW radar. To measure the surroundings and obtain azimuth angle information, two radar sensing techniques are commonly employed.

Spinning radars contain a mechanically spinning device with a transmitter-receiver antenna setup. This allows the sensor to obtain measurements in a 360° surround-view around the car [11]. As it will be presented in Chapter 3, while

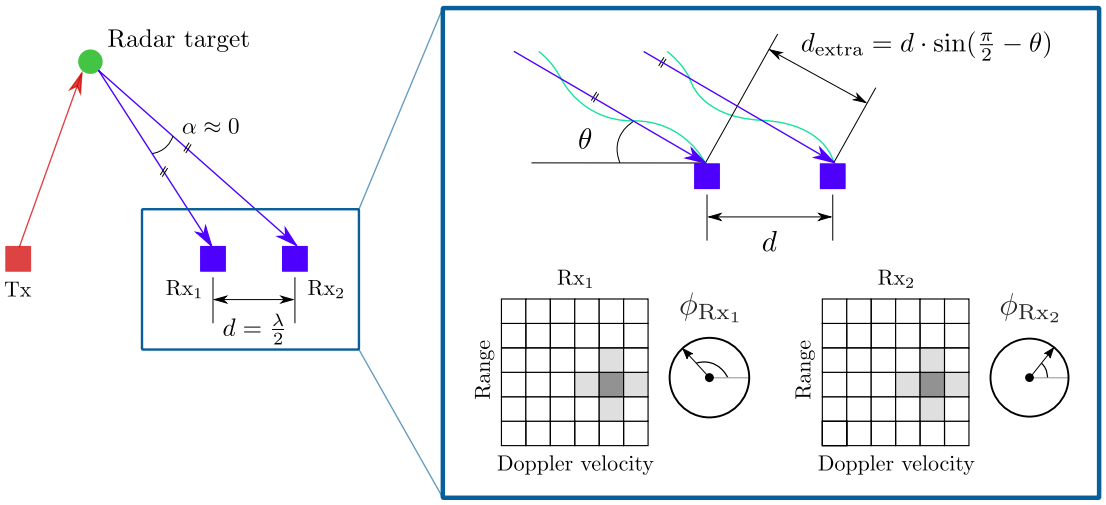


Figure 2.7: Range-Doppler values and phase of one target using one transmitter (Tx) and two receiver (Rx) antennas. Since the signal travels different distances to the two receiver antennas, this can be measured as a change in phase of the received signal between the two antennas.

several works employ spinning radars for pose estimation, the design leads to bulky sensor setups which are unsuitable for consumer vehicles.

Multiple-input-multiple-output radars, commonly referred to as automotive radars in the context of robotics, are more compact and affordable, suitable for integration within end-user vehicles. Instead of containing a mechanically spinning device, automotive radars are composed of an array of transmitter-receiver antennas that measure the delay in the received signals between them. The separation distance between receiver antennas is fixed to $d = \frac{\lambda}{2}$.

Figure 2.7 shows the simplest configuration of an automotive radar, with a single transmitter antenna and two receiver antennas. The transmitter antenna emits a signal that is reflected off a target, being measured by the receiver antennas at different distances. Subsequently, the wave needs to travel longer distances to reach receiver antennas that are farther away from the target. Due to the small signal wavelength (1-10 nm) compared to the distance to the target (1-200 m), it can be assumed that the reflected signal reaches all the receiver antennas in a parallel manner.

The resulting extra distance traveled by the reflected signal to reach the next receiver antenna depends on the azimuth angle θ, and can be measured using the phase shift between the reflected signals in both receiver antennas following

$$d_{\text{extra}} = d \cdot \sin\left(\frac{\pi}{2} - \theta\right). \tag{2.9}$$

Since the difference in the travelled distance is small, in the order of millimeters, measuring this distance with each receiver antenna leads to a similar range-Doppler matrix with a shift in the received signal's phase. Combining them using an FFT, known as the "azimuth FFT" results in the estimation of

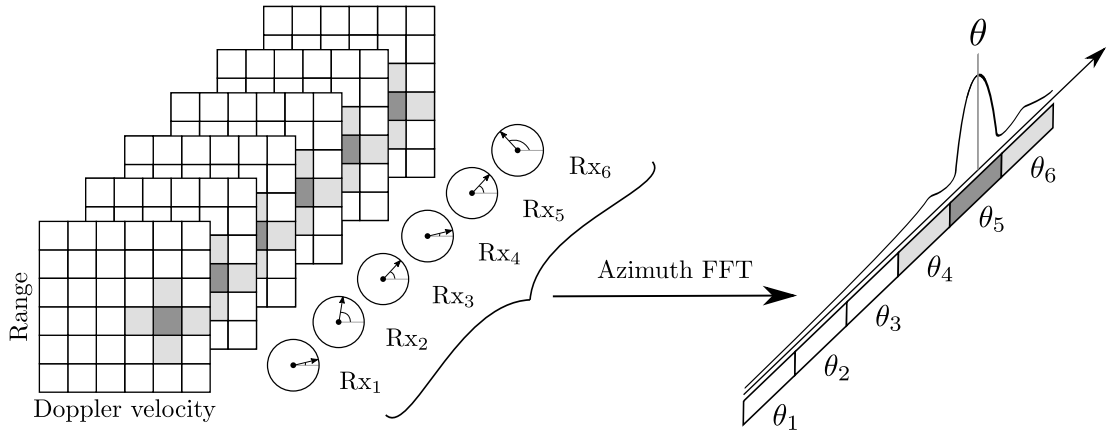


Figure 2.8: Azimuth angle measurement of a single dynamic target from six receiver antennas. The different phases measured by each receiver antenna can be exploited with the azimuth FFT to extract the azimuth of a target. The peak of the FFT indicates the target's azimuth angle.

the azimuth angle θ for each receiver channel, with a peak indicating the target's azimuth angle. As a result, a higher number of antennas leads to a higher azimuth angle resolution. This process is illustrated in Figure 2.8 for the case of a single dynamic target over the measurement of six receiver radar antennas.

Although it is common to refer to radars that measure range, azimuth, and Doppler as 3D sensors, since their geometric output lies in a 2D plane, we refer to them as 2D radars. This naming deviates from convention but improves clarity when comparing it with radars that provide a 3D geometric output.

2.1.6 Elevation Angle Information From FMCW radars

While some radars only provide 2D radar measurements, the latest automotive radars can also provide elevation angle information. The methodology for achieving this follows the same principle from the previous Section 2.1.5. However, the antennas must be aligned vertically rather than horizontally.

A combination of both horizontal and vertical alignment of transmitter and receiver antennas in a 2D matrix leads to an automotive radar that provides range, Doppler, azimuth, and elevation angles of the measured targets. Radar sensors with these four output channels are commonly known as 4D radars.

Although the output of a radar sensor that measures range, azimuth, elevation, and Doppler information is four-dimensional, and the literature sometimes refers to them as 4D radars, since their geometric output remains three-dimensional, range-azimuth-elevation, which can be transformed to x-y-z cartesian coordinates, in this work we will refer to these sensors as 3D radars, for an easier distinction between 2D and 3D point coordinates.

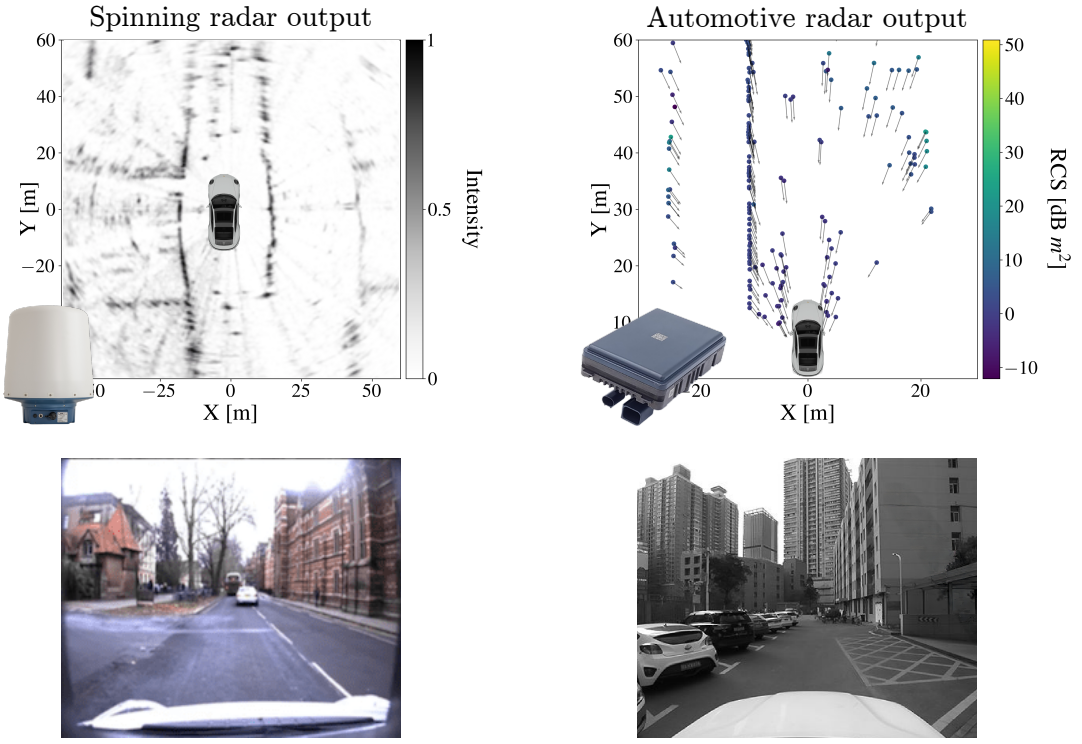


Figure 2.9: Comparison between spinning [11] and automotive radars [78] with a reference camera image. While spinning radars are bulky and output a 2D intensity image, automotive radars are compact and output a point cloud with Doppler velocities and RCS information.

2.1.7 Radar Sensors for Autonomous Driving

The information provided by FMCW radars can vary depending on the sensor type. We can group radars into two main categories based on their output and working principle: spinning radars and automotive radars, shown in Figure 2.9.

Spinning radars consist of a mechanically spinning antenna that computes a range measurement for each azimuth bin. They output a 2D image, as displayed in Figure 2.9 (left), where each pixel represents the intensity of the measurement. Spinning radar sensors provide high accuracy and 360 degree field-of-view, and researchers have leveraged them to perform odometry [5, 6, 30, 83], place recognition [56, 166, 197] and SLAM [74]. However, they are costly, and their dimensions make them hard to mount in consumer vehicles.

Automotive radars output a point cloud containing the spherical coordinates of a point (range, azimuth, elevation), which are usually converted to Cartesian coordinates (x, y, z). They also provide the radar cross section of each point and the Doppler velocity, which corresponds to the projection of the target's velocity in the radial direction of the measurement. Automotive radars have also shown outstanding results in object segmentation [198], object tracking [200], odometry [89], and place recognition [22]. The raw output of automotive radar sensors is a range-azimuth-Doppler 3D matrix representation of the environment for

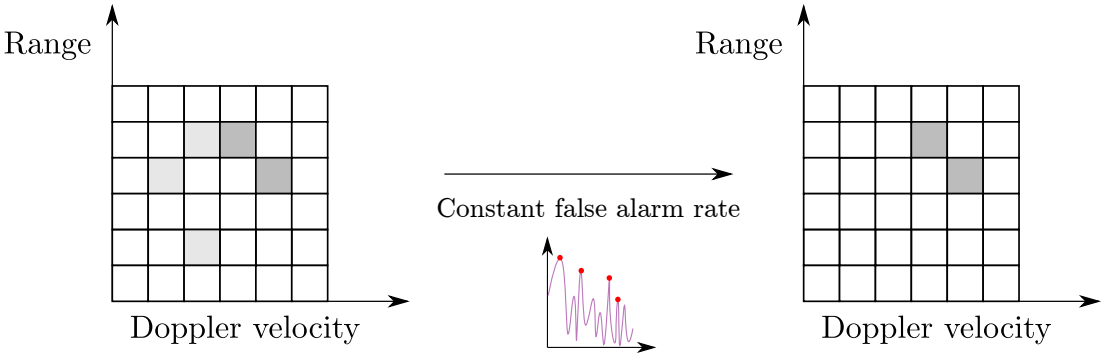


Figure 2.10: The constant false alarm rate algorithm estimates peaks of maximum intensity. It applies a threshold resulting in a binary grid where a target is likely to be present.

each sensor channel. Some researchers [104, 153] directly use the range-azimuth-doppler matrix for their radar solutions. However, raw data is not usually provided by the radar sensor manufacturers as it can reveal their specific signal processing techniques. Instead, most manufacturers directly apply a constant false alarm rate [152] detection algorithm to obtain a point cloud from the scans. As illustrated in Figure 2.10, we can identify the detection peaks and discard measurements with a lower intensity value. The output results in a point cloud, as illustrated in Figure 2.9 (right). However, despite the algorithm leading to a point cloud containing the main peak reflections, challenges still remain when it comes to dealing with clutter and multi-path reflections, and it can be challenging to discriminate between direct measurements of objects and those coming from clutter or reflections. The difficulty becomes even greater due to the low number of points per scan, as it can be challenging to identify objects and structural elements within the environment.

Some research points in the direction of employing machine learning to differentiate between noise and non-noise measurements to filter the point cloud [105] or using rule-based geometric models [88, 103, 147, 154] that measure the distance traveled by the radar signal and identify points that are farther away than a certain threshold. Other methods aim to approximate radar scans to LiDAR point clouds using occupancy grids [110, 183] or voxel maps [153]. However, the sparse and noisy nature of radar scans makes it difficult to directly apply LiDAR solutions for localization and mapping. These methodologies also suffer from the limitations of restricting the output points to those that match the properties of the LiDAR sensors. Subsequently, cars visible to radar but invisible to LiDARs will be discarded. Moreover, they are bound to the point coordinates of the LiDAR scans, without exploiting the radar Doppler velocities and RCS information provided by the sensor.

Table 2.1: ARS548, ZF FRGen21 radars and Ouster OS0-128 LiDAR sensor characteristics.

Parameter	ARS548 Radar			ZF FRGen21 Radar			Ouster OS0-128 LiDAR		
	Range	Resolution	Accuracy	Range	Resolution	Accuracy	Range	Resolution	Accuracy
Distance [m]	0.20 - 301	0.22	±0.15	0 - 100	0.20	±0.02	0.25 - 75	0.005	±0.03
Azimuth [°]	±60	1.68	±0.5	±75	1.50	±0.15	±180	0.18	±0.05
Elevation [°]	±14	2.30	±0.1	±15	1.50	±0.3	±45	0.703	±0.05
Velocity [km/h]	-400 - 200	0.35	±0.1	±140	0.36	±0.036	-	-	-

2.1.8 Noise, Multi-Path Propagation, and Sparsity

Despite being able to measure range, azimuth, elevation, and Doppler velocity information from radar sensors, obtaining a radar point cloud composed solely of true target detections remains a significant challenge. As outlined in the automotive radar noise characterization by Chan et al. [31], false detections may come from various sources.

Two of the main causes of radar noise are the hardware components and the processing of the received signal. Automotive radars are susceptible to heating of the electronic components, the material properties of the radar sensor casing, and the component inefficiencies of the radar system. Moreover, the bumpers of the autonomous vehicle, which contain the radar sensors, also have an effect on the noise present in the output point cloud. Additional random noise in automotive radars arises from the signal processing procedures. In particular, the FFTs used to extract the range, azimuth, elevation, and Doppler velocities can introduce sidelobes or spectral leakage, manifesting as false detections. This is represented in Figures 2.3–2.8 and Figure 2.10, as the variation of the intensity values in the different grid cells, which can lead to additional peaks when applying the constant false alarm rate algorithm. Nevertheless, some of the false detections originated from hardware and signal processing are random points inconsistent across multiple radar measurements. As it will be introduced in Chapter 4, these can be filtered out by measuring the spatial consistency between radar scans.

False noise measurements coming from multi-path propagations are more challenging to identify. These radar targets originate from measured radar signals that reflect off multiple surfaces before returning to the radar, and typically come from objects with high reflectivity, such as guardrails or cars. These indirect paths usually create what is known as “ghost” targets, as the radar interprets these measurements at incorrect distances or angles. Unlike random noise, multi-path propagation may be consistent across multiple measurements, leading to problems in differentiating them from real objects in the scene. Our place recognition method from Chapter 6 exploits these observations to identify which points in the radar scan are reliable, hence belonging to true radar detections.

Another challenge in automotive radars is the sparsity. To keep the sensor small, manufacturers limit the number of transmitter and receiver antennas. As

shown in Table 2.1, this results in reduced measurement resolutions compared to high-definition LiDAR sensors, especially in the vertical component. We tackle these problems in Chapter 4 and Chapter 5, and despite these limitations, we manage to achieve accuracy comparable to LiDAR sensors with our pose estimation strategies by exploiting the additional Doppler and RCS information provided by automotive radars.

2.2 Relative Body Motion

Radar sensors measure the Doppler velocity of each point. However, due to the Doppler effect, these velocities only account for the radial component of the velocity with respect to the measurement direction. Therefore, the radar does not provide the full measurement of the targets' velocities. In this section, we derive the basics of relative motion analysis to compute the velocity of a point given the sensor velocity, and how this can be leveraged to project the car velocities into the radar's radial measurement direction. This has two main applications detailed in Chapter 4 that include: leveraging the Doppler velocities for pose estimation, and dynamic outlier detection.

2.2.1 Velocity of a Point Given the Sensor Velocity

To compute the velocity of a point given the sensor velocity, we define two coordinate frames: an absolute inertial frame $\{I\}$, which remains static, and a moving sensor frame $\{S\}$, as illustrated in Figure 2.11. The goal is to find the velocity of a point $\mathbf{p}_k$ in the inertial frame, assuming that the absolute velocity of the sensor origin is known.

The position vector of $\mathbf{p}_k$ relative to the sensor origin, expressed in the sensor frame $\{S\}$, is defined as

$$ {}^{S}\mathbf{t}^{S}_{\mathbf{p}_k} = {}^{S}\mathbf{t}^{I}_{\mathbf{p}_k} - {}^{S}\mathbf{t}^{I}_{S}. \tag{2.10} $$

By taking the derivative over time of Equation (2.10) we compute the corresponding velocities expressed in the sensor frame $\{S\}$ following

$$ \frac{d({}^{S}\mathbf{t}^{S}_{\mathbf{p}_k})}{dt} = {}^{S}\boldsymbol{v}_{\mathbf{p}_k} - {}^{S}\boldsymbol{v}_{S}, \tag{2.11} $$

where ${}^{S}\boldsymbol{v}_{S}$ and ${}^{S}\boldsymbol{v}_{\mathbf{p}_k}$ are the absolute linear velocities of the sensor origin and the point, and $\frac{d({}^{S}\mathbf{t}^{S}_{\mathbf{p}_k})}{dt}$ is the absolute time derivative of the relative position vector, all expressed in $\{S\}$.

Moreover, note that ${}^{S}\mathbf{t}^{S}_{\mathbf{p}_k}$ from Equation (2.10) can also be expressed as the coordinates of point $\mathbf{p}_k$ with respect to the sensor frame $\{S\}$ as

$$ {}^{S}\mathbf{t}^{S}_{\mathbf{p}_k} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}, \tag{2.12} $$

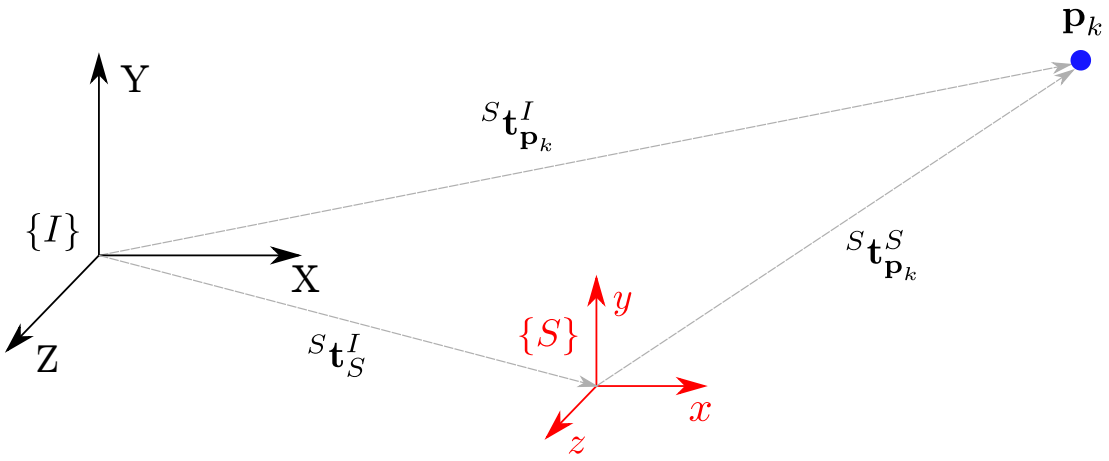


Figure 2.11: Visualization of the relative motion analysis between an absolute inertial frame $\{I\}$, a moving sensor frame $\{S\}$, a point measured by the radar sensor $\mathbf{p}_k$, and the translation vectors between them expressed in the sensor frame denoted as ${}^S\mathbf{t}^S_{\mathbf{p}_k}$, ${}^S\mathbf{t}^I_{\mathbf{p}_k}$, and ${}^S\mathbf{t}^I_S$.

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the unit vectors defining the sensor frame $\{S\}$, and x, y, z are the components measured in $\{S\}$. As a result, the derivative of Equation (2.12) accounts for the change in components and the rotation of the basis vectors of the moving sensor frame following

$$\frac{d({}^S\mathbf{t}^S_{\mathbf{p}_k})}{dt} = \left[\frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} + \frac{dz}{dt}\mathbf{k}\right] + \left[x\frac{d\mathbf{i}}{dt} + y\frac{d\mathbf{j}}{dt} + z\frac{d\mathbf{k}}{dt}\right]. \tag{2.13}$$

The first bracket represents the relative velocity of $\mathbf{p}_k$ measured by the sensor and expressed in the sensor frame $\{S\}$, which we denote

$${}^S\boldsymbol{v}_{\mathbf{p}_k/S} = \frac{dx}{dt}\mathbf{i} + \frac{dy}{dt}\mathbf{j} + \frac{dz}{dt}\mathbf{k}. \tag{2.14}$$

The second bracket represents the contribution due to the rotation of $\{S\}$, which depends on the angular velocity of the sensor frame ${}^S\boldsymbol{\omega}_S$ following

$$x\frac{d\mathbf{i}}{dt} + y\frac{d\mathbf{j}}{dt} + z\frac{d\mathbf{k}}{dt} = {}^S\boldsymbol{\omega}_S \times {}^S\mathbf{t}^S_{\mathbf{p}_k}. \tag{2.15}$$

Combining Equations (2.11)–(2.15) leads to

$${}^S\boldsymbol{v}_{\mathbf{p}_k} - {}^S\boldsymbol{v}_S = {}^S\boldsymbol{v}_{\mathbf{p}_k/S} + {}^S\boldsymbol{\omega}_S \times {}^S\mathbf{t}^S_{\mathbf{p}_k}, \tag{2.16}$$

which can be reordered to isolate the absolute velocity of point $\mathbf{p}_k$, resulting in

$${}^S\boldsymbol{v}_{\mathbf{p}_k} = {}^S\boldsymbol{v}_S + {}^S\boldsymbol{v}_{\mathbf{p}_k/S} + {}^S\boldsymbol{\omega}_S \times {}^S\mathbf{t}^S_{\mathbf{p}_k}. \tag{2.17}$$

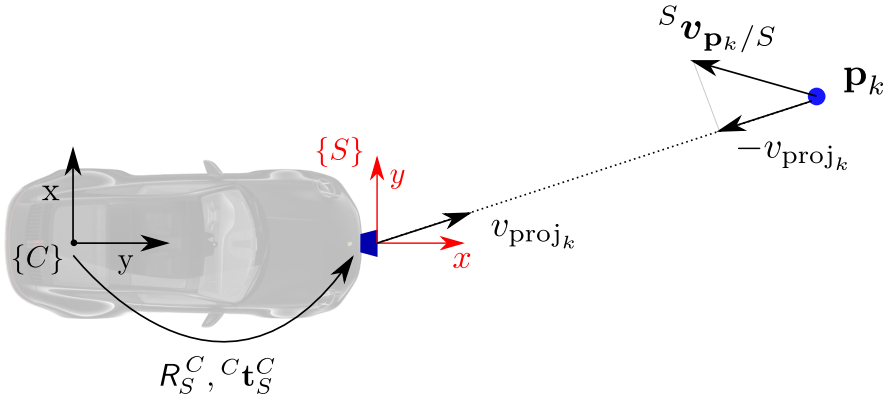


Figure 2.12: Geometry of the problem of projecting a known car velocity into radar measurement direction. If the point $\mathbf{p}_k$ is static, the measured radial velocity will have the same magnitude as the projection of the car velocity onto the radar line of sight with opposite sign.

2.2.2 Projecting the Car Velocity Onto a Point

While Equation (2.17) computes the velocity of a point knowing the sensor velocity, it can be interesting to compare the measured radial velocities with the vehicle's ego-velocity, as detailed in Chapter 4. To achieve this, we project the car velocities into the direction of the measurement. Therefore, in the following derivation, we assume a known car velocity. We follow the procedure of Hexsel et al. [69] assuming rigid body motion between the vehicle and the sensor, see Figure 2.12. Additionally, since pose estimation often relies on a static environment assumption, we focus on measurements with zero absolute velocity, where ${}^S\boldsymbol{v}_{\mathbf{p}_k} = 0$. The relative velocity of a static point measured by the radar and expressed in the sensor frame $\{S\}$ simplifies to

$$ {}^S\boldsymbol{v}_{\mathbf{p}_k/S} = -\left({}^S\boldsymbol{v}_S + {}^S\boldsymbol{\omega}_S \times {}^S\mathbf{t}_{\mathbf{p}_k}^S\right). \tag{2.18} $$

Note, however, that due to the Doppler effect, the radar measurement does not directly include the full measurement of the target's velocity ${}^S\boldsymbol{v}_{\mathbf{p}_k}$. Instead, it provides the projection of this velocity in the radial direction of the point measurement, which we denote $v_{\text{proj}_k} \in \mathbb{R}$. We project the velocity of the static target into the radial direction following

$$ v_{\text{proj}_k} = {}^S\boldsymbol{d}_{\mathbf{p}_k}^S \, {}^S\boldsymbol{v}_{\mathbf{p}_k/S} = -{}^S\boldsymbol{d}_{\mathbf{p}_k}^S \left({}^S\boldsymbol{v}_S + {}^S\boldsymbol{\omega}_S \times {}^S\mathbf{t}_{\mathbf{p}_k}^S\right), \tag{2.19} $$

where the direction vector ${}^S\boldsymbol{d}_{\mathbf{p}_k}^S$ from the radar sensor to the target point $\mathbf{p}_k$ expressed in the sensor frame is given as

$$ {}^S\boldsymbol{d}_{\mathbf{p}_k}^S = \frac{{}^S\mathbf{t}_{\mathbf{p}_k}^S}{||{}^S\mathbf{t}_{\mathbf{p}_k}^S||}. \tag{2.20} $$

Since ${}^S\boldsymbol{d}_{\mathbf{p}_k}^S$ is orthogonal to ${}^S\boldsymbol{\omega}_S \times {}^S\mathbf{t}_{\mathbf{p}_k}^S$, Equation (2.19) simplifies to

$$ v_{\text{proj}_k} = -({}^S\boldsymbol{d}_{\mathbf{p}_k}^S)^\top \, {}^S\boldsymbol{v}_S. \tag{2.21} $$

With Equation (2.21), we have achieved the projection of the radar sensor velocity into the radial direction of the measurement. The goal now is to represent the projection of the linear ${}^{C}\boldsymbol{v}_{C}$ and angular ${}^{C}\boldsymbol{\omega}_{C}$ car velocities, which are expressed in the car frame $\{C\}$, in the direction of the measurement.

Given a car-to-sensor extrinsic calibration including a rotation $\mathbf{R}_{S}^{C} \in \mathbb{R}^{3\times3}$ and a translation ${}^{C}\mathbf{t}_{S}^{C} \in \mathbb{R}^{3}$, we can express the sensor velocity ${}^{C}\boldsymbol{v}_{S}$ in the car frame by rotating it with

$$ {}^{C}\boldsymbol{v}_{S} = \mathbf{R}_{S}^{C}\, {}^{S}\boldsymbol{v}_{S}. \tag{2.22} $$

We rewrite Equation (2.21) reordering terms as in Equation (2.22) leading to

$$ v_{\mathrm{proj}_k} = -({}^{S}\boldsymbol{d}_{\mathbf{p}_k}^{S})^{\top} \left(\mathbf{R}_{C}^{S}\, {}^{C}\boldsymbol{v}_{S}\right) = -\left(({}^{S}\boldsymbol{d}_{\mathbf{p}_k}^{S})^{\top} (\mathbf{R}_{S}^{C})^{\top}\right) {}^{C}\boldsymbol{v}_{S} \tag{2.23} $$

$$ = -\left(\mathbf{R}_{S}^{C}\, {}^{S}\boldsymbol{d}_{\mathbf{p}_k}^{S}\right)^{\top} \boldsymbol{v}_{S} = -({}^{C}\boldsymbol{d}_{\mathbf{p}_k}^{S})^{\top}\, {}^{C}\boldsymbol{v}_{S}. \tag{2.24} $$

Furthermore, assuming that the car is a rigid body and the distance between the car's center of mass and the radar remains constant, the velocity of the sensor with respect to the vehicle is given as

$$ {}^{C}\boldsymbol{v}_{S} = {}^{C}\boldsymbol{v}_{C} + {}^{C}\boldsymbol{\omega}_{C} \times {}^{C}\mathbf{t}_{S}^{C}. \tag{2.25} $$

Finally, we can combine Equation (2.24) and Equation (2.25) to obtain the velocity of a static point in the environment as a function of the car velocities according to

$$ v_{\mathrm{proj}_k} = -({}^{C}\boldsymbol{d}_{\mathbf{p}_k}^{S})^{\top} \left({}^{C}\boldsymbol{v}_{C} + {}^{C}\boldsymbol{\omega}_{C} \times {}^{C}\mathbf{t}_{S}^{C}\right). \tag{2.26} $$

Knowing the projection of the car velocities into the direction of the measurement can be exploited for pose estimation and dynamic outlier detection, as introduced later in Chapter 4.

2.3 Least Squares

Several mathematical problems in geodesy, robotics, and computer science can be formulated as minimization problems with a larger number of equations than unknowns. As a result, we need to find the best solution to a given number of constraints, defined as an overdetermined system. Least squares is a commonly used mathematical optimization technique for finding the best-fit solution of overdetermined systems in the sense of minimizing the sum of squared errors. These systems can usually be represented using a function that consists of a sum of squared residuals between observed and predicted data. Examples where least squares is used include radar scan matching and factor graph optimization, which will be introduced later in Section 2.5.

2.3.1 Problem Formulation

The goal of least squares is to find an optimal state $\mathbf{x}^* \in \mathbb{R}^D$ that minimizes the sum of squared residuals between observed data and predicted values. The observed data is represented as $\mathbf{z}_i \in \mathbb{R}^M$ and usually comes from the output of a sensor or the result of an algorithm. The predicted data $\hat{\mathbf{z}}_i \in \mathbb{R}^M$ is obtained from a prediction function $f_i(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^M$, for example, a function that predicts the relative transformation between two point clouds. The error function between the observed and predicted data is given as

$$\mathbf{e}_i(\mathbf{x}) = \mathbf{z}_i - f_i(\mathbf{x}) = \mathbf{z}_i - \hat{\mathbf{z}}_i. \tag{2.27}$$

The squared error of a measurement $e_i \in \mathbb{R}$ is given as

$$e_i(\mathbf{x}) = \mathbf{e}_i(\mathbf{x})^\top \mathbf{W}_i \mathbf{e}_i(\mathbf{x}), \tag{2.28}$$

where $\mathbf{W}_i$ is a symmetric information matrix for that measurement. It represents the inverse of the covariance matrix $\mathbf{W}_i = \Sigma_i^{-1}$ and quantifies the precision or confidence of the measurement. Higher values in $\mathbf{W}_i$ indicate lower uncertainty and greater weight of the corresponding residual.

We want to find the optimal state $\mathbf{x}^*$ that minimizes the error given all measurements where the final objective function $F(\mathbf{x})$ is the sum of the squared errors

$$F(\mathbf{x}) = \sum_{i=1}^{N} e_i(\mathbf{x}). \tag{2.29}$$

The minimization problem is given as

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\, F(\mathbf{x}). \tag{2.30}$$

If $f_i(\mathbf{x})$ is linear, the resulting objective function is convex. We, therefore, can obtain the optimal solution $\mathbf{x}^*$ by finding the point of zero gradient that fulfills

$$\frac{\partial F}{\partial \mathbf{x}} = 0. \tag{2.31}$$

Deriving the global error and the globally optimal solution, however, may not be possible due to nonlinearities in the prediction functions $f_i(\mathbf{x})$, hence it is common to employ numerical approaches to compute locally optimal solutions to the problem in an iterative manner.

2.3.2 Linear Least Squares

For the simplest scenario, where the prediction functions $F_i(\mathbf{x})$ are linear, the predictions are given as linear equations following

$$f_i(\mathbf{x}) = \mathbf{F}_i \mathbf{x}, \tag{2.32}$$

where $\mathbf{F}_i \in \mathbb{R}^{M \times D}$ is a matrix of constant scalars. The residual is then given as

$$\mathbf{e}_i(\mathbf{x}) = \mathbf{z}_i - \mathbf{F}_i\mathbf{x}. \tag{2.33}$$

We can expand Equation (2.29) with the new residuals following

$$F(\mathbf{x}) = \sum_{i=1}^{N} e_i(\mathbf{x}) = \sum_{i=1}^{N} (\mathbf{z}_i - \mathbf{F}_i\mathbf{x})^\top \mathbf{W}_i (\mathbf{z}_i - \mathbf{F}_i\mathbf{x}) \tag{2.34}$$

$$= \sum_{i=1}^{N} \left(\mathbf{z}_i^\top \mathbf{W}_i \mathbf{z}_i - 2\mathbf{z}_i^\top \mathbf{W}_i \mathbf{F}_i \mathbf{x} + \mathbf{x}^\top \mathbf{F}_i^\top \mathbf{W}_i \mathbf{F}_i \mathbf{x} \right). \tag{2.35}$$

Grouping the terms in Equation (2.35) we can simplify the equation as

$$F(\mathbf{x}) = \mathbf{c} - 2\mathbf{b}^\top \mathbf{x} + \mathbf{x}^\top \mathbf{H} \mathbf{x}, \tag{2.36}$$

where

$$\mathbf{c} = \sum_i \mathbf{z}_i^\top \mathbf{W}_i \mathbf{z}_i, \tag{2.37}$$

$$\mathbf{b} = \sum_i \mathbf{F}_i^\top \mathbf{W}_i \mathbf{z}_i, \tag{2.38}$$

$$\mathbf{H} = \sum_i \mathbf{F}_i^\top \mathbf{W}_i \mathbf{F}_i, \tag{2.39}$$

In linear least squares, the objective function $F(\mathbf{x})$ is a convex quadratic function of $\mathbf{x}$, i.e., a global minimum exists. We obtain the optimal state $\mathbf{x}^*$ that achieves the global minimum of $F(\mathbf{x})$ at the location of zero-gradient, as in Equation (2.31), following

$$\frac{\partial F}{\partial \mathbf{x}} = -2\mathbf{b} + 2\mathbf{H}\mathbf{x}^* = 0. \tag{2.40}$$

Rearranging the terms, we obtain the final system of linear equations as

$$\mathbf{H}\mathbf{x}^* = \mathbf{b}. \tag{2.41}$$

Assuming the problem is well constrained and the matrix $\mathbf{H}$ is positive definite, the solution to the linear least squares problem $\mathbf{x}^*$ is given by

$$\mathbf{x}^* = \mathbf{H}^{-1}\mathbf{b}. \tag{2.42}$$

2.3.3 Nonlinear Least Squares

Linear prediction functions lead to a globally optimal solution of the least squares problem. However, in the field of robotics, it is common to find prediction functions $f_i(\mathbf{x})$ that are nonlinear. In such cases, the derivative in Equation (2.31)

is not straightforward to compute. We employ numerical methods to achieve a locally optimal solution by iteratively reducing the objective $F(\mathbf{x})$.

Given an initial estimate $\mathbf{x}_0$, for the k^{th} iteration we find an incremental value $\Delta\mathbf{x}_k$ such that the value of the objective function $F(\mathbf{x})$ is reduced. The new state $\mathbf{x}_{k+1}$ is given as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}_k. \tag{2.43}$$

We repeat the process until the increment $\Delta\mathbf{x}_k$ is smaller than a given threshold, which indicates that we have reached a local minimum. It is important to note how the problem has changed from estimating the globally optimal solution of the least squares problem, to finding an increment $\Delta\mathbf{x}_k$ that reduces the optimization function. Common techniques to solve the problem include gradient descent, Gauss-Newton, and Levenberg-Marquardt.

2.3.3.1 Gradient Descent

The gradient descent method estimates the increment $\Delta\mathbf{x}_k$ iteratively by following the negative gradient of the objective function. The new state update $\Delta\mathbf{x}_k$ is given by the negative gradient of the objective function $\nabla F(\mathbf{x}_k)$ following

$$\Delta\mathbf{x}_k = -\alpha\nabla F(\mathbf{x}_k), \tag{2.44}$$

where α is a parameter indicating the step size. Bigger α values will cause bigger jumps in the optimization, and smaller α values will lead to smaller jumps with slower convergence. The gradient $\nabla F(\mathbf{x}_k)$ is given as

$$\nabla F(\mathbf{x}_k) = \frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} = 2\sum_{i=1}^{N} \mathbf{J}_i^\top \mathbf{W}_i \mathbf{e}_i(\mathbf{x}), \tag{2.45}$$

where $\mathbf{J}_i = \frac{\partial \mathbf{e}_i}{\partial \mathbf{x}}$ is the jacobian of the residual function $\mathbf{e}_i(\mathbf{x})$.

2.3.3.2 Gauss-Newton Method

The Gauss-Newton method estimates the increment $\Delta\mathbf{x}_k$ by linearizing the error function around a point $\mathbf{x}_k$. To find $\Delta\mathbf{x}_k$ in the current iteration, if $f_i(\mathbf{x}_k)$ is nonlinear, we can approximate the error functions around an initial guess $\mathbf{x}_k$ using the Taylor expansion

$$\mathbf{e}_i(\mathbf{x}_k + \Delta\mathbf{x}_k) \approx \mathbf{e}_i(\mathbf{x}_k) + \mathbf{J}_i(\mathbf{x}_k)\Delta\mathbf{x}_k = \mathbf{e}_i + \mathbf{J}_i\Delta\mathbf{x}_k, \tag{2.46}$$

where $\mathbf{J}_i = \frac{\partial \mathbf{e}_i}{\partial x}\Big|_{\mathbf{x}_k}$ is the Jacobian of $\mathbf{e}_i$ with respect to $\mathbf{x}_k$. We denote $\mathbf{e}_i(\mathbf{x}_k) = \mathbf{e}_i$ and $\mathbf{J}_i(\mathbf{x}_k) = \mathbf{J}_i$ for better readability.

The new squared error term $e_i(\mathbf{x}_k)$ is given as

$$e_i(\mathbf{x}_k + \Delta\mathbf{x}_k) = (\mathbf{e}_i + \mathbf{J}_i\Delta\mathbf{x}_k)^\top \mathbf{W}_i(\mathbf{e}_i + \mathbf{J}_i\Delta\mathbf{x}_k). \tag{2.47}$$

We can expand Equation (2.29) with the new residuals following

$$F(\mathbf{x}_k + \Delta\mathbf{x}_k) = \sum_{i=1}^{N} e_i(\mathbf{x}) = \sum_{i=1}^{N} (\mathbf{e}_i + \mathbf{J}_i\Delta\mathbf{x}_k)^\top \mathbf{W}_i(\mathbf{e}_i + \mathbf{J}_i\Delta\mathbf{x}_k) \tag{2.48}$$

$$= \sum_{i=1}^{N} \left(\mathbf{e}_i^\top \mathbf{W}_i\mathbf{e}_i + 2\mathbf{e}_i^\top \mathbf{W}_i\mathbf{J}_i\Delta\mathbf{x}_k + \Delta\mathbf{x}_k^\top \mathbf{J}_i^\top \mathbf{W}_i\mathbf{J}_i\Delta\mathbf{x}_k\right). \tag{2.49}$$

$$= \sum_{i=1}^{N} \left(\mathbf{e}_i^\top \mathbf{W}_i\mathbf{e}_i\right) + 2\sum_{i=1}^{N} \left(\mathbf{e}_i^\top \mathbf{W}_i\mathbf{J}_i\right)\Delta\mathbf{x}_k + \Delta\mathbf{x}_k^\top \sum_{i=1}^{N} \left(\mathbf{J}_i^\top \mathbf{W}_i\mathbf{J}_i\right)\Delta\mathbf{x}_k. \tag{2.50}$$

Grouping the terms in Equation (2.50) we can simplify the equation as

$$F(\mathbf{x}_k + \Delta\mathbf{x}_k) = \mathbf{c} - 2\mathbf{b}^\top\Delta\mathbf{x}_k + \Delta\mathbf{x}_k^\top\mathbf{H}\Delta\mathbf{x}_k, \tag{2.51}$$

where

$$\mathbf{c} = \sum_i \mathbf{e}_i^\top \mathbf{W}_i\mathbf{e}_i, \tag{2.52}$$

$$\mathbf{b} = -\sum_i \mathbf{J}_i^\top \mathbf{W}_i\mathbf{e}_i, \tag{2.53}$$

$$\mathbf{H} = \sum_i \mathbf{J}_i^\top \mathbf{W}_i\mathbf{J}_i. \tag{2.54}$$

Note that the Hessian $\mathbf{H}$ approximates the second derivative of the cost function, which determines the curvature of the error surface.

To find the optimal increment $\Delta\mathbf{x}^*$, we find the point with zero gradient of the linearized objective function following

$$\frac{\partial F(\mathbf{x}_k + \Delta\mathbf{x}_k^*)}{\partial\Delta\mathbf{x}_k} = -2\mathbf{b} + 2\mathbf{H}\Delta\mathbf{x}_k^* = 0. \tag{2.55}$$

Rearranging the terms, we obtain the final system of linear equations as

$$\mathbf{H}\Delta\mathbf{x}_k^* = \mathbf{b}. \tag{2.56}$$

We can solve this system using Cholesky decomposition or QR factorization under the assumption that the problem is well constrained and the Hessian $\mathbf{H}$ is positive definite.

2.3.4 Levenberg-Marquardt Method

The Levenberg-Marquardt method is intended to improve stability and convergence of the Gauss-Newton technique. The new system of linear equations introduces a damping factor λ. This factor is applied by augmenting the diagonal of

the Hessian with a scaled identity matrix $\mathbf{I}$, particularly when $\mathbf{H}$ is poorly conditioned or near-singular. Following Equation (2.41), the new system of linear equations is defined as

$$(\mathbf{H} + \lambda \mathbf{I})\Delta \mathbf{x}_k^* = \mathbf{b}. \tag{2.57}$$

For large values of λ, the equation approximates gradient descent. The steps are small and slow, ensuring stability in regions of high nonlinearity or poor initial guesses, similar to gradient descent. It follows that

$$\Delta \mathbf{x}_k^* \approx \frac{1}{\lambda}\mathbf{b} \approx -\frac{1}{\lambda}\nabla F(\mathbf{x}_k). \tag{2.58}$$

For small values of λ, the equation is similar to the Gauss-Newton method, which may lead to fast convergence but instability if $\mathbf{H}$ is ill-conditioned, following

$$\mathbf{H}\Delta \mathbf{x}_k^* \approx \mathbf{b}. \tag{2.59}$$

2.4 Lie Theory

In the robotics domain, we frequently encounter geometric state estimation problems that involve translations and rotations within the 3D space. Although the rotations can be represented by Euler angles, this representation can lead to singularities, commonly known as gimbal lock, which can result in failures during pose optimization and scan alignment. To avoid encountering singularities and achieving smooth pose updates, as described by Sola et al. [164], a rotation R in the 3D space is typically represented as part of a smooth and continuous manifold known as the special orthogonal group $SO(3)$, which fulfills the properties

$$SO(3) = \left\{ R \in \mathbb{R}^{3\times 3} \;\middle|\; R^\top R = \mathbf{I},\ \det(R) = 1 \right\}. \tag{2.60}$$

Since the movement in 3D usually involves a translation, the transformation that represents a rotation R and translation $\mathbf{t}$ operations is defined as part of a smooth and continuous manifold known as the special Euclidean group $SE(3)$, defined as

$$SE(3) = \left\{ \mathsf{T} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \;\middle|\; R \in SO(3), \quad \mathbf{t} \in \mathbb{R}^3 \right\}. \tag{2.61}$$

The special orthogonal group $SO(3)$ and the special Euclidean group $SE(3)$, are both Lie groups, since they fulfill the group properties that (i) multiplying two matrices gives another valid matrix belonging to the group, (ii) matrix multiplication is associative, (iii) the identity matrix acts as a neutral element, and (iv) every matrix has an inverse within the group. Nevertheless, $SO(3)$ and $SE(3)$ represent curved nonlinear manifolds. To optimize objective functions

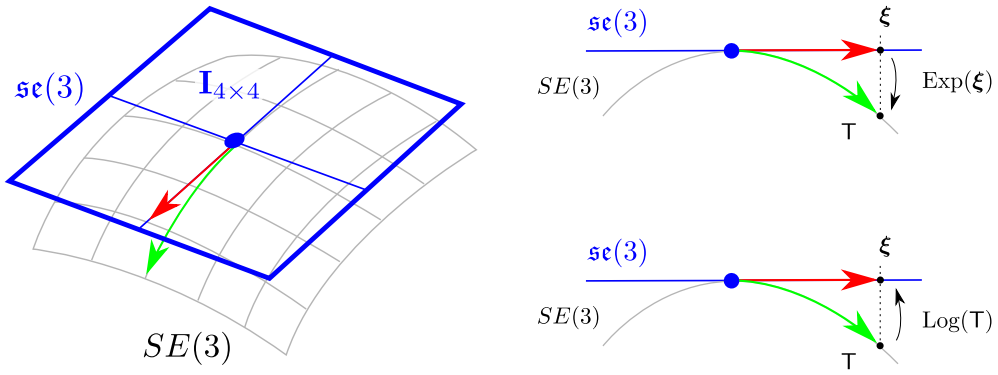


Figure 2.13: Mapping between the $SE(3)$ manifold and the $\mathfrak{se}(3)$ Lie algebra tangent space at the identity $\mathbf{I} \in \mathbb{R}^{4\times4}$. It involves the logarithm and exponential functions $\mathrm{Log}(\cdot)$ and $\mathrm{Exp}(\cdot)$.

that involve $SO(3)$ and $SE(3)$ exploiting least squares techniques, as described in Section 2.3.3, it is necessary to linearize the function around a point. This linearization of the Lie group at the identity is known as the Lie algebra.

The Lie algebra of the special orthogonal group, $\mathfrak{so}(3)$, is the tangent plane at the identity of the $SO(3)$ manifold. It is parametrized by the axis-angle representation of rotations around the x,y, and z coordinate axes $\boldsymbol{\phi} \in \mathbb{R}^3$. Furthermore, to map the vector $\boldsymbol{\phi}$ to the matrix form, we use the skew-symmetric operator $^\wedge$. Formally, the Lie group $\mathfrak{so}(3)$ is defined as

$$\mathfrak{so}(3) = \left\{ \boldsymbol{\phi} \in \mathbb{R}^3 \ , \ \boldsymbol{\phi}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \right\}. \tag{2.62}$$

The function that maps from the group manifold $SO(3)$ to the tangent plane of the Lie algebra $\mathfrak{so}(3)$ such that $R \in \mathbb{R}^{3\times3} \rightarrow \boldsymbol{\phi} \in \mathbb{R}^3$ is known as the logarithmic mapping. It is given as

$$\mathrm{Log}(R) \ : \ SO(3) \rightarrow \mathfrak{so}(3). \tag{2.63}$$

The inverse, mapping from the Lie algebra tangent plane to the group manifold, such that $\boldsymbol{\phi} \in \mathbb{R}^3 \rightarrow R \in \mathbb{R}^{3\times3}$ is given as the exponential mapping following

$$\mathrm{Exp}(\boldsymbol{\phi}) = \exp(\boldsymbol{\phi}^\wedge) \ : \ \mathfrak{so}(3) \rightarrow SO(3). \tag{2.64}$$

Similarly, the Lie algebra of the special Euclidean group, $\mathfrak{se}(3)$, is the tangent plane at the identity of the $SE(3)$ manifold. It is parametrized with $\boldsymbol{\xi} \in \mathbb{R}^6$, which combines a translation component $\boldsymbol{\rho}$ with the rotation represented in the axis-angle representation $\boldsymbol{\phi}^\wedge$ following

$$\mathfrak{se}(3) = \left\{ \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix} \in \mathbb{R}^6 \ , \ \boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \ \middle| \ \boldsymbol{\phi}^\wedge \in \mathfrak{so}(3), \ \boldsymbol{\rho} \in \mathbb{R}^3 \right\}. \tag{2.65}$$

The function that maps from the group manifold $SE(3)$ to the tangent plane of the Lie algebra $\mathfrak{se}(3)$ such that $\mathsf{T} \in \mathbb{R}^{4\times4} \rightarrow \boldsymbol{\xi} \in \mathbb{R}^6$ is known as the logarithmic mapping. It is given as

$$\text{Log}(\mathsf{T}) \ : \ SE(3) \rightarrow \mathfrak{se}(3). \tag{2.66}$$

The inverse, mapping from the Lie algebra tangent plane to the group manifold, such that $\boldsymbol{\xi} \in \mathbb{R}^6 \rightarrow \mathsf{T} \in \mathbb{R}^{4\times4}$, is given as the exponential mapping following

$$\text{Exp}(\boldsymbol{\xi}) = \exp(\boldsymbol{\xi}^\wedge) \ : \ \mathfrak{se}(3) \rightarrow SE(3). \tag{2.67}$$

This mapping between the $SE(3)$ manifold and the $\mathfrak{se}(3)$ tangent space is illustrated in Figure 2.13, which is inspired by Engel et al. [45] and Sola et al. [164].

In summary, Lie theory provides a mathematical framework that allows problems in robotics to handle transformations involving rotations and translations more efficiently. In practice, this is helpful for tasks such as scan registration, enabling smooth and accurate pose estimation and motion modelling.

2.5 Simultaneous Localization and Mapping

Knowing the pose of an autonomous vehicle is crucial for motion planning and navigation. Generally, GNSS systems can be used to determine the location of a robot. However, satellite visibility is not available in indoor scenarios such as tunnels and parking lots. Moreover, GNSS accuracy degrades substantially in environments with tall skyscrapers. Simultaneous localization and mapping (SLAM) involves estimating the pose of a robot while simultaneously building a map of the environment. SLAM relies on local sensing, without the need for global positioning techniques. While online SLAM strategies estimate the pose of the robot at the current time, the full SLAM approach estimates the entire trajectory followed by the robot and constructs a map of the environment. It can be defined as described by Thrun et al. [168].

For a time interval T, given the robot's controls $\mathbf{u}_{1:T} = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3..., \mathbf{u}_T\}$ and measurements $\mathbf{z}_{1:T} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3..., \mathbf{z}_T\}$ of arbitrary dimensions, the goal of SLAM is to build the map of the environment $\mathcal{M}$ and estimate the path followed by the robot $\mathbf{x}_{0:T} = \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2...\mathbf{x}_T \in \mathbb{R}^6$. This can be formulated probabilistically as

$$p(\mathbf{x}_{0:T}, \mathcal{M} \mid \mathbf{z}_{1:T}, \mathbf{u}_{1:T}). \tag{2.68}$$

We provide insights into how to estimate Equation (2.68) employing factor graphs in Section 2.5.4.1.

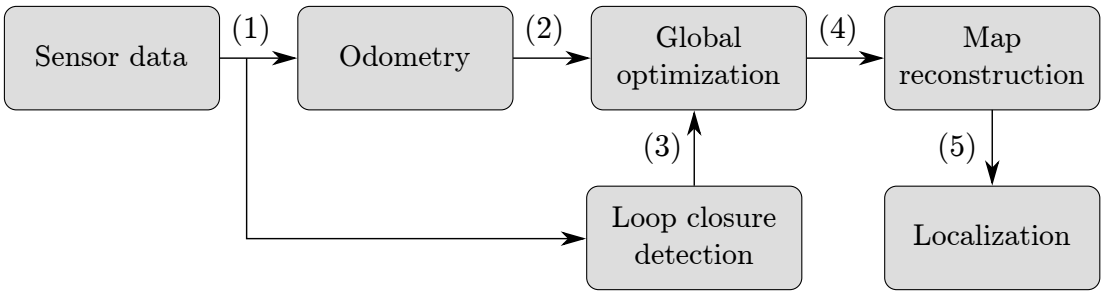


Figure 2.14: Common components of a SLAM system in a robot. The sensor data (1) is processed by the odometry module to estimate the relative motion over time, and by the loop closure detection module to identify revisited locations. The odometry (2) and loop closure (3) constraints are jointly optimized within the global map. The map reconstruction module uses the optimized poses (4) to enable multi-session mapping and map maintenance to obtain a globally consistent map (5), which is leveraged during map-based localization.

2.5.1 SLAM Systems

From a practical perspective, the solution to the SLAM problem can be divided into smaller components within a pipeline, displayed in Figure 2.14. The sensor data corresponds to real-world measurements coming, for example, from a camera, LiDAR, or radar. In this work, we exploit automotive radars described in Section 2.1 as our main source of information. The sensor data is used by the odometry estimation to obtain relative pose estimates of the robot over time. However, simply estimating the relative motion of the robot leads to errors accumulating over time. We correct this drift by identifying previously visited places using a loop closure detection module and passing the information to a global optimization procedure. Finally, a map reconstruction module maintains the map over longer periods of time. This module may also enable multi-session SLAM by merging maps from multiple sessions and identifying differences between them. The constructed map can be further leveraged for accurate map-based localization, exploiting prior information from previously recorded maps.

2.5.2 Odometry

The goal of odometry is to estimate the relative pose of the robot over time, solely relying on sensor data. Depending on the sensor, odometry can be categorized into LiDAR odometry, visual odometry, or radar odometry. Each of these approaches has its advantages and trade-offs. As explained in Section 2.1, automotive radars are resilient to adverse weather but are sparse and noisy. In this work, we focus on one particular technique used for odometry called scan matching, where the transformation that aligns two consecutive radar scan measurements also represents the change in pose between the two measurements. The most common approach for scan matching is based on the iterative closest point

algorithm, where the transformation between two frames is obtained by iteratively minimizing the distance between corresponding points. In the following, we describe its formulation as a least-squares problem, as it plays a crucial role in the development of our radar SLAM system.

2.5.2.1 Iterative Closest Point Using Least Squares

The iterative closest point (ICP) algorithm [16] aims to obtain a transformation that minimizes a distance metric between two point clouds. Based on the objective function, variations of this algorithm include point-to-point ICP [172], point-to-plane ICP [124], generalized ICP [159], and combinations between them [113]. Later, in Chapter 4, we will observe the benefits of employing point-to-point ICP on radar scans and how augmenting it with radar Doppler velocities enhances pose estimation accuracy.

The aim of ICP is to estimate the transformation $\mathsf{T}^* \in SE(3)$ that minimizes the distance between the points of a source point set $\mathbf{p} \in \mathcal{P}$, and a the points in a target point set $\mathbf{q} \in \mathcal{Q}$ such that

$$\mathsf{T}^* = \underset{\mathsf{T}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathsf{T}\mathbf{p} - \mathbf{q}\| \,. \tag{2.69}$$

The problem defined in Equation (2.69) can be solved iteratively. To achieve this, the set of correspondences is obtained at every k^{th} iteration by finding the closest point correspondences between the target scan and the source scan transformed with the pose estimate at the previous iteration T_{k-1} according to

$$\mathcal{C} = \{(\mathbf{p}, \mathbf{q})^* \,|\, \mathbf{p} \in \mathcal{P}, \mathbf{q}^* = \underset{\mathbf{q} \in \mathcal{Q}}{\operatorname{argmin}} \|\mathsf{T}_{k-1}\mathbf{p} - \mathbf{q}\|\}. \tag{2.70}$$

Subsequently, the point-to-point residual $\mathbf{e}_i(\mathsf{T}) \in \mathbb{R}^3$ between two matches is given as

$$\mathbf{e}_i(\mathsf{T}_k) = \mathsf{T}_k\mathbf{p}_i - \mathbf{q}_i. \tag{2.71}$$

Note that Equation (2.71) has the same form as the least squares residual from Equation (2.27), where $\mathbf{q}_i$ is the measurement and $\mathsf{T}_k\mathbf{p}_i$ is the prediction. We can formulate the objective function for the least squares problem following Equation (2.29) as

$$F(\mathsf{T}_k) = \sum_{i=1}^{N} e_i(\mathsf{T}_k), \tag{2.72}$$

where

$$e_i(\mathsf{T}_k) = \mathbf{e}_i(\mathsf{T}_k)^\top \mathbf{W}_i \mathbf{e}_i(\mathsf{T}_k). \tag{2.73}$$

If T_k was linear, we could directly compute the solution to the least squares problem following Section 2.3.2. However, as it contains a rotation, it belongs to

the non-Euclidean manifold of $SE(3)$. We use the Lie algebra parametrization, which facilitates smooth linearization of rotation and translation.

Instead of directly estimating the incremental update $\Delta\mathsf{T}_k \in SE(3)$, we express the linearization of the error $\mathbf{e}_i(\mathsf{T}_k)$ in the tangent space $\mathfrak{se}(3)$ with an exponential mapping $\text{Exp}(\cdot) : \boldsymbol{\xi}_k \in \mathfrak{se}(3) \rightarrow \mathsf{T}_k \in SE(3)$ as explained in Section 2.4, according to

$$\mathbf{e}_i(\boldsymbol{\xi}_k) = \text{Exp}(\boldsymbol{\xi}_k)\mathbf{p}_i - \mathbf{q}_i. \tag{2.74}$$

.

The linearization of $\mathbf{e}_i(\boldsymbol{\xi}_k)$ using the Taylor expansion is now given as

$$\mathbf{e}_i(\boldsymbol{\xi}_k + \Delta\boldsymbol{\xi}_k) \approx \mathbf{e}_i(\boldsymbol{\xi}_k) + \mathbf{J}_i(\boldsymbol{\xi}_k)\Delta\boldsymbol{\xi}_k, \tag{2.75}$$

where the Jacobian $\mathbf{J}_i(\boldsymbol{\xi}_k) \in \mathbb{R}^{3\times 6}$ is given by

$$\mathbf{J}_i(\boldsymbol{\xi}_k) = \left.\frac{\partial \mathbf{e}_i(\boldsymbol{\xi}_k)}{\partial \boldsymbol{\xi}_k}\right|_{\boldsymbol{\xi}_k} = \begin{bmatrix}\mathbf{I}_{3\times 3} & -(\mathsf{T}\mathbf{p}_i)^\wedge\end{bmatrix}. \tag{2.76}$$

The goal is now to estimate the optimal update $\Delta\boldsymbol{\xi}_k^* \in \mathfrak{se}(3)$. The final system of equations has the same form as Equation (2.56) and is given as

$$\mathbf{H}\Delta\boldsymbol{\xi}_k^* = \mathbf{b}. \tag{2.77}$$

The optimal incremental update $\Delta\mathsf{T}_k^*$ can be obtained with the exponential mapping as

$$\Delta\mathsf{T}_k^* = \text{Exp}(\Delta\boldsymbol{\xi}_k^*). \tag{2.78}$$

The source point cloud is then transformed by $\Delta\mathsf{T}_k^*$ using left increments according to

$$\mathsf{T}_{k+1} = \Delta\mathsf{T}_k^*\,\mathsf{T}_k, \tag{2.79}$$

and a new iteration begins. This process is repeated until the increment $\Delta\boldsymbol{\xi}_k^*$ is smaller than a certain threshold, indicating that a local minimum has been achieved, and a locally optimal transformation T has been computed.

2.5.3 Loop Closure Detection

Solely relying on odometry for pose estimation will lead to drift over long periods of time. It is essential to correct this drift when estimating the trajectory of a robot and creating a map of the environment. The goal of loop closure detection is to identify locations that have already been visited to correct the accumulated drift. The loop detection module relies on feature extraction and matching techniques to compare the current sensor measurements with previously recorded data. This is typically achieved by leveraging place recognition, where

previously visited locations are identified within a map database. We can categorize existing place recognition approaches into heuristic-based or learning-based. Heuristic-based methods encode the sensor measurements into a spatial representation based on features like maximum height of the measurements or intensity [94, 179]. However, these approaches will underperform in locations that are difficult to differentiate based on the basic criteria. Learning-based approaches learn patterns in the environment useful to describe each location [101, 185]. More existing loop closure detection approaches will be introduced in Section 3.3.

2.5.4 Global Optimization

Once revisited locations have been identified with the loop closure detection module, this information can be exploited to correct the accumulated drift for global pose estimation. During operation, we construct a factor graph to model the SLAM problem. This graph integrates the relative motion estimates from the odometry module and the loop closure information that corrects accumulated drift. In the following section, we provide more details on how to leverage factor graphs to conveniently formulate SLAM as a least-squares problem.

2.5.4.1 Factor Graph Optimization Using Least Squares

Factor graphs are bipartite graphs that can be used in robotics to model complex estimation problems. They consist of factors representing probabilistic constraints, and variables representing the unknowns in the estimation problem. To estimate the entire vehicle trajectory, incremental approaches like iSAM2 [87] and g2o [108] build upon the probabilistic formulation in Equation (2.68). They construct and maintain a sparse factor-graph representation of the posterior such that new measurements only affect the portions of the graph corresponding to their poses, leading to efficient optimization of the full trajectory. Furthermore, factor graphs explicitly represent the relationships between variables and measurements, leading to an intuitive formulation of the problem.

We can formulate the SLAM problem introduced in Equation (2.68) as the optimization of a factor graph following the derivations by Dellaert et al. [39] and Montemerlo et al. [132]. The joint probability model is given as

$$p(\mathbf{x}_{0:T}, \mathcal{M} \mid \mathbf{z}_{1:T}, \mathbf{u}_{1:T}) = p(\mathbf{x}_0) \prod_{t=1}^{T} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) \prod_{t=1}^{T} p(\mathbf{z}_t \mid \mathbf{x}_t, \mathcal{M}), \tag{2.80}$$

where $p(\mathbf{x}_0)$ denotes the initial state, the motion model $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t)$ represents the relative motion of the robot between two consecutive timestamps given an input $\mathbf{u}_t$, and the measurement model $p(\mathbf{z}_t \mid \mathbf{x}_t, \mathcal{M})$ relates the independent sensor measurements $\mathbf{z}_t$ to the robot pose at time t. In practice, the input $\mathbf{u}_t$ is often

associated with a motion estimate, such as the computed transformation between two poses from ICP in Section 2.5.2.1.

To better understand how the motion model and measurement model represent the evolution of the system over time, we describe the relationship between the probabilistic models and the state space equations. Assuming that the motion model $g_t(\mathbf{x}_{t-1}, \mathbf{u}_t)$ and the measurement model $h_t(\mathbf{x}_t, \mathcal{M})$ are affected by Gaussian noise $w_t \sim \mathcal{N}(0, \Sigma_w)$ and $v_t \sim \mathcal{N}(0, \Sigma_v)$, respectively, the probabilistic models can be represented with their corresponding the state-space equations as

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) \iff \mathbf{x}_t = g_t(\mathbf{x}_{t-1}, \mathbf{u}_t) + w_t, \tag{2.81}$$

$$p(\mathbf{z}_t \mid \mathbf{x}_t, \mathcal{M}) \iff \mathbf{z}_t = h_t(\mathbf{x}_t, \mathcal{M}) + v_t. \tag{2.82}$$

We can now represent the motion factor $f_t^{\text{mt.}}(\mathbf{x}_{t-1}, \mathbf{x}_t; \mathbf{u}_t)$ and measurement factor $f_t^{\text{ms.}}(\mathbf{x}_t, \mathcal{M}; \mathbf{z}_t)$ following

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t) \propto \exp\left(-\frac{1}{2}\|f_t^{\text{mt.}}(\mathbf{x}_{t-1}, \mathbf{x}_t; \mathbf{u}_t)\|_{\Sigma_w}\right), \tag{2.83}$$

$$p(\mathbf{z}_t \mid \mathbf{x}_t, \mathcal{M}) \propto \exp\left(-\frac{1}{2}\|f_t^{\text{ms.}}(\mathbf{x}_t, \mathcal{M}; \mathbf{z}_t)\|_{\Sigma_v}\right), \tag{2.84}$$

where

$$f_t^{\text{mt.}}(\mathbf{x}_{t-1}, \mathbf{x}_t; \mathbf{u}_t) = \mathbf{x}_t - g_t(\mathbf{x}_{t-1}, \mathbf{u}_t), \tag{2.85}$$

$$f_t^{\text{ms.}}(\mathbf{x}_t, \mathcal{M}; \mathbf{z}_t) = \mathbf{z}_t - h_t(\mathbf{x}_t, \mathcal{M}). \tag{2.86}$$

The solution to the SLAM problem from Equation (2.80) is given as the maximum a posteriori estimate of the joint probability distribution. This involves finding the optimal trajectory of the robot $\mathbf{x}^* = \mathbf{x}_{0:T}^*$ and the map $\mathcal{M}^*$ given the observed measurements $\mathbf{z}_{1:T}$ and control inputs $\mathbf{u}_{1:T}$ following

$$(\mathbf{x}^*, \mathcal{M}^*) = \underset{\mathbf{x}_{0:T}, \mathcal{M}}{\operatorname{argmax}}\ p(\mathbf{x}_{0:T}, \mathcal{M} \mid \mathbf{z}_{1:T}, \mathbf{u}_{1:T}). \tag{2.87}$$

However, optimizing joint probabilities in the range $[0, 1]$ requires multiplying many small terms, which quickly leads to numerical underflow. Instead, we can consider that the logarithmic function is monotonically increasing. Therefore, optimizing Equation (2.87) is equal to computing the maximum likelihood estimate according to

$$(\mathbf{x}^*, \mathcal{M}^*) = \underset{\mathbf{x}_{0:T}, \mathcal{M}}{\operatorname{argmax}}\ \log p(\mathbf{x}_{0:T}, \mathcal{M} \mid \mathbf{z}_{1:T}, \mathbf{u}_{1:T}), \tag{2.88}$$

$$= \underset{\mathbf{x}_{0:T}, \mathcal{M}}{\operatorname{argmin}}\ -\log p(\mathbf{x}_{0:T}, \mathcal{M} \mid \mathbf{z}_{1:T}, \mathbf{u}_{1:T}). \tag{2.89}$$

By performing the optimization in Equation (2.89), which combines the joint probability distribution of Equation (2.80), the motion model derived in Equation (2.83), and the measurement model from Equation (2.84), the nonlinear

least-squares problem is formulated as the minimization of sums, following

$$(\mathbf{x}^*, \mathcal{M}^*) = \underset{\mathbf{x}_{0:T}, \mathcal{M}}{\operatorname{argmin}} \Big[\sum_{t=1}^{T} \big\| f_t^{\text{mt.}}(\mathbf{x}_{t-1}, \mathbf{x}_t; \mathbf{u}_t) \big\|_{\Sigma_w} + \sum_{t=1}^{T} \| f_t^{\text{ms.}}(\mathbf{x}_t, \mathcal{M}; \mathbf{z}_t) \|_{\Sigma_v} \Big]. \quad (2.90)$$

We can reorganize the terms to formulate the SLAM problem as the least-squares problem in Equation (2.29), resulting in

$$F(\mathbf{x}) = \sum_{t=1}^{T} e_t^{\text{mt.}}(\mathbf{x}) + \sum_{t=1}^{T} e_t^{\text{ms.}}(\mathbf{x}), \quad (2.91)$$

where

$$e_t^{\text{mt.}}(\mathbf{x}) = f_t^{\text{mt.}}(\mathbf{x}_{t-1}, \mathbf{x}_t; \mathbf{u}_t)^\top \, \Sigma_w \, f_t^{\text{mt.}}(\mathbf{x}_{t-1}, \mathbf{x}_t; \mathbf{u}_t), \quad (2.92)$$

$$e_t^{\text{ms.}}(\mathbf{x}) = f_t^{\text{ms.}}(\mathbf{x}_t, \mathcal{M}; \mathbf{z}_t)^\top \, \Sigma_v \, f_t^{\text{ms.}}(\mathbf{x}_t, \mathcal{M}; \mathbf{z}_t). \quad (2.93)$$

Sometimes, however, a factor graph may only contain pose factors that are associated to on an arbitrary number of states $\mathbf{x}_{t_1}, \mathbf{x}_{t_2} ..., \mathbf{x}_{t_M} \in \mathbb{R}^6$, which are jointly constrained by a single observation variable o_t. This factor is independent of the map, and it can be represented as $f_t(\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, ..., \mathbf{x}_{t_M}; o_t)$. The resulting factor graph is typically called a pose graph, since it depends only on the vehicle's poses. It is represented as

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{t} \| f_t(\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_M}; o_t) \|_{\Sigma_w}. \quad (2.94)$$

The optimization problem can be solved with the gradient descent, Gauss-Newton, or Levenberg-Marquardt algorithms presented in Section 2.3.3.

By leveraging this least squares formulation of the SLAM problem, we perform global optimization of the robot trajectory and correct the drift accumulated during odometry, as it will be presented later in Chapter 7.

2.5.5 SLAM Evaluation Metrics

Evaluating the performance of SLAM systems is critical before deployment in real-world driving scenarios. This is done by comparing the estimated trajectory with a ground truth, commonly obtained with high-accuracy sensing techniques including centimeter-level real-time kinematic GPS systems [97] or a combination of GNSS, inertial sensors, and high-resolution LiDAR maps [21]. Note, however, that these high-accuracy positioning techniques cannot be easily integrated within robots or end-user vehicles, typically due to price and size constraints. Therefore, the goal is to develop a SLAM system that achieves the closest performance to these reference trajectories, only relying on onboard sensing.

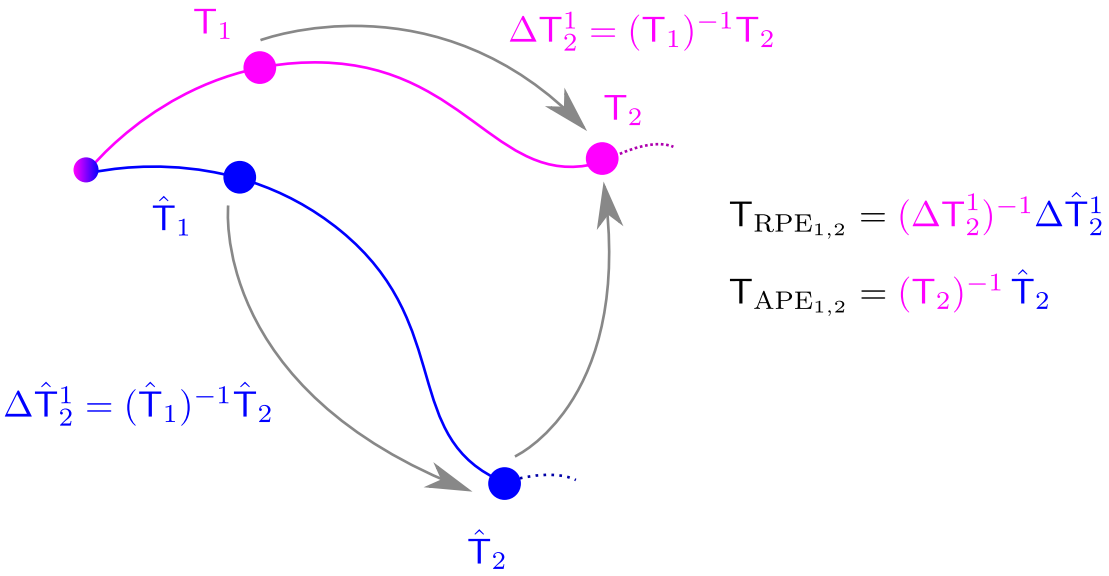


Figure 2.15: Illustration of the relative (RPE) and absolute (APE) pose errors in a trajectory. While the relative error serves as a way of evaluating the local consistency of the trajectory, the absolute errors provide insights into the global consistency of the estimated vehicle poses.

There are two main metrics that measure the accuracy of the estimated trajectory of a vehicle, the relative pose error (RPE) and the absolute pose error (APE), displayed in Figure 2.15. Since all the absolute poses described in this section are expressed with respect to the inertial frame, we omit the superscript $(\cdot)^I$ for better readability.

The relative pose error evaluates the performance of a SLAM system between consecutive measurements and measures the local consistency of the trajectory. A SLAM system with high RPE will accumulate error over time, leading to a high-drift trajectory. Meanwhile, if the RPE is low, the estimated trajectory is highly accurate, and low drift will accumulate over time. Hence, the RPE is mainly used to evaluate the odometry component of a SLAM system, which estimates the vehicle's relative motion over time.

The relative error between two consecutive poses $\mathsf{T}_{\mathrm{RPE}_{i-1,i}}$ is defined as the relative difference between pose changes in the ground truth, $\mathsf{T}_i, \mathsf{T}_{i-1} \in \mathrm{SE}(3)$, and the pose estimates, $\hat{\mathsf{T}}_i, \hat{\mathsf{T}}_{i-1} \in \mathrm{SE}(3)$ according to

$$\mathsf{T}_{\mathrm{RPE}_{i-1,i}} = (\Delta\mathsf{T}_i^{i-1})^{-1}\Delta\hat{\mathsf{T}}_i^{i-1} = (\mathsf{T}_{i-1}^{-1}\mathsf{T}_i)^{-1}(\hat{\mathsf{T}}_{i-1}^{-1}\hat{\mathsf{T}}_i). \tag{2.95}$$

Subsequently, the relative translation error t_{RPE} and the relative rotation error r_{RPE} for a trajectory of $N+1$ poses are defined as

$$t_{\text{RPE}} = \frac{1}{N}\sum_{i=1}^{N} \left\| \text{trans}(\mathsf{T}_{\text{RPE}_{i-1,i}}) \right\|, \tag{2.96}$$

$$r_{\text{RPE}} = \frac{1}{N}\sum_{i=1}^{N} \left\| \text{Log}(\text{rot}(\mathsf{T}_{\text{RPE}_{i-1,i}})) \right\|, \tag{2.97}$$

where the $\text{trans}(\cdot)$ and $\text{rot}(\cdot)$ operators extract the translation and rotation component from a homogeneous matrix, respectively.

Sometimes, however, some works provide the relative error KITTI metric [61] over multiple segments $l \in \mathcal{L}$ of fixed path lengths. The error associated to each pose $\mathsf{T}_{\text{KITTI}_{i,l}} \in SE(3)$ is given as

$$\mathsf{T}_{\text{KITTI}_{i,l}} = (\Delta\mathsf{T}^{i}_{i+l})^{-1}\Delta\hat{\mathsf{T}}^{i}_{i+l} = (\mathsf{T}_i^{-1}\mathsf{T}_{i+l})^{-1}(\hat{\mathsf{T}}_i^{-1}\hat{\mathsf{T}}_{i+l}). \tag{2.98}$$

The KITTI translation t_{KITTI} and rotation r_{KITTI} errors for a trajectory of N poses over $|\mathcal{L}|$ segments are computed as

$$t_{\text{KITTI}} = \frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}} \frac{1}{N-l}\sum_{i=1}^{N-l} \left\| \text{trans}(\mathsf{T}_{\text{KITTI}_{i,l}}) \right\|, \tag{2.99}$$

$$r_{\text{KITTI}} = \frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}} \frac{1}{N-l}\sum_{i=1}^{N-l} \left\| \text{Log}(\text{rot}(\mathsf{T}_{\text{KITTI}_{i,l}})) \right\|. \tag{2.100}$$

The absolute pose error evaluates the global accuracy of the predicted trajectory. The APE is commonly used to evaluate full SLAM systems that include loop closure with reduced odometry drift. While having an accurate odometry will lead to good local consistency, maps used for localization must also be globally consistent to be useful. The APE measures the absolute difference between each pose in the ground truth and the predicted trajectory following

$$\mathsf{T}_{\text{APE}_i} = (\mathsf{T}_i)^{-1}\hat{\mathsf{T}}_i = \mathsf{T}_i^{-1}\hat{\mathsf{T}}_i. \tag{2.101}$$

The absolute translation t_{APE} and rotation r_{APE} errors are computed as

$$t_{\text{APE}} = \frac{1}{N}\sum_{i=1}^{N} \left\| \text{trans}(\mathsf{T}_{\text{APE}_i}) \right\|, \tag{2.102}$$

$$r_{\text{APE}} = \frac{1}{N}\sum_{i=1}^{N} \left\| \text{Log}(\text{rot}(\mathsf{T}_{\text{APE}_i})) \right\|. \tag{2.103}$$

In sum, the pose evaluation metrics allow us to estimate the quality of the predicted trajectories. While relative pose metrics provide information about the local consistency of the trajectory, absolute errors measure the global consistency of the estimated path.

2.6 Summary

In this chapter, we have introduced the basic techniques underlying our work, which serve as the foundation of our proposed approaches for radar pose estimation and mapping. We began in Section 2.1 by discussing the key differences between automotive radars and other common sensors such as LiDARs and cameras. We outlined the working principles of radar sensors, highlighting their main challenges due to their high noise and sparsity. In addition, in Section 2.2 we provided a summary of relative body motion and how it can be used with automotive radar data, as it will be exploited for odometry in Chapter 4 during odometry. Then, in Section 2.3 and Section 2.4, we briefly introduced the least-squares problem formulation and Lie theory as key elements present throughout this thesis, including Chapter 4 during scan matching and in Chapter 7 for factor graph optimization. Finally, in Section 2.5 we presented the simultaneous localization and mapping problem and how it can be divided into sub-components, including sensor input, odometry, loop detection, global optimization, and map reconstruction. We also introduced the main metrics for measuring the quality of the SLAM trajectory estimates, serving as a benchmark for comparing the performance between different approaches.

Chapter 3

Related Work

The goal of our work is to develop an automotive radar-based system that is able to estimate the pose of a vehicle, build maps of the environment, maintain them, and localize within them, without relying on external GNSS information during operation. Existing odometry techniques leverage LiDAR and/or camera sensors to perform relative pose estimation. Sensor-based SLAM approaches are able to additionally perform loop closure detection by recognizing previously visited locations and correcting accumulated drift. Some LiDAR approaches also remove non-persistent points from maps and perform localization within them.

LiDARs, however, have a downside. They are bulky, costly, not easy to integrate into consumer vehicles, and their performance degrades under adverse weather conditions. Similarly, cameras are not resilient to environmental conditions, including low light, heavy rain, fog, and snowfall. Radars, on the contrary, are easy to pack, low-cost, and resilient to adverse weather and lighting conditions. However, the sparse and noisy nature of automotive radar scans presents challenges when adapting LiDAR techniques to the radar domain. In the first section of this chapter, we introduce state-of-the-art LiDAR and camera approaches and address their limitations when applied to radar sensors. We then introduce existing radar approaches and their main operating principles. Given the limited number of methods specifically designed for automotive radars, we also include related research from the spinning radar domain. Our discussion focuses on four key areas: methods that estimate the pose of the vehicle and construct a map of the environment; techniques that extract ground plane data as additional information for pose estimation; place recognition approaches suitable for loop closure detection; and long-term solutions for map maintenance and localization.

3.1 Pose Estimation and Mapping

Pose estimation and mapping address the problem of determining a vehicle's position and orientation relative to a reference frame and leveraging the sensor measurements to construct a map of the environment. The most popular approaches in the LiDAR domain are based on the iterative closest point algorithm (ICP) [16] and matching specific elements between two consecutive scans. These elements are usually pointwise correspondences, like KISS-ICP [172], plane correspondences, like CT-ICP [40] and MAD-ICP [51], or a combination of both, such as GenZ-ICP [113] and G-ICP [159]. Zhang et al. [202] also minimize the distance between corners and edges in the LiDAR point cloud, and Behley et al. [14] optimize surfels extracted from the LIDAR point cloud, aggregating them to build a feature-based environment representation.

Nevertheless, as highlighted by Tuna et al. [169], solely relying on LiDAR may degrade in scenarios where no meaningful features can be extracted, which can be addressed by introducing information from additional sensors like the odometers from Kinematic-ICP [66] or the inertial measurement unit (IMU) from LIO-SAM [161]. The high-frequency accelerometer and gyroscope readings can be integrated over time as proposed by Foster et al. [53] to provide an additional constraint during odometry. For example, Xu et al. [188] exploit this by combining LiDAR and IMU measurements with an iterated Kalman filter to estimate the pose of the robot. Others employ pose graphs [116, 148, 161] combining IMU and LiDAR odometry information that can be jointly optimized within a single graph. A major disadvantage of single-factor graph approaches like LIO-SAM [161] is that, in their implementation, they do not combine the IMU and LiDAR information within the global graph. Instead, they use the IMU as a source of high-frequency odometry but construct the backend graph containing only the poses from LiDAR scan registration. More recent approaches by Koide et al. [99, 100] propose to integrate a local submap and a global map, maintaining global consistency. Grisetti et al. [64] also propose a hierarchical optimization method that enables accurate odometry by combining multiple graphs.

However, LiDAR sensors are affected by adverse weather and are hard to pack within autonomous consumer vehicles. Moreover, due to the sparse and noisy properties of radar point clouds, applying LiDAR methods directly to radar scans reduces pose estimation performance [26]. While LiDAR scans are commonly dense and accurate, with hundreds of thousands of points per scan, radar point clouds are comparably sparser and noisier, typically containing only a few hundred points per scan. This can be problematic for point matching and odometry, resulting in existing LiDAR pose estimation methods being easily outperformed as demonstrated by Haggag et al. [68] and Kung et al. [109].

Radar odometry and mapping approaches leverage radar sensors to estimate the transformation between two sparse and noisy radar point clouds and store the measurements into a map. These may not necessarily require point correspondence matching between the scans, thanks to the radar's Doppler velocity information as proposed by Kellner et al. [89, 90] and others [13, 142, 184]. Many of the existing approaches are based on the 2D spinning radar and are usually either feature-based [2, 12, 18, 19, 20, 29, 30, 86, 118, 122], where key-points are extracted and matched across the output intensity images, similar to the constant false alarm rate algorithm described in Figure 2.10 on page 19, or signal processing based [13, 142, 184], where the car's ego-pose is estimated using the correlation operation between frames. In order to correct the odometry drift, full SLAM approaches with loop closure have also been proposed for spinning radars, proving successful in adverse weather scenarios [1, 70, 73, 176]. Nevertheless, similar to LiDARs, spinning radars are bulky and too expensive to be integrated in consumer vehicles. In contrast, automotive radar sensors are smaller and provide a sparse and noisy 3D point cloud that also contains the Doppler velocity and RCS information of each point. In the context of automotive radar odometry, Aldera et al. [4] and Ding et al. [42, 43] take advantage of combining radars with LiDARs for operation or training. Zhuo et al. [207] combine radars with cameras, and others combine automotive radars with IMUs to improve odometry accuracy. Specifically, Milli-RIO [7] leverages a combination of scan matching via normal distribution transform and IMU integration within an unscented Kalman filter procedure to perform odometry, EKF-RIO [44] performs 6-degree-of-freedom pose estimation combining IMU measurements with radar Doppler velocities, and 4D-iRIOM [206] estimates the poses of a robot with a cluster-based matching approach and a Kalman filter to handle radar sparsity.

Techniques that specifically deal with outliers include probabilistic approaches like the normal distribution transform by Rapp et al. [149, 150] or the Gaussian mixture model by Haggag et al. [68], which accounts for the uncertainty of a measurement as a way of dealing with the high amount of outliers. Others approach the problem based only on the Doppler velocity information [58, 89, 107], and/or on the vehicle's motion kinematics [125, 151, 156]. However, discarding the positional information and solely relying on the radar Doppler velocities can lead to increased drift in the trajectory estimate [26]. Moreover, methods that only rely on odometry suffer from error accumulation over time, leading to drift in the trajectory estimates.

Full automotive radar SLAM approaches that optimize a pose graph have also been proposed, but have some limitations. Holder et al. [72] require wheel odometers, and Isele et al. [81] require semantic segmentation of the radar point clouds to be performed online. Li et al. [117] and Park et al. [143] propose

to include the Doppler velocity information into the pose graph as a way of constraining odometry. This is later leveraged by Wang et al. [178] to constrain the IMU motion within the factor graph. Similarly, Zhang et al. [204] integrate point measurement uncertainty into their system. However, as shown in the experiments of our work, one major limitation of these approaches is that they rely on loop closure designed for LiDARs, limiting place recognition accuracy.

In addition, the sparsity of radar scans may lead to noisy trajectory estimates, which can be improved by incorporating additional input from an IMU into the full SLAM system. Some methods that combine automotive radar and IMU information utilize Kalman filters, like Zhuang et al. [206], or continuous-time optimization to estimate the pose of the vehicle over time, like the approach proposed by Ng et al. [137]. Recent research from Girod et al. [63] and Wang et al. [178] has shown how graph-based methods are advantageous in terms of accuracy and simplicity of the system. However, these approaches integrate all the measurement information into a single optimizable graph adapted from LiDAR [98], and face limitations in terms of accuracy, highlighting the need for techniques specifically designed for automotive radars.

3.2 Ground Points as a Source of Information

One of the main challenges of automotive radar pose estimation is the limited vertical resolution of these sensors. This limitation can be observed in the sensor comparison from Table 2.1 on page 20. Nevertheless, the ground plane is a feature commonly present in most LiDAR and radar scans when mounted on a vehicle. We can leverage this information to help reduce vertical estimation errors as shown by Koide et al. [98] in the LiDAR domain. LiDAR ground plane segmentation and detection approaches exist to identify the points in a scan that belong to the ground and estimate the ground plane parameters, commonly using heuristics. Himmelsbach et al. [71] and Steinhauser et al. [165] fit lines to point sets within the scan, classifying points as ground or non-ground based on the properties of the line segments. A more recent work by Zermas et al. [201] proposes to use principal component analysis [52] to estimate the final ground plane. However, they rely on the assumption that the lowest points within a scan belong to the ground, which is not true in radar point clouds due to the high amount of noise and multi-path propagation. Based on the cylindrical geometry of LiDAR point clouds, other approaches like Patchwork [121] and Patchwork++ [114] divide the scans into multiple concentric regions that are segmented separately. Narski et al. [134] also leverage the LiDAR ring properties for their segmentation method. Moreover, Koide et al. [98] present a ground-plane fitting approach where they include a horizontal ground plane within a pose graph. Nevertheless,

these methods have been developed for LiDAR sensors, relying on minimal noise, little multi-path propagation below the ground, and concentric point clouds from spinning laser devices. Solely one approach by Chen et al. [33] extracts the velocity of the ground points to estimate the vehicle's ego-motion. However, it provides little insight into how this strategy affects vertical pose estimation, underscoring the importance of ground extraction approaches that enhance vertical motion estimation accuracy.

3.3 Place Recognition

Place recognition identifies whether the current location has already been visited before. In practice, this is similar to finding the closest matching scene in a map database. A key aspect of place recognition is finding a compact scene descriptor that accurately represents a scene. As a result, scene descriptors belonging to the same place have a high similarity, while scene descriptors that belong to different places have a low similarity. In the case of camera-based solutions, they encode the images into handcrafted features like ORB [133], learned features such as the approaches by Arandjelovic et al. [8] and Izquierdo et al. [82], or enhance their system exploiting image sequence information, such as the techniques by Milford et al. [130] or by Vysotska and Stachniss [174, 175]. LiDAR approaches commonly compress the scan into a more compact representation based on geometric or learned features. Namely, LiDAR geometric approaches usually transform the point cloud into a polar representation, as seen in ScanContext [94] and its multiple variations [48, 85, 93, 179], or into density grid maps, introduced by Gupta et al. [67]. The main concern with such methods is that they rely on structural assumptions about the environments, like height, point density, intensity, or contours. Advances in LiDAR place recognition show that enhancing geometric approaches with neural network-based solutions yields higher recognition recall, like DiSCO [189] and the method by Kim et al. [91]. Another group of approaches discretizes the space into a voxel grid [32, 102, 163, 187], which leads to information loss, which is especially harmful for sparse point clouds in radar scans. Other methods like PointNetVLAD [171] and KPPR [185] consider each individual point by using a pointwise feature encoder, which is more suitable for our task of sparse radar scan place recognition. They aggregate their local point features into a global descriptor representative of the scene. Furthermore, the technique by Di Giammarino et al. [41] incorporates LiDAR intensity information. In terms of autonomous driving, bad weather conditions and difficult packaging are the primary concerns when incorporating these sensors into end-user vehicles. Moreover, as shown by Cai et al. [22], adapting LiDAR place recognition methods [94, 101] directly for automotive radars can lead to a signif-

icant decrease in performance due to the sparsity and the low number of points when projecting the scans into an image.

Radar-based place recognition techniques address these challenges by exploiting the additional Doppler velocity and radar cross section information provided by the sensor. Several methods employ 2D spinning radars using handcrafted features [1, 37, 84], learned features [12, 55, 155, 170], or contrastive learning, as proposed by Yuan et al. [197]. They use the intensity image provided by the spinning radar, encoding it into a descriptor that represents each place. Alternatively, solutions exist where the robot carrying a radar localizes within maps collected using different sensor types, such as LiDARs [128, 135, 193], satellite images [166], and binary maps [181]. This is particularly useful in locations where data can be previously collected with high-accuracy sensing techniques, but these maps are not always available. New approaches by Fu et al. [54] and Garcia-Hernandez et al. [59] also combine multi-modal sensor fusion using camera and radars for place recognition. Methods by Li et al. [117] and Zhang et al. [204] that exclusively rely on automotive radar generally leverage a variant of intensity ScanContext [179] as an additional component to their SLAM pipeline. However, their primary focus is on achieving a complete radar SLAM system, with minimal emphasis on the place recognition component. Notably, Autoplace [22] and TransLoc4D [145] aim for high-accuracy place recognition using automotive radars. Autoplace, however, imposes limitations on geometric understanding between the points and restricts the features to a planar space. Additionally, it relies on point cloud aggregation, which requires an additional odometer and the availability of radar sub-maps for place recognition. Autoplace's resulting descriptor vector is also high-dimensional, compromising storage efficiency.

3.4 Long-Term Mapping and Localization

Long-term mapping and localization are crucial for the navigation of autonomous vehicles within previously recorded maps. The goal is to achieve an accurate map that can be extended over time and perform pose estimation within it. Long-term mapping and localization involve three main steps: multi-session mapping, map maintenance, and map-based localization.

Multi-session mapping aims to construct maps from data acquired at different points in time and to align them to a common reference frame. Various approaches exist, proposed by Blanco-Claraco [17] and Lazaro et al. [111], to maintain global consistency of maps in large-scale environments. Additionally, Gupta et al. [67] and Lazaro et al. [112] address the problem of efficient submap alignment. In particular, Kim et al. [96] and Yang et al. [191] address multi-session mapping on LiDAR sensors. They align multiple maps by perform-

ing ScanContext [94] place recognition on a reference session, creating a high dependence on the pose accuracy and the coverage of the reference map. Moreover, they do not perform loop verification, which can lead to incorrect matches, especially in sparse radar point clouds [24]. To date, there are no multi-session mapping approaches specifically designed for radar data, requiring specialized methods that effectively handle the noise and sparsity of radar scans.

Once the different maps are aligned to a common reference frame, map maintenance involves identifying map variations over time. Objects can be categorized into dynamic, volatile, and persistent. Dynamic objects, such as pedestrians and moving cars, are commonly identified online using learning-based techniques as proposed by Zeller et al. [198] or the dynamic point outlier rejection strategies by Kellner et al. [89]. Volatile objects may or may not move during a mapping session, hence they are not always identifiable at runtime. Reasons for this may include false negatives from failing object detectors or the absence of certain objects within the training dataset. Offline LiDAR approaches by Gil et al. [62], Kim et al. [95, 96], and Lim et al. [119, 120], detect volatile objects and erase them from existing maps by estimating the consistency between measurements, where multiple LiDAR maps are combined and their difference is extracted. Furthermore, volatile objects may also include new construction sites and new buildings, which need to be accounted for within the map. On the contrary, persistent objects do not vary over time, and they are considered highly reliable for localization [62]. A focus is also placed on how eliminating temporal changes affects the final map size, map compression, or the localization performance [77, 96, 126, 196]. These methods, however, are specifically designed for removing outliers from dense and accurate LiDAR scans. Radars, on the contrary, output sparse scans with high amounts of random noise, clutter, and multi-path propagation, leading to challenges when computing scan-to-map point correspondences and when identifying differences between maps built at different points in time.

After performing multi-session map alignment and map maintenance, we can leverage the information of the map to improve the pose estimation accuracy by performing map-based localization. Approaches can be categorized into absolute and relative localization. Absolute localization estimates the absolute pose of the robot on the entire map and is generally used for computing the first pose for relative localization. Absolute localization approaches typically leverage place recognition, described in Section 3.3. LiDAR approaches work well with dense and accurate point clouds, such as the ones proposed by Chen et al. [36], Kim et al. [94], and Wiesmann et al. [185]. Meanwhile, radar methods exploit the nature of radar scans, achieving higher performance in the radar domain [24] or localizing in different map modalities as suggested by Ma et al. [128], Nayak et al. [135], and Yin et al. [193]. Relative localization methods assume a

known initial pose estimate. Although there is little published work specific to relative localization, Wen et al. [182] and Yoneda et al. [195] match LiDAR scans to high-definition maps. Meanwhile, Alamos et al. [3], Baek et al. [9] and Blanco-Claraco [17] directly perform scan matching to the prior map. This kind of localization is generally achieved using scan-to-map point cloud registration [172]. However, during localization, incorrect point correspondences hinder scan matching performance. These failures often result from movable elements in the scene that have changed since the map was created. Some long-term localization approaches by Fang et al. [49] and Peng et al. [144] alternate between odometry and localization based on the number of point correspondences, and Hroob et al. [75] predict stable points of the environment during operation. These methods, however, rely on scan-to-map matching of dense and accurate LiDAR scans and face limitations when applied to sparse radar scans.

Chapter 4

Radar-Only Odometry and Mapping

Sensor odometry plays a key role in the navigation of autonomous vehicles in unfamiliar terrain. It allows autonomous vehicles to track their own pose solely leveraging local sensor data, without relying on GNSS. This process enables the vehicle to determine its pose in scenarios with reduced satellite visibility, such as urban canyons and under dense foliage, or in indoor scenarios like tunnels and parking lots. Accurate pose estimation is also crucial for tasks that require knowing the vehicle's pose over time, including motion planning and obstacle avoidance. Additionally, mapping enables the vehicle to construct a representation of the environment over time. This information can be later leveraged for trajectory planning and to improve localization accuracy.

Odometry and mapping are commonly achieved using cameras or LiDARs. Existing approaches typically track features across multiple camera images [23] or perform scan registration by matching elements within consecutive LiDAR point clouds [172, 180]. However, as introduced in Section 2.1, these sensors are affected by adverse weather such as heavy rain, snow, and fog. In contrast, automotive radars are resilient to environmental conditions [31]. They also provide additional information, such as the Doppler velocity and radar cross section of the measurements. Nevertheless, the noise and sparsity of their output challenge the accuracy of radar odometry and mapping methodologies. State-of-the-art approaches by Kellner et al. [89, 90] discussed in Section 3.1 assume a mostly static environment and use only the Doppler velocities to estimate the vehicle's ego-motion. Moreover, velocity-based approaches often rely on estimating the vehicle's motion from a single scan. This can fail in edge cases where a large dynamic object blocks the sensor's field of view, as very few measurements belong to the actual static environment. Other approaches, including the works by Kung et al. [109] and Aldera et al. [6], rely only on positional information of the

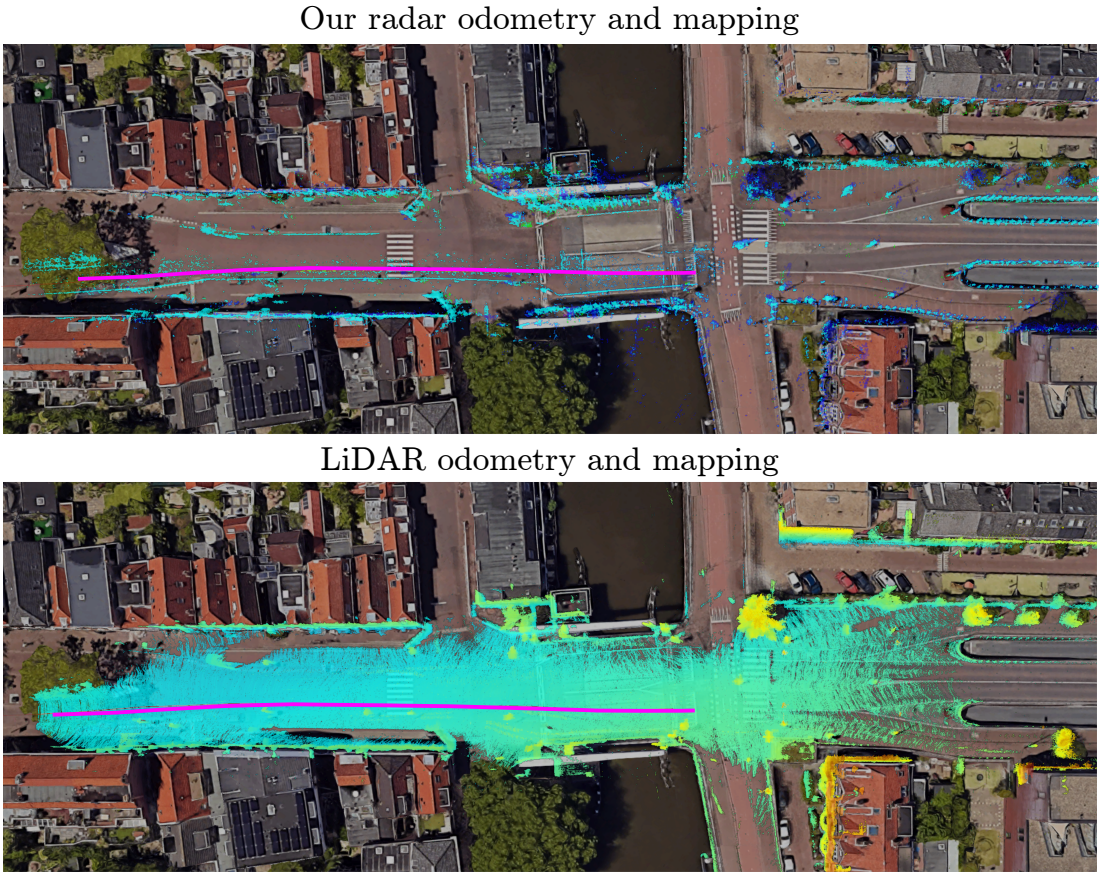


Figure 4.1: Satellite view of the odometry and mapping from our radar-ICP compared to aggregated LiDAR data. Although the radar map is noticeably sparser, the estimated trajectory by our radar-only method closely resembles the trajectory estimated using a LiDAR sensor.

measurements. Furthermore, very few approaches tackle the mapping task, and only the methods by Kung et al. [110] and Weston et al. [183] address the radar scan denoising task. However, their predictions focus on spinning radars, which are hard to integrate into consumer vehicles.

In our work, we take inspiration from the LiDAR domain [69, 172] and demonstrate how combining both velocity and positional information enhances radar-only odometry. Moreover, we present a velocity pre-filtering strategy that eliminates outliers when the radar sensor is being blocked by large moving objects, leading to enhanced trajectory estimation over time. In this chapter, we also introduce a filtering strategy that uses velocity and point matches across scans to eliminate outliers, and compare the results against its relative LiDAR map, as shown in Figure 4.1.

The main contribution of this chapter is the introduction of three techniques for online odometry and mapping that exploit the capabilities of automotive radar sensors. We focus on two ego-motion estimation approaches and a novel mapping step. The first method presents a novel point-to-point iterative clos-

est point (ICP) technique specifically designed to leverage the Doppler velocity measurements provided by 3D automotive radar sensors. The second method is tailored for low-speed scenarios where the radar is being covered by large moving obstacles such as trucks or buses. This configuration is particularly advantageous for 2D radar sensors with a low number of points. Our approach combines a constant velocity filter and Doppler velocity measurements to estimate the vehicle's ego-motion. We also give weight to the mapping stage and introduce a simple, yet effective, filtering step to improve the quality of the resulting map. Finally, we conduct an extensive evaluation using public real-world datasets.

In sum, we make three key claims. Our approach: (i) achieves state-of-the-art results in 3D radar odometry using a point-to-point radar ICP approach; (ii) accurately estimates odometry in 2D radars, particularly in scenarios where most of the radar is blocked by large dynamic objects; and (iii) provides a simple yet effective filtering step for radar map creation leading to a high similarity compared to its corresponding LiDAR map.

4.1 Our Approach to Radar Odometry

In this chapter, we estimate the pose of the vehicle and store the sensor readings in a map of the environment. For this, we introduce two new odometry methods for 3D and 2D automotive radars. For the 3D case, we demonstrate how our scan-to-map point-to-point ICP strategy achieves state-of-the-art accuracy using the positional and velocity information of the measurements. For the 2D case, we improve previous approaches by addressing scenarios where most of the radar's field of view is being covered by a moving obstacle. Throughout this chapter, all measurements and velocities are expressed in the car frame; therefore, we omit the left superscript ${}^{C}(\cdot)$ to simplify notation.

4.1.1 Point-to-Point Radar ICP

One of the main challenges present in automotive radars is the sparsity of their output point cloud. As a result, scan matching strategies that involve finding point correspondences have been commonly outperformed by the normal distribution transform [109] and other probabilistic approaches as introduced by Haggag et al. [68]. However, we demonstrate how Doppler velocity information provided by the radar sensors can be leveraged to augment the point-to-point ICP algorithm, leading to high registration accuracy. We demonstrate how the method benefits from the higher number of points and the additional axis provided by 3D radars, while still providing accurate results for 2D radar sensors. As introduced in Section 2.5.2.1, the goal of point-to-point ICP is to obtain the

transformation $\mathsf{T} \in SE(3)$ that minimizes the distance between a source point set $\mathbf{p} \in \mathcal{P}$, and a target point set $\mathbf{q} \in \mathcal{Q}$. In our method, we address the sparsity of radar point clouds by building $\mathcal{Q}$ as a submap that aggregates the previous scans. Each iteration involves two steps.

First, we obtain the set of N point-to-point correspondences between the two point clouds based on the Euclidean distance between matches. Then we minimize the point geometric error function E_t between all correspondences following

$$E_t(\mathsf{T}) = \sum_{k=1}^{N} \rho_t(||\mathsf{T}\mathbf{p}_k - \mathbf{q}_k||), \tag{4.1}$$

where ρ_t refers to the Geman McClure kernel to reduce the effect of outliers, similar to the work by Vizzo et al. [172].

As in Equation (2.76), we derive the Jacobian of the position error $\mathbf{J}_{t_k} \in \mathbb{R}^{3\times 6}$ for the k^{th} point using its Lie algebra formulation from Section 2.4 as

$$\mathbf{J}_{t_k} = \begin{bmatrix} \mathbf{I}_{3\times 3} & -(\mathsf{T}\mathbf{p}_k)^{\wedge} \end{bmatrix}. \tag{4.2}$$

Equation (4.1) is used with regular LiDAR sensors, which often include only positional information. Nevertheless, radars also provide additional Doppler velocity information of each point that can be exploited to improve pose estimation accuracy. In our approach, we follow the derivation presented in Section 2.2 on page 21 to express the vehicle velocity projected to the radial direction of the measurement. In this case, however, we cannot directly compute the projection of the vehicle velocity into the direction of the measurement $v_{\text{proj}_k} \in \mathbb{R}$, since we do not know the true linear and angular velocities of the vehicle $\boldsymbol{v}_C, \boldsymbol{\omega}_C \in \mathbb{R}^3$. Instead, we leverage the estimated velocities at the current ICP iteration, $\hat{\boldsymbol{v}}_C, \hat{\boldsymbol{\omega}}_C \in \mathbb{R}^3$, to compute the estimated projection $\hat{v}_{\text{proj}_k} \in \mathbb{R}$. We then compare it with the measured Doppler velocity by the radar sensor v_{r_k} for each point to correct the point-to-point ICP estimation error.

To obtain $\hat{\boldsymbol{v}}_C$ and $\hat{\boldsymbol{\omega}}_C$, we first express the current transformation T as its Lie algebra using the logarithmic mapping from Section 2.4 following

$$\text{Log}(\mathsf{T}) : SE(3) \rightarrow \hat{\boldsymbol{\xi}} = \begin{bmatrix} \hat{\boldsymbol{\rho}}_C \\ \hat{\boldsymbol{\phi}}_C \end{bmatrix} \in \mathbb{R}^6. \tag{4.3}$$

The estimated velocities of the vehicle are computed given the time increment between two radar scans, following

$$\dot{\hat{\boldsymbol{\xi}}} = \begin{bmatrix} \hat{\boldsymbol{v}}_C \\ \hat{\boldsymbol{\omega}}_C \end{bmatrix} = \frac{1}{\Delta t} \begin{bmatrix} \hat{\boldsymbol{\rho}}_C \\ \hat{\boldsymbol{\phi}}_C \end{bmatrix}. \tag{4.4}$$

We leverage the estimated vehicle velocities to get the estimated projection following Equation (2.26), which adapts to

$$\hat{v}_{\text{proj}_k} = -(\boldsymbol{d}^S_{\mathbf{p}_k})^{\top} \left(\hat{\boldsymbol{v}}_C + \hat{\boldsymbol{\omega}}_C \times \mathbf{t}^C_S \right) = -\frac{1}{\Delta t} (\boldsymbol{d}^S_{\mathbf{p}_k})^{\top} \left(\hat{\boldsymbol{\rho}}_C + \hat{\boldsymbol{\phi}}_C \times \mathbf{t}^C_S \right). \tag{4.5}$$

The unit vector $\boldsymbol{d}^S_{\mathbf{p}_k}$ indicates the direction vector of the sensor measurement with respect to the radar given in the car frame.

The final velocity error E_v is defined as the residual between the measured Doppler velocity by the radar v_{r_k}, and the estimated projection $\hat{v}_{\text{proj}_k}$ of the vehicle velocity into the measurement direction according to

$$E_v(\mathsf{T}) = \sum_{k=1}^{N} \rho_v(||v_{r_k} - \hat{v}_{\text{proj}_k}||), \tag{4.6}$$

where the Geman McClure kernel ρ_v is tuned empirically based on the distribution of the velocity residuals in a radar scan.

We compute the Jacobian of the vehicle velocity error $\mathbf{J}_{v_k} \in \mathbb{R}^{1\times 6}$ by deriving Equation (4.5), which results in

$$\mathbf{J}_{v_k} = \begin{bmatrix} \dfrac{-\boldsymbol{d}^S_{\mathbf{p}_k}}{\Delta t} & \dfrac{-\boldsymbol{d}^S_{\mathbf{p}_k} \times \mathbf{t}^C_S}{\Delta t} \end{bmatrix}. \tag{4.7}$$

The optimization problem outputs the locally optimal transform between the previous pose and the current one $\mathsf{T}^* \in SE(3)$. The final error function depends on the point positions and velocities following

$$\mathsf{T}^* = \underset{\mathsf{T}}{\operatorname{argmin}} \Big[(1-\gamma) E_t(\mathsf{T}) + \gamma E_v(\mathsf{T}) \Big], \tag{4.8}$$

where parameter $\gamma \in [0, 1]$ allows us to weigh each error type individually. Note that a value of $\gamma = 0$ would result in simple point-to-point ICP without taking into consideration the Doppler velocities.

The final Jacobian $\mathbf{J}_{(t,v)_k} \in \mathbb{R}^{4\times 6}$ is obtained by combining the point-to-point ICP Jacobian with the velocity Jacobian following

$$\mathbf{J}_{(t,v)_k} = \begin{bmatrix} \mathbf{I}_{3\times 3} & -(\mathsf{T}\mathbf{p}_k)^\wedge \\ \dfrac{-\boldsymbol{d}^S_{\mathbf{p}_k}}{\Delta t} & \dfrac{-\boldsymbol{d}^S_{\mathbf{p}_k} \times \mathbf{t}^C_S}{\Delta t} \end{bmatrix}. \tag{4.9}$$

Given the residual from Equation (4.8) and the Jacobian from Equation (4.9), we can formulate the optimization as a least-squares problem which can be optimized with the Gauss-Newton method as described in Section 2.3.3.2.

4.1.2 Single-Scan Odometry and Velocity Filter

As we have already seen, the Doppler velocities provided by the radars can be exploited for pose estimation. In the 2D radar domain, and under the assumptions of a mostly static environment, Kellner et al. [89] propose a least squares solution to directly estimate the vehicle's odometry without performing scan matching. Their approach leverages RANSAC [52] as a dynamic outlier rejection technique.

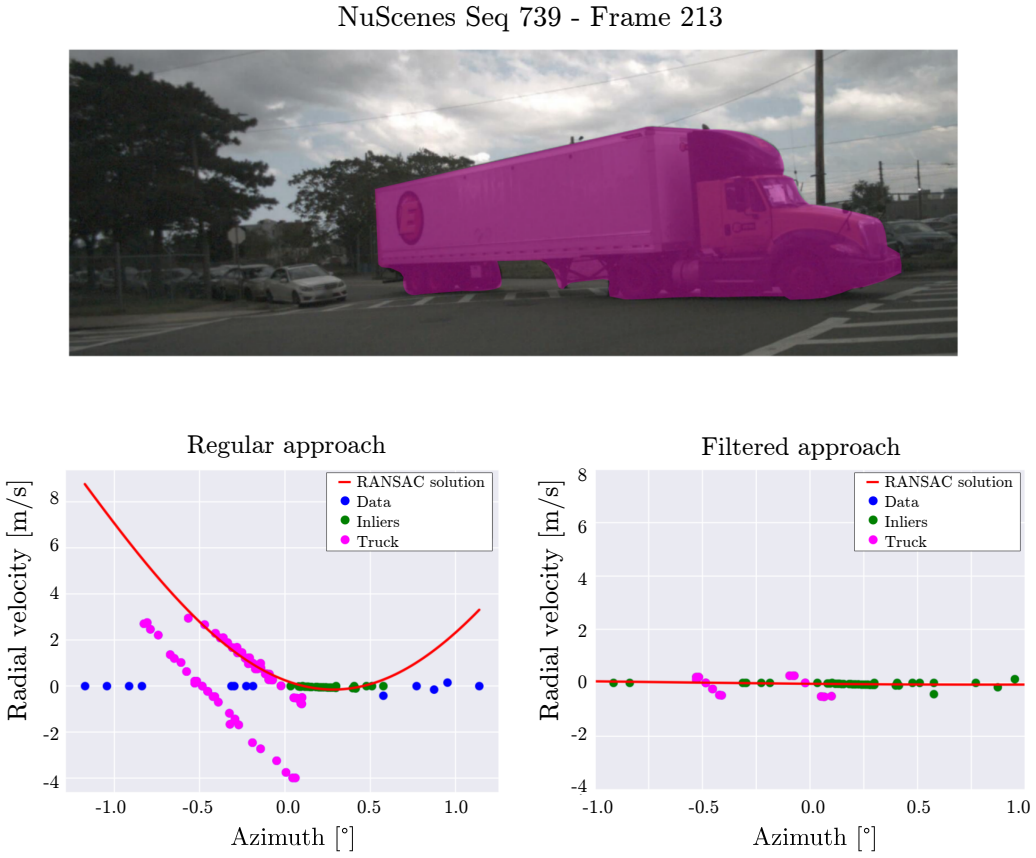


Figure 4.2: Scenario where a truck is moving in front of a static radar sensor. The method by Kellner et al. [89] fails by considering the truck's wheels as inliers. Our filter eliminates points that do not match the vehicle's motion, leading to an accurate ego-velocity estimation.

It requires a minimum of two velocity measurements from stationary targets to output a solution, making it appropriate for radars with a low number of points like the 2D Continental ARS408 sensor used in the nuScenes [21] dataset, which has a maximum of 125 points per scan. However, solely relying on the current scan as proposed by Kellner et al. [89] makes the system vulnerable to scenarios where most of the points measured by the radar sensor come from a large moving object, as illustrated in Figure 4.2. Our method addresses these edge cases by introducing a filtering step prior to RANSAC outlier removal. This filter only keeps feasible samples according to the car's previous movement following the constant velocity motion model, under the assumption that a vehicle will have no substantial changes in velocity between two consecutive scans, which automotive radars typically capture in 50 ms intervals.

To achieve this, we approximate the translation and angular velocities in the current time step by leveraging the pose estimates from the two previous scans, T_{i-1} and $\mathsf{T}_{i-2} \in SE(3)$. We compute a constant velocity estimate of the

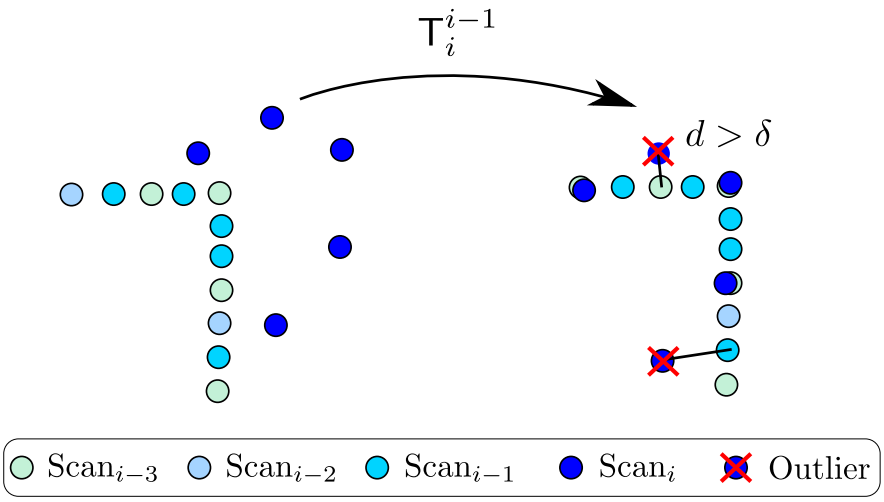


Figure 4.3: The local map is built by comparing the current scan to the previous $M = 3$ scans. Comparing the distance d of each point in the current scan to its closest neighbor in the local submap, we determine if a point is an outlier if the distance is higher than threshold δ.

car velocities following

$$\begin{bmatrix} \tilde{\boldsymbol{v}}_C \\ \tilde{\boldsymbol{\omega}}_C \end{bmatrix} = \frac{\mathrm{Log}(\mathsf{T}_{i-2}^{\top}\,\mathsf{T}_{i-1})}{\Delta t}, \tag{4.10}$$

where Δt is the difference in timestamps between the previous and the current scan, and $\mathrm{Log}(\cdot)$ extracts the linear translation for each axis, as well as the axis-angle representation of the rotation, as described by the Lie theory, Section 2.4.

Similar to Equation (4.5), the predicted velocity in the radial direction for each k^{th} point in the current scan can be estimated by computing the projection of the sensor velocity in the vehicle frame to the direction of the measurement according to

$$\tilde{v}_{\mathrm{proj}_k} = -(\boldsymbol{d}_{\mathbf{p}_k}^S)^{\top} \left(\tilde{\boldsymbol{v}}_C + \tilde{\boldsymbol{\omega}}_C \times \mathbf{t}_S^C\right). \tag{4.11}$$

Then, it is possible to filter each point with a measured Doppler velocity v_{r_k} that does not match the predicted point velocity within a threshold following

$$v_{r_k} - \tilde{v}_{\mathrm{proj}_k} < \varepsilon, \tag{4.12}$$

where ε is adjusted empirically based on the radar velocity resolution.

This simple thresholding method serves as a first-step filtering executed before RANSAC [52], helping to reduce the risk of failure by rejecting outliers.

4.2 Constructing a Radar Map

In the previous sections, we presented our methods for accurately estimating the pose of the vehicle and filtering out dynamic point outliers. During pose estimation, we can construct the environment map by aggregating the radar scans within

the estimated poses. Nevertheless, as radar point clouds are noisy, it is important that only valuable points are stored for future localization. We propose a simple yet effective strategy to improve the consistency of the points in the resulting map, filtering out clutter and noise from the radar scans. We approach this using the principle of scan-to-submap matching introduced by Newcombe et al. [136].

In our method, illustrated in Figure 4.3, we initialize the map with the first M measured unfiltered scans. We then compare the new radar measurements with the previous M aggregated scans and find the point correspondences between them within a radius δ, similar to the ICP matching step in Section 4.1.1. Every point that has a correspondence will be considered as an inlier, and those points without correspondences will be discarded. We obtain the final radar map by aggregating all the filtered scans within a sequence, resulting in a clean map with a low number of noise and clutter outliers.

4.3 Implementation Details

In our implementation, we build upon KISS-ICP [172] to support radar point cloud data. We add the Doppler velocities to the ICP formulation with a Doppler weight $\gamma = 0.1$, as well as our proposed RANSAC outlier pre-filter, where we set a threshold of $\varepsilon = 0.5\,\mathrm{m/s}$ for the nuScenes dataset [21] and $\varepsilon = 0.1\,\mathrm{m/s}$ for the View of Delft [139] dataset. As for constructing the radar map, we aggregate the previous $M = 3$ scans and retain point correspondences within a radius $\delta = 1.5\,\mathrm{m}$ in our evaluation.

4.4 Experimental Evaluation

We present our experiments to show the capabilities of our work. The results of our evaluation support our key claims that our work: (i) achieves state-of-the-art results on 3D radar odometry using a point-to-point radar ICP approach; (ii) accurately estimates odometry in 2D radars, particularly in scenarios where most of the radar is blocked by large dynamic objects; and (iii) provides a simple yet effective filtering step for radar map creation leading to a high similarity compared to its corresponding LiDAR map.

4.4.1 Experimental Setup and Metrics

We evaluate our strategies in real-world scenarios, leveraging the nuScenes [21] and View of Delft [139] datasets for 2D and 3D automotive radars, respectively. We first evaluate our work on odometry by comparing it to the dataset reference trajectories and to other radar and LiDAR methods. Moreover, we focus on

planar trajectories, removing all vertical movement from the pose estimates. We denote our ICP approach from Section 4.1.1 as "Radar ICP" and our constant velocity filter method from Section 4.1.2 as "Velocity filtered". The best results on radar data are shown in **bold**. We show the results for generic scenarios as well as for some edge cases that particularly benefit from our work. We employ the evo library [65] to compute the relative translation and rotation error metrics t_{RPE} and r_{RPE}, from Section 2.5.5, and report the mean, root-mean-squared error (RMSE), and standard deviation (std.) results.

We evaluate the filtered radar map with respect to the LiDAR map by constructing the reference map and removing dynamic objects from the LiDAR point clouds based on ground truth annotations and the poses estimated by our odometry. As we are only using the front-facing radars in both datasets, we only consider LiDAR points within the camera field-of-view. Moreover, in nuScenes [21], the radar range of 250 m is much larger than the LiDAR's of 100 m. Thus, we only consider radar points within the LiDAR range for evaluation. We use the symmetric chamfer distance $\text{SCD}(\mathcal{R} \leftrightarrows \mathcal{L})$ to measure similarity between the radar map points $\mathcal{R}$ and the LiDAR map points $\mathcal{L}$. It measures the average bidirectional closest-point matching error between radar and LiDAR maps. A smaller distance indicates better results, i.e., two point clouds are similar to each other, while a larger distance indicates a big difference between them. It is defined as

$$\text{SCD}(\mathcal{R} \leftrightarrows \mathcal{L}) = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \min_{\mathbf{l} \in \mathcal{L}} \|\mathbf{r} - \mathbf{l}\|^2 + \frac{1}{|\mathcal{L}|} \sum_{\mathbf{l} \in \mathcal{L}} \min_{\mathbf{r} \in \mathcal{R}} \|\mathbf{l} - \mathbf{r}\|^2. \tag{4.13}$$

Additionally, in order to reduce the effect of 2D radar noise points being matched with the LiDAR ground points in nuScenes, we also show the asymmetric chamfer distance $\text{ACD}(\mathcal{R} \rightarrow \mathcal{L})$ measuring only the similarity from the radar to the LiDAR map, defined as

$$\text{ACD}(\mathcal{R} \rightarrow \mathcal{L}) = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \min_{\mathbf{l} \in \mathcal{L}} \|\mathbf{r} - \mathbf{l}\|. \tag{4.14}$$

4.4.2 Performance on View of Delft 3D Radar Dataset

The first experiment evaluates the accuracy of our 3D radar odometry estimation pipeline in comparison to other methods. This evaluation supports our claim that leveraging Doppler velocity information with point-to-point ICP can yield high accuracy, obtaining state-of-the-art results in 3D radar odometry. We evaluate both of our methods in the 3D radar dataset [139] and compare them with other approaches in Table 4.1, including point-to-point ICP [16], KISS-ICP [172] on radar and LiDAR data, the ego-motion estimation approach by Kellner et al. [89], point-to-plane DICP [69], and the deep learning-based approach 4DRVO-Net [207], which combines camera and radar data.

Table 4.1: Our approach sets a new benchmark for radar odometry on View of Delft [139].

	t_{RPE} [m]			r_{RPE} [°]		
	Mean	RMSE	std.	Mean	RMSE	std.
KISS-ICP (LiDAR) [172]	0.048	0.060	0.036	0.051	0.080	0.061
4DRVO-Net [207]	0.080	-	-	0.070	-	-
DICP [69]	0.079	0.101	0.061	0.073	0.168	0.145
Kellner et al. [89]	0.078	0.116	0.059	0.097	0.171	0.120
Point-to-point ICP [16]	0.067	0.088	0.055	0.064	0.100	0.073
KISS-ICP (Radar) [172]	0.061	0.073	0.040	0.074	0.114	0.080
Velocity filtered (Ours)	0.078	0.115	0.059	0.097	0.170	0.116
Radar ICP (Ours)	**0.049**	**0.062**	**0.036**	**0.059**	**0.089**	**0.066**

Table 4.2: Our approach sets a new benchmark for radar odometry on nuScenes [21].

	t_{RPE} [m]			r_{RPE} [°]		
	Mean	RMSE	std.	Mean	RMSE	std.
KISS-ICP (LiDAR) [172]	0.013	0.016	0.012	0.035	0.051	0.051
CREME [68]	-	0.012	-	-	**0.100**	-
Kellner et al. [89]	0.010	**0.011**	0.009	0.108	0.174	0.206
Point-to-point ICP [16]	0.354	0.641	2.246	0.468	0.875	4.871
KISS-ICP (Radar) [172]	0.133	0.162	0.104	0.149	0.196	0.140
Velocity filtered (Ours)	**0.009**	**0.011**	**0.008**	0.088	0.117	0.110
Radar ICP (Ours)	0.012	0.015	0.009	**0.085**	0.126	**0.089**

Overall, our radar ICP approach gives the best scan-to-scan t_{RPE} and r_{RPE} results compared to other radar methods, achieving accuracy comparable to LiDAR. Note that KISS-ICP is a state-of-the-art method for LiDAR scan registration, based on classical point-to-point matching. The good results of KISS-ICP on radar data show the importance of point-to-point positional information, improved by our technique that exploits the Doppler velocity measurements.

4.4.3 Performance in Presence of Large Dynamic Objects

The second experiment studies the performance of our second odometry method in 2D radars when big moving objects are blocking the radar's field of view. While the View of Delft dataset [139] is good for evaluating odometry and mapping, the driving scenarios are limited to a specific area in a city, and most outliers

Table 4.3: Our filtering improves the symmetric chamfer distanc (SCD) and the asymmetric chamfer distance (ACD), expressed in [m^2], between LiDAR and radar maps.

	View of Delft		**NuScenes**	
	SCD($\mathcal{R} \leftrightarrows \mathcal{L}$)	ACD($\mathcal{R} \rightarrow \mathcal{L}$)	SCD($\mathcal{R} \leftrightarrows \mathcal{L}$)	ACD($\mathcal{R} \rightarrow \mathcal{L}$)
No filter	10.98	1.20	44.22	2.44
Random	10.91	1.20	45.36	2.44
Ours	**3.61**	**0.66**	**40.47**	**1.92**

belong to small objects that can be easily removed by RANSAC [52]. However, the nuScenes dataset [21] has a wider variety of cases, some of which include trucks and buses almost blocking the complete field of view of the radar, as seen previously in Figure 4.2.

Our results from Table 4.4 show scenes where a big truck or bus is directly turning in front of the estimating vehicle. We also evaluate our method on the full dataset in Table 4.2 comparing it against point-to-point ICP [16], KISS-ICP [172] on radar and LiDAR, the approach by Kellner et al. [89], and CREME [68].

As shown in Table 4.4, our 2D odometry strategy accurately estimates the trajectory by reducing the influence of outliers from large moving objects. This results in enhanced pose estimates with respect to its baseline [89]. In addition, despite our technique not having a significant impact on the View of Delft dataset, see Table 4.1, Table 4.2 shows that our filtering does improve the overall accuracy over the nuScenes dataset, particularly benefiting 2D automotive radars.

4.4.4 Mapping Results

The last experiment evaluates our mapping stage, showing how our filtering technique achieves maps with reduced outliers. We report quantitatively in Table 4.3 and qualitatively in Figure 4.4 that our filtered map obtains improved similarity with the LiDAR map. We also display the LiDAR maps for comparison with the unfiltered and filtered radar maps.

The qualitative results from Figure 4.4 show how our strategy removes a substantial amount of outliers that do not belong to structural elements. Additionally, dynamic outliers are also being filtered out thanks to the Doppler velocity provided by the radar. Furthermore, Table 4.3 shows that our approach leads to an improvement in symmetric and asymmetric chamfer distances, and while it removes a high amount of noise and dynamic outliers, it preserves the structures in the environment. Subsequently, the filtered map has a remarkably higher similarity than the unfiltered map and the map filtered using random sampling.

Table 4.4: Our 2D odometry achieves improved results on nuScenes [21] sequences with the presence of big moving objects. t_{RPE} in [m] and r_{RPE} in [°].

	0224		**0242**		**0587**		**0739**		**0749**		**0875**		**0998**		**Mean**	
	t_{RPE}	r_{RPE}	t_{RPE}	r_{RPE}	t_{RPE}	r_{RPE}	t_{RPE}	r_{RPE}	t_{RPE}	r_{RPE}	t_{RPE}	r_{RPE}	t_{RPE}	r_{RPE}	t_{RPE}	r_{RPE}
Kellner et al. [89]	0.0075	0.3088	0.0088	0.4698	0.0031	0.1087	0.0066	0.1832	0.0060	0.4157	0.0071	0.2390	0.0120	0.3103	0.0071	0.2859
CREME [68]	0.0044	0.0812	0.0051	0.0867	0.0027	0.0675	0.0056	0.0714	0.0030	0.0630	0.0032	0.0528	0.0066	0.0800	0.0044	0.0718
Velocity filtered (Ours)	**0.0036**	**0.0435**	**0.0031**	**0.0672**	**0.0020**	**0.0294**	**0.0052**	**0.0514**	**0.0022**	**0.0359**	**0.0024**	**0.0434**	**0.0060**	**0.0653**	**0.0034**	**0.0472**
Radar ICP (Ours)	0.0164	0.3288	0.0224	0.3461	0.0104	0.1494	0.0106	0.1161	0.0178	0.0617	0.0123	0.204	0.0328	0.2621	0.0175	0.2097

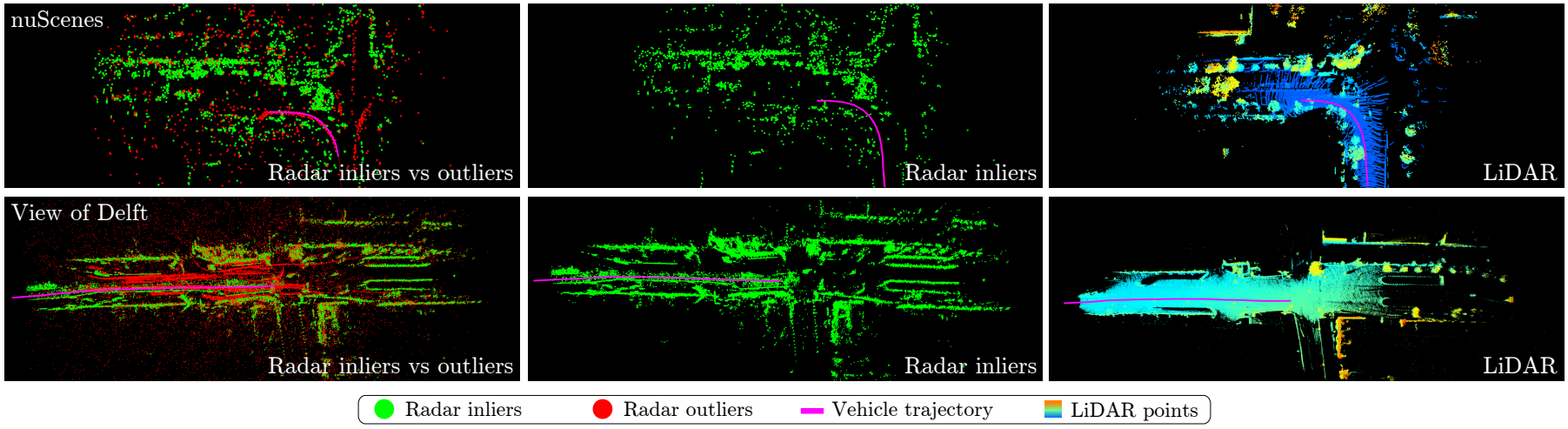


Figure 4.4: Our filtering procedure removes a high amount of dynamic point outliers and radar clutter on nuScenes [21] and View of Delft [139] maps.

4.5 Conclusion

In this chapter, we presented three components that enhance online odometry and mapping accuracy relying solely on automotive radars. Our work encompasses two novel ego-motion estimation approaches and a mapping step, collectively enhancing the capabilities of radar-driven navigation. Our 3D radar method uses classical point-to-point ICP, leveraging the Doppler velocity measurements provided by the radar. We include this information in the ICP objective function, leading to enhancements in trajectory estimation accuracy. Our 2D radar approach exploits a constant velocity filter for scenarios where most of the sensor's field of view is covered by large dynamic objects. It assumes a constant-velocity motion model between two consecutive radar scans to pre-filter the radar point cloud and prevent drifting trajectories due to moving elements in the scene. Moreover, our filtering steps also lead to a highly outlier-free map that preserves the structural elements within the scene. We implemented and evaluated our approach on two datasets that include 3D and 2D automotive radars, and compared it against the state of the art on LiDAR and radar odometry. The experiments suggest that our approach achieves state-of-the-art odometry estimates and demonstrates how simple filtering can lead to improvements in mapping accuracy. Leveraging our odometry techniques, we accurately estimate the pose of an autonomous vehicle navigating an unknown environment solely relying on automotive radar sensors. Furthermore, our map filtering approach helps reduce the noise in radar maps, resulting in cleaner maps for localization.

It should be noted, however, that the evaluations conducted in this chapter are primarily focused on planar motion. A key limitation of these odometry strategies is that automotive radars suffer from reduced vertical resolution, as already mentioned in Chapter 2. This can lead to substantial drift along the vertical direction. In the following chapter, we address this issue by exploring additional information present in radar scans and leveraging the ground plane as an odometry constraint.

Chapter 5

Ground-Aware Radar Odometry

Odometry approaches estimate the pose of an autonomous vehicle over time without relying on external satellite signals or a map. In the previous Chapter 4, we evaluated radar odometry techniques for vehicles mainly moving on a plane. However, real vehicles typically navigate on roads with small slope variations rather than completely flat terrains, also requiring accurate motion estimation in the vertical direction. A key challenge in radar odometry arises from the reduced vertical resolution of 3D automotive radar sensors and the lack of vertical features within radar point clouds. Subsequently, matching radar scans along the vertical axis can degrade odometry accuracy, resulting in substantial vertical drift.

A straightforward approach to handle this limitation is to mount two radar sensors close to each other, as proposed by Huang et al [79]. The two sensors can be oriented vertically and horizontally, mitigating the sparsity and lack of vertical features. Although they achieve accurate odometry in the vertical direction, this solution can lead to bulky sensor setups. As introduced in Section 3.2, researchers in the LiDAR domain constrain the vertical pose estimation problem by leveraging the ground as an additional source of information [98]. Since this issue is also partially present in LiDAR-only odometry, prior ground plane computation approaches by Himmelsbach et al. [71] and Steinhauser et al. [165] rely on heuristics to fit lines and planes to point sets within the LiDAR scan, classifying points as ground or non-ground. In more recent work, Zermas et al. [201] propose to use principal component analysis, and others divide the scan into multiple concentric regions [114, 121, 134], leveraging the LiDAR ring properties for their segmentation method. These methods, however, have been developed for LiDAR sensors, relying on low noise, little multi-path propagation below the ground, and concentric point clouds from spinning laser devices. This is not the case with automotive radar sensors, which output sparse point clouds with high amounts of noise and multi-path propagation.

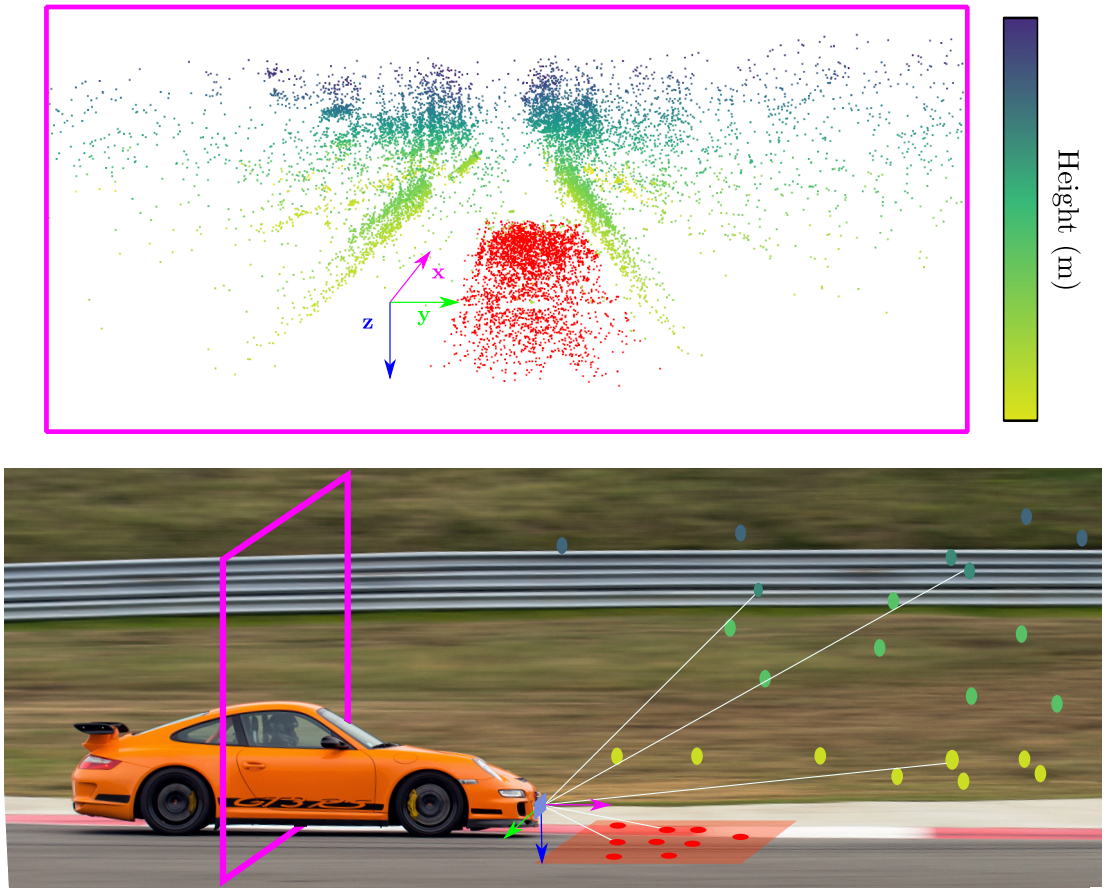


Figure 5.1: Our ground segmentation approach on sparse and noisy 3D radar scans. The top image displays a point cloud with the segmented ground points from the car's perspective. We use this information for vehicle odometry estimation. Below we show a graphic representation of the environment captured by the radar, with the ground points colored in **red** [160].

In our work, we leverage automotive radars for ground segmentation and accurate vertical motion estimation, see Figure 5.1. We propose a heuristic-based technique to extract the ground plane from automotive radar scans. We also introduce a novel point-to-ground-plane residual to reduce drift during ICP scan matching, and leverage a sliding window optimization approach of the poses together with the ground plane, leading to improved vertical motion estimation. We put our odometry to the test using the SJTURadarDataset [117]. Our findings illustrate an improvement in pose accuracy accompanied by a noticeable enhancement of the vertical motion pose estimates.

The main contribution of this chapter is an effective radar odometry approach that exploits the ground plane within automotive radar scans, without relying on any additional sensors. We achieve this by removing points that do not belong to a feasible ground region, leveraging the RCS information provided by the radar, extracting the ground plane from the scans, and matching it across consecutive frames in a point-to-ground-plane manner. We also use sliding window-based

optimization within a factor graph, including a ground plane node, which leads to an improvement in our radar-only odometry results. In sum, we make three key claims. Our approach: (i) achieves state-of-the-art performance in automotive radar odometry; (ii) successfully exploits the RCS feature of automotive radars to estimate the ground plane from single radar scans; (iii) improves odometry accuracy by leveraging the ground plane through scan-to-map matching during scan registration, and across multiple scans via factor graph optimization.

This work has been performed in close collaboration with Franz Kaschner. My main contribution regards the novel design of the RCS ground filtering step in Section 5.1.1 and ground plane matching in Section 5.1.2. His contribution lies in the ground plane pose graph optimization discussed in Section 5.1.3 and implementation details. We report it here for the sake of completeness.

5.1 Our Approach to Ground-Aware Radar Odometry

Our technique aims to achieve accurate radar odometry with improved estimates of the vehicle's vertical motion. The process involves three steps. First, we exploit the radar properties to estimate the ground plane within a single scan, Section 5.1.1. Then, we match the ground plane between two consecutive radar scans during ICP optimization, Section 5.1.2. We also leverage the ground as an additional constraint during factor graph optimization, where the measured ground plane is compared against a global ground, Section 5.1.3. In our approach, we assume typical driving scenarios with small changes in elevation, imposing a softer constraint that is less restrictive than a complete flat-world assumption. Since most measurements and velocities within this chapter are expressed in the car frame, we omit the left superscript ${}^{C}(\cdot)$ for better readability.

5.1.1 Ground Plane Segmentation and Detection

The sparse nature of radar point clouds poses challenges in extracting meaningful features from radar scans. However, the combination of the mounting position and the radar field of view can return multiple points belonging to the ground plane. Although LiDAR ground plane extraction algorithms by Lim et al. [121] and Pan et al. [140] exist, these methods often do not account for the characteristics of radar scans, which are noisier, sparser, and contain fewer ground points than LiDAR scans. We propose a simple strategy that exploits the properties of the measured ground points, segmenting them from the rest and extracting the parameters of the ground plane at the current time, as illustrated in Figure 5.2. Given a radar point cloud $\mathcal{P}_t$ at timestep t, the goal of ground plane segmentation

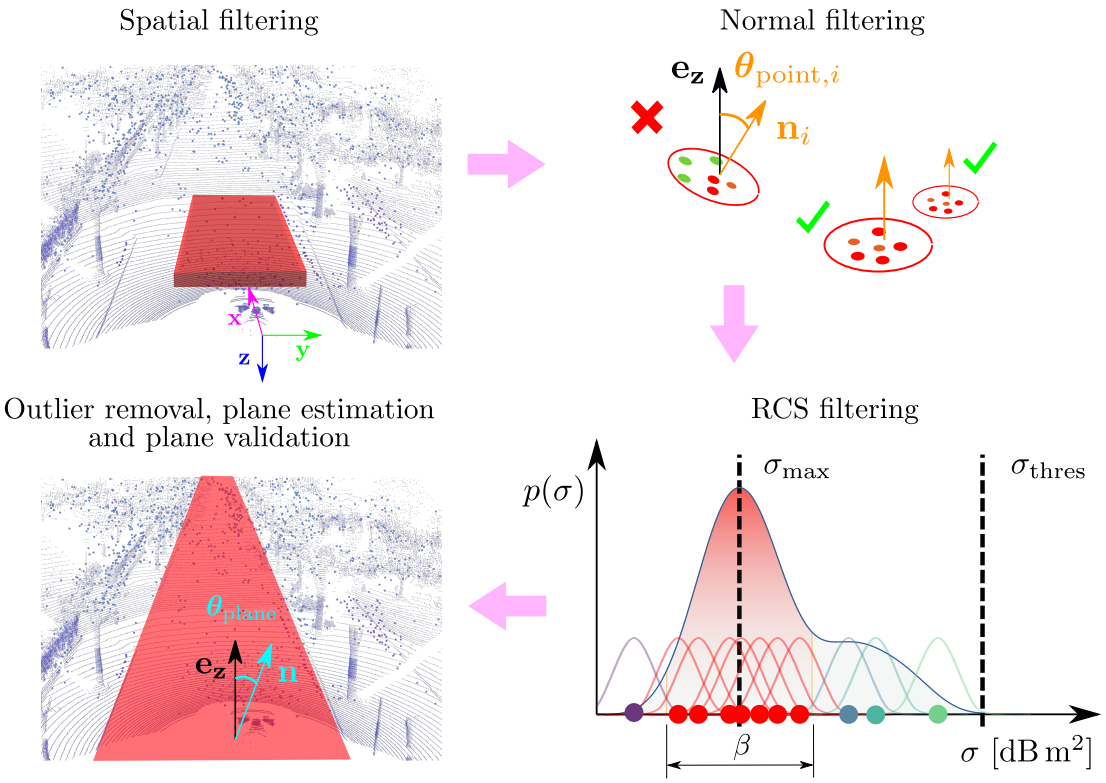


Figure 5.2: Steps of our ground plane segmentation and estimation. First, spatial filtering discards points outside of predefined physical boundaries. Then, a normal filtering step keeps individual points that form a horizontal plane with its neighbors, followed by an RCS filter that removes points with RCS values outside of a defined range β. Finally, outliers not belonging to the ground plane are discarded, and the ground plane is estimated and validated.

is to obtain a set of points $\mathcal{G}_t$ that correspond to ground measurements, which we can leverage to estimate a ground plane $\boldsymbol{\pi}_t = [\mathbf{n}_t, d_t]^\top$, defined by its normal $\mathbf{n}_t \in \mathbb{R}^3$, with $\|\mathbf{n}_t\| = 1$, and the distance to the origin $d_t \in \mathbb{R}$. Our approach consists of five main steps. During spatial filtering, Section 5.1.1.1, points that do not belong to a feasible ground region are discarded. The points in the region are then checked for verticality with their neighbors in Section 5.1.1.2. The remaining points are then checked for their RCS value, and only measurements within a specific range are kept, see Section 5.1.1.3. Finally, we estimate the ground plane using RANSAC in Section 5.1.1.4, and validate it under the assumption that its normal is within a feasible range in Section 5.1.1.5.

5.1.1.1 Spatial Filtering

We perform spatial filtering based on the sensor mounting height h_{sensor} and discard points outside a predefined feasible region where the ground plane can be detected. The LiDAR approach by Zermas et al. [201] relies on the assumption that the points with minimum height within the scan most likely belong to the ground. Furthermore, Patchwork, by Lim et al. [121], considers scans that

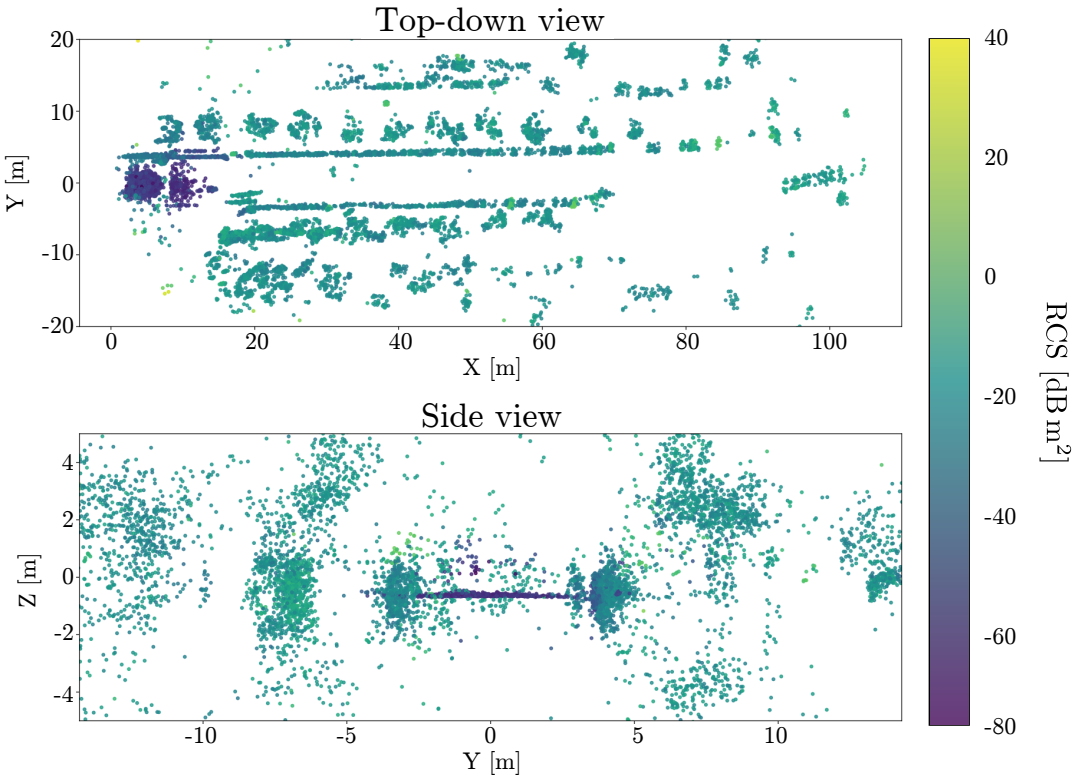


Figure 5.3: Aggregated scans from the SJTURadarDataset [117] colored by their RCS values. The ground points are flat and have specific RCS values compared to other radar targets.

provide a full 360-degree field-of-view. However, radar scans often contain reflections below the ground level due to multi-path propagation, with most points concentrated in front of the sensor. Thus, the lowest points cannot be used for estimating the ground plane. To capture ground points, we propose to use a bounding box that extends within a range $[d_{\text{min},x}, d_{\text{max},x}]$ to the front in the x-direction, a symmetric range to the sides in the y-direction $[-d_y, d_y]$ and a height tolerance range in the vertical direction $[-h_{\text{sensor}} - d_z, -h_{\text{sensor}} + d_z]$. Using the logical AND operator $\wedge$, the resulting set of points $\mathcal{G}_t'''$ is defined by

$$\mathcal{G}_t''' = \{\mathbf{p} \in \mathcal{P}_t \mid x \in [d_{\text{min},x}, d_{\text{max},x}] \ \wedge \ |y| \leq d_y \ \wedge \ |z + h_{\text{sensor}}| \leq d_z\}\,. \tag{5.1}$$

5.1.1.2 Normal Estimation and Filtering

To remove measurements corresponding to noise, clutter, or surrounding objects, we compute the normal vector of the feasible ground points $\mathcal{G}_t'''$. We estimate the normal $\mathbf{n}_i$ of each point by computing the covariance of all neighbors within a radius and performing principal component analysis. A point is retained if the angle $\theta_{\text{point},i}$ between the normal and the z-axis is below a threshold $\theta_{\text{point,max}}$. Given the unit vector in the vertical direction $\mathbf{e}_z$, we keep points that fulfill

$$\mathcal{G}_t'' = \left\{\mathbf{p}_i \in \mathcal{G}_t''' \;\middle|\; \mathbf{n}_i^\top \mathbf{e}_z > \cos(\theta_{\text{point,max}})\right\}. \tag{5.2}$$

5.1.1.3 RCS Filtering

After finding the points belonging to the horizontal plane, we leverage the radar cross section (RCS) property measured by the sensor. As described in the radar sensor fundamentals, Section 2.1, the RCS is a feature measured by automotive radars for each point that represents the reflectivity properties of the measured target by the radar. It depends on various properties of the target, including its material, the incidence and reflection angle, and the size of the target. During our experiments, we observed that the RCS of the ground differs from the RCS of other objects, as shown in Figure 5.3. Consequently, we refine the set of potential ground points $\mathcal{G}_t''$ by leveraging the distribution of radar cross section values.

To achieve this, given that each point in $\mathcal{G}_t''$ has an associated RCS value σ_i, we approximate the underlying RCS distribution in a continuous manner with a kernel density estimate. Specifically, we build a continuous RCS probability distribution $p(\sigma)$ using $G = |\mathcal{G}_t''|$ Gaussian kernels, each centered at σ_i following

$$p(\sigma) = \frac{1}{G} \sum_{i=1}^{G} \frac{1}{\sqrt{2\pi}\nu} \exp\left(-\frac{(\sigma - \sigma_i)^2}{2\nu^2}\right), \tag{5.3}$$

where the standard deviation ν is determined using Scott's rule [158] for one-dimensional data, i.e., $\nu = N^{-\frac{1}{5}}$. Then, we determine the maximum RCS value from the probability density function as

$$\sigma_{\max} = \underset{\sigma}{\operatorname{argmax}}\ p(\sigma), \tag{5.4}$$

and remove all points outside of a manually set range β. Additionally, we keep the points where the RCS value of a point is below a threshold σ_{thres}, resulting in the set

$$\mathcal{G}_t' = \left\{ \mathbf{p}_i \in \mathcal{G}_t'' \;\middle|\; \left|\sigma_i - \sigma_{\max}\right| \leq \beta \ \wedge\ \sigma_i < \sigma_{\text{thres}} \right\}. \tag{5.5}$$

5.1.1.4 RANSAC-Based Filtering and Plane Estimation

We leverage RANSAC [52] to estimate the final ground plane parameters, including the normal $\mathbf{n}_t$ and distance to the origin d_t. We discard all remaining points that are over a certain distance γ from the estimated plane, resulting in the final set of ground points $\mathcal{G}_t$ defined as

$$\mathcal{G}_t = \left\{ \mathbf{p} \in \mathcal{G}_t' \;\middle|\; \left|\mathbf{n}_t^\top \mathbf{p} + d_t\right| \leq \gamma \right\}. \tag{5.6}$$

5.1.1.5 Ground Plane Validation

The final step of our approach is to verify that the estimated ground plane normal remains within a feasible range with respect to the vertical axis. Due to the

vehicle's pitch and roll angles as well as the road's slope, small deviations from the vertical direction are possible. We define the ground plane as valid if the angle θ_{plane} between its normal $\mathbf{n}_t$ and the unit vector in z-direction $\mathbf{e}_z$ is below a threshold $\theta_{\text{plane,max}}$, such that

$$\mathbf{n}_t^\top \mathbf{e}_z > \cos(\theta_{\text{plane,max}}). \tag{5.7}$$

When this condition is not met, we determine that the ground plane cannot be successfully estimated and exclude it from scan registration. This may occur due to a low number of measurements or steep slopes. Note that while the normal filtering step described above applies the verticality check to each point, here the final ground plane parameters are being verified.

5.1.2 Exploiting the Ground Plane During ICP

Rather than relying on the flat world assumption during scan matching, we leverage the extracted ground plane for ICP optimization. Our ground-based scan matching module combines a point-to-point error metric with a point-to-ground-plane error to find the transformation that aligns the current radar scan to the previous radar scans in the local map. For point-to-point ICP, we adopt the position residual from Equation (4.1), where the goal is to obtain the transformation $\mathsf{T} \in SE(3)$ that minimizes the distance between a source point set $\mathbf{p} \in \mathcal{P}$, and a target point set $\mathbf{q} \in \mathcal{Q}$. Our point-to-ground-plane ICP error function E_{g} minimizes the distance between a ground point $\mathbf{p}_g \in \mathcal{G}_t$ from the current scan, and the ground plane of the previous scan $\boldsymbol{\pi}_{t-1} = [\mathbf{n}_{t-1}, d_{t-1}]^\top$ according to

$$E_{\text{g}}(\mathsf{T}) = \sum_{\mathbf{p}_g \in \mathcal{G}_t} \rho\left(||\mathbf{n}_{t-1}^\top (\mathsf{T}\mathbf{p}_g) + d_{t-1}||\right), \tag{5.8}$$

with the Geman McClure kernel ρ to potentially reduce the effect of outliers as employed by Vizzo et al. [172].

For the point-to-ground-plane residual, the Jacobian $\mathbf{J}_g \in \mathbb{R}^{1\times 6}$ is defined as

$$\mathbf{J}_g = \begin{bmatrix} \mathbf{n}_{t-1}^\top & -\mathbf{n}_{t-1}^\top (\mathsf{T}\mathbf{p}_g)^\wedge \end{bmatrix}. \tag{5.9}$$

The final optimization problem, which combines the positional and point-to-ground-plane error functions, is similar to Equation (4.8). We define it with a hand-tuned weight $\alpha \in [0, 1]$ that measures the influence of each residual following

$$\mathsf{T}^* = \underset{\mathsf{T}}{\operatorname{argmin}} \Big[(1-\alpha) E_t(\mathsf{T}) + \alpha E_g(\mathsf{T}) \Big]. \tag{5.10}$$

The final Jacobian $\mathbf{J}_{t,g_k} \in \mathbb{R}^{4\times 6}$ that combines the point-to-point residual of point $\mathbf{p}_k$ and the point-to-ground residual of a ground point $\mathbf{p}_g$ is defined as

$$\mathbf{J}_{t,g_k} = \begin{bmatrix} \mathbf{I}_{3\times 3} & -(\mathsf{T}\mathbf{p}_k)^\wedge \\ \mathbf{n}_{t-1}^\top & -\mathbf{n}_{t-1}^\top (\mathsf{T}\mathbf{p}_g)^\wedge \end{bmatrix}. \tag{5.11}$$

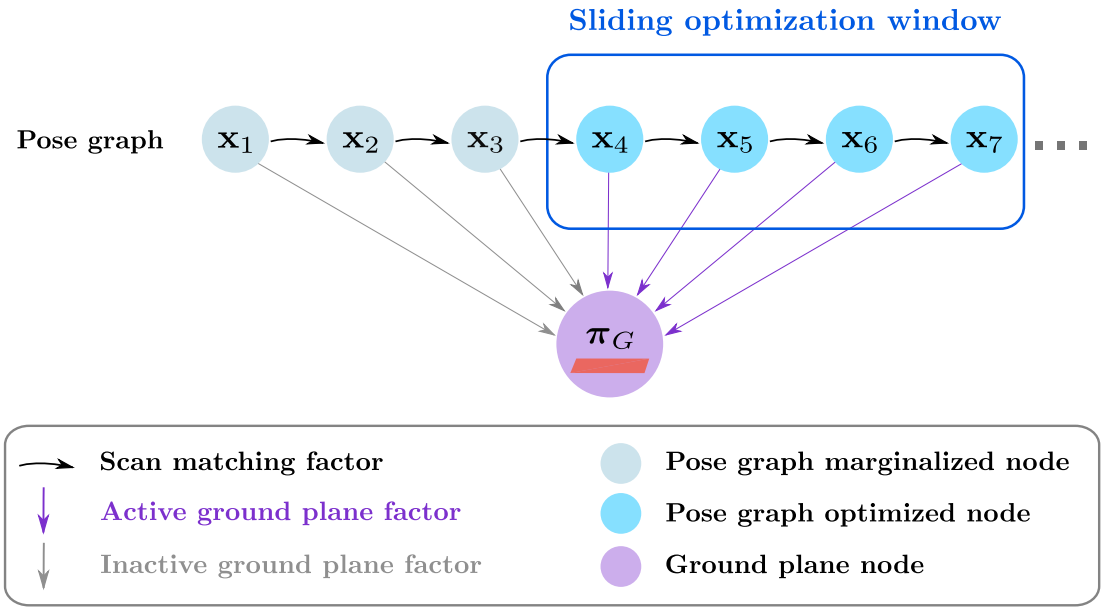


Figure 5.4: Factor graph containing odometry and ground constraints. Each pose within an optimization window has a ground plane that is matched with a global ground plane $\boldsymbol{\pi}_G$.

5.1.3 Ground Plane Pose Graph Optimization

The previous section explained how adding ground information to the ICP algorithm helps reduce the error between the ground points $\mathcal{G}_t$ in the current scan and the ground plane $\boldsymbol{\pi}_{t-1}$ in the previous scan. We now add an additional constraint to our system that considers a global ground plane over an entire sequence, similar to the LiDAR approach by Koide et al. [98]. This acts as a dampening effect of drifting vertical motion over time. To achieve this, we create a factor graph, shown in Figure 5.4, with local constraints between the pose estimations using odometry factors, and a global constraint between the estimated ground plane and a global horizontal ground plane node. We follow the factor graph formulation introduced previously in Section 2.5.4.1 on page 35.

The odometry factor residual is dependent on the states at the previous and the current timesteps, $\mathbf{x}_{t-1}, \mathbf{x}_t \in \mathbb{R}^6$. It is defined as

$$f_{\text{odom}}(\mathbf{x}_{t-1}, \mathbf{x}_t; \mathsf{T}_C^I \mathsf{T}), \tag{5.12}$$

where $\mathsf{T}_C^I \in SE(3)$ represents the estimated transformation from the inertial frame to the previous pose of the vehicle, and T, which was defined previously, expresses the relative transformation between the previous and the current pose.

Regarding the plane factors, we assume that the trajectory has no significant variations in slope within an optimization window. This assumption is reasonable for typical driving scenarios such as highways and rural roads. Consequently, the normal of the global ground plane coincides with the vertical z-direction ${}^I\mathbf{e}_z$ and

the distance to the origin is the sensor mounting height h_{sensor}, such that

$$ {}^{I}\boldsymbol{\pi}_G = [{}^{I}\mathbf{n}_G, d_G]^\top = [{}^{I}\mathbf{e}_z, h_{\text{sensor}}]^\top. \tag{5.13} $$

To compare the estimated ground plane with the global ground, we first express the local ground plane $\boldsymbol{\pi}_t$ with respect to the inertial frame following

$$ {}^{I}\boldsymbol{\pi}_t = (\mathsf{T}_I^C)^\top \boldsymbol{\pi}_t. \tag{5.14} $$

Given the minimal plane parameterization introduced by Ma et al. [127] that includes the azimuth and elevation of the normal, and its distance to the origin, which is defined as

$$ \boldsymbol{\tau}(\boldsymbol{\pi}) = \left[\text{atan2}\left(n_y, n_x\right), \text{atan2}\left(\ n_z, \sqrt{n_x^2 + n_y^2}\right), d\right]^\top, \tag{5.15} $$

we can compute the factor between the global ground plane ${}^{I}\boldsymbol{\pi}_G$ and the local ground plane observation ${}^{I}\boldsymbol{\pi}_t$ expressed in the inertial frame following

$$ f_{\text{ground}}(\mathbf{x}_t; \boldsymbol{\tau}({}^{I}\boldsymbol{\pi}_G) - \boldsymbol{\tau}({}^{I}\boldsymbol{\pi}_t)). \tag{5.16} $$

We leverage window-based optimization considering the last N scans to optimize the factor graph. The final factor graph cost function is given as

$$ \mathbf{x}^* = \underset{\mathbf{x}}{\text{argmin}} \left[\sum_{i=t-N}^{t} \left\| f_{\text{odom}}(\mathbf{x}_{i-1}, \mathbf{x}_i; \mathsf{T}_C^I \mathsf{T}) \right\|_{\Sigma_{\text{odom}_i}} + \right. \tag{5.17} $$

$$ \left. \sum_{i=t-N}^{t} \left\| f_{\text{ground}}\big(\mathbf{x}_i;\ \boldsymbol{\tau}({}^{I}\boldsymbol{\pi}_G) - \boldsymbol{\tau}({}^{I}\boldsymbol{\pi}_i)\big) \right\|_{\Sigma_{\text{ground}_i}} \right], \tag{5.18} $$

with their corresponding odometry and ground plane factor covariance matrices, Σ_{odom_i} and Σ_{ground_i}, respectively. In cases of an invalid ground plane estimation at step i, the ground plane factor is skipped, and only the odometry factor is added to the factor graph.

5.2 Implementation Details

The chosen parameters for the evaluation on the SJTURadarDataset [117] are listed in Table 5.1. We tune the spatial filtering parameters empirically by visualizing the radar scans and finding the area most likely to have ground points. Similarly, the RCS filter is also set empirically by observing the RCS values of the ground points. Regarding the verticality conditions $\theta_{\text{point,max}}$ and $\theta_{\text{plane,max}}$, we restrict our approach to locations with slight slope variations. Moreover, we assign a high value to the ICP weighting parameter α as a way of compensating for the small amount of ground points present within a single radar scan.

Table 5.1: Hyperparameters of our method for the SJTURadarDataset [117].

Spatial filtering	sensor height	h_{sensor}	0.663 m
	minimum x-value	$d_{\text{min},x}$	0.5 m
	maximum x-value	$d_{\text{max},x}$	9.0 m
	y-range	d_y	2.0 m
	z-range	d_z	0.2 m
Normal filtering	point vertical threshold	$\theta_{\text{point,max}}$	15.0°
RCS filtering	threshold	σ_{thres}	-35.0 dB m^2
	range	β	20.0 dB m^2
Outlier removal	RANSAC threshold	γ	0.02 m
Ground plane validation	plane vertical threshold	$\theta_{\text{plane,max}}$	1.0°
Graph optimization	sliding window size	N	20
Ground-ICP	weighting factor	α	0.993

5.3 Experimental Evaluation

This chapter focuses on developing an automotive radar odometry method that exploits the ground plane information as an additional feature for pose estimation. We present our experiments to show the capabilities of our method. The results support our key claims that our approach: (i) achieves state-of-the-art performance in automotive radar odometry; (ii) successfully exploits the RCS feature of automotive radars to estimate the ground plane from single radar scans; and (iii) improves odometry accuracy by leveraging the ground plane through scan-to-map matching during scan registration, and across multiple scans via factor graph optimization.

5.3.1 Experimental Setup

For the evaluation of our approach, we run experiments on the publicly available SJTURadarDataset [117]. Its 3D radar sensor is mounted on the front bumper of the car, where part of the output points belong to the ground. In our evaluation, rather than measuring the ground segmentation accuracy, we focus on the relative errors from the downstream odometry task. In our experiments, we first compare our approach against the state-of-the-art radar and LiDAR odometry, defining our reference values as the RTK-GPS measurements provided by the dataset. We also show qualitative results of the estimated trajectories and the segmented ground planes within each sequence. Our second experiment performs an ab-

Table 5.2: Our radar odometry approach achieves state-of-the-art pose estimation accuracy in the SJTURadarDataset [117] dataset. t_{KITTI} in [%], r_{KITTI} in [°/m], and t_{APE} in [m].

	Campus 1			**Campus 2**			**Campus 3**			**Campus 4**		
	t_{KITTI}	r_{KITTI}	t_{APE}	t_{KITTI}	r_{KITTI}	t_{APE}	t_{KITTI}	r_{KITTI}	t_{APE}	t_{KITTI}	r_{KITTI}	t_{APE}
LeGO-LOAM [162]	1.96	0.020	2.27	3.19	0.012	2.57	2.68	0.010	3.33	-	-	-
4DRadarSLAM [204]	16.0	0.103	60.5	39.1	0.115	351.5	36.0	0.096	205.7	42.0	0.127	949.3
SLAM4D [117]	2.32	0.021	2.28	3.13	0.020	**3.79**	3.06	0.024	3.83	-	-	-
SLAM4D (Odom.) [117]	2.52	0.025	4.14	3.84	0.019	10.2	3.29	0.027	9.08	-	-	-
Radar-ICP (Chap. 4)	1.56	**0.013**	6.50	2.80	0.012	40.0	2.86	0.017	16.9	8.91	0.041	118.3
KISS-ICP (Radar) [172]	1.55	0.014	3.71	2.00	0.010	11.6	2.06	**0.013**	14.7	6.95	**0.039**	71.6
Ours	**1.52**	0.014	**1.78**	**1.92**	**0.009**	7.29	**1.91**	**0.013**	**1.69**	**6.86**	**0.039**	**13.4**

lation study of our system, demonstrating how each component contributes to the final odometry estimation result and to the runtime. We employ the metrics introduced in Section 2.5.5, and, contrary to Chapter 4, we include the vertical motion in the evaluation results. Since we are using the 4DRadarSLAM results directly from their paper [117], we present the KITTI metric evaluation results, along with the absolute position errors introduced in Section 2.5.5.

5.3.2 Comparison With the State of the Art

The first experiment evaluates the performance of our method and demonstrates that it achieves state-of-the-art results in automotive radar odometry comparable to LiDAR approaches. We compare our method against the LiDAR feature matching odometry method LeGO-LOAM [162] applied to LiDAR scans, the point-to-point matching technique KISS-ICP [172] on radar point clouds, two radar SLAM frameworks 4DRadarSLAM by Zhang et al. [204] and SLAM4D by Li et al. [117] with and without loop closure, and our radar odometry approach Radar-ICP [26] introduced in Chapter 4. For LeGO-LOAM [162] and SLAM4D [117], we get the results directly from Li et al. [117]. Since they only provide the KITTI relative error metrics, we measure the relative errors using the relative translation, t_{KITTI}, and rotation, r_{KITTI}, KITTI metric as well as the absolute trajectory error, t_{APE}, introduced in Section 2.5.5. The experimental evaluation is shown in Table 5.2 with the best results in **bold**. We also show qualitative results of the ground plane segmentation in Figure 5.5.

Our approach outperforms other methods in terms of t_{APE} for most sequences. Likewise, our method shows comparable performance to using a 3D LiDAR sensor in most scenarios. Note, however, that our previous approach from Chapter 4 decreases performance despite its similarities with KISS-ICP [172]. Its main limitation is the reliance on the vehicle-to-radar calibration parameters, which are not available in the SJTURadarDataset [117]. We, therefore, attempt to manually set the parameters for the evaluation, leading to pose inaccuracies.

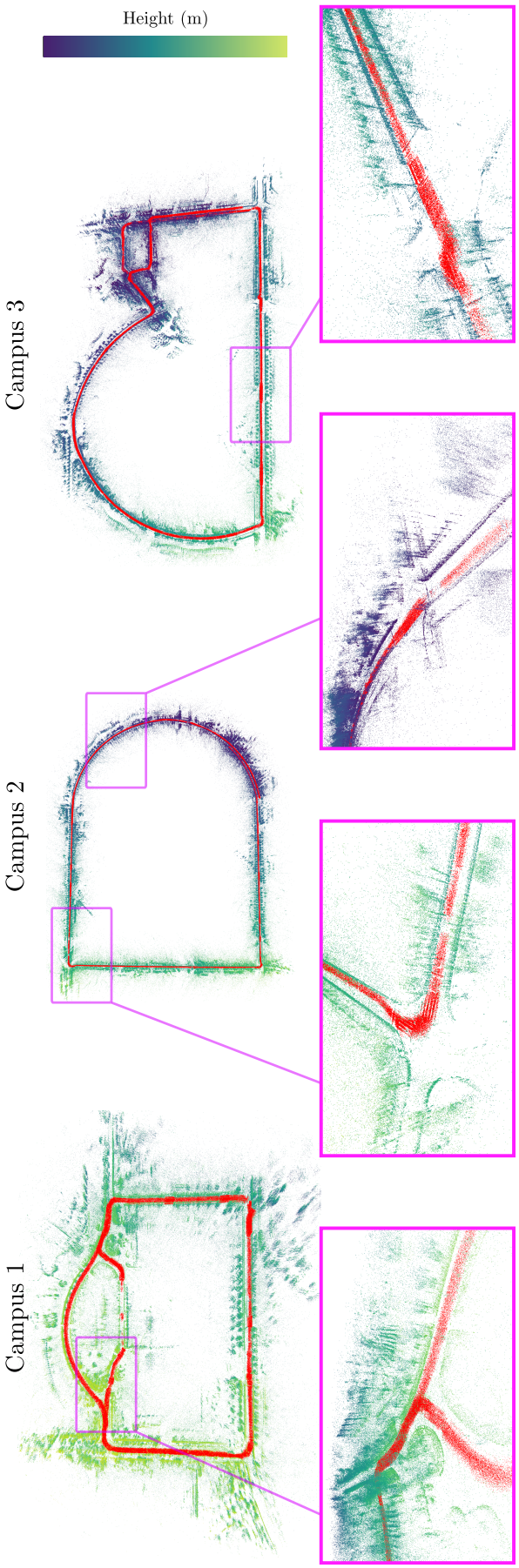


Figure 5.5: Our ground segmentation approach successfully identifies the ground points, shown in **red**, in the SJTURadarDataset [117].

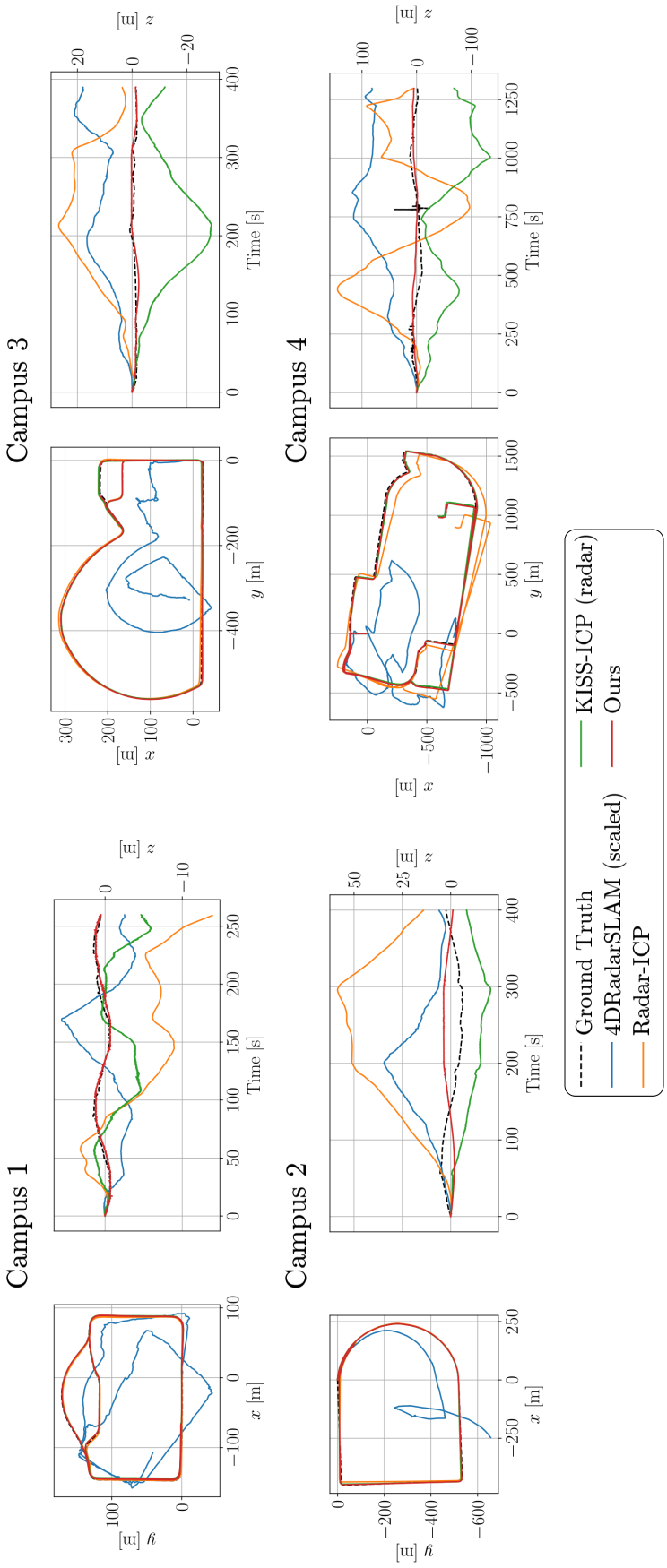


Figure 5.6: Our approach achieves state-of-the-art results in radar pose estimation with low drift in the vertical axis in the SJTURadarDataset [117].

Table 5.3: Comparison of the influence of different components of our ground-based ICP contributing to the final accuracy in SJTURadarDataset [117]. t_{KITTI} in [%] and t_{APE} in [m].

	Campus 1		**Campus 2**		**Campus 3**	
	t_{KITTI} [%]	t_{APE} [m]	t_{KITTI} [%]	t_{APE} [m]	t_{KITTI} [%]	t_{APE} [m]
P2P-ICP	1.55	3.71	2.00	11.6	2.06	14.7
P2P-ICP with $z = 0$	1.81	1.97	2.27	7.83	3.32	3.11
Ground-ICP	**1.52**	3.09	2.11	8.70	2.07	8.22
P2P-ICP + Graph	**1.52**	**1.75**	1.95	7.62	1.92	3.31
Ground-ICP + Graph	**1.52**	1.78	**1.92**	**7.29**	**1.91**	**1.69**

Table 5.4: Comparison of the runtimes of our radar odometry system in different configurations.

	P2P-ICP	Ground-ICP	Ground-ICP + Graph
Scan matching	17.1 ms	16.5 ms	16.5 ms
Ground plane estimation	–	1.4 ms	1.4 ms
Pose graph construction & optimization	–	–	0.7 ms
Total	17.1 ms	17.9 ms	18.6 ms

5.3.3 Ablation Studies

The second experiment evaluates how each component of our system contributes to the final accuracy. We perform the evaluation on the sequences "Campus 1", "Campus 2", and "Campus 3", as "Campus 4" contains jumps in the reference trajectory. The results are presented in Table 5.3. The baseline method, point-to-point ICP (P2P-ICP), is based on KISS-ICP [172] applied to radar data. We also provide evaluation results when the poses are restricted to the horizontal plane (P2P-ICP with z = 0). When we add ground plane detection and optimization to ICP, we refer to this as "Ground-ICP". If we include the ground plane as a node in the graph for P2P-ICP, this method is labeled as "P2P-ICP + Graph". Combining the "Ground-ICP" with the ground plane as an additional factor graph node is termed "Ground-ICP + Graph". We observe that removing the z-component from the poses leads to the worst result, since this does not consider vertical deviations in areas where the ground is not perfectly planar. The approach that combines "Ground-ICP + Graph" results in the highest average accuracy, showing how each component contributes to the final accuracy. We additionally measure the impact on the runtime of each component in Table 5.4, showing that our ground detection and graph optimization modules lead to minimal runtime increases.

5.4 Conclusion

In this chapter, we presented a novel approach to estimate the ego-motion of a vehicle by leveraging the ground plane using radars. By performing point-to-ground-plane matching with the ground and integrating the ground information into a factor graph, we relax the assumption of a flat ground and constrain the movement of the vehicle in the vertical direction. Our approach consists of multiple filtering steps that exploit the properties of radar sensors to segment ground points from the rest of the scan. These filters include spatial, geometric, and RCS-based techniques that exploit the characteristics of the sensor. After extracting the ground-plane parameters, we match the detected ground plane over multiple scans in a point-to-plane manner and integrate its information into a global factor graph, leading to a reduced drift in the z-axis. We implemented and evaluated our approach on a real-world dataset and provided comparisons to existing techniques that do not exploit ground information, supporting all claims made in this chapter. The experiments suggest that our ground-aware odometry approach enhances pose estimation performance, achieving LiDAR-level accuracy with a single automotive radar sensor.

Our odometry, however, estimates the pose of the vehicle mainly relying on the most recent radar measurements. This leads to accumulating drift over time, which can reduce trajectory accuracy and produce inconsistencies in the constructed radar map. Instead, we can leverage information collected over the whole trajectory to recognize previously visited places and detect loop closures, a technique we explore in the following chapter.

Chapter 6

Radar Place Recognition

Identifying previously visited places is a fundamental aspect of autonomous vehicle localization, as it enables the temporal association between measurements collected at different times. This procedure is known as place recognition, and it can be exploited in two distinctive ways explored later in Chapter 7 and Chapter 8. First, it enables SLAM to detect loops and perform loop closure optimization, reducing trajectory drift. Second, it extends the mapping capabilities of the SLAM system to operate over long periods of time, enabling the construction of radar maps across multiple mapping sessions.

During place recognition, every new radar measurement is encoded into a descriptor representation and stored within a database, as shown in Figure 6.1. These descriptors can be efficiently compared using a distance metric, such as the $L2$ norm, to identify similar radar scans that belong to the same location. The main challenge is to achieve a compact encoding that reliably captures the information within radar scans, resulting in descriptors with high similarity for coinciding locations and high dissimilarity for non-coinciding locations.

As outlined in Section 3.3, existing sensor-based place recognition techniques beyond GNSS mainly rely on cameras and LiDAR sensors [8, 94]. However, relatively few strategies focus specifically on automotive radars. The high amounts of noise and clutter in radar scans hinder the extraction of valuable features for place recognition, and the low number of points per scan makes it hard to identify objects and structural elements in the environment. This poses a challenge when adapting heuristic-based LiDAR approaches such as ScanContext [94] and its variations [48, 85, 93, 179], which rely on handcrafted features that may not always be present in the scenes. In contrast, neural network-based methods can learn patterns in the environment, leading to improved place recognition accuracy [34, 185]. One notable approach in learning-based radar place recognition is Autoplace [22], which projects the radar point cloud into a planar image and employs an image encoder to obtain the descriptor. However, this strategy en-

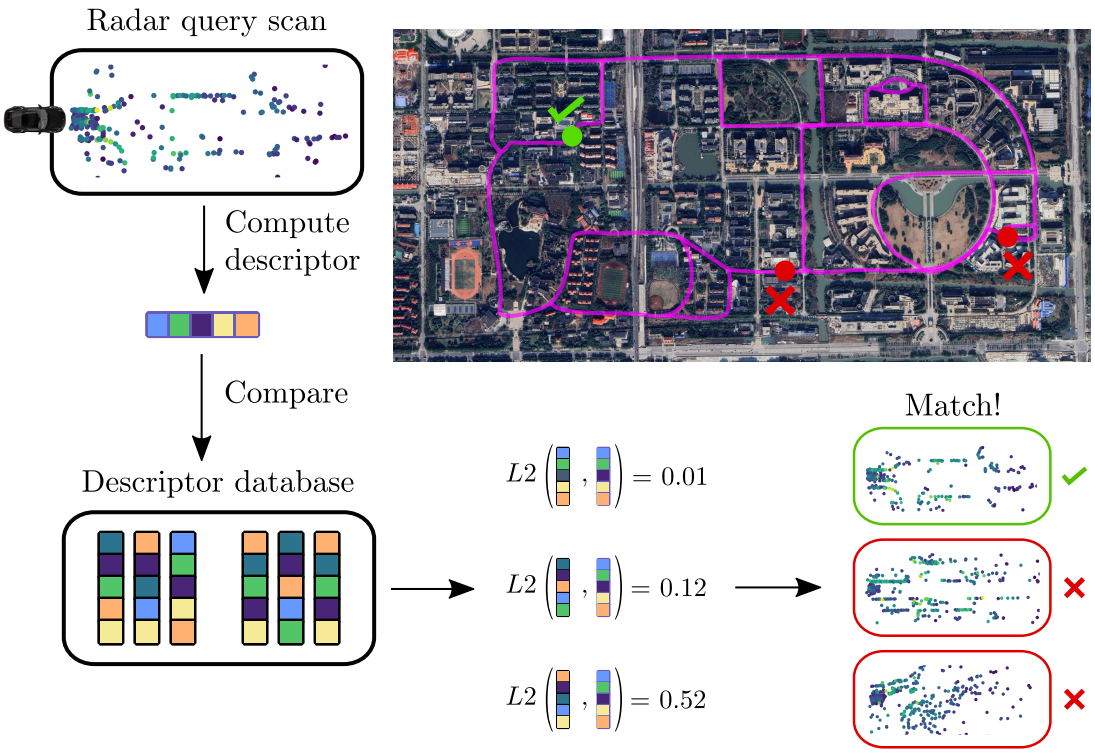


Figure 6.1: Procedure to perform place recognition in radar point clouds. The query scan is encoded by our network and compared to the map database using the $L2$ distance metric. Place recognition is performed by identifying the closest matching descriptor from the database.

codes information of the whole image into a single descriptor vector, resulting in a high-dimensional descriptor with substantial storage requirements. Moreover, the scan-to-image projection in Autoplace does not account for the elevation provided by 3D automotive radars.

In our work, we introduce a neural network specifically designed for automotive radar sensors. Our approach addresses radar sparsity by handling the scan in a pointwise manner and measures the contribution of each point for place recognition with our proposed point importance estimation module. Moreover, we introduce a novel radar cross section module that encodes the RCS information of the scene. The resulting descriptor is compact, yet informative, achieving state-of-the-art place recognition results in public radar datasets.

The main contribution of this chapter is a novel neural network architecture that leverages single radar scans for place recognition. Our model is based on a point encoder that extracts features from the original point clouds, a network that encodes the RCS values, a point scoring module that estimates the importance of a point for place recognition, and a global descriptor extractor for spatial clustering of the global descriptor vectors. Our method achieves state-of-the-art results for 2D and 3D automotive radar place recognition using a comparably smaller scene descriptor. In sum, we make three key claims. Our place recognition

approach: (i) achieves state-of-the-art performance on automotive radar single-scan place recognition while keeping a compact scene representation; (ii) provides a novel procedure to utilize RCS information to describe the scene, improving accuracy; and (iii) enhances feature extraction by estimating the importance of points within the scan for place recognition.

6.1 Our Approach to Radar Place Recognition

Our approach aims to achieve single-scan radar place recognition as a way of extracting meaningful information from sparse and noisy radar point clouds. The process involves comparing scan descriptors stored in a database with current scans, as shown in Figure 6.1. Initially, radar sensors capture the environment during the robot's first pass at a location. We convert each scan into a place descriptor using our encoder from Figure 6.2, which combines geometric and RCS point information from the measured point cloud. Then, we store the descriptor in a map database. During navigation, we query the database with current measurements converted into descriptors using the same encoder-descriptor network from Figure 6.2. This enables us to identify locations with high similarity based on our scoring function. Our approach utilizes the Doppler velocity provided by the radar in order to filter dynamic point outliers. More specifically, we encode the 3D radar point information with the scan encoder and our point importance estimation module. The feature vector is then transformed using NetVLAD to a global descriptor and combined with the output of our RCS network module. The resulting descriptor can then be stored in the database when visiting an unknown environment, or queried from the database for place recognition.

6.1.1 Dynamic Point Pre-Filtering

Before encoding a point cloud into a descriptor, it is crucial to filter out dynamic points, as they can cause the appearance of a location to vary over time. While this presents a significant challenge in LiDAR sensors, one of the key features of automotive radars is that they measure the Doppler velocity of the target. This velocity measurement allows us to differentiate between static and dynamic points, as emphasized by Zeller et al. [198]. We follow Cai et al. [22], to focus solely on the static points of the radar scan for place recognition. The main idea is that static points of the environment should match the ego-vehicle's velocity. Points with different velocities are likely to correspond to moving objects, and are thus considered outliers. We preprocess all the scans following our procedure from Equation (4.12) on page 55, resulting in filtered point clouds containing only static points within the scene.

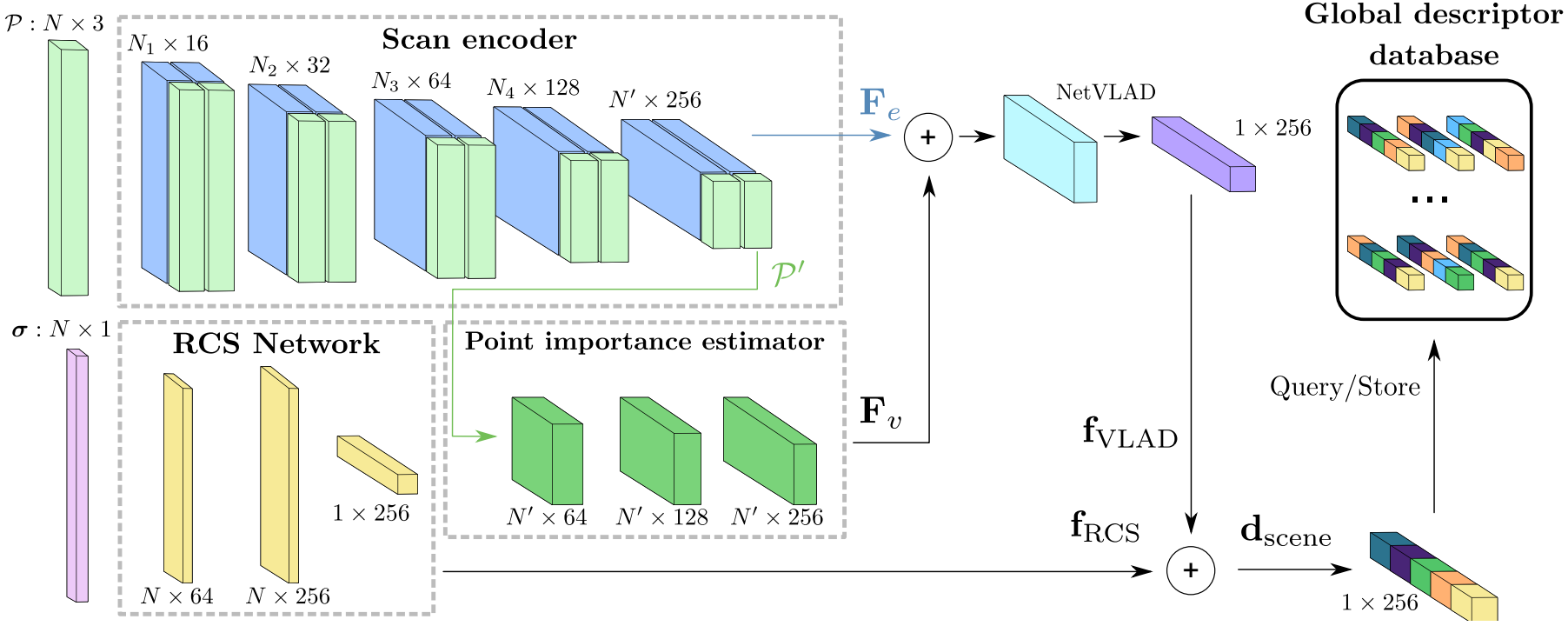


Figure 6.2: The 3D radar scan point information is encoded, resulting in $\mathbf{F}_e \in \mathbb{R}^{N' \times D}$. The individual contribution of each downsampled point for place recognition $\mathbf{F}_v \in \mathbb{R}^{N' \times D}$ is computed with our point importance estimation module. The feature vector is then transformed using NetVLAD to a global descriptor, and combined with the output of our RCS network module, which takes as input the point RCS values, $\boldsymbol{\sigma}$. The resulting descriptor contains geometric and RCS information from the radar scan. It can be stored in the database during the first pass or queried from the database for place recognition.

6.1.2 Scan Encoder

The goal of the scan encoder is to obtain a feature representation $\mathbf{F}_e \in \mathbb{R}^{N' \times D}$ of the filtered static radar point cloud $\mathcal{P}$. In the case of 2D radars, we set the z dimension to be zero. The encoding should contain spatial data descriptive enough for place recognition, which involves capturing contextual information of points at different scales. Autoplace [22] achieves this by projecting its radar point cloud into a 2D image and encoding it with a convolutional neural network. As a consequence, most vertical and geometric information is lost. Similar to KPPR [185], we focus on capturing 3D contextual information of individual points directly from the radar scan using rigid kernel point convolutions (KPConv) [167]. For a point $\mathbf{p}_i^l \in \mathcal{P}^l$ at layer l, the convolution of features $\mathbf{F}^{l-1} \in \mathbb{R}^{N_{l-1} \times D_{l-1}}$ with kernel g^l follows

$$(\mathbf{F}^{l-1} * g^l)(\mathbf{p}_i^l) = \sum_{\mathbf{p}_k^{l-1} \in \mathcal{N}^l(\mathbf{p}^l)} g^l(\mathbf{p}_k^{l-1} - \mathbf{p}_i^l)\, \mathbf{f}_k^{l-1}, \tag{6.1}$$

where $\mathcal{N}^l(\mathbf{p}_i^l) = \{\mathbf{p}_k^{l-1} \mid \left\|\mathbf{p}_k^{l-1} - \mathbf{p}_i^l\right\| < r^l\}$ denotes the neighboring points of $\mathbf{p}_i^l$ within a radius $r^l \in \mathbb{R}$ such that $r^l > 0$, and $\mathbf{f}_k^{l-1}$ represents the k^{th} convolution feature from layer $l-1$.

Our encoder is composed of a sequence of five convolutional and downsampling layers that capture local features $\mathbf{F}^l$ at different levels with different radii. We use grid downsampling at different scales, and contrary to KPConv [167], we find that downsampling before computing convolution helps process contextual information in the points. We also increase the radius of points to consider in $\mathcal{N}^l(\mathbf{p}_i^l)$ in every new convolution-downsampling block for an extended receptive field. Although we also test adding the RCS as an additional channel to the input point cloud, we observe little improvement in the place recognition performance. Instead, we propose an additional network module that accounts for the distribution of RCS values within the scene.

6.1.3 Radar Cross Section Network

To maximize the potential of radar sensors, we propose an additional component that utilizes radar cross section information to enhance the final feature descriptor. The RCS measures how detectable an object is by the radar based on the object's properties and the measurement angle. It provides additional information for each point in $\mathcal{P}$ about the properties of a specific location, making it a valuable attribute for place recognition. However, unlike research in object detection by Palffy et al. [139], our experiments show that adding the RCS as an additional channel minimally enhances place recognition performance.

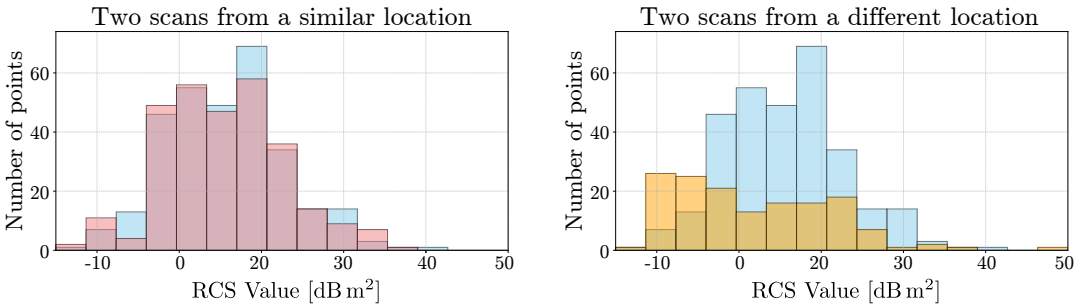


Figure 6.3: Comparison of the RCS distributions at similar locations (left) and different locations (right). Locations that are close to each other have RCS histograms that are similar to each other, while different locations often have RCS distributions that differ substantially.

Instead, following Cai et al. [22], we observe how the distribution of RCS values is similar between coinciding locations and dissimilar if the scans belong to different places, as shown in Figure 6.3. However, rather than comparing the scene RCS as an additional postprocessing step as suggested by Cai et al. [22], we propose an RCS network module that learns the RCS feature representation of the entire radar scan $\mathbf{f}_{\text{RCS}} \in \mathbb{R}^D$ and is permutation invariant. This enhances the global descriptor vector, making it more informative than a descriptor only containing point information.

To capture the RCS information $\boldsymbol{\sigma} \in \mathbb{R}^N$ from the scan, we propose to use a two layer MLP that encodes the RCS value of each point σ_i into a feature encoding $\mathbf{f}_{\sigma_i} \in \mathbb{R}^D$ such that

$$\mathbf{f}_{\sigma_i} = \text{MLP}_r(\sigma_i). \tag{6.2}$$

We then aggregate and normalize the features from all points over the feature dimension, making the result permutation invariant following

$$\mathbf{f}_{\text{RCS}} = \frac{\sum_{i=1}^{N} \mathbf{f}_{\sigma_i}}{||\sum_{i=1}^{N} \mathbf{f}_{\sigma_i}||}. \tag{6.3}$$

The resulting $\mathbf{f}_{\text{RCS}}$ acts as a global descriptor containing the distribution of RCS values for that particular scan. While Autoplace [22] uses an additional stage that performs histogram re-ranking after their network prediction, we integrate our RCS network module inside the descriptor encoder network, avoiding the additional postprocessing step.

6.1.4 Point Importance Estimator

Due to the sparsity and high amount of outliers coming from noise and multi-path reflections in radar point clouds, it is crucial that the place recognition network is able to identify valuable anchor points within the scan. Knowing that radar noise can randomly appear and disappear between single scans, we propose a point

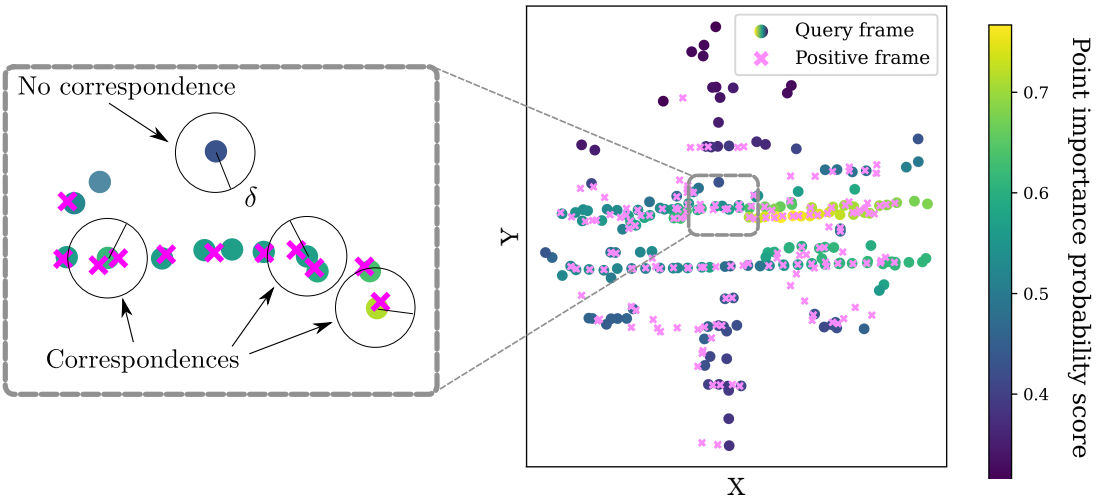


Figure 6.4: Our point importance estimator prioritizes correspondences from persistent elements. (left) During training, those query points that have a correspondence in the positive frame within a radius δ, represented as a circle, are considered valuable for place recognition. Those that do not have a correspondence can be highly inconsistent between scans, thus their predicted importance should be reduced. Although all points are checked, we only display the radius on a small subset for clear visualization. (right) Resulting probabilities of our point importance estimation module. Those areas with higher densities and geometric line patterns are considered more important than random points located far away from the sensor.

importance estimation module to guide the training of the network, focusing on points that are relevant for place recognition. As a result of finding valuable measurements, this module also helps focus on those points that we are more certain that exist, such as high-density locations, patterns in the environment, and the point distribution within the scans. During training, those query points that have a correspondence in the positive frame within a radius δ, represented in Figure 6.4 as a circle, are considered valuable for place recognition. Those that do not have a correspondence can be highly inconsistent between scans, thus their predicted importance should be reduced. As can be observed in Figure 6.4, areas with higher densities and geometric line patterns are considered more important, and thus, have a higher probability of existence, than random points located far away from the sensor.

Using the subsampled scan $\mathcal{P}'$ to estimate the importance of their corresponding encoded local features, we encode the probability of a point $\mathbf{p}'_j \in \mathcal{P}'$ being valuable for place recognition into a feature vector $\mathbf{f}_{v_j} \in \mathbb{R}^D$ using an MLP with three learnable layers, ReLU activations, and layer normalization such that

$$\mathbf{f}_{v_j} = \mathrm{MLP}_v(\mathbf{p}'_j). \tag{6.4}$$

The importance probability of a point is within the range $[0, 1]$. It indicates how valuable the point is for place recognition. It is estimated as

$$P(\mathbf{p}'_j) = \mathrm{Sigmoid}\left(\mathrm{Linear}\left(\mathbf{f}_{v_j}\right)\right), \tag{6.5}$$

where Sigmoid$(\cdot)$ indicates the sigmoid activation function and Linear$(\cdot)$ denotes the linear projection of the feature vector from $\mathbb{R}^D$ to $\mathbb{R}$. We use this scoring later in Equation (6.9) to compute the loss during training between points that only exist in one of the scans that are being compared. The encoded probabilities $\mathbf{f}_v$ from all downsampled points are stacked into the feature vector $\mathbf{F}_v \in \mathbb{R}^{N' \times D}$, and added to the output of the point encoder, resulting in $\mathbf{F}_{e+v} \in \mathbb{R}^{N' \times D}$, which is defined as

$$\mathbf{F}_{e+v} = \mathbf{F}_e + \mathbf{F}_v. \tag{6.6}$$

During our ablation studies, we compare addition and multiplication of the encoded probability features in Equation (6.6), with addition leading to better results. We believe that the low probability of incorrect predictions may lead to the loss of information, which can degrade place recognition performance on automotive radar data.

6.1.5 Global Descriptor Database

The descriptor extraction layer aggregates local features into a single global descriptor vector. We exploit a NetVLAD layer [171] to aggregate the local features $\mathbf{F}_{e+v}$ from Equation (6.6) into C learnable cluster centers, resulting in $\mathbf{f}_{\text{VLAD}} \in \mathbb{R}^D$. These learnable centers represent points calculated from groups of similar local descriptors. As a result, descriptors from coinciding locations will be clustered together. We combine the NetVLAD descriptor with the RCS descriptor, leading to the final global descriptor vector $\mathbf{d}_{\text{scene}} \in \mathbb{R}^D$ following

$$\mathbf{d}_{\text{scene}} = \frac{\mathbf{f}_{\text{VLAD}} + \mathbf{f}_{\text{RCS}}}{\|\mathbf{f}_{\text{VLAD}} + \mathbf{f}_{\text{RCS}}\|}. \tag{6.7}$$

The resulting descriptor represents the location measured by a radar scan. It contains local information obtained from the point encoder and the point importance estimator, as well as global features extracted using the RCS network and NetVLAD. During the map recording, every scan is stored as a descriptor $\mathbf{d}^{\text{m}}_{\text{scene}} \in \mathbb{R}^D$ in a KDTree [15] map database $\mathcal{M} = \{\mathbf{d}^{\text{m}}_{\text{scene}_1}, \mathbf{d}^{\text{m}}_{\text{scene}_2}, ..., \mathbf{d}^{\text{m}}_{\text{scene}_M}\}$. The query scan is also encoded as the feature descriptor vector $\mathbf{d}^{\text{q}}_{\text{scene}}$, and compared to those stored in $\mathcal{M}$ using the $L2$ distance metric. We use superscript "m" to indicate that the feature vector belongs to the map database.

6.1.6 Metric Learning for Place Recognition

The goal of metric learning is to obtain the parameters of our descriptor encoder network that lead to a meaningful and compressed representation of the environment. For a query descriptor $\mathbf{d}^{\text{q}}_{\text{scene}}$, the feature descriptor vector must be similar

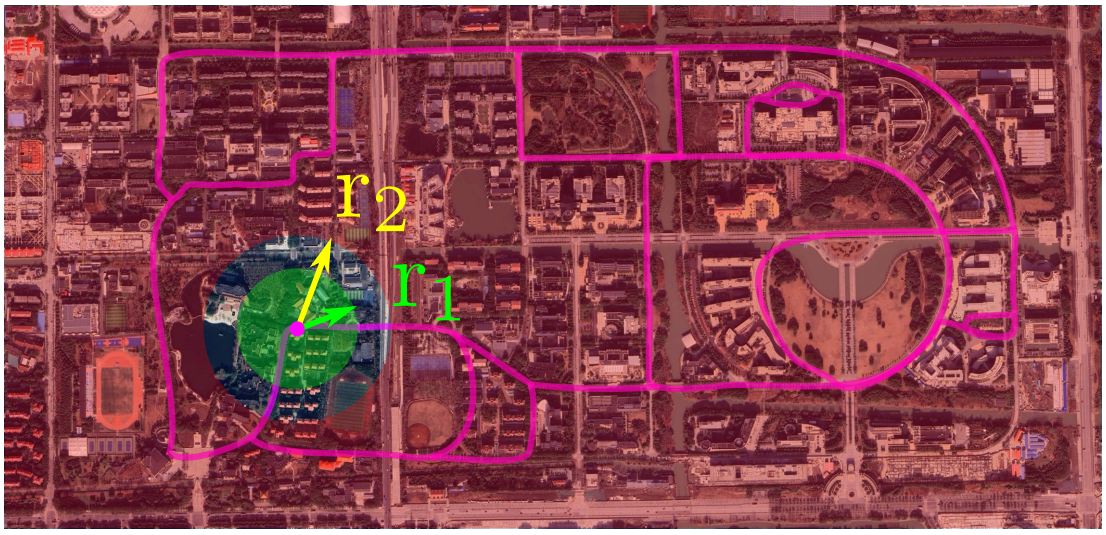


Figure 6.5: GNSS labels training samples as positive or negative based on pose proximity. Samples that are within a radius r_1 are considered positive samples, and those farther away than a radius r_2 are considered negative samples. We leave a margin between r_1 and r_2 to set a higher separation between negative and positive scene descriptors in the feature space.

for observations of the same location, named positive samples $\mathbf{d}^{\text{pos}}_{\text{scene}}$, and dissimilar for observations of different locations, denoted as negative samples $\mathbf{d}^{\text{neg}}_{\text{scene}}$. We define positive and negative samples during training based on GNSS pose information as shown in Figure 6.5. However, during operation, we no longer require GNSS for the operation of our place recognition network.

We define positive samples as those within a radius distance r_1 to the query measured with the GNSS location. Furthermore, due to occlusions or different viewing perspectives, the positive samples taken from a close geographic location may look very different from the queries themselves. Therefore, during training, the selected positive samples are those within r_1 that have the lowest $L2$ distance between descriptors. We also observe an increase in performance when training on multiple positives for the same location, as the network learns that the same location can be measured in multiple ways and from different viewpoints.

Meanwhile, negative samples are farther away from a bigger radius r_2 such that $r_2 > r_1$. However, as the dataset may contain very different scans from different locations, randomly selecting negative samples may lead to low discrimination and generalization capabilities. As proposed by Uy et al. [171], we use a hard negative mining strategy to find the most similar negative sample to the query descriptor $\mathbf{d}^{\text{q}}_{\text{scene}}$ in the feature space. This helps the network learn to discriminate scenes where a wrong database match closely resembles the query.

We leverage the triplet margin loss by Balntas et al. [10] that minimizes the $L2$ distance between the query and a positive sample $d_{\text{pos}} = L2(\mathbf{d}^{\text{q}}_{\text{scene}}, \mathbf{d}^{\text{pos}}_{\text{scene}})$, and maximizes the distance with respect to the $h = 1, \ldots, H$ hardest negative descrip-

tors $d_{\text{neg}_h} = L2(\mathbf{d}^{\,\text{q}}_{\text{scene}}, \mathbf{d}^{\,\text{neg}_h}_{\text{scene}})$. The triplet margin loss $\mathcal{L}_{\text{TM}}(\mathbf{w})$ that optimizes the network parameters $\mathbf{w}$ for a margin α is given as

$$\mathcal{L}_{\text{TM}}(\mathbf{w}) = \sum_{h=1}^{H} \max(d_{\text{pos}} - d_{\text{neg}_h} + \alpha, 0). \tag{6.8}$$

Additionally, to account for the estimated value of the points in our point importance estimator from Section 6.1.4, we propose to measure the nearest point correspondences between the query and its associated positive scan. First, we align both scans by transforming them to GNSS global coordinates. Then, as shown in Figure 6.4, we introduce a radius distance hyperparameter δ that determines whether a given point in the query scan has a correspondence in the positive scan. We consider valuable points for place recognition only those that demonstrate consistency, thus existing in both scans. Note that the GNSS signal is only required for computing the loss during the training procedure, but is not required during operation.

Interpreting this as a binary classification problem, we assign those points in the query scan that have a correspondence in the positive scan a label $\hat{P}(\mathbf{p}'_j) = 1$, and $\hat{P}(\mathbf{p}'_j) = 0$ to those points without a correspondence. Denoting $P_j = P(\mathbf{p}'_j)$ for brevity, the final binary cross-entropy loss is defined as

$$\mathcal{L}_{\text{BCE}}(\mathbf{w}) = -\sum_{j=1}^{N'} P_j \log(\hat{P}_j) + (1 - P_j) \log(1 - \hat{P}_j). \tag{6.9}$$

The final loss $\mathcal{L}(\mathbf{w})$ is a combination between the triplet margin loss defined in Equation (6.8) and the point importance loss in Equation (6.9) weighted by a tunable parameter μ following

$$\mathcal{L}(\mathbf{w}) = \mathcal{L}_{\text{TM}}(\mathbf{w}) + \mu\, \mathcal{L}_{\text{BCE}}(\mathbf{w}). \tag{6.10}$$

The loss minimization problem that finds the optimal network parameters $\mathbf{w}^*$ for the training dataset is given as

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmin}}\ \mathcal{L}(\mathbf{w}). \tag{6.11}$$

The cost function is minimized iteratively with stochastic gradient descent, where the gradients of the loss with respect to the descriptor encoder network parameters $\nabla_{\mathbf{w}}\mathcal{L}$ are computed using backpropagation. These gradients then update the parameters in the direction that reduces the loss, such that

$$\mathbf{w}' \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}, \tag{6.12}$$

where η refers to the learning rate, which measures the optimization step size.

6.2 Implementation Details

We train the model for 80 epochs using a batch size of 4. Following KPPR [185], we set the descriptor size $D = 256$, a learning rate $\eta = 5 \times 10^{-6}$ and a learning rate decay of 0.95 every 20 steps. We define the triplet loss margin parameter as $\alpha = 0.1$. We set the KPConv radius at each layer to $r = [3, 6, 12, 24, 48]\,\mathrm{m}$ and a grid downsampling of sizes $0.25, 0.5, 1, 2, 4\,\mathrm{m}$. The kernel size is set to 5 points for nuScenes and 7 points for the SJTURadarDataset. The number of NetVLAD cluster centers C is set to 64.

NuScenes provides insight into how our approach behaves over long periods of time on 2D automotive radars. We follow a similar procedure to Cai et al. [22], training on scans from "Boston sea-port" obtained in the first 105 days and validating on the measurements performed after that time. We split our training and test sets with a $4:1$ ratio and consider a prediction correct if it is located within $\mathrm{r}_1 = 9\,\mathrm{m}$ of the actual location. During training, negative samples are taken from outside a $\mathrm{r}_2 = 18\,\mathrm{m}$ radius. Moreover, we train on 5 positive samples for each location for improved place recognition performance.

The SJTURadarDataset [117] evaluates the loop closure detection capabilities of our method using automotive 3D radars, as the recordings were taken within a short time span with minimal variations in the environment. We use "Campus 1" and "Campus 4" for training, and evaluate on "Campus 2" and "Campus 3". Additionally, we consider a prediction correct if it is within $\mathrm{r}_1 = 2\,\mathrm{m}$. During training, negative samples are taken from outside a $\mathrm{r}_2 = 4\,\mathrm{m}$ radius.

6.3 Experimental Evaluation

This chapter aims to achieve single-scan place recognition using automotive radars with the goal of identifying revisited locations. We present our experiments to show the capabilities of our system. The results of our evaluation support our key claims that our approach: (i) achieves state-of-the-art performance on automotive radar single-scan place recognition while keeping a compact scene representation; (ii) provides a novel procedure to utilize RCS information to describe the scene, improving accuracy; and (iii) enhances feature extraction by estimating the importance of points within the scan for place recognition.

6.3.1 Experimental Setup

Our approach is able to reliably retrieve a corresponding scene descriptor in a given map based on a single radar scan. During the evaluation of our method, we run experiments on real-world driving scenarios, using 2D and 3D automotive

Table 6.1: We outperform previous work in nuScenes [21] with a smaller descriptor. The abbreviation "TE*" indicates the multi-scan temporal encoding used by Autoplace [22].

	R@1% [%]	R@1/5/10 [%]	Output dim.
ScanContext RCS [92]	6.38	3.58 / 4.84 / 5.22	1200
FFT-RadVLAD RCS [57]	11.2	0.19 / 0.58 / 1.64	32768
FFT-RadVLAD [57]	16.3	0.29 / 1.74 / 3.29	32768
RadVLAD RCS [57]	17.2	0.97 / 3.19 / 4.84	32768
ScanContext [179]	24.7	15.3 / 20.8 / 22.2	1200
RadVLAD [57]	33.5	1.74 / 6.09 / 10.5	32768
KPPR [185]	88.0	66.1 / 78.1 / 81.6	**256**
Autoplace [22]	83.0	60.8 / 72.8 / 76.5	9216
Autoplace+TE* [22]	86.4	73.4 / 81.2 / 83.0	4096
Ours	**88.2**	**76.2** / **82.3** / **83.9**	**256**

Table 6.2: We outperform previous work in SJTURadarDataset [117] with a smaller descriptor.

	R@1% [%]	R@1/5/10 [%]	Output dim.
ScanContext RCS [92]	94.7	94.0 / 94.7 / 94.7	1200
FFT-RadVLAD [57]	50.4	20.0 / 26.5 / 30.8	32768
FFT-RadVLAD RCS [57]	57.3	21.5 / 28.8 / 34.2	32768
RadVLAD RCS [57]	99.3	94.6 / 97.4 / 98.2	32768
ScanContext [179]	87.7	79.7 / 86.6 / 87.4	1200
RadVLAD [57]	99.3	91.9 / 96.3 / 97.3	32768
KPPR [185]	99.4	95.4 / 99.2 / 99.2	**256**
Autoplace [22]	99.9	94.4 / 99.4 / **99.8**	9216
Autoplace+TE* [22]	99.5	**97.7** / 99.0 / 99.1	4096
Ours	**100.0**	97.1 / **99.6** / **99.8**	**256**

radar datasets, nuScenes [21], and the SJTURadarDataset [117], respectively. We first evaluate our work on single-scan radar place recognition and compare it to state-of-the-art solutions. We provide quantitative and qualitative results for the comparison. Then, we carry out an ablation study of our system to analyze the contribution of each module to the final result, as well as how the new hyperparameters have an effect on place recognition.

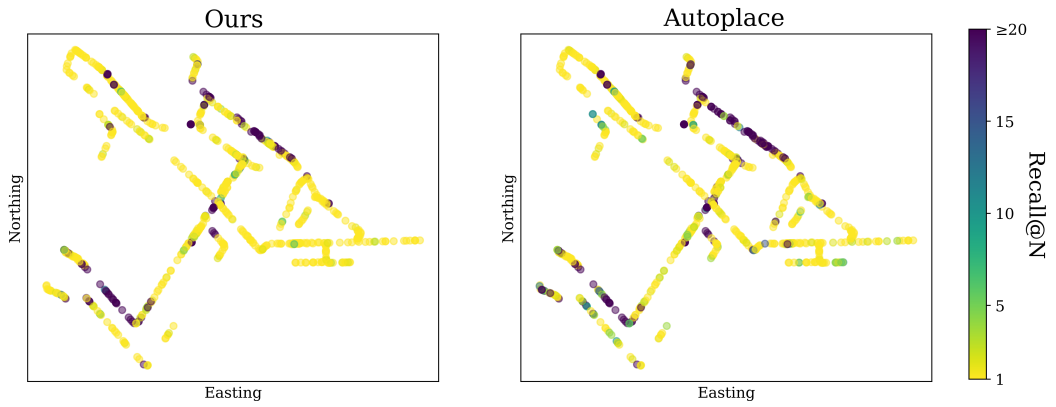


Figure 6.6: Our approach achieves higher single scan radar place recognition recall when comparing it with our baseline Autoplace [22] on the nuScenes [21] autonomous driving dataset.

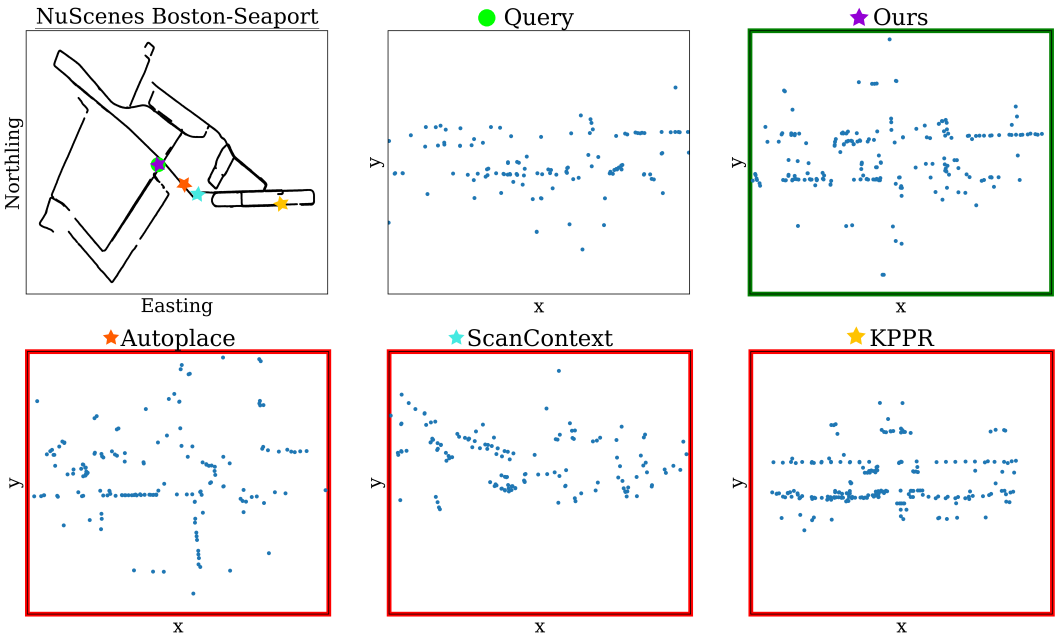


Figure 6.7: Example of single scan place recognition of various state-of-the-art approaches. Other methods fail to correctly classify certain locations due to the high radar noise and sparsity.

6.3.2 Comparison With the State of the Art

The first experiment evaluates the accuracy and the resulting descriptor size of our single-scan radar place recognition system in comparison to other methods in 2D and 3D radar datasets. Similar to other works, we denote recall as "R" and measure the R@1/5/10 and R@1% metrics, which represent if the selected query by the system ranks among the top 1, 5, 10, or 1% candidates in the database. We also display the output descriptor dimensionality, describing the size of the descriptor being stored in the database. A higher-dimensional descriptor requires higher storage space, while smaller descriptors can be stored more efficiently.

We show a quantitative comparison of our method with state-of-the-art techniques in Table 6.1 and Table 6.2, and qualitative visualizations in Figure 6.6 and Figure 6.7. We evaluate it against the feature-based LiDAR methods of ScanContext [179] and RCS ScanContext [92], a neural network-based LiDAR approach KPPR [185], a spinning radar frequency method FFT-RadVLAD [57], and a clustering-based approach RadVLAD [57] with and without RCS values, and a learning-based automotive radar solution Autoplace [22]. We place special focus on Autoplace [22] as the only automotive radar baseline available at the time of publication. The original implementation is done with 7 aggregated radar point clouds, and their LSTM-based temporal encoding considers 3 consecutive aggregated scans. In addition to the results in their paper, we compare their results in nuScenes for one sweep with and without temporal encoding. The best results are displayed in **bold**.

Our method achieves state-of-the-art place recognition recall for both datasets, achieving comparable performance to multi-scan approaches while using a more compact scene descriptor. In nuScenes, the low number of points per radar scan makes it difficult for non-learning-based methods to extract useful patterns from the environment, such as ScanContext [94] and RadVLAD [57]. The discrete nature of projected point clouds also poses a challenge for spinning radar approaches like FFT-RadVLAD. For the SJTURadarDataset, our method applied to a single scan obtains a similar R@1 to Autoplace using multi-scan temporal encoding and 3 scans. This experiment demonstrates how our compact descriptor remains highly informative for place recognition.

6.3.3 Ablation Studies

The second set of experiments supports our claim that integrating RCS into the network and estimating the importance of each point enhances place recognition performance. We carry out the experiments on nuScenes [21] test set from Section 6.3.2. In Table 6.3, we evaluate how each component contributes to the final result and how it affects inference runtime. The main modules are the scan encoder, which accepts as inputs point coordinates (x, y, z) or point coordinates with an additional RCS channel (x, y, z, RCS), the proposed RCS network, and the point importance estimation module. We also experiment with multiplication $\otimes$, and addition $\oplus$ of the point importance features in Equation (6.6). Furthermore, we also test our training strategy, where five positive samples are used for each query, and show how scan aggregation affects the runtime.

The results illustrate how each component contributes to the final result, with the biggest effect being caused by the RCS network module. We also observe how addition is preferred over multiplication in the point importance estimation module, as it enhances the useful points for place recognition without affecting

Table 6.3: Each component of our network contributes to the final performance. Configurations include: scan encoder (Encoder) with coordinate inputs (x, y, z), with RCS as an additional channel (x, y, z, RCS), our RCS network (RCSN), our point importance estimator (PIE), training with five positive samples per query (5 Pos.), and aggregation of seven scans (Aggr.).

Encoder	RCSN	PIE	5 Pos.	Aggr.	R@1/5 [%]	Runtime [ms]
(x, y, z)	✓	$\oplus$	✓	✓	**77.8** / **81.9**	269
(x, y, z)					62.7 / 76.8	150
(x, y, z, RCS)					63.2 / 78.0	172
(x, y, z)	✓				70.7 / 80.8	159
(x, y, z)		$\oplus$			63.1 / 76.5	159
(x, y, z)	✓	$\otimes$			70.9 / 81.0	168
(x, y, z)	✓	$\oplus$			73.3 / **82.2**	167
(x, y, z, RCS)	✓	$\oplus$	✓		73.9 / 80.1	167
(x, y, z)	✓	$\oplus$	✓		**74.3** / 81.6	167

Table 6.4: Ablation studies for the δ and μ hyperparameters in our point importance estimator.

Radius δ [m]	R@1/5/10 [%]	Loss weight μ	R@1/5/10 [%]
1.0	71.4 / **81.1** / **83.6**	0.00	71.0 / 80.0 / 82.6
2.0	71.5 / 80.2 / 82.2	0.10	71.6 / 80.6 / 83.1
3.0	**71.7** / 80.6 / 83.0	0.50	**73.3** / **82.2** / **84.6**
4.0	70.8 / 80.7 / 82.5	1.00	71.4 / 81.1 / 83.6

the remaining point cloud. Moreover, the runtime is minimally affected by the RCS network and point importance estimation modules ($< 20\,\text{ms}$) but greatly increases with scan aggregation. Adding the RCS as an additional channel leads to a slight improvement in performance compared to the RCS network module. This highlights the importance of using the RCS information of the entire scan, rather than adding it as an additional feature to each point.

Additionally, in Table 6.4, we test the influence of the distance parameter δ and loss function weight μ introduced by our point importance estimation module. High radii δ cause wrong correspondences, while low radii lead to not finding any correspondence, resulting in an equal weighting of all points. This demonstrates how the R@1 performance can be improved by focusing on those points that are important for place recognition and focusing less on the noise points. Additionally, varying μ shows the influence caused by the point importance estimation module, and how the multi-objective loss from Equation (6.10) affects the final result.

6.4 Conclusion

The aim of this chapter was to develop a mechanism that can be leveraged to identify previously visited locations by exploiting the characteristics of automotive radar data. To accomplish this, we encode the radar measurements into scene descriptors that can be compared within a map database. Our novel point-based neural network architecture encodes local and global information of the scene into a single compressed descriptor. We achieve this by encoding the local information using point convolutions and combining it with the RCS data of the points within a scene. Additionally, we propose a point importance estimation module that helps the network learn radar targets that are useful for place recognition. This is particularly useful due to the high noise and sparsity in automotive radar point clouds. We implemented and evaluated our approach on different datasets and provided comparisons to other existing techniques, supporting all claims made in this chapter. The experiments suggest that our method achieves high performance for estimating the global location of a car within a map using single automotive radar scans, while keeping a compact scene representation.

Having a reliable and efficient way to perform place recognition is crucial within our localization and mapping system for two main reasons. First, as detailed in the next Chapter 7, it can be leveraged to perform loop detection while navigating unknown environments, resulting in reduced trajectory drift. Second, as it will be introduced in Chapter 8, place recognition can be exploited to identify coinciding locations across different maps recorded at different points in time, enabling multi-session mapping.

Chapter 7

Radar-Inertial SLAM for Autonomous Vehicles

Simultaneous localization and mapping play a crucial role in the operation of autonomous vehicles within unknown environments. Estimating the pose of an autonomous car with respect to its surroundings is necessary for path planning, and collecting surrounding information helps to localize and make more informed decisions when a place is being revisited. In Chapters 4 and 5, we explored the capabilities of radar-only odometry leveraging scan matching. However, relative motion estimation between consecutive radar scans results in the accumulation of errors over time. A way to minimize this drift is to perform loop closure, which requires recognizing previously visited locations within a trajectory. In this chapter, we integrate our radar odometry presented in Chapter 4 with our place recognition system from Chapter 6 into a full SLAM system with loop closure for large-scale mapping, see Figure 7.1. Furthermore, we introduce an additional source of information from an inertial measurement unit (IMU) to our system, further enhancing our pose estimation. These sensors are present in most modern vehicles today and help smooth the noisy trajectories estimated during scan matching of automotive radar scans.

Most existing full-SLAM systems presented in Section 3.1 leverage LiDAR sensors [100] or spinning radars [70] to estimate the vehicle trajectory and map. However, integrating LiDARs and spinning radars into consumer vehicles is challenging due to their increased size and manufacturing cost. Automotive radars, however, are compact and affordable. Some automotive radar SLAM approaches propose to optimize a pose graph with loop closures [72, 117, 204] and/or to combine radar sensors with IMU information [178, 204] to estimate the pose of the vehicle over time. However, these techniques rely on a single global factor graph that contains all the measurement information and utilize adapted LiDAR techniques for loop closure that can result in incorrect loop detections.

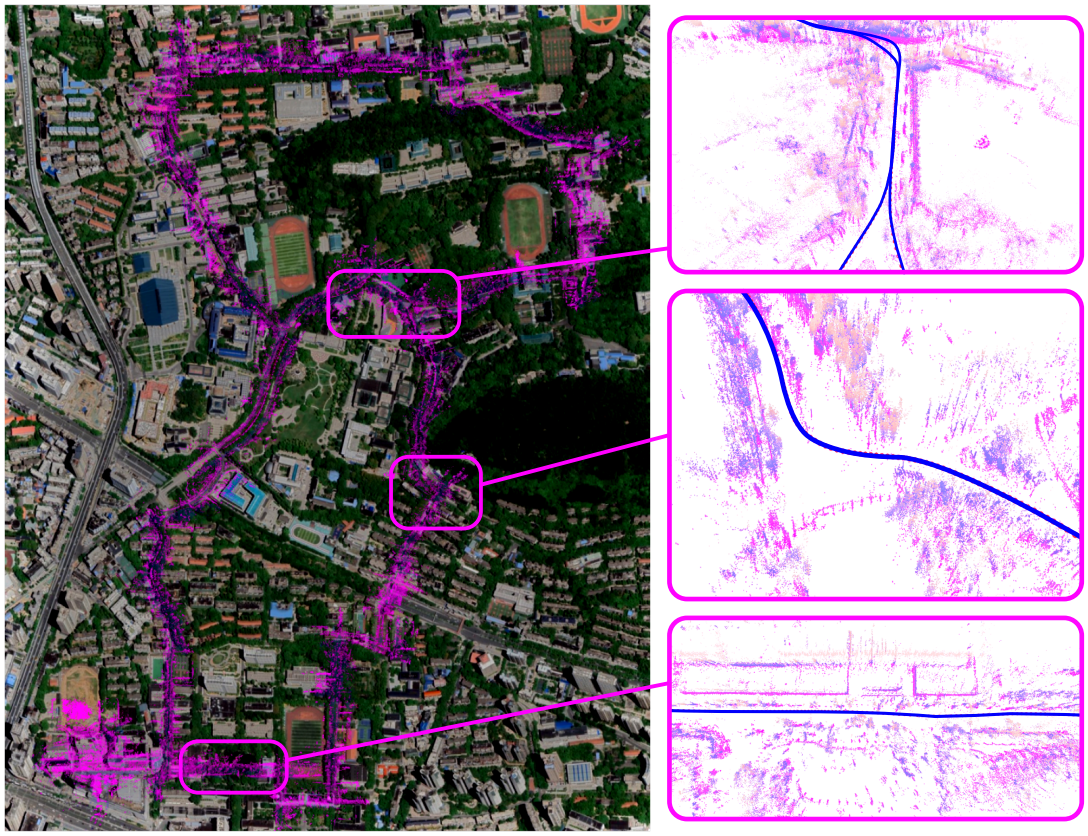


Figure 7.1: Qualitative results of our radar-inertial SLAM system on the SNAIL-Radar dataset [78]. (left) Radar point cloud map (**purple**) projected into a satellite view of the location of the dataset. (right) Different sections of the created map are shown with corresponding colored boxes in the left image, with a visualization of the estimated trajectory (**blue**).

In our work, we propose a novel SLAM pipeline specific to automotive radars. We take inspiration from the global mapping approaches by Koide et al. [99, 100] and Grisetti et al. [64] that maintain multiple pose graphs, and leverage hierarchical pose graph optimization to achieve trajectory estimates that are locally and globally consistent in large-scale datasets. Additionally, we propose a way to utilize the radar velocities to enhance trajectory accuracy to initialize scan matching. We also introduce a loop closure verification score that leverages the characteristics of sparse and noisy automotive radar scans.

The main contribution of this chapter is a novel radar-inertial SLAM system for autonomous vehicles that fully leverages the characteristics of automotive radars for odometry estimation and loop closure. We formulate the problem in a modular manner, with three distinct components: a local pose graph, a global pose graph, and a loop closure detector. The local pose graph integrates local information from the IMU and radar scan registration, leveraging the Doppler

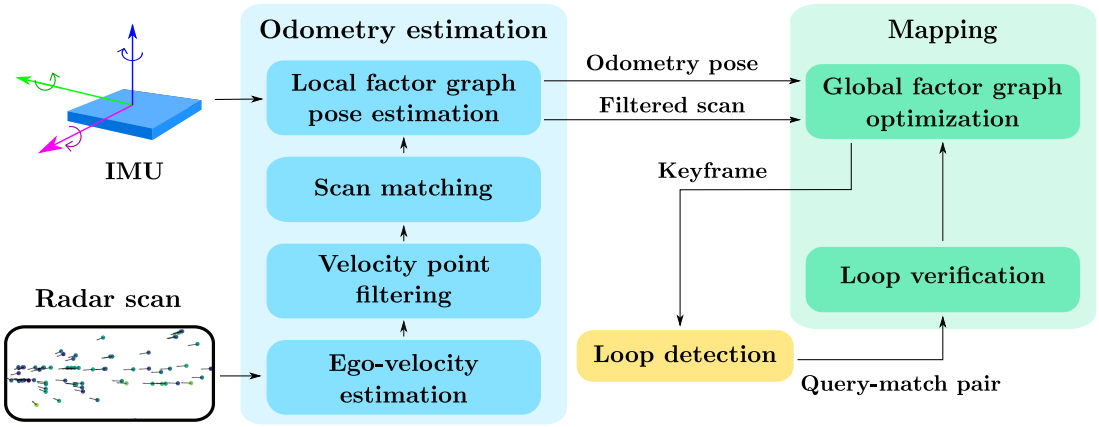


Figure 7.2: Structure of our radar-inertial SLAM system. It consists of three main modules. The odometry estimation module optimizes a local graph for radar-inertial odometry and scan filtering. The mapping module optimizes a global graph with odometry and loop closures. The loop detection module recognizes revisited places and adds them to the global graph.

velocities for scan matching. This serves as an initial pose estimate for the global pose graph. We perform global pose graph optimization, including the loop closures from our loop detector. Our system achieves state-of-the-art performance in radar-inertial SLAM using real-world autonomous driving data. In sum, we make three key claims. Our approach: (i) achieves state-of-the-art results in radar-inertial SLAM for autonomous driving on publicly available data; (ii) introduces a two-layered system structure for radar-inertial SLAM, comprising a local graph and a global graph leading to enhanced accuracy; and (iii) exploits the velocity and radar cross section to improve SLAM accuracy.

7.1 Our Approach to Radar-Inertial SLAM

Our approach estimates the pose of an autonomous vehicle over time and simultaneously constructs a map of the environment. Our system architecture is shown in Figure 7.2 and comprises three core modules. The odometry estimation module optimizes a local graph for radar-inertial odometry point cloud filtering. The mapping module optimizes a global graph containing odometry information and loop closures. The loop detection module recognizes revisited places and adds them to the global graph. Furthermore, having a double-graph structure as illustrated in Figure 7.3 enables our system to incorporate short-term and large-scale information effectively. In the odometry estimation frontend, the local factor graph collects the most recent information from the IMU and radar scan matching and marginalizes nodes outside of a fixed window to maintain computational efficiency. In the backend, the global factor graph collects the optimized data from the local pose graph, verifies loop detections, and integrates loop closure

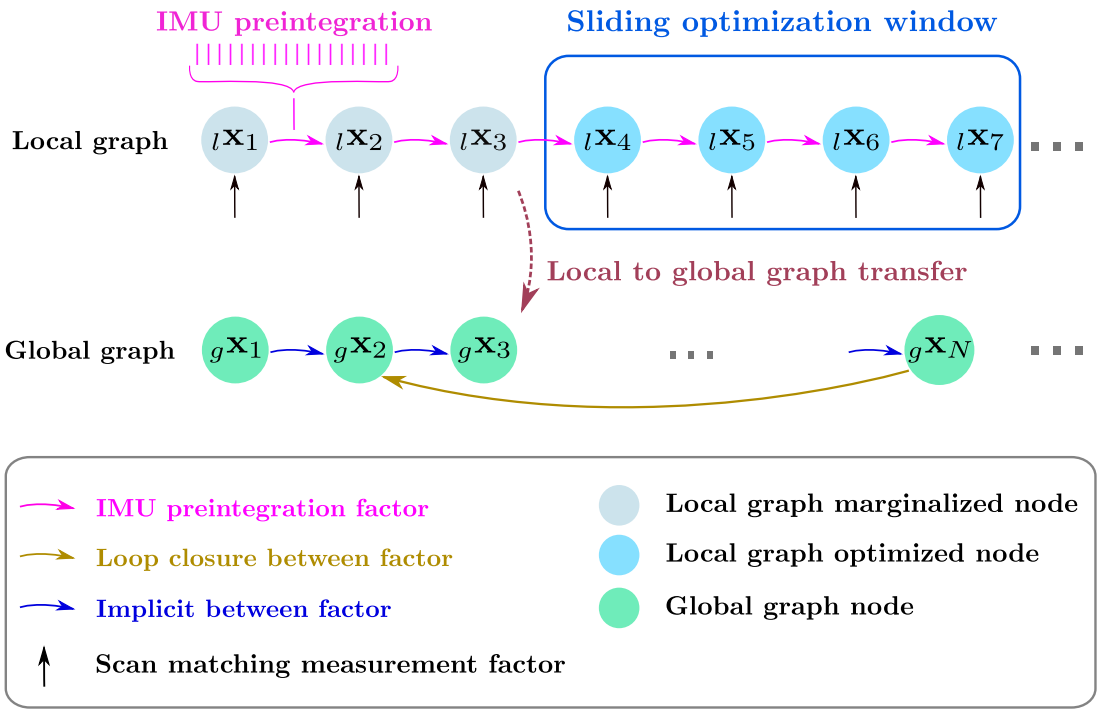


Figure 7.3: Our system integrates local and global information in a double-graph architecture. The scan matching and IMU information are collected within the optimization window. The global graph incorporates this data and optimizes the loops within the estimated poses.

information, improving large-scale consistency. This reduces odometry drift and performs global optimization by considering all available data.

We employ incremental smoothing and mapping by Kaess et al. [87] to optimize the local and global graphs that compute the vehicle's pose. The optimization problem for a single graph can be expressed following the formulation from Section 2.5.4.1 and Equation (2.94) on page 37. In the following, we represent states and observations from the local and global graphs using left-subscripts ${}_l(\cdot)$ and ${}_g(\cdot)$, respectively.

7.1.1 Local Factor Graph

The goal of the local factor graph is to estimate the pose of the vehicle by combining scan matching information with pre-integrated accelerometer and gyroscope measurements. We exploit the robustness of the scan matcher while leveraging the short-term accuracy provided by high-frequency IMU sensors employing two separate factors. The local factor graph is a way of extending the odometry introduced in Chapter 4 by adding an additional IMU sensor to the pipeline, aiming to improve trajectory accuracy. However, it still contains inherent drift, which is corrected with our loop closure procedure in the global graph.

7.1.1.1 Scan Matching

In our system, the primary source of odometry is derived from estimating the relative transform between radar scans using scan-to-map registration with Doppler velocities. However, an incorrect alignment of a single radar scan can lead to an incorrect absolute trajectory estimate. We mitigate this by utilizing the radar Doppler velocity information to initialize our ICP scan alignment procedure, extending our Radar-ICP approach presented in Chapter 4. Following Kellner et al. [89], given the azimuth angle of each measurement θ_i and its Doppler velocity v_i, we first estimate the planar velocity of the radar sensor ${}^{S}\boldsymbol{v}_S = [v_{s,x}, v_{s,y}, 0]^\top$ as a least-squares problem following

$$\begin{bmatrix} v_{r,1} \\ \vdots \\ v_{r,N} \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \vdots & \vdots \\ \cos(\theta_N) & \sin(\theta_N) \end{bmatrix} \begin{bmatrix} v_{s,x} \\ v_{s,y} \end{bmatrix}. \tag{7.1}$$

Assuming a vehicle with no vertical and lateral movement, the linear ${}^{C}\boldsymbol{v}_C \in \mathbb{R}^3$ and angular ${}^{C}\boldsymbol{\omega}_C \in \mathbb{R}^3$ velocities can be computed following

$${}^{C}\boldsymbol{v}_C = \mathbf{R}_S^C \, {}^{S}\boldsymbol{v}_S - {}^{C}\boldsymbol{\omega}_C \times {}^{C}\mathbf{t}_S^C, \tag{7.2}$$

where $\mathbf{R}_S^C \in SO(3)$ and ${}^{C}\mathbf{t}_S^C \in \mathbb{R}^3$ are the calibration parameters from the car to the radar sensor expressed in the car frame. Given the time difference Δt between the current and previous scan, we can estimate the Lie algebra $\hat{\boldsymbol{\xi}} \in \mathfrak{se}(3)$ of the current pose given

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} {}^{C}\hat{\boldsymbol{\rho}}_C \\ {}^{C}\hat{\boldsymbol{\phi}}_C \end{bmatrix} = \Delta t \begin{bmatrix} {}^{C}\hat{\boldsymbol{v}}_C \\ {}^{C}\hat{\boldsymbol{\omega}}_C \end{bmatrix}. \tag{7.3}$$

We can map $\hat{\boldsymbol{\xi}}$ to the homogeneous matrix $\mathsf{T}_{\text{init}} \in SE(3)$ following

$$\mathsf{T}_{\text{init}} = \text{Exp}(\hat{\boldsymbol{\xi}}). \tag{7.4}$$

We leverage $\mathsf{T}_{\text{init}} \in SE(3)$ as the initial estimate for ICP and perform scan-to-map matching, which helps to handle incorrect correspondences between radar scans. The scan-to-map factor associated to node ${}_{l}\mathbf{x}_i \in \mathbb{R}^6$ contains its corresponding odometry measurement ${}_{l}o_{\text{odom}_i}$ associated to the ICP transformation estimate $\mathsf{T}_i \in SE(3)$ according to

$${}_{l}f_{\text{odom}}({}_{l}\mathbf{x}_i; {}_{l}o_{\text{odom}_i}) = {}_{l}f_{\text{odom}}({}_{l}\mathbf{x}_i; \mathsf{T}_i). \tag{7.5}$$

7.1.1.2 IMU Preintegration

While we consider scan matching as our main source of information within the global graph, the accelerometer and gyroscope readings from the IMU help to

smooth the trajectory and provide high-frequency pose estimation in environments with few geometric features. We follow the procedure by Foster et al. [53] to pre-integrate the IMU measurements and combine them with the odometry measurements in the factor graph.

The raw measurements from the sensor defined in the IMU frame $\{M\}$ are given as

$$
\begin{aligned}
{}^{M}\tilde{\boldsymbol{\omega}}_t &= {}^{M}\boldsymbol{\omega}_t + {}^{M}\mathbf{b}_t^g + {}^{M}\boldsymbol{\eta}_t^g, && (7.6)\\
{}^{M}\tilde{\boldsymbol{a}}_t &= (\mathsf{R}_t^I)^\top({}^{M}\boldsymbol{a}_t - {}^{M}\mathbf{g}) + {}^{M}\mathbf{b}_t^a + {}^{M}\boldsymbol{\eta}_t^a, && (7.7)
\end{aligned}
$$

which consist of the gyroscope and accelerometer measurement ${}^{M}\boldsymbol{\omega}_t, {}^{M}\boldsymbol{a}_t \in \mathbb{R}^3$ an added white noise ${}^{M}\boldsymbol{\eta}_t^g, {}^{M}\boldsymbol{\eta}_t^a \in \mathbb{R}^3$, an added bias ${}^{M}\mathbf{b}_t^g, {}^{M}\mathbf{b}_t^a \in \mathbb{R}^3$, the gravity vector ${}^{M}\mathbf{g} \in \mathbb{R}^3$, and the rotation $\mathsf{R}_t^I \in SO(3)$ of the IMU at time t with respect to the inertial frame.

Using the IMU measurements, we estimate the rotation $\mathsf{R}_{t+\Delta t}^I \in SO(3)$, velocity ${}^{M}\boldsymbol{v}_{t+\Delta t} \in \mathbb{R}^3$, and position ${}^{M}\boldsymbol{p}_{t+\Delta t} \in \mathbb{R}^3$ after a time increment Δt following the integration of IMU measurements over time with

$$
\begin{aligned}
\mathsf{R}_{t+\Delta t}^I &= \mathsf{R}_t^I \operatorname{Exp}\left(({}^{M}\tilde{\boldsymbol{\omega}}_t - {}^{M}\mathbf{b}_t^g - {}^{M}\boldsymbol{\eta}_t^g)\Delta t\right), && (7.8)\\
{}^{M}\boldsymbol{v}_{t+\Delta t} &= {}^{M}\boldsymbol{v}_t + {}^{M}\mathbf{g}\Delta t + \mathsf{R}_t^I({}^{M}\tilde{\boldsymbol{a}}_t - {}^{M}\mathbf{b}_t^a - {}^{M}\boldsymbol{\eta}_t^a)\Delta t, && (7.9)\\
{}^{M}\boldsymbol{p}_{t+\Delta t} &= {}^{M}\boldsymbol{p}_t + {}^{M}\boldsymbol{v}_t\Delta t + \left({}^{M}\mathbf{g} + \mathsf{R}_t^I({}^{M}\tilde{\boldsymbol{a}}_t - {}^{M}\mathbf{b}_t^a - {}^{M}\boldsymbol{\eta}_t^a)\right)\frac{1}{2}\Delta t^2. && (7.10)
\end{aligned}
$$

The estimate of the relative motion of the sensor ${}^{C}\Delta\mathsf{R}_{i,j}$, ${}^{C}\Delta\boldsymbol{v}_{i,j}$, and ${}^{C}\Delta\boldsymbol{p}_{i,j}$ measured by the IMU between two radar scans at times i and j is given by

$$
\begin{aligned}
{}^{C}\Delta\mathsf{R}_{i,j} &= \mathsf{R}_M^C \prod_{k=i}^{j-1} \operatorname{Exp}\left(({}^{M}\tilde{\boldsymbol{\omega}}_k - {}^{M}\mathbf{b}_k^g + {}^{M}\boldsymbol{\eta}_k^g)\Delta t\right), && (7.11)\\
{}^{C}\Delta\boldsymbol{v}_{i,j} &= \mathsf{R}_M^C \sum_{k=i}^{j-1} {}^{M}\Delta\mathsf{R}_{i,k}\left({}^{M}\tilde{\boldsymbol{a}}_k - {}^{M}\mathbf{b}_k^a - {}^{M}\boldsymbol{\eta}_k^a\right)\Delta t, && (7.12)\\
{}^{C}\Delta\boldsymbol{p}_{i,j} &= \mathsf{R}_M^C \sum_{k=i}^{j-1} \left[{}^{M}\Delta\boldsymbol{v}_{i,k}\,\Delta t + {}^{M}\Delta\mathsf{R}_{i,k}({}^{M}\tilde{\boldsymbol{a}}_k - {}^{M}\mathbf{b}_k^a - {}^{M}\boldsymbol{\eta}_k^a)\frac{1}{2}\Delta t^2\right], && (7.13)
\end{aligned}
$$

where R_M^C represents the extrinsic calibration from the car to the IMU frame.

The relative motion information estimated from the IMU is added to the local factor graph as an additional constraint for the optimization, in a similar manner to LIO-SAM [161]. Our approach, however, is adapted to incorporate radar data that is optimized together with the IMU factors. Following the formulation in Equation (2.94), this results in the factors with the corresponding state nodes ${}_{l}\mathbf{x}_{i-1}, {}_{l}\mathbf{x}_i \in \mathbb{R}^6$ and associated observations of the rotation ${}_{l}o_{\mathsf{R}_{i-1,i}}$,

velocity ${}_{l}o_{\boldsymbol{v}_{i-1,i}}$, and position ${}_{l}o_{\boldsymbol{p}_{i-1,i}}$ defined as

$$ {}_{l}f_{\text{IMU}}({}_{l}\mathbf{x}_{i-1}, {}_{l}\mathbf{x}_{i}; {}_{l}o_{R_{i-1,i}}) = {}_{l}f_{\text{IMU}}({}_{l}\mathbf{x}_{i-1}, {}_{l}\mathbf{x}_{i}; {}^{C}\Delta R_{i-1,i}), \tag{7.14} $$

$$ {}_{l}f_{\text{IMU}}({}_{l}\mathbf{x}_{i-1}, {}_{l}\mathbf{x}_{i}; {}_{l}o_{\boldsymbol{v}_{i-1,i}}) = {}_{l}f_{\text{IMU}}({}_{l}\mathbf{x}_{i-1}, {}_{l}\mathbf{x}_{i}; {}^{C}\Delta \boldsymbol{v}_{i-1,i}), \tag{7.15} $$

$$ {}_{l}f_{\text{IMU}}({}_{l}\mathbf{x}_{i-1}, {}_{l}\mathbf{x}_{i}; {}_{l}o_{\boldsymbol{p}_{i-1,i}}) = {}_{l}f_{\text{IMU}}({}_{l}\mathbf{x}_{i-1}, {}_{l}\mathbf{x}_{i}; {}^{C}\Delta \boldsymbol{p}_{i-1,i}). \tag{7.16} $$

We optimize the local graph with scan matching factors from Equation (7.5), and the IMU factors from Equations (7.14–7.16) in a sliding window manner. All factors outside the marginalization window are excluded from the optimization, keeping a bounded size of the optimizable graph.

7.1.2 Global Factor Graph

The global factor graph corrects the drift accumulated in the local factor graph by performing loop closure optimization. Two key processes are performed in the global graph, as illustrated in Figure 7.3. First, we transfer the information from the local to the global factor graph. This results in a global factor graph with implicit factors that contain information about the IMU and scan registration. We then include loop closure factors into the graph to correct odometry drift.

7.1.2.1 Transfer From Local to Global Graph

The data transfer from the local to the global graph is performed on the latest node of the windowed optimization. We create an implicit factor between the nodes ${}_{g}\mathbf{x}_{i-1}, {}_{g}\mathbf{x}_{i} \in \mathbb{R}^6$ containing the relative pose between the current and the previous scan, coming from both, scan-to-map matching and IMU, following

$$ {}_{g}f({}_{g}\mathbf{x}_{i-1}, {}_{g}\mathbf{x}_{i}; {}_{g}o_{i}) = {}_{g}f({}_{g}\mathbf{x}_{i-1}, {}_{g}\mathbf{x}_{i}; \mathsf{T}_{i}^{i-1}), \tag{7.17} $$

where $\mathsf{T}_{i}^{i-1} \in SE(3)$ represents the transformation between the last two optimized poses within the local graph.

7.1.2.2 Loop Closure

Including the implicit factors in the global graph does not solve the problem of a drifting trajectory. We introduce loop closure detection and optimization to correct the accumulated errors with three main criteria. First, our radar place recognition module from Chapter 6 finds a matching location and returns a similarity score. Then, an odometry distance measurement estimates the feasibility of the loop closure in space. Finally, we propose an intuitive point correspondence metric that replaces the standard ICP distance score used by Shan et al. [161] to measure matching quality.

The place recognition module identifies locations that have been visited in the past. While other works use ScanContext [178, 179], in Chapter 6 we demonstrated how learning-based approaches show superior place recognition performance with minimal storage requirements. Moreover, as mentioned previously, our radar descriptor encoder network captures point-neighbor information from the radar scan and an overall RCS distribution of the point cloud in a vector $\mathbf{d} \in \mathbb{R}^{256}$ that contributes to a high place recognition recall. We match two radar scans if

$$\|\mathbf{d}_{\text{query}} - \mathbf{d}_{\text{match}}\| < \delta_{\text{PR}}, \tag{7.18}$$

where δ_{PR} is a predefined threshold and $\mathbf{d}_{\text{query}}, \mathbf{d}_{\text{match}} \in \mathbb{R}^{256}$ are the encoded radar descriptors of the query and matching scans.

The odometry distance metric is inspired by TBV-SLAM [1] and verifies that the position of the query frame and matching frame $\mathbf{t}_{\text{query}}, \mathbf{t}_{\text{match}} \in \mathbb{R}^3$ are within a reasonable spatial threshold δ_{odom} proportional to the accumulated length of the current trajectory d_{length}. It is defined as

$$\frac{\|\mathbf{t}_{\text{query}} - \mathbf{t}_{\text{match}}\|}{d_{\text{length}}} < \delta_{\text{odom}}. \tag{7.19}$$

After a loop candidate has been detected based on similarity and odometry distance, we use ICP to estimate the transformation $\mathsf{T}_m^q \in SE(3)$ between the match and the query frames, as introduced earlier in Section 2.5.2.1. However, we observe that the distance score used in LiDAR approaches is not a reliable matching quality measure in automotive radars. While in LiDAR, a low average distance between point correspondences indicates a good scan alignment, this is not always the case in sparse and noisy radar point clouds. Good alignments may still return high average distance values due to outliers, bad alignments may still return reasonable average distance values, and if two radar scans are taken at different times, the amount of noise and reflection outliers may vary.

To address these limitations, we propose a new scoring method for loop-matching quality. Instead of using the mean distance of corresponding points, we measure that the number of points after ICP registration that have a correspondence within a radius r is greater than a threshold δ_{d}. This gives an intuitive and reliable estimate of the matching quality. The comparison of the score with $\delta_{\text{d}} \in [0, 1]$ is given as

$$\Big[\frac{1}{|\mathcal{C}|} \sum_{(\mathbf{q},\mathbf{m})\in\mathcal{C}} \mathbb{I}\{\|\mathbf{q} - \mathbf{m}\| < r\}\Big] > \delta_{\text{d}}, \tag{7.20}$$

where $(\mathbf{q}, \mathbf{m}) \in \mathcal{C}$ is the set of correspondences between query $\mathbf{q}$ and matching scan $\mathbf{m}$, and $\mathbb{I}\{c\}$ is the indicator function, which returns 1 if condition c is true, and 0 otherwise. If there are no close correspondences, i.e., $|\mathcal{C}| = 0$, no loop closure takes place.

Once the three criteria have been verified and the transformation between query and match T_m^q has been computed, we express the final loop factor between nodes ${}_g\mathbf{x}_q, {}_g\mathbf{x}_m \in \mathbb{R}^6$ with the observation ${}_g o_{\mathrm{loop}_i}$ following

$$ {}_g f({}_g\mathbf{x}_q, {}_g\mathbf{x}_m; {}_g o_{\mathrm{loop}_i}) = {}_g f({}_g\mathbf{x}_q, {}_g\mathbf{x}_m; \mathsf{T}_m^q). \tag{7.21} $$

The global graph contains the implicit factors from Equation (7.17) and the loop factors from Equation (7.21). To keep the real-time capabilities of the odometry frontend, we optimize the global graph independently in the backend, following the procedure from Equation (2.94).

7.2 Implementation Details

We implement our approach using ROS2 for communication between modules and a GTSAM [39] factor graph optimization framework. We leverage a fixed-lag smoother for the local graph frontend and employ the IMU preintegration factor from Foster et al. [53]. For the global optimization, we create a keyframe with every scan when the vehicle is not static, and perform place recognition for each keyframe. We run global optimization every 10 seconds in a separate thread.

7.3 Experimental Evaluation

The main focus of this chapter is to develop a full radar-inertial SLAM system for autonomous vehicles that exploits the information provided by radar sensors for large-scale pose estimation and mapping. We present our experiments to show the capabilities of our method. The results explicitly support our key claims that our approach: (i) achieves state-of-the-art results in radar-inertial SLAM for autonomous driving on publicly available data; (ii) introduces a two-layered system structure for radar-inertial SLAM, comprising a local graph and a global graph leading to enhanced accuracy; and (iii) exploits the velocity and radar cross section to improve SLAM accuracy.

7.3.1 Experimental Setup

We evaluate our system on the SNAIL-Radar [78] and HeRCULES [97] datasets, since other datasets only have sequences with no loop closures [21, 139], or do not have any IMU sensors, such as the SJTURadarDataset employed earlier [117]. Within the SNAIL-Radar dataset, we select sequences 20240113/3, 20240113/1, 20240115/2, 20240123/2, and 20240123/3, which cover the entire area of the dataset. From the HeRCULES [97] dataset, we select five diverse sequences containing loop closures, “Mountain Day 1”, “Library Day 1”, “Sports Complex

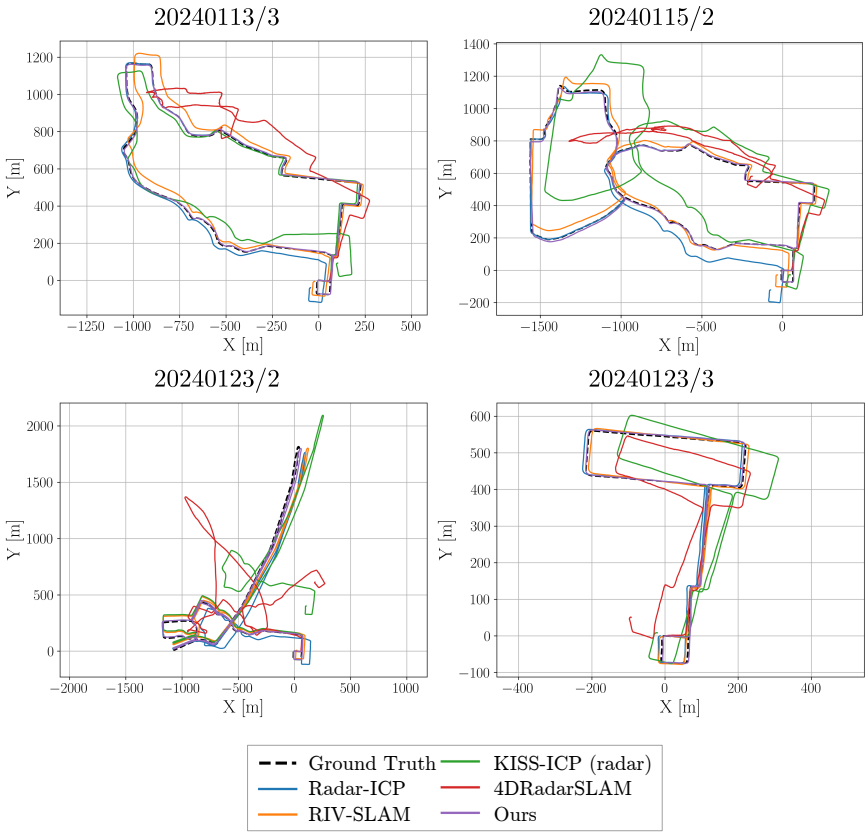


Figure 7.4: Our approach achieves state-of-the-art results on the SNAIL-Radar [78] dataset, showing reduced drift and smaller absolute errors than others in large-scale trajectories.

Day 1", "Parking Lot 3 Night", and "Street Day 1". We use the data from the ARS548 radar for the evaluation of all methods except in the comparison with 4DRadarSLAM [204], whose parameters have been optimized by the authors for the Oculii Eagle. The main difference between these two sensors is that, while the ARS548 has been designed to be integrated in consumer vehicles, the Oculii is bulkier and performs point cloud post-processing to enhance the density of the radar scans. For our loop closure module, we train the radar descriptor encoder from Chapter 6 on the SJTURadarDataset [117] using the ZF FRGen21 radar, a different radar sensor than the SNAIL-Radar dataset, indicating the generalizability of our system. We evaluate the trajectories on the plane, discarding all vertical movement. The metrics used for evaluation consist of the t_{RPE}, r_{RPE} and t_{APE} metrics presented in Section 2.5.5, measuring short-term accuracy as well as global consistency. The best results on automotive radars are shown in **bold**.

7.3.2 Comparison With the State of the Art

The first experiment evaluates the performance of our method and demonstrates that it achieves state-of-the-art results in publicly available data. We compare

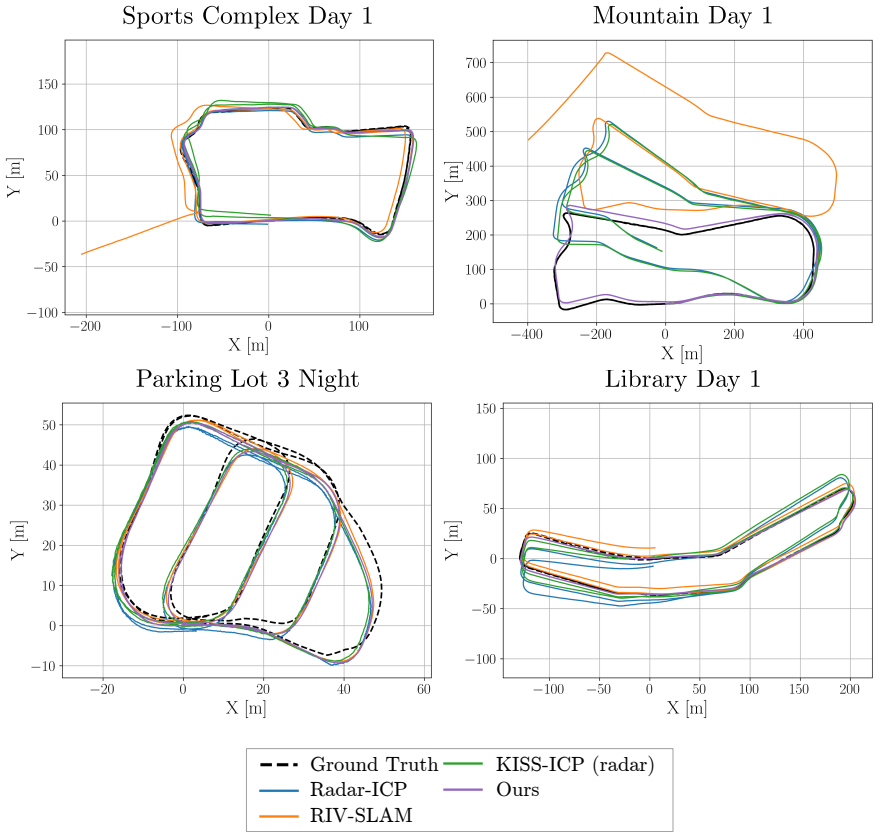


Figure 7.5: Our approach achieves state-of-the-art accuracy on the HeRCULES [97] dataset, showing reduced drift and smaller absolute errors than others in shorter trajectories.

our method against other approaches in the SNAIL-Radar dataset [78], and then compare the best-performing methods in the HeRCULES dataset [97]. Qualitative results are shown in Figures 7.4, 7.5, and 7.6, and quantitative results are shown in Tables 7.1 and 7.2.

The baseline methods employed in the comparison are RIV-SLAM [178], which leverages a single graph integrating all sensor data; Graph-RIO [63], which uses a factor graph but does not include scan registration and loop closure; 4DRadarSLAM [204], which relies solely on the Oculii radar data without IMU information; our Radar ICP approach introduced in Chapter 4; KISS-ICP [172] applied directly on radar and LiDAR point clouds; and LIO-SAM [161], a LiDAR-inertial odometry system used in this work to benchmark sensor performance.

As it can be observed in Figure 7.4 and Figure 7.5, LiDAR approaches like KISS-ICP applied to radar data have an increased drift, sometimes failing in scenarios with few geometric features, like in the long road from sequence 2024123/2. Odometry-only Radar-ICP has less drift but lacks loop closure to correct it, visible in sequences like “Mountain Day 1”. RIV-SLAM achieves high overall relative and absolute accuracy but accumulates drift over longer sequences, as it relies on intensity ScanContext [179] and the regular ICP distance score for loop closure.

Table 7.1: Our method achieves state-of-the-art results on radar odometry and SLAM on the SNAIL-Radar dataset [78]. t_{RPE} t_{APE}, in [m] and r_{RPE} in [°].

	20240113/3			**20240113/1**			**20240115/2**			**20240123/2**			**20240123/3**			**Mean**		
	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}
LIO-SAM (LiDAR) [161]	0.014	0.066	52.6	0.015	0.131	0.3	0.019	0.084	48.5	5.983	0.786	57.6	0.013	0.080	5.9	1.209	0.229	33.0
KISS-ICP (Radar) [172]	0.240	0.155	68.4	0.118	0.179	4.4	0.232	0.134	147.1	0.269	0.117	167.8	0.222	0.155	45.9	0.216	0.148	86.7
4DRadarSLAM [204]	0.737	1.170	53.2	0.460	1.074	8.9	0.663	1.179	491.2	0.864	0.901	454.5	0.503	0.983	142.1	0.645	1.061	230.0
Graph-RIO [63]	-	-	-	0.169	0.172	9.5	**0.195**	0.172	763.4	-	-	-	0.266	0.168	497.0	0.210	0.170	423.3
Radar-ICP (Chap. 4)	0.238	0.156	18.2	0.120	0.174	3.9	0.229	0.131	31.6	0.252	0.112	37.5	0.221	0.151	7.9	0.212	0.145	19.8
RIV-SLAM [178]	**0.213**	**0.142**	30.2	**0.113**	0.171	4.1	0.219	0.128	33.1	**0.224**	**0.101**	35.5	**0.201**	0.140	6.1	**0.194**	0.137	21.8
Ours	0.242	0.147	**4.7**	0.115	**0.164**	**3.4**	0.220	**0.117**	**7.9**	0.256	0.104	**8.4**	0.219	**0.139**	**3.5**	0.210	**0.134**	**5.6**

Table 7.2: Our method achieves state-of-the-art results on radar odometry and SLAM on the HeRCULES dataset [97]. t_{RPE}, t_{APE}, in [m] and r_{RPE} in [°].

	Mountain Day 1			**Library Day 1**			**Sports Complex Day 1**			**Parking Lot 3 Night**			**Street Day 1**			**Mean**		
	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}
KISS-ICP (LiDAR) [172]	0.064	0.068	21.9	0.057	0.065	8.4	0.057	0.084	8.3	0.077	0.146	2.1	-	-	-	0.064	0.091	10.2
KISS-ICP (Radar) [172]	0.057	0.067	113.3	0.046	0.061	8.7	0.046	0.069	8.2	0.065	0.101	2.5	0.033	0.044	**5.5**	0.049	0.069	27.7
Radar-ICP (Chap. 4)	0.055	0.067	118.6	0.049	0.064	10.2	0.049	0.071	7.1	0.058	0.089	3.4	0.022	0.028	11.7	0.029	0.064	30.2
RIV-SLAM [178]	0.077	0.084	206.9	**0.014**	0.064	4.2	-	-	-	**0.020**	0.075	2.4	0.010	0.042	10.7	0.030	0.066	56.0
Ours	**0.019**	**0.024**	**11.2**	0.019	**0.021**	**2.1**	**0.020**	**0.023**	**3.9**	0.022	**0.033**	**2.2**	**0.008**	**0.005**	11.5	**0.018**	**0.021**	**6.2**

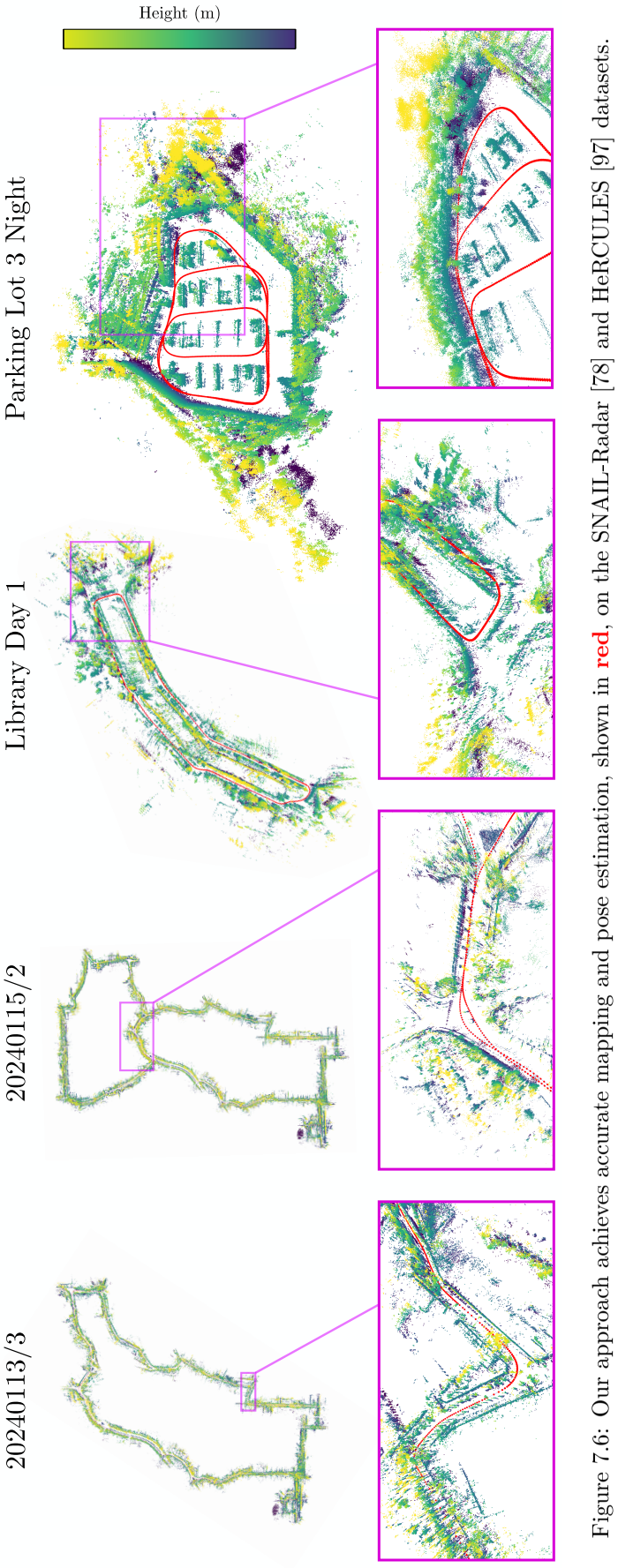


Figure 7.6: Our approach achieves accurate mapping and pose estimation, shown in **red**, on the SNAIL-Radar [78] and HeRCULES [97] datasets.

Table 7.3: Each component of our radar-inertial SLAM system contributes to the final accuracy in sequences from SNAIL-Radar [78] and HeRCULES [97]. t_{RPE}, t_{APE}, in [m] and r_{RPE} in [°].

	20240115/2			**Library Day 1**		
	t_{RPE}	r_{RPE}	t_{APE}	t_{RPE}	r_{RPE}	t_{APE}
Scan matching	0.247	0.258	112.4	0.087	0.361	18.47
Scan matching with vel. prior	0.244	0.252	40.04	0.085	0.346	17.03
Local graph with IMU	**0.222**	**0.145**	39.90	**0.024**	**0.210**	16.99
Local and global without IMU	0.252	0.253	**5.642**	0.087	0.362	3.652
Local and global with IMU	0.229	0.146	5.726	**0.024**	**0.210**	**3.389**

However, ScanContext and the regular ICP distance score fail to detect certain loops and correct accumulated errors. Our method, which relies on a radar-oriented loop detection procedure, identifies the loops and effectively integrates local and global information, leading to a lower mean t_{APE} while maintaining competitive accuracy for relative pose estimation. RIV-SLAM also fails in sequence "Sports Complex Day 1", where an incorrect point cloud alignment leads to an inconsistent trajectory. This is prevented with our velocity-based ICP initialization. Additionally, our radar-inertial SLAM technique is on par with LiDAR approaches. Specifically, it presents an advantage in the highly dynamic scenario from "Street Day 1", where LiDAR odometry fails to complete the sequence due to the high amount of point outliers corresponding to moving objects.

7.3.3 Ablation Studies

The second experiment evaluates how our two-layered graph structure, and how exploiting the characteristics of radar data for odometry and loop closure, contribute to the final accuracy. We evaluate different configurations of our system on the SNAIL-Radar [78] dataset sequence 20240115/2 and on the HeRCULES dataset [97] sequence "Library Day 1" as good examples of long trajectories that include loops. The main components are the velocity prior for scan matching, the local graph with and without IMU, and the global loop closure for drift correction. The results are shown in Table 7.3.

Employing the estimated velocity as the prior for ICP results in a notable improvement of the absolute pose errors. Without the velocity prior, wrong point correspondences can lead to an incorrect trajectory for the following measurements, affecting the absolute error. Furthermore, adding the IMU measurements reduces relative error but preserves a similar absolute error due to the way the local factor graph is constructed. Adding the loop closure additionally corrects accumulated drift, resulting in a system with reduced local and global pose errors.

7.4 Conclusion

In this chapter, we integrated our odometry and place recognition modules into a radar-specific system to achieve high-accuracy pose estimation and mapping in unknown environments. Our modularized approach consists of three main components. First, we combine our radar odometry approach with an IMU within a local factor graph to estimate the relative ego-motion between consecutive radar measurements. Second, we perform loop closure detection using our radar place recognition module, which exploits the sparse and noisy characteristics of radar point clouds. Third, we verify the detected loops using our odometry and point correspondence metrics, and optimize a global graph that contains loop and odometry information. We implemented and evaluated our approach on real-world scenarios supporting all claims made in this chapter. The experiments suggest that our method achieves high performance for estimating the global pose of the vehicle by solely relying on onboard radar-inertial sensing, with each component of the system contributing to the final pose estimation accuracy. As demonstrated by our experimental evaluation, our approach enables simultaneous localization and mapping leveraging radar and IMU sensors, resulting in a trajectory that is locally and globally consistent. Beyond estimating the trajectory, our SLAM system also builds a map of the environment by aggregating radar scans within each pose, which can serve as a prior for the localization system of the vehicle.

Our radar-inertial SLAM system builds maps from specific areas of the environment; hence, covering new regions requires re-running SLAM. In the next chapter, we integrate long-term mapping and localization capabilities within our system. This is crucial for merging maps with different coverages, identifying changes between them, and leveraging the maps to improve pose estimation accuracy during navigation.

Chapter 8

Long-Term Radar Mapping and Localization

Long-term mapping and localization aim to achieve an accurate map that can be extended over time and perform pose estimation within it. Although our SLAM procedure from Chapter 7 produces a radar point cloud map, such maps typically involve specific regions of the environment. Therefore, we require a strategy that can merge multiple SLAM sessions performed at different times to map the entire drivable area. This results in two main challenges, illustrated in Figure 8.1. First, during SLAM, the trajectory is estimated relative to the starting point, which results in each map having a different origin if the starting location differs between sessions. Furthermore, environments change over time, such as parked cars changing locations. This can result in inconsistent maps when combining multiple overlapping maps. Since radar scans are very sparse, it is crucial that the map only contains correct measurements to minimize wrong point-to-point correspondences during localization.

Existing approaches detailed in Section 3.4 have tackled the challenge of multi-session map construction and map maintenance using LiDARs. Kim et al. [96] introduced an approach that performs place recognition between multiple sessions and aligns them to a common reference frame. Others like Lazaro et al. [111] focus on the problem of pose graph optimization from maps recorded by multiple robots. Furthermore, LiDAR methods by Gil et al. [62] and Lim et al. [119, 120] identify map differences either by measuring the distance between point correspondences in different maps, or by projecting the LiDAR scans to range-azimuth images and identifying the changes between images. Nevertheless, very few approaches tackle the problem of localization within existing maps. Fang et al. [49] and Peng et al. [144] alternate between odometry and localization based on the number of point correspondences, and Hroob et al. [75] predict non-persistent points during operation. These methods, however, rely heavily on the high den-

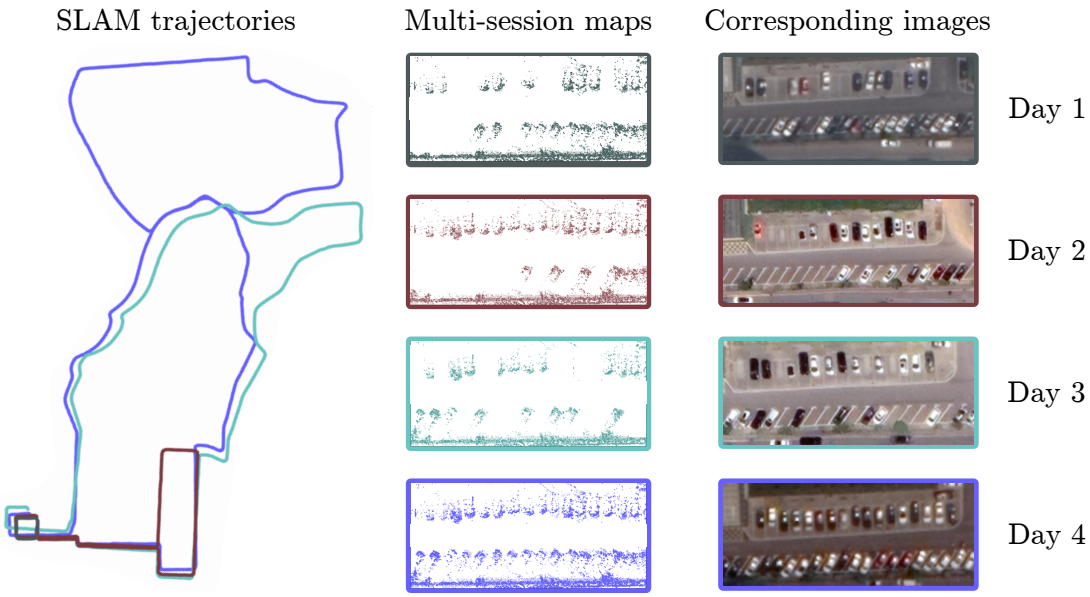


Figure 8.1: Two main challenges during long-term mapping. (left) Trajectories recorded during different SLAM sessions are initially misaligned. (right) Variations of elements in the scene, such as parked cars, result in map inconsistencies that need to be considered before localization. We show the radar map for each of the sessions and its corresponding satellite image for reference.

sity and accuracy of LiDAR sensors. The low number of points in radar scans can lead to false positives from the place recognition module, and computing map differences by measuring point correspondences can result in incorrect matches due to radar noise. Moreover, directly performing scan-to-map localization is heavily affected by radar sparsity, leading to inaccurate trajectories.

In our work, we overcome incorrect place recognition detections by proposing a strategy to verify detections between sequences. We also present a novel map maintenance module for noisy radar point clouds, which can identify and label persistent elements within the environment. Moreover, we present a simple, yet effective, localization strategy within sparse radar maps that reduces drift compared to single-session pose estimation approaches.

The main contribution of this chapter is a multi-session mapping and localization approach that merges multiple maps and localizes within them, relying on automotive radars. We leverage our approach from Chapter 7 to perform multiple SLAM sessions that cover different regions within the environment. Then, we address the problem of different starting points by identifying coinciding locations between sessions using our place recognition module, presented in Chapter 6, and aligning them to a common reference frame. Additionally, our proposed grid-based strategy identifies differences between maps. Finally, we present a method to perform localization in radar maps. In sum, we make three key claims. Our work: (i) achieves state-of-the-art alignment accuracy of multiple radar SLAM sessions recorded at different points in time; (ii) merges maps from multiple

sessions, preserving persistent and reliable points within a compact map; and (iii) achieves accurate radar pose estimation in the resulting map by combining odometry and localization for scan matching.

8.1 Our Approach to Multi-Session Mapping and Localization

Our approach achieves multi-session mapping and localization by leveraging automotive radars. We first perform online radar SLAM leveraging the technique described in Chapter 7 over multiple sequences with overlapping locations. We find the overlap by performing place recognition between all recorded sequences following our procedure introduced in Chapter 6 and align the maps employing a global offline optimization approach. We then perform map maintenance by comparing the sequences recorded at different points in time and identifying the existence probability of each point. Finally, we achieve low-drift localization by solely leveraging the points with the highest probability of existence.

8.1.1 Multi-Session Mapping

Rather than relying on a single recording to construct a map of the environment, the goal of multi-session mapping is to obtain a globally aligned, consistent, and accurate map by combining recordings taken at different points in time. To achieve this, we build upon the LiDAR-based multi-session alignment approach from LT-Mapper [96] and adapt it to operate with automotive radars. We perform SLAM to build maps of multiple overlapping sequences and identify overlapping scans based on our place recognition strategy, creating additional constraints between sessions. We construct a global pose graph containing all odometry and loop constraints and perform a preliminary optimization step. We then remove infeasible constraints and perform a second optimization step to refine the global pose graph, enhancing alignment accuracy.

We leverage our radar-inertial SLAM procedure from Chapter 7 to perform pose estimation and mapping on M sequences with overlapping regions. Each sequence, $\mathcal{S}_m$, is defined as a set of K keyframes $\mathcal{K}_{m,k} \in \mathcal{S}_m$, where each individual keyframe $\mathcal{K}_{m,k} = (\mathsf{T}_k^m, \mathbf{x}_{m,k}, \mathcal{P}_{m,k}, \mathbf{d}_{m,k})$ contains the pose $\mathsf{T}_k^m \in SE(3)$ estimated by the single-session SLAM approach and its associated node in the factor graph $\mathbf{x}_{m,k} \in \mathbb{R}^6$, the individual scans $\mathcal{P}_{m,k}$ filtered by Doppler velocities, and the place recognition point cloud descriptor $\mathbf{d}_{m,k} \in \mathbb{R}^{256}$ computed by the loop closure detector. Additionally, each sequence $\mathcal{S}_m$ has an associated set of factors $\mathcal{F}_m$ from its own global factor graph representing the odometry and loop constraints introduced in our SLAM approach from Chapter 7.

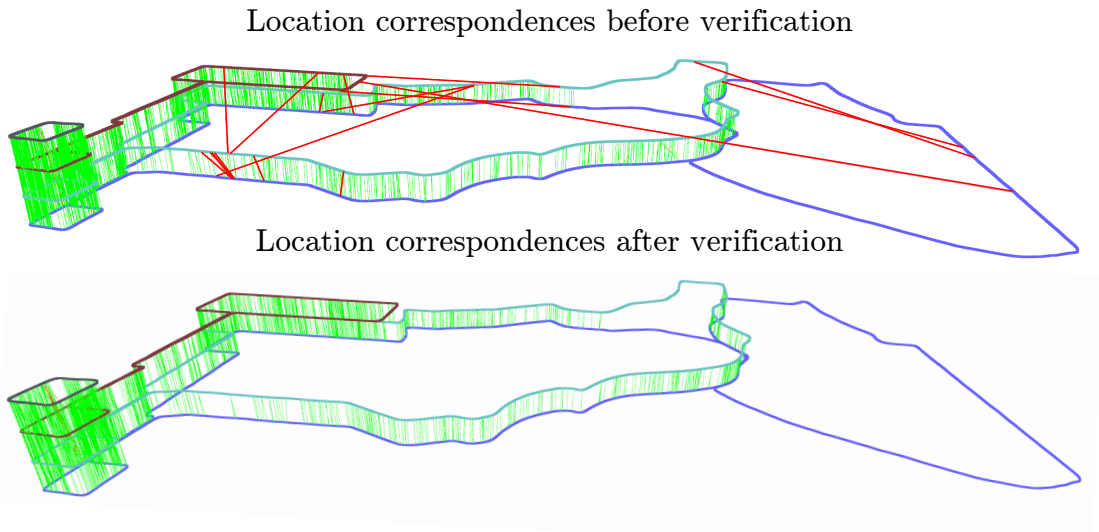


Figure 8.2: Our three-step loop verification strategy removes incorrect place recognitions (**red**).

Each sequence is recorded independently, with the estimated poses given relative to the start of the sequence. Inspired by the LiDAR method by Kim et al. [96], we need to identify overlapping keyframes between the sequences and align them to a common reference frame. In our work, we leverage the place recognition module presented in Chapter 6 which exploits the radar RCS information to identify one-to-one location correspondences. Additionally, to remove false positive loop detections, as shown in Figure 8.2, we propose to leverage three criteria: descriptor similarity, point correspondences, and geometric distances between matches.

First, we measure the $L2$ distance between descriptors with Equation (7.18) on page 102. We then leverage the iterative closest point algorithm presented in Section 2.5.2.1 to align both scans and perform a point correspondence check to mitigate false positives due to perceptual ambiguity. Unlike LT-Mapper [96], we leverage the correspondence matching quality introduced in Equation (7.20), which has shown to be more effective in the radar domain than the ICP fitness score. We classify a location as similar if the number of correspondences within a radius r is higher than a threshold $\delta_{\mathrm{d}} \in [0, 1]$, following

$$\left[\frac{1}{|\mathcal{C}|}\sum_{(\mathbf{q},\mathbf{p})\in\mathcal{C}} \mathbb{I}\{\|\mathbf{q}-\mathbf{p}\| < r\}\right] > \delta_{\mathrm{d}}, \tag{8.1}$$

where $\mathcal{C} = \{(\mathbf{q}, \mathbf{p})\}$ is the set of closest point correspondences between the aligned query points $\mathbf{q} \in \mathcal{Q}$ and the matching point cloud $\mathbf{p} \in \mathcal{P}$, and $\mathbb{I}\{c\}$ is the indicator function, returning 1 if the condition c is true, and 0 otherwise. Furthermore, we count the number of corresponding locations between sequences. If no matches are found for a particular sequence, it may either indicate that there is no overlap or that the place recognition failed to detect valid correspondences. In such cases, we set the sequence aside for possible alignment with future SLAM sessions.

After identifying inter-session loop detections, we construct a factor graph that contains the odometry and loop constraints $\mathcal{F}_m$ from each session coming from the corresponding global factor graphs in Section 7.1.2. Now each pair of matched nodes, $\mathbf{x}_{i,k}, \mathbf{x}_{j,k} \in \mathbb{R}^6$, will have a corresponding pose expressed with respect to the sequences' own reference frame, denoted as $\mathsf{T}_k^i, \mathsf{T}_k^j \in SE(3)$.

To obtain an arbitrary pose k of sequences i and j with respect to the inertial frame $\mathsf{T}_{i,k}^I, \mathsf{T}_{j,k}^I \in SE(3)$, we follow Kim et al. [96] and define their corresponding anchor nodes, $\Delta\mathbf{x}_i, \Delta\mathbf{x}_j \in \mathbb{R}^6$, that represent the transformation between the inertial frame and each sequence such that

$$\mathsf{T}_{i,k}^I = \mathsf{T}_i^I \, \mathsf{T}_k^i, \tag{8.2}$$

$$\mathsf{T}_{j,k}^I = \mathsf{T}_j^I \, \mathsf{T}_k^j. \tag{8.3}$$

Subsequently, the relative transformation between pose k in both sequences is given by

$$\mathsf{T}_{j,k}^{i,k} = (\mathsf{T}_{i,k}^I)^{-1} \, \mathsf{T}_{j,k}^I. \tag{8.4}$$

The transformation from Equation (8.4) is compared with the estimated ICP transform between the two scans $\hat{\mathsf{T}}_{j,k}^{i,k} \in SE(3)$ within the inter-session factor, which is also associated with the observation $o_{i,j,k}$ following

$$f(\Delta\mathbf{x}_i, \Delta\mathbf{x}_j, \mathbf{x}_{i,k}, \mathbf{x}_{j,k}; o_{i,j,k}) = f(\Delta\mathbf{x}_i, \Delta\mathbf{x}_j, \mathbf{x}_{i,k}, \mathbf{x}_{j,k}; \hat{\mathsf{T}}_{j,k}^{i,k}). \tag{8.5}$$

Once the factor graph containing all sequences is built, we perform an initial optimization step to align the sequences in a common reference frame. However, incorrect location correspondences may lead to inaccuracies in the estimated trajectories. We propose to perform an additional simple, yet effective, verification check to remove ambiguous location correspondences by measuring the Euclidean distance between them. We delete all keyframe correspondences that have a distance greater than a threshold δ_{E} and perform a second global optimization step, leading to more accurate trajectories. We further extend LT-Mapper [96] by performing one-to-one pairing between all sequences. This strategy enables direct merging of multiple recordings into a single map, reducing potential deviations caused by an inaccurate central sequence. Additionally, instead of re-optimizing all trajectories when new sequences are recorded, the map can be updated incrementally by pairing each new session with the existing map as a reference.

8.1.2 Map Maintenance

Environments are dynamic, with elements such as vehicles arriving and departing, changes caused by constructions, and other scene elements evolving over time. To have reliable localization and prevent incorrect point matches during scan registration, it is essential to retain only persistent points within the map, making

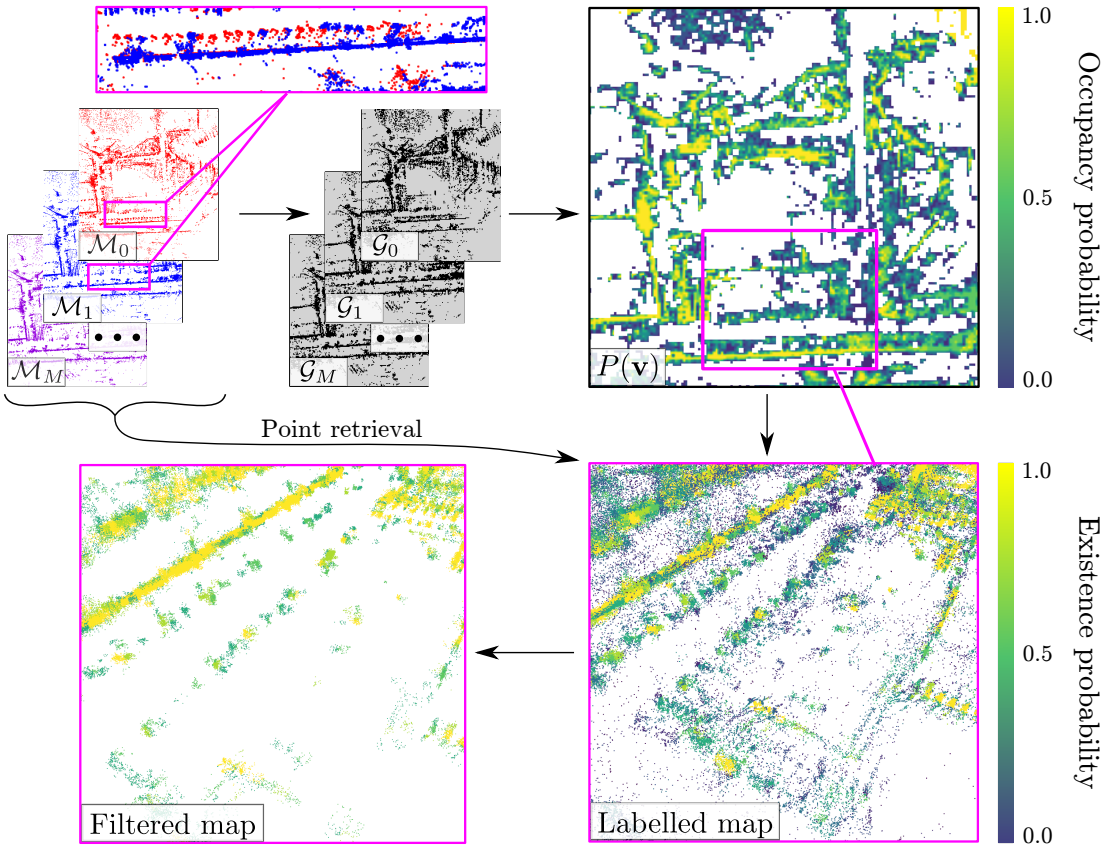


Figure 8.3: Our map maintenance procedure. We convert the point cloud maps into occupancy grids, estimate the occupancy probability of each voxel, and retrieve the labeled points with their probability. Then we discard points with an existence probability lower than a threshold.

map maintenance a critical step for multi-session mapping. Existing LiDAR methods compute map changes by leveraging dense and accurate point clouds, typically measuring the nearest-neighbor distances [75] or projecting scans to images [96]. However, radar maps often contain noise outliers and sparse point clouds, resulting in unreliable point correspondences.

Inspired by Hroob et al. [76] and Gil et al. [62], we differentiate between three types of points in a map: dynamic, volatile, and persistent. Dynamic points belong to objects that are moving during SLAM. Unlike LiDARs [95, 119, 120], radars provide per-point Doppler velocities. To filter out dynamic objects, we compute the vehicle's velocity and discard points with a motion-compensated velocity higher than a threshold δ_{dyn} following Equation (4.12) on page 55. Volatile points belong to objects that change their location across sessions, such as radar noise, stopped buses, standing people, and parked cars. Additionally, persistent points belong to objects that remain in the scene over time, which can be associated with structures like buildings or elements that have not changed across sessions, such as occupied parking spots. However, volatile and persistent points may not be directly identifiable at runtime, requiring specialized map

maintenance procedures. We propose a strategy, illustrated in Figure 8.3, that leverages 3D grid maps to compute the map occupancy probability over multiple sessions. We then identify volatile points with low probability of existence, and only leverage persistent points during localization.

To represent the environment after multi-session alignment, we convert each radar point cloud map $\mathcal{M}_m$ into a 3D binary occupancy voxel grid $\mathcal{G}_m$. The goal is to compute the occupancy probability of the voxels corresponding to each map. However, not all maps have the same coverage. While maps will partially overlap, some maps may contain poses covering a larger area than others. Therefore, it is essential that only maps that cover a specific voxel are taken into account when estimating its probability of being occupied.

Given the set of voxels present at least in one occupancy map $\mathcal{V}$ with each voxel defined by its center position $\mathbf{v} \in \mathbb{R}^3$, we denote the map coverage $C_m(\mathbf{v})$ as a binary indicator of whether any k^{th} position of map $\mathcal{M}_m$ is within the radar range R of voxel $\mathbf{v}$ such that

$$C_m(\mathbf{v}) = \mathbb{I}\{\|\text{trans}(\mathsf{T}_k^m) - \mathbf{v}\| < R\}, \tag{8.6}$$

where the binary indicator function $\mathbb{I}\{c\}$ checks whether any k^{th} position from map m^{th} is within the radar measurement range R, and $\text{trans}(\mathsf{T}_k^m)$ extracts the translation component of the pose k from map m that has already been optimized during multi-session alignment.

We compute the number of maps $N_{\text{cov}}(\mathbf{v})$ covering a specific voxel following

$$N_{\text{cov}}(\mathbf{v}) = \sum_{m \in \mathcal{M}} C_m(\mathbf{v}), \tag{8.7}$$

and the number of maps that cover an occupied voxel $N_{\text{cov+occ}}(\mathbf{v})$ according to

$$N_{\text{cov+occ}}(\mathbf{v}) = \sum_{m \in \mathcal{M}} C_m(\mathbf{v})\ \mathcal{G}_m(\mathbf{v}), \tag{8.8}$$

where $\mathcal{G}_m(\mathbf{v})$ is 1 if an occupied voxel $\mathbf{v}$ exists in the grid map, and 0 otherwise.

We estimate the final occupancy probability with respect to the number of maps that cover each voxel $P(\mathbf{v})$ with

$$P(\mathbf{v}) = \frac{N_{\text{cov+occ}}(\mathbf{v})}{N_{\text{cov}}(\mathbf{v})}. \tag{8.9}$$

Note that our approach also accounts for new structural elements. The occupancy probability increases as more sessions contain that occupied voxel, eventually considering the point as persistent. Moreover, by storing two hash maps for $N_{\text{cov}}(\mathbf{v})$ and $N_{\text{cov+occ}}(\mathbf{v})$, the probability $P(\mathbf{v})$ of affected voxels can be incrementally updated with new incoming sessions.

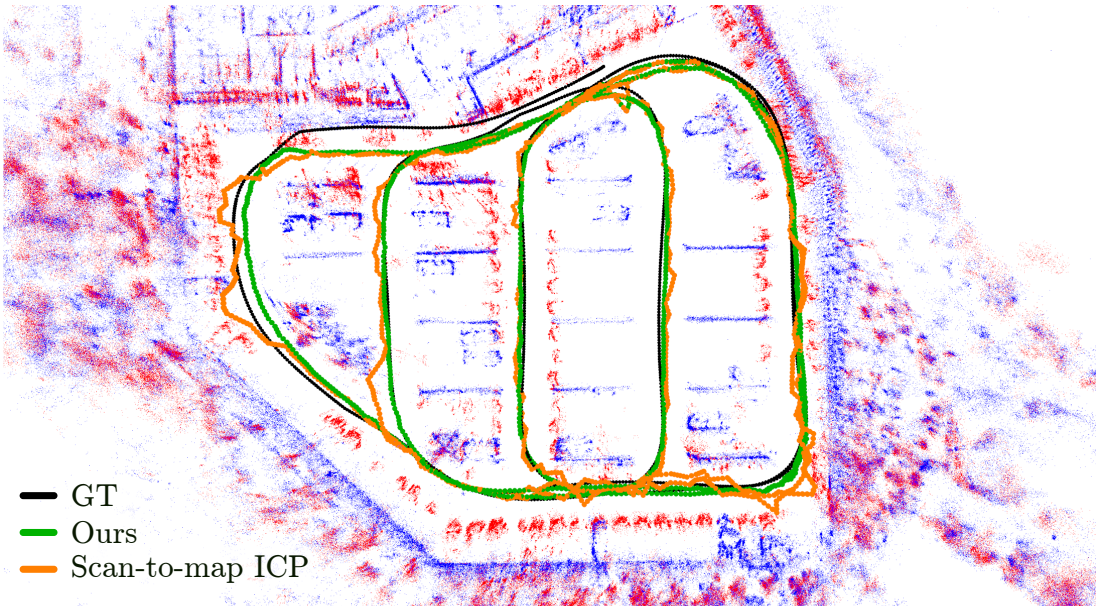


Figure 8.4: Matching between current radar scans (**blue**) and map (**red**) is error-prone due to scene changes and sparsity. Combining localization and odometry enhances trajectory accuracy.

To obtain the final global map $\mathcal{M}_G$, we retrieve the points from their corresponding voxels and assign to each point the voxel's occupancy probability as its existence probability. During localization, we discard points with an existence probability below a threshold β.

8.1.3 Radar Localization

The goal of the previous steps was to create a map suitable for long-term localization. Contrary to odometry, which solely relies on an online map, having an accurate pre-built map reduces uncertainty during pose estimation. Some approaches exist in the LiDAR domain [3, 9] that directly match the current measured scan with the pre-built map, or perform mode-switching between odometry and localization [49]. However, radar scans are sparse, and direct scan-to-map matching leads to estimation errors in changing environments, see Figure 8.4. Our proposed radar localization system combines scan-to-map localization with odometry and leverages radar Doppler velocities to achieve a consistent and accurate trajectory.

Our simple, yet effective, scan-to-map localization strategy fuses the short-term pose accuracy of radar odometry with the long-term temporal consistency of a maintained global map, leading to reduced drift. We register the current scan $\mathbf{p} \in \mathcal{P}$ jointly against two spatial representations: the locally accumulated odometry map $\mathbf{m} \in \mathcal{M}_L$, and the globally consistent map $\mathbf{m}_G \in \mathcal{M}_G$ obtained following our map maintenance procedure in Section 8.1.2. However, a naive combination of global and local maps into the same voxel map [172] can lead to a bias towards odometry, as its points tend to have more matches due to

higher similarity between consecutive scans, neglecting the global prior map. We approach this with a weighted scan-to-map registration strategy that explicitly biases the optimization toward the global map, while still leveraging information from the local map. During ICP alignment, we assign a higher weight to the correspondences from the global map $\mathcal{M}_G$, thereby guiding the solution toward globally consistent poses. We estimate the relative transformation $\mathsf{T}^* \in SE(3)$ that combines the global and local map following

$$\mathsf{T}^* = \underset{\mathsf{T}\in \mathrm{SE}(3)}{\operatorname{argmin}} \Big[w_L \cdot \sum_{(\mathbf{p},\mathbf{m}_L)\in\mathcal{C}_L} \rho\big(|\mathbf{p} - \mathsf{T}\mathbf{m}_L|\big) + w_G \cdot \sum_{(\mathbf{p},\mathbf{m}_G)\in\mathcal{C}_G} \rho\big(|\mathbf{p} - \mathsf{T}\mathbf{m}_G|\big) \Big], \quad (8.10)$$

where $\mathcal{C}_L$, $\mathcal{C}_G$ are the sets of correspondences between the current scan and the local and global map, respectively, w_L, w_G, are balancing weights for the local and global maps, and ρ is a robust Geman McClure kernel [172] to handle point correspondence outliers. Additionally, we leverage the Doppler velocities provided by the radar to compute the initial ICP estimate as described in Section 7.1.1.1.

8.2 Implementation Details

Our implementation of the long-term mapping and localization approach consists of three main components: the multi-session mapping procedure presented in Section 8.1.1, our map maintenance procedure from Section 8.1.2, and our localization strategy from Section 8.1.3. We implement our system employing ROS2 for communication between modules and GTSAM [39] for factor graph optimization. For place recognition, we exploit our approach from Chapter 6 trained on the SJTURadarDataset [117]. The evaluation parameters include the place recognition threshold $\delta_{\mathrm{PR}} = 0.6$, the point matching radius $r = 0.5\,\mathrm{m}$ and ratio $\delta_{\mathrm{d}} = 0.3$, the Euclidean distance correspondence threshold $\delta_{\mathrm{E}} = 5\,\mathrm{m}$, the dynamic velocity threshold $\delta_{\mathrm{dyn}} = 0.1\,\mathrm{m/s}$, a voxel grid size of 1 m, the radar range $R = 50\,\mathrm{m}$, a point probability threshold $\beta = 0.6$ and the global and local map weights $w_G = 10$ and $w_L = 1$ for the SNAIL-Radar [78] and $w_G = 2$, $w_L = 1$ for the HeRCULES [97] datasets. Furthermore, during map generation, we remove all vertical movement and assume a planar vehicle trajectory. The evaluation measures the absolute trajectory (t_{APE}) and rotation (r_{APE}) error with respect to the reference trajectories as introduced in Section 2.5.5.

8.3 Experimental Evaluation

The main focus of this chapter is a long-term mapping and localization approach that combines maps built at different times, performs map maintenance to keep only persistent points, and localizes within the resulting map. We present our

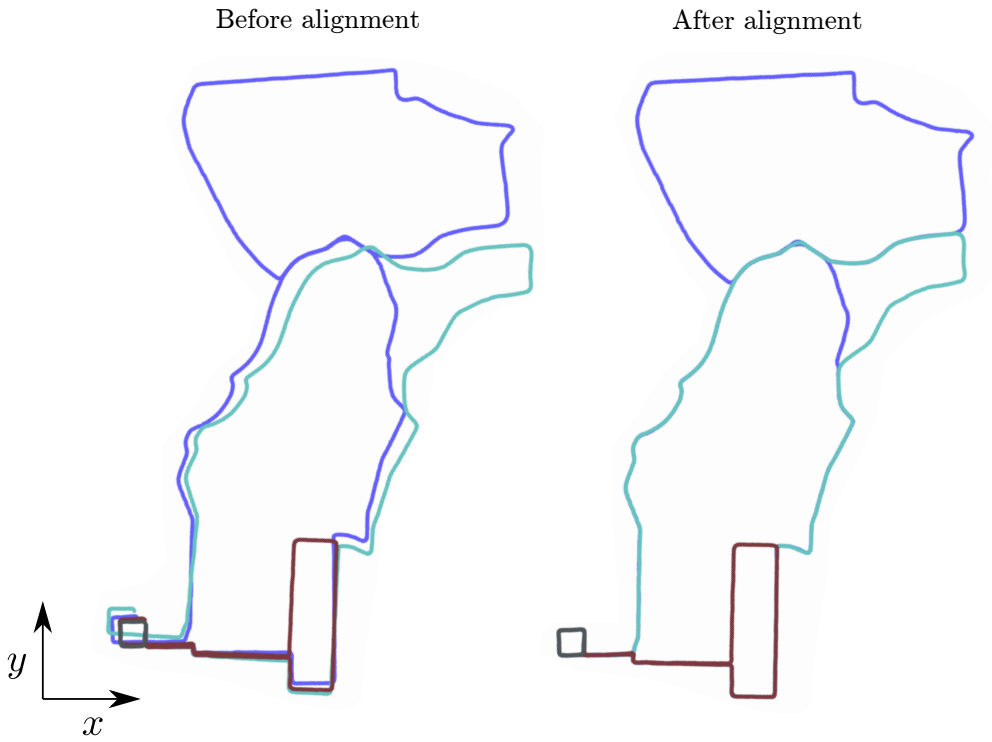


Figure 8.5: Before trajectory alignment, each sequence has a different origin with variations in the rotation. Our approach accurately brings all trajectories into a common reference frame.

experiments to show the capabilities of our method. The results explicitly support our claims that our approach: (i) achieves state-of-the-art alignment accuracy of multiple radar SLAM sessions recorded at different points in time; (ii) merges maps from multiple sessions, preserving persistent and reliable points within a compact map; and (iii) achieves accurate radar pose estimation in the resulting map by combining odometry and localization for scan matching.

8.3.1 Experimental Setup

We evaluate our system on the SNAIL-Radar [78] and HeRCULES [97] datasets using the ARS548 radar sensor. For multi-session alignment, we perform single-session SLAM on the SNAIL-Radar SUV sequences "20240113/1", "20240123/3", "20240113/3", "20240115/2", and "20240116/2" and perform our two-step global optimization between them. We also use the aforementioned sequences during our ablations to analyze the contribution of each component. Additionally, we evaluate parking sequences from the HeRCULES dataset including "Parking D.1", "Parking N.", "Parking D.2.1", and "Parking D.2.2". We then perform multi-session alignment across all sequences for each dataset, treating every sequence as reference. In each sequence, we report the mean error across all alignments. Furthermore, to observe the impact of the SLAM system on our multi-session alignment, we evaluate our approach leveraging the RIV-SLAM [178] and RaI-SLAM

Table 8.1: We achieve state-of-the-art multi-session alignment perfomance on the SNAIL-Radar dataset [78] within multiple sequences of varying sizes. t_{APE} in [m] and r_{APE} in [°].

	20240113/1		**20240123/3**		**20240113/3**		**20240115/2**		**20240116/2**		**Mean**	
	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}
RaI-SLAM (Chap. 7)	0.193	0.907	3.013	1.221	12.49	1.051	6.758	0.749	45.11	2.179	13.51	1.220
RIV-SLAM [178]	0.418	1.076	4.275	1.016	3.120	0.887	5.991	0.649	N/A	N/A	-	-
LT-Mapper (RIV) [96]	0.454	1.253	1.849	1.132	3.390	1.203	3.310	0.753	N/A	N/A	-	-
Ours (RIV)	**0.206**	**1.075**	**1.568**	**1.049**	**2.512**	**1.009**	**2.544**	**0.582**	N/A	N/A	-	-
LT-Mapper (RaI) [96]	**0.197**	**0.924**	1.821	1.096	2.648	0.666	2.575	0.698	**8.998**	**1.160**	3.248	0.900
Ours (RaI)	0.242	0.956	**1.607**	**0.998**	**2.153**	**0.610**	**2.180**	**0.634**	9.472	1.229	**3.131**	**0.885**

Table 8.2: We achieve state-of-the-art multi-session alignment performance on the HeRCULES dataset [97] within multiple sequences in a parking lot. t_{APE} in [m] and r_{APE} in [°].

	Parking D.1		**Parking D.2.1**		**Parking D.2.2**		**Parking N.**		**Mean**	
	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}
RaI-SLAM (Chap. 7)	0.622	2.575	0.466	2.412	0.576	1.015	0.774	3.256	0.609	2.314
RIV-SLAM [178]	0.772	2.146	0.465	2.687	6.602	2.722	0.783	3.520	2.155	2.769
LT-Mapper (RIV) [96]	3.999	9.516	1.057	4.825	**1.283**	6.546	**1.015**	**3.384**	1.838	6.068
Ours (RIV)	**0.740**	**2.833**	**0.347**	**2.745**	3.692	**5.327**	1.064	4.658	**1.461**	**3.891**
LT-Mapper (RaI) [96]	2.001	5.353	0.990	3.323	0.660	1.874	**0.842**	**3.232**	1.123	3.445
Ours (RaI)	**0.610**	**2.516**	**0.571**	**2.530**	**0.620**	**1.383**	1.275	4.196	**0.769**	**2.656**

backends from Chapter 7. We modify them to output the same pose, point cloud, and descriptor formats, and select the most reliable backend for the rest of the experiments. To evaluate our localization approach in the SNAIL-Radar dataset, we localize sequences, "20240116_eve/5" and "20240113/1", within a map built with "20231208/4", "20231213/1", "20231213/4", "20231213/5", "20240115/3", "20240116/5" and "20240123/3". We also localize sequences "20231208/5", and "20231213/2" within a map built from sequences "20231201/3", "20240113/1" and "20240113/3". In the HeRCULES dataset, we build, maintain, and localize "Parking D.1", "Parking N.", "Parking D.2.1", and "Parking D.2.2". Similarly, we compare localization accuracy in maintained and not maintained maps, localizing "20240116_eve/5", "20240113/1", "20231208/5", and "20231213/2" in the maps built for localization, and performing uniform voxel downsampling of the maps at 0.5, 1.0, and 1.5 m to evaluate accuracy at different resolutions. In our evaluation, we omit the mean results of methods with failed runs.

8.3.2 Multi-Session Trajectory Alignment

The first experiment supports our claim that our multi-session mapping approach achieves state-of-the-art alignment accuracy of multiple radar SLAM sessions recorded at different points in time. We compare our method against a state-of-the-art LiDAR approach, LT-Mapper [96] for multi-session alignment. Since

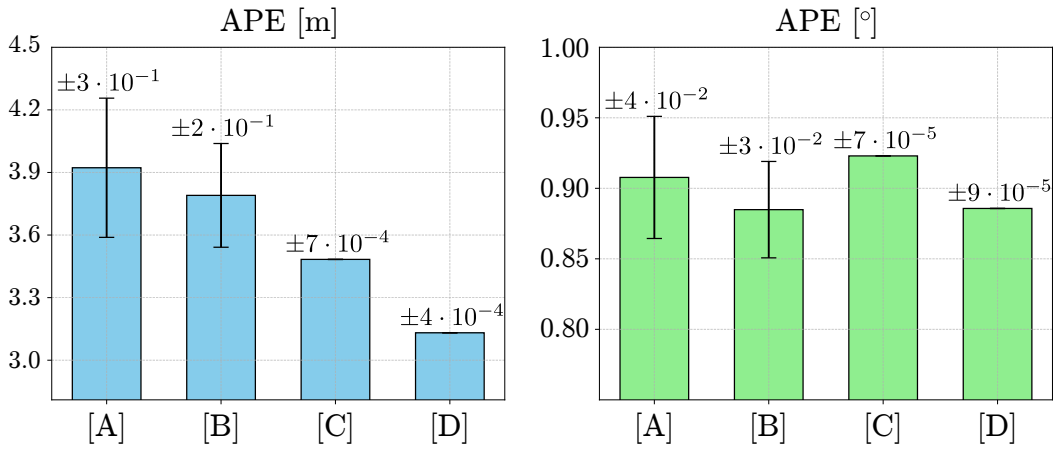


Figure 8.6: Ablations of our alignment strategy for single sequence matching [A], single sequence with verification [B], multi-sequence matching [C], and multi-sequence with verification [D].

LT-Mapper is designed for LiDAR sensors, we modify its input to take the poses coming from the radar SLAM backends. Moreover, we swap the ScanContext [94] LiDAR place recognition approach with our radar place recognition method introduced in Chapter 6 for fair comparison. Additionally, we provide the online trajectory results of single-sequence RaI-SLAM from Chapter 7 and RIV-SLAM [178]. The results are displayed in Table 8.1 and Table 8.2, where **bold** indicates the best alignment for each backend. Our ablations in Figure 8.6 report the mean and standard deviation under different configurations. In Figure 8.5 we also provide a qualitative comparison of the trajectories in the SNAIL-Radar [78] dataset before and after performing multi-session alignment.

The results demonstrate how our method enhances the average alignment accuracy in both datasets, with the RaI-SLAM backend showing superior performance. Moreover, RIV-SLAM fails in sequence "20240116/2", meaning that this sequence cannot be considered during alignment. Furthermore, multi-session alignment presents a clear advantage over single-session SLAM as it is able to correct for loop closures that may not have occurred. In addition, our ablations demonstrate how the loop verification contributes to the overall accuracy. While the improvement in positional accuracy is clear, the orientation error reflects a trade-off from multi-sequence configurations, which constrain the poses with respect to all trajectories. In such cases, the optimization may compromise a well-aligned trajectory to improve consistency across sessions. Therefore, single-sequence alignment may present superior performance when relying on a single session with high trajectory accuracy, but will substantially degrade if the reference session is inaccurate. Our approach, on the contrary, can mitigate reference trajectory inaccuracies, visible in the reduced standard deviation.

Table 8.3: Localization performance in the SNAIL-Radar dataset [78] improves in maintained maps (Maint.) and degrades when performing raw scan aggregation (Not maint.) at different voxel resolutions. "N / A" indicates a failed run. Map size in [MB], t_{APE} in [m] and r_{APE} in [°].

		20240116_eve/5			**20240113/1**			**20231208/5**			**20231213/2**		
Voxel size	Map type	Size	t_{APE}	r_{APE}	Size	t_{APE}	r_{APE}	Size	t_{APE}	r_{APE}	Size	t_{APE}	r_{APE}
None	Not maint.	149.3	1.293	**1.144**	165.7	0.476	**1.119**	159.4	**1.929**	0.809	167.9	N/A	N/A
	Maint.	**106.8**	**1.254**	1.161	**119.6**	**0.420**	1.190	**124.6**	1.934	**0.797**	**131.1**	**1.816**	**1.313**
0.5 m	Not maint.	58.89	1.288	1.188	62.03	0.522	1.245	81.39	**1.924**	0.805	83.83	N/A	N/A
	Maint.	**34.34**	**1.259**	**1.171**	**35.92**	**0.413**	**1.233**	**57.10**	1.925	**0.800**	**58.64**	**1.817**	**1.318**
1.0 m	Not maint.	24.42	1.276	1.170	25.21	0.432	1.085	38.57	**1.912**	**0.799**	39.27	**1.811**	1.353
	Maint.	**12.21**	**1.274**	**1.155**	**12.46**	**0.360**	**0.977**	**24.23**	1.917	0.816	**24.58**	1.812	**1.340**
1.5 m	Not maint.	12.99	**1.286**	**1.122**	13.31	0.476	1.089	22.03	1.926	0.892	22.35	1.825	1.424
	Maint.	**6.478**	1.339	1.177	**6.572**	**0.464**	**1.040**	**13.49**	**1.914**	**0.835**	**13.64**	**1.806**	**1.371**

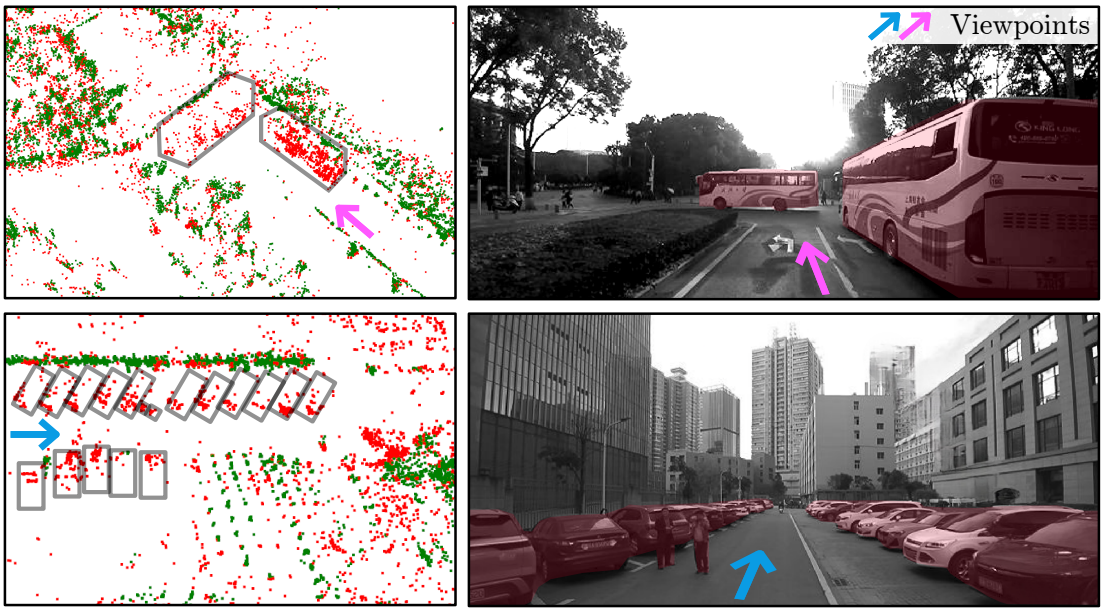


Figure 8.7: (left) Labeled map after map maintenance with volatile points in **red** and persistent points in **green**. (right) Corresponding camera image with movable objects masked in red.

8.3.3 Maintained Maps for Localization and Storage

The second experiment supports our claim that our map maintenance approach merges maps from multiple sessions, preserving persistent and reliable points within a compact map. To validate the reliability of our maintained maps, we perform standard scan-to-map localization without relying on odometry and verify the performance on maintained maps, which include only persistent points, and not maintained maps, which are constructed by merging all session maps without filtering. We display the results in Figure 8.7 and Table 8.3, where **bold** highlights the best performance. The evaluation demonstrates that our map maintenance

Table 8.4: Our approach improves pose estimation accuracy over scan-to-map matching and achieves reduced drift on the HeRCULES dataset [97]. t_{APE} in [m] and r_{APE} in [°].

	Parking D.1		**Parking D.2.1**		**Parking D.2.2**		**Parking N.**		**Mean**	
	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}
RIV-SLAM [178]	0.787	2.230	0.487	2.799	6.644	2.920	0.847	3.054	2.191	2.751
Radar ICP Odom. (Chap. 4)	3.164	4.528	1.788	1.297	2.202	1.957	3.619	1.557	2.536	1.604
RaI-SLAM (Chap. 7)	1.770	2.379	1.778	1.181	1.161	1.772	2.385	1.374	1.774	1.442
Radar ICP Loc. (Chap. 7)	1.273	2.776	1.586	1.986	1.344	2.711	2.345	6.698	1.401	2.491
Ours	**1.116**	**2.064**	**1.466**	**1.574**	**1.332**	**1.770**	**2.038**	**6.314**	**1.305**	**1.803**

Table 8.5: Our approach improves pose estimation accuracy and achieves reduced drift on the SNAIL-Radar dataset [78]. "N / A" indicates a failed run. t_{APE} in [m] and r_{APE} in [°].

	20240116_eve/5		**20240113/1**		**20231208/5**		**20231213/2**		**Mean**	
	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}	t_{APE}	r_{APE}
RIV-SLAM [178]	7.109	0.880	0.266	3.454	6.291	1.678	5.736	0.839	4.851	1.713
Radar ICP Odom. (Chap. 4)	4.437	1.533	1.723	2.040	18.18	1.692	24.04	2.317	12.09	1.896
RaI-SLAM (Chap. 7)	3.832	1.365	0.427	1.365	23.25	2.024	26.40	2.453	13.49	1.802
Radar ICP Loc. (Chap. 7)	**1.262**	1.190	0.393	**1.298**	1.929	0.800	N/A	N/A	-	-
Ours	1.270	**1.172**	**0.328**	1.342	**1.926**	**0.776**	**1.854**	**1.387**	**1.345**	**1.170**

approach removes volatile points and reduces map size while maintaining superior overall localization accuracy. Interestingly, in sequence "20231213/2", localization in the non-maintained map fails at low compression rates due to the high amount of incorrect point correspondences, but works after heavy downsampling, when many of these wrong correspondences are removed.

8.3.4 Localization in Radar Maps

The third experiment supports our claim that our radar localization method achieves accurate pose estimates in the resulting map by combining odometry and localization for scan matching. Due to the lack of map-based localization methods, we adapt Radar-ICP from Chapter 4 to perform scan matching against a pre-built map and use it as our baseline. We display the best map-based localization approach in **bold**, and also provide online pose estimation methods that do not leverage a prior map, including the original Radar-ICP from Chapter 4, RaI-SLAM from Chapter 7, and RIV-SLAM [178]. The results are displayed in Figure 8.4, Table 8.4, and Table 8.5. The evaluation shows that odometry, which operates without any prior map, leads to substantial drift. Online SLAM approaches, on the contrary, achieve high accuracy in shorter parking lot sequences, but can result in drift over time. Localization approaches, however, can achieve reduced drift by leveraging a pre-computed map. Moreover, as previously displayed in Figure 8.4, our method overcomes the sparsity of radar scans and substantial environmental changes, where simple scan-to-map matching fails.

8.4 Conclusion

In this chapter, we presented a novel approach for long-term mapping and localization of autonomous vehicles that leverages automotive radars. We address the limitations of our single-session SLAM technique introduced in Chapter 7 and propose an approach that merges and localizes by exploiting multiple maps recorded at different times. Specifically, our radar-oriented strategy integrates the place recognition module from Chapter 6 to perform loop detection and verification between sessions, reducing the influence of incorrect place recognition detections. Moreover, by identifying inter-session constraints across multiple sequences, our approach reduces the dependence on a single reference sequence. We leverage the aligned sessions to perform map maintenance with a novel technique that constructs a temporal occupancy grid to filter out noise and movable objects from noisy radar maps. We then localize in the maintained maps with a reliable vehicle pose estimation approach that combines localization, odometry, and radar Doppler velocities, reducing the influence of point outliers. We implemented and evaluated our approach on real-world scenarios supporting all claims made in this chapter. We demonstrated how our multi-session alignment technique achieves state-of-the-art performance on automotive radar maps and how our map maintenance procedure identifies temporal changes in the map. Finally, our localization method achieves accurate pose estimation in sparse radar maps. Our work enables combining previous knowledge from the environment and leveraging it to achieve accurate localization, collectively enhancing the capabilities of long-term navigation using automotive radar data.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

Autonomous vehicles have the potential to mitigate human error and reduce traffic fatalities. To achieve reliable and safe navigation, these vehicles must be able to determine their pose independently of the driving scenario. Localization approaches that rely on GNSS degrade under dense foliage, in urban canyons, and in indoor scenarios such as tunnels and parking lots. Camera-based solutions face limitations under low-light and adverse weather conditions. And LiDAR systems are affected by adverse weather and high production costs. The main contributions of this thesis are novel approaches that leverage automotive radar sensors for pose estimation and mapping. These sensors are compact, robust against bad weather, and come at considerably lower production costs than LiDARs. Nevertheless, they present unique challenges such as high sparsity and noise in their output point clouds, which are specifically addressed in our proposed methods. In our work, we divide the task of radar localization and mapping into distinct components and address them individually.

In Chapter 4, we advance radar-only odometry and mapping. We propose a novel scan matching approach that combines the Doppler velocity information provided by automotive radars with point-to-point ICP, leading to state-of-the-art results comparable to LiDAR sensors. In addition, our velocity pre-filtering strategy handles scenarios where the sensor's field of view is being blocked by large dynamic objects. We also address the online mapping problem, where our simple yet effective approach reduces the number of outliers present in the radar map. The methods of this chapter focus on motion estimation and mapping on the plane. However, autonomous vehicles often navigate on roads with minor slope variations, requiring accurate vertical motion estimation.

We tackle this challenge in Chapter 5, where we propose a heuristic-based procedure that leverages the radar cross section information provided by the

sensor to segment the ground plane within radar scans. We leverage this ground plane during pose estimation, which mitigates vertical drift and achieves state-of-the-art pose estimation accuracy with minimal runtime overhead.

Our odometry techniques, however, do not leverage information from the past, resulting in the accumulation of errors over time. To exploit previously recorded radar data, we present in Chapter 6 one of the first radar-specific place recognition approaches that identifies previously visited locations. We propose a descriptor encoder network that exploits the radar cross section information of the entire scan to measure the similarity between locations, and addresses radar sparsity and noise by identifying the scan points that are more reliable for place recognition. Our approach achieves state-of-the-art results and produces descriptors that are substantially more compact than existing methods. In addition, despite training our network only once for 3D radar sensors, the model is able to generalize to other datasets and radar sensors during SLAM, as introduced in Chapter 7, and multi-session mapping presented in Chapter 8.

We integrate our odometry and place recognition modules into a complete SLAM system described in Chapter 7, which leverages loop closure detection and the Doppler velocities provided by radars to achieve reduced drift. Moreover, we add an inertial measurement unit to enhance local trajectory estimates. Compared to existing radar-SLAM approaches, our method combines a local graph with a global graph, enhancing local and global trajectory consistency. Our system achieves state-of-the-art pose estimation accuracy for single-session radar-inertial SLAM.

Radar odometry and SLAM enable accurate trajectory estimation and mapping. However, these methods are designed for operating online. In Chapter 8 we introduce one of the first long-term map construction, map maintenance, and localization systems suitable for radar data. Our proposed approach combines multiple SLAM sessions into a single reference frame and filters out incorrect correspondences, achieving accurate trajectory alignment. Furthermore, we focus on the challenge of detecting changes in sparse radar maps, such as parked vehicles or new constructions. These map points can hinder localization performance by causing incorrect point correspondences between the current radar scan and the map. In our work, we present a novel approach to maintain maps over time and localize within the resulting maps, achieving improved localization accuracy.

All in all, our contributions to SLAM exploiting the properties of automotive radars have the potential to shape the future of radar-based pose estimation and mapping for autonomous driving. Our radar-specific techniques achieve state-of-the-art performance on odometry and place recognition. We further enhance our system by integrating these two modules into a full-SLAM system. We integrate an inertial sensor to improve our local trajectory performance and perform

loop closure for drift correction. Additionally, our work tackles the challenge of multi-session alignment, accounting for changes between different maps and performing localization within them. We evaluated and tested all our approaches using publicly available real-world datasets and demonstrated that our radar-specific methods surpass the state of the art in radar pose estimation and map construction. In summary, this work provides practical solutions to enhance localization and mapping using radar, which can help autonomous vehicles become safer and reduce traffic fatalities under all weather conditions.

9.2 Future Work

While our contributions provide promising directions toward achieving vehicle autonomy by exploiting the capabilities of automotive radars, future novel techniques will provide further improvements in terms of data availability and pose estimation accuracy. Moreover, semantic understanding within a map can be beneficial to enhance safe autonomous vehicle navigation.

One of the remaining challenges in radar localization and mapping for autonomous vehicles is the limited availability of radar data to the general public. Although several automotive radar datasets have been published during the recent years [21, 50, 78, 97, 139, 177, 203], most of them are short sequences, such as the nuScenes [21], truckScenes [50], and View of Delft [139] datasets, or present some inconsistencies in the ground truth, like the NTURadLM [203]. Moreover, only nuScenes and truckScenes provide extrinsic calibrations of the sensors with respect to the vehicle, which is necessary for certain velocity-based approaches like the ones based on the formulation from Kellner et al. [89]. Therefore, assembling a sensor platform that includes accurate extrinsic calibration and high-accuracy sensors for ground truth generation remains a crucial step in the development of reliable radar localization and mapping algorithms.

Another remaining challenge is to further improve radar pose estimation accuracy to consistently achieve LiDAR-level results. While the performance of the presented methods within this thesis has pushed forward the state of the art, the low resolution of radar point clouds can remain a challenge in sharp turns, where the radar is affected by substantial environment variations. Moreover, despite our ground-aware odometry approach providing encouraging results in pose estimation performance with slope variations, radar manufacturers sometimes label ground points as noise and remove them during the internal signal processing of the radar. We believe that estimating the local gravity vector from inertial measurements as proposed by Malladi et al. [129] in the LiDAR domain can aid in vertical motion estimation. The combination of radars with cameras can also help address this issue. As an additional component to radar scan matching, extract-

ing translation and rotation-invariant features from images that can be tracked across frames can help improve odometry accuracy, especially in the z-axis.

Another crucial aspect where the camera can be beneficial is during online navigation. In this thesis, we have focused on point cloud maps, since they are the direct output of automotive radar sensors. Nonetheless, even after filtering radar scans and performing map maintenance, radar point clouds may still contain outliers coming from noise or reflections. This can lead to errors during navigation where driveable areas are incorrectly considered occupied. By leveraging cameras, spatial occupancy estimation can be enhanced, and driveable regions can be better estimated than when relying solely on radar point clouds. The fusion of radar and camera sensors also results in broader possibilities in terms of place recognition. Multi-modal information can be combined within the place recognition system, where a weighting procedure could bias the output towards each sensor modality based on the lighting and weather conditions. Existing methods [54] directly fuse the information of both sensor modalities, which can enhance place recognition performance. Moreover, rather than using single radar scans for loop closure detection, leveraging temporal information from sequences of multiple scans [173] could also result in more reliable place recognition.

Finally, it can be beneficial for vehicle navigation to have an understanding of the elements within a map. This can support scene understanding and obstacle avoidance within pre-recorded maps. Semantic mapping techniques could add such information, enhancing the understanding of the surroundings. Moreover, since maps created by autonomous vehicles can be highly diverse, ranging from parking lots to logistics warehouses, it can be of great interest to leverage the characteristics of open-world vision models trained on large-scale data to integrate their information within radar maps. These models are typically trained on billions of images and text information. A dedicated labeling system can project the radar point clouds onto their corresponding camera images, transferring existing knowledge from camera images to radar point clouds. This has already been achieved for RGB-D camera sensors [146, 190], and can provide essential ways for 3D radar map understanding.

All in all, the methods presented in this thesis provide a foundation for radar localization and mapping for autonomous vehicles. While there are still open challenges regarding data availability, high-accuracy vertical pose estimation, and camera-radar sensor fusion, our approaches have shown a step forward toward achieving reliable pose estimation and mapping leveraging automotive radars.

Bibliography

[1] D. Adolfsson, M. Karlsson, V. Kubelka, M. Magnusson, and H. Andreasson. TBV Radar SLAM: Trust But Verify Loop Candidates. *IEEE Robotics and Automation Letters (RA-L)*, 8(6):3613–3620, 2023.

[2] D. Adolfsson, M. Magnusson, A. Alhashimi, A.J. Lilienthal, and H. Andreasson. LiDAR-Level Localization With Radar? The CFEAR Approach to Accurate, Fast, and Robust Large-Scale Radar Odometry in Diverse Environments. *IEEE Trans. on Robotics (TRO)*, 39(2):1476–1495, 2022.

[3] I. Alamanos, G.P. Moustris, and C.S. Tzafestas. Localization and Offline Mapping of High-Voltage Substations in Rough Terrain Using a Ground Vehicle. In *Proc. of the Mediterranean Conf. on Control and Automation (MED)*, 2024.

[4] R. Aldera, D. De Martini, M. Gadd, and P. Newman. Fast Radar Motion Estimation With a Learnt Focus of Attention Using Weak Supervision. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[5] R. Aldera, D. De Martini, M. Gadd, and P. Newman. What Could Go Wrong? Introspective Radar Odometry in Challenging Environments. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2019.

[6] R. Aldera, M. Gadd, D. De Martini, and P. Newman. What Goes Around: Leveraging a Constant-Curvature Motion Constraint in Radar Odometry. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7865–7872, 2022.

[7] Y. Almalioglu, M. Turan, C.X. Lu, N. Trigoni, and A. Markham. Milli-RIO: Ego-Motion Estimation With Low-Cost Millimetre-Wave Radar. *Sensors*, 21(3):3314–3323, 2020.

[8] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] J. Baek, J. Park, S. Cho, and C. Lee. 3D Global Localization in the Underground Mine Environment Using Mobile LiDAR Mapping and Point Cloud Registration. *Sensors*, 22(8):2873, 2022.

[10] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning Local Feature Descriptors With Triplets and Shallow Convolutional Neural Networks. In *Proc. of British Machine Vision Conf. (BMVC)*, 2016.

[11] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner. The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[12] D. Barnes and I. Posner. Under the Radar: Learning to Predict Robust Keypoints for Odometry Estimation and Metric Localisation in Radar. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[13] D. Barnes, R. Weston, and I. Posner. Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information. In *Proc. of the Conf. on Robot Learning (CoRL)*, 2019.

[14] J. Behley and C. Stachniss. Efficient Surfel-Based SLAM Using 3D Laser Range Data in Urban Environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[15] J. Bentley. Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*, 18(9):509–517, 1975.

[16] P. Besl and N. McKay. A Method for Registration of 3D Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992.

[17] J.L. Blanco-Claraco. A Modular Optimization Framework for Localization and Mapping. In *Proc. of Robotics: Science and Systems (RSS)*, 2019.

[18] K. Burnett, A.P. Schoellig, and T.D. Barfoot. Do We Need to Compensate for Motion Distortion and Doppler Effects in Spinning Radar Navigation? *IEEE Robotics and Automation Letters (RA-L)*, 6(2):771–778, 2021.

[19] K. Burnett, Y. Wu, D.J. Yoon, A.P. Schoellig, and T.D. Barfoot. Are We Ready for Radar to Replace LiDAR in All-Weather Mapping and Localization? *IEEE Robotics and Automation Letters (RA-L)*, 7(4):10328–10335, 2022.

[20] K. Burnett, D.J. Yoon, A.P. Schoellig, and T.D. Barfoot. Radar Odometry Combining Probabilistic Estimation and Unsupervised Feature Learning. *arXiv preprint*, arXiv:2105.14152, 2021.

[21] H. Caesar, V. Bankiti, A. Lang, S. Vora, V. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[22] K. Cai, B. Wang, and C.X. Lu. Autoplace: Robust Place Recognition With Single-Chip Automotive Radar. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[23] C. Campos, R. Elvira, J.J.G. Rodríguez, J.M. Montiel, and J.D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. on Robotics (TRO)*, 37(6):1874–1890, 2021.

[24] D. Casado Herraez, L. Chang, M. Zeller, L. Wiesmann, J. Behley, M. Heidingsfeld, and C. Stachniss. SPR: Single-Scan Radar Place Recognition. *IEEE Robotics and Automation Letters (RA-L)*, 9(10):9079–9086, 2024.

[25] D. Casado Herraez, F. Kaschner, M. Zeller, D. Muhle, J. Behley, M. Heidingsfeld, D. Cremers, and C. Stachniss. Ground-Aware Automotive Radar Odometry. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2025.

[26] D. Casado Herraez, M. Zeller, L. Chang, I. Vizzo, M. Heidingsfeld, and C. Stachniss. Radar-Only Odometry and Mapping for Autonomous Vehicles. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[27] D. Casado Herraez, M. Zeller, D. Wang, J. Behley, M. Heidingsfeld, and C. Stachniss. Multi-Session Mapping and Long-Term Localization for Autonomous Vehicles Using Radar. *IEEE Robotics and Automation Letters (RA-L)*, 11(3):2522–2529, 2025.

[28] D. Casado Herraez, M. Zeller, D. Wang, J. Behley, M. Heidingsfeld, and C. Stachniss. RaI-SLAM: Radar-Inertial SLAM for Autonomous Vehicles. *IEEE Robotics and Automation Letters (RA-L)*, 10(6):5257–5264, 2025.

[29] S. Cen and P. Newman. Precise Ego-Motion Estimation With MMWave Radar Under Diverse and Challenging Conditions. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[30] S. Cen and P. Newman. Radar-Only Ego-Motion Estimation in Difficult Settings via Graph Matching. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[31] P.H. Chan, S.S. Roudposhti, X. Ye, and V. Donzella. A Noise Analysis of 4D Radar: Robust Sensing for Automotive? *Sensors*, 25(10):18291–18301, 2025.

[32] M.Y. Chang, S. Yeon, S. Ryu, and D. Lee. SpoxelNet: Spherical Voxel-Based Deep Place Recognition for 3D Point Clouds of Crowded Indoor Spaces. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[33] H. Chen, Y. Liu, and Y. Cheng. DRIO: Robust Radar-Inertial Odometry in Dynamic Environment. *IEEE Robotics and Automation Letters (RA-L)*, 8(9):5918–5925, 2023.

[34] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss. OverlapNet: Loop Closing for LiDAR-based SLAM. In *Proc. of Robotics: Science and Systems (RSS)*, 2020.

[35] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss. SuMa++: Efficient LiDAR-based Semantic SLAM. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[36] X. Chen, I. Vizzo, T. Läbe, J. Behley, and C. Stachniss. Range Image-Based LiDAR Localization for Autonomous Vehicles. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[37] B. Choi, H. Kim, and Y. Cho. ReFeree: Radar-Based Efficient Global Descriptor Using a Feature and Free Space for Place Recognition. In *Proc. of the ICRA Workshop on Radar in Robotics*, 2024.

[38] J. Civera, A.J. Davison, and J.M. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Trans. on Robotics (TRO)*, 24(5):932–945, 2008.

[39] F. Dellaert and M. Kaess. Factor Graphs for Robot Perception. *Foundations and Trends in Robotics*, 6(1):1–139, 2017.

[40] P. Dellenbach, J.E. Deschaud, B. Jacquet, and F. Goulette. CT-ICP: Real-time Elastic LiDAR Odometry With Loop Closure. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[41] L. Di Giammarino, I. Aloise, C. Stachniss, and G. Grisetti. Visual Place Recognition Using LiDAR Intensity Information. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[42] F. Ding, A. Palffy, D.M. Gavrila, and C.X. Lu. Hidden Gems: 4D Radar Scene Flow Learning Using Cross-Modal Supervision. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[43] F. Ding, Z. Pan, Y. Deng, J. Deng, and C.X. Lu. Self-Supervised Scene Flow Estimation With 4D Automotive Radar. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):8233–8240, 2022.

[44] C. Doer and G.F. Trommer. An EKF-Based Approach to Radar Inertial Odometry. In *Proc. of the Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2020.

[45] J. Engel, V. Koltun, and D. Cremers. Direct Sparse Odometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(3):611–625, 2018.

[46] European Commission. Road Safety Statistics 2024 in More Detail. *Mobility and Transport*, 2025.

[47] J.M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera. CAM-Convs: Camera-Aware Multi-Scale Convolutions for Single-View Depth. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[48] Y. Fan, X. Du, L. Luo, and J. Shen. Fresco: Frequency-Domain Scan Context for LiDAR-Based Place Recognition With Translation and Rotation Invariance. In *Proc. of the Intl. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, 2022.

[49] Y. Fang, Y. Li, K. Qian, F. Tombari, Y. Wang, and G.H. Lee. LiLoc: Lifelong Localization Using Adaptive Submap Joining and Egocentric Factor Graph. *arXiv preprint*, arXiv:2409.10172, 2024.

[50] F. Fent, F. Kuttenreich, F. Ruch, F. Rizwin, S. Juergens, L. Lechermann, C. Nissler, A. Perl, U. Voll, M. Yan, and M. Lienkamp. Man Truckscenes: A Multimodal Dataset for Autonomous Trucking in Diverse Conditions. In *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*, 2024.

[51] S. Ferrari, L.D. Giammarino, L. Brizi, and G. Grisetti. MAD-ICP: It is All About Matching Data Robust and Informed LiDAR Odometry. *IEEE Robotics and Automation Letters (RA-L)*, 9(11):9175–9182, 2024.

[52] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting With Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[53] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Trans. on Robotics (TRO)*, 33(1):1–21, 2016.

[54] S. Fu, Y. Duan, Y. Li, C. Meng, Y. Wang, J. Ji, and Y. Zhang. CRPlace: Camera-Radar Fusion With BEV Representation for Place Recognition. *arXiv preprint*, arXiv:2403.15183, 2024.

[55] M. Gadd, D. De Martini, and P. Newman. Look Round You: Sequence-based Radar Place Recognition With Learned Rotational Invariance. In *Proc. of the IEEE Symp. on Position, Location and Navigation*, 2020.

[56] M. Gadd, D. De Martini, and P. Newman. Contrastive Learning for Unsupervised Radar Place Recognition. In *Proc. of the Intl. Conf. on Advanced Robotics (ICAR)*, 2021.

[57] M. Gadd and P. Newman. Open-RadVLAD: Fast and Robust Radar Place Recognition. In *Proc. of the IEEE Radar Conf. (RadarConf)*, 2024.

[58] A. Galeote-Luque, V. Kubelka, M. Magnusson, J.R. Ruiz-Sarmiento, and J. Gonzalez-Jimenez. Doppler-Only Single-Scan 3D Vehicle Odometry. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[59] A. Garcia-Hernandez, R. Giubilato, K.H. Strobl, J. Civera, and R. Triebel. Unifying Local and Global Multimodal Features for Place Recognition in Aliased and Low-Texture Environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[60] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[61] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics: The KITTI Dataset. *Intl. Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013.

[62] H. Gil, D. Lee, G. Kim, and A. Kim. Ephemerality Meets LiDAR-Based Lifelong Mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2025.

[63] R. Girod, M. Hauswirth, P. Pfreundschuh, M. Biasio, and R. Siegwart. A Robust Baro-Radar-Inertial Odometry M-Estimator for Multicopter Navigation in Cities and Forests. In *Proc. of the Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2024.

[64] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical Optimization on Manifolds for Online 2D and 3D Mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2010.

[65] M. Grupp. evo: Python package for the evaluation of odometry and SLAM. `https://github.com/MichaelGrupp/evo`, 2017.

[66] T. Guadagnino, B. Mersch, I. Vizzo, S. Gupta, M.V. Malladi, L. Lobefaro, G. Doisy, and C. Stachniss. Kinematic-ICP: Enhancing LiDAR Odometry With Kinematic Constraints for Wheeled Mobile Robots Moving on Planar Surfaces. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2025.

[67] S. Gupta, T. Guadagnino, B. Mersch, I. Vizzo, and C. Stachniss. Effectively Detecting Loop Closures Using Point Cloud Density Maps. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[68] K. Haggag, S. Lange, T. Pfeifer, and P. Protzel. A Credible and Robust Approach to Ego-Motion Estimation Using an Automotive Radar. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):6020–6027, 2022.

[69] B. Hexsel, H. Vhavle, and Y. Chen. DICP: Doppler Iterative Closest Point Algorithm. In *Proc. of Robotics: Science and Systems (RSS)*, 2022.

[70] M. Hilger, N. Mandischer, and B. Corves. RaNDT SLAM: Radar SLAM Based on Intensity-Augmented Normal Distributions Transform. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2024.

[71] M. Himmelsbach, F. Hundelshausen, and H. Wuensche. Fast segmentation of 3D point clouds for ground vehicles. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2010.

[72] M. Holder, S. Hellwig, and H. Winner. Real-Time Pose Graph SLAM Based on Radar. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[73] Z. Hong, Y. Petillot, A. Wallace, and S. Wang. RadarSLAM: A Robust Simultaneous Localization and Mapping System for All Weather Conditions. *Intl. Journal of Robotics Research (IJRR)*, 41(5):519–542, 2022.

[74] Z. Hong, Y. Petillot, and S. Wang. RadarSLAM: Radar Based Large-Scale SLAM in All Weathers. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[75] I. Hroob, S. Molina, R. Polvara, G. Cielniak, and M. Hanheide. Learned Long-Term Stability Scan Filtering for Robust Robot Localisation in Continuously Changing Environments. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, 2023.

[76] I. Hroob, S. Molina, R. Polvara, G. Cielniak, and M. Hanheide. Adaptive Robot Localization in Dynamic Environments through Self-Learnt Long-Term 3D Stable Points Segmentation. *Journal on Robotics and Autonomous Systems (RAS)*, 181:104786, 2024.

[77] X. Hu, J. Wu, J. Jiao, B. Jiang, W. Zhang, W. Wang, and P. Tan. MS-Mapping: An Uncertainty-Aware Large-Scale Multi-Session LiDAR Mapping System. *arXiv preprint*, arXiv:2408.03723, 2024.

[78] J. Huai, B. Wang, Y. Zhuang, Y. Chen, Q. Li, Y. Han, and C. Toth. Snail-Radar: A Large-Scale Diverse Dataset for the Evaluation of 4D-Radar-Based SLAM Systems. *arXiv preprint*, arXiv:2407.11705, 2024.

[79] J.T. Huang, R. Xu, A. Hinduja, and M. Kaess. Multi-Radar Inertial Odometry for 3D State Estimation Using mmWave Imaging Radar. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[80] J.J. Hwang, R. Xu, H. Lin, W.C. Hung, J. Ji, K. Choi, D. Huang, T. He, P. Covington, B. Sapp, Y. Zhou, J. Guo, D. Anguelov, and M. Tan. EMMA: End-to-End Multimodal Model for Autonomous Driving. *Trans. on Machine Learning Research (TMLR)*, 2025.

[81] S. Isele, F. Haas-Fickinger, and M. Zöllner. SERALOC: SLAM on Semantically Annotated Radar Point-Clouds. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2021.

[82] S. Izquierdo and J. Civera. Optimal Transport Aggregation for Visual Place Recognition. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[83] H. Jang, M. Jung, M.H. Jeon, and A. Kim. LODESTAR: Maritime Radar Descriptor for Semi-Direct Radar Odometry. *IEEE Robotics and Automation Letters (RA-L)*, 9(2):1684–1691, 2024.

[84] H. Jang, M. Jung, and A. Kim. RAPLACE: Place Recognition for Imaging Radar Using Radon Transform and Mutable Threshold. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[85] B. Jiang and S. Shen. Contour Context: Abstract Structural Distribution for 3D LiDAR Loop Detection and Metric Pose Estimation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[86] E. Jose and M.D. Adams. Relative Radar Cross Section Based Feature Identification With Millimeter Wave Radar for Outdoor SLAM. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2004.

[87] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental Smoothing and Mapping With Fluid Relinearization and Incremental Variable Reordering. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[88] A. Kamann, P. Held, F. Perras, P. Zaumseil, T. Brandmeier, and U. Schwarz. Automotive Radar Multipath Propagation in Uncertain Environments. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2018.

[89] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer. Instantaneous Ego-Motion Estimation Using Doppler Radar. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2013.

[90] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer. Instantaneous Ego-Motion Estimation Using Multiple Doppler Radars. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[91] G. Kim, B. Park, and A. Kim. 1-Day Learning, 1-Year Localization: Long-Term LiDAR Localization Using Scan Context Image. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1948–1955, 2019.

[92] G. Kim, Y. Park, Y. Cho, J. Jeong, and A. Kim. Mulran: Multimodal range dataset for urban place recognition. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[93] G. Kim, S. Choi, and A. Kim. Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments. *IEEE Trans. on Robotics (TRO)*, 38(3):1856–1874, 2021.

[94] G. Kim and A. Kim. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[95] G. Kim and A. Kim. Remove, Then Revert: Static Point Cloud Map Construction Using Multiresolution Range Images. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[96] G. Kim and A. Kim. Lt-mapper: A Modular Framework for LiDAR-Based Lifelong Mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[97] H. Kim, M. Jung, C. Noh, S. Jung, H. Song, W. Yang, H. Jang, and A. Kim. HeRCULES: Heterogeneous Radar Dataset in Complex Urban Environment for Multi-session Radar SLAM. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2025.

[98] K. Koide, J. Miura, and E. Menegatti. A Portable Three-dimensional LIDAR-based System for Long-term and Wide-area People Behavior Measurement. *Intl. Journal of Advanced Robotic Systems*, 16(2):1–16, 2019.

[99] K. Koide, M. Yokozuka, S. Oishi, and A. Banno. Globally Consistent and Tightly Coupled 3D LiDAR Inertial Mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[100] K. Koide, M. Yokozuka, S. Oishi, and A. Banno. GLIM: 3D Range-Inertial Localization and Mapping With GPU-Accelerated Scan Matching Factors. In *Proc. of Robotics: Science and Systems (RSS)*, 2024.

[101] J. Komorowski. Minkloc3d: Point Cloud Based Large-Scale Place Recognition. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[102] J. Komorowski. Improving Point Cloud Based Place Recognition With Ranking-Based Loss and Large Batch Training. In *Proc. of the Intl. Conf. on Pattern Recognition (ICPR)*, 2022.

[103] J. Kopp, D. Kellner, A. Piroli, and K. Dietmayer. Fast Rule-Based Clutter Detection in Automotive Radar Data. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2021.

[104] A. Kramer, K. Harlow, C. Williams, and C. Heckman. Coloradar: The Direct 3D Millimeter Wave Radar Dataset. *Intl. Journal of Robotics Research (IJRR)*, 41(4):351–360, 2022.

[105] F. Kraus, N. Scheiner, W. Ritter, and K. Dietmayer. Using Machine Learning to Detect Ghost Images in Automotive Radar. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2020.

[106] M. Kronauge and H. Rohling. New Chirp Sequence Radar Waveform. *IEEE Trans. on Aerospace and Electronic Systems*, 50(4):2870–2877, 2014.

[107] V. Kubelka, E. Fritz, and M. Magnusson. Do We Need Scan-Matching in Radar Odometry? In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[108] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[109] P.C. Kung, C.C. Wang, and W.C. Lin. A Normal Distribution Transform-Based Radar Odometry Designed for Scanning and Automotive Radars. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[110] P.C. Kung, C.C. Wang, and W.C. Lin. Radar Occupancy Prediction With LiDAR Supervision While Preserving Long-Range Sensing and Penetrating Capabilities. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):2637–2643, 2022.

[111] M. Lázaro, R. Capobianco, and G. Grisetti. Efficient Long-Term Mapping in Dynamic Environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[112] M.T. Lazaro, L.M. Paz, P. Pinies, J.A. Castellanos, and G. Grisetti. Multi-robot SLAM Using Condensed Measurements. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[113] D. Lee, H. Lim, and S. Han. GenZ-ICP: Generalizable and Degeneracy-Robust LiDAR Odometry Using an Adaptive Weighting. *IEEE Robotics and Automation Letters (RA-L)*, 10(1):152–159, 2024.

[114] S. Lee, H. Lim, and H. Myung. Patchwork++: Fast and Robust Ground Segmentation Solving Partial Under-Segmentation Using 3D Point Cloud. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.

[115] J. Li, B. Li, Z. Tu, X. Liu, Q. Guo, F. Juefei-Xu, R. Xu, and H. Yu. Light the Night: A Multi-Condition Diffusion Framework for Unpaired Low-Light Enhancement in Autonomous Driving. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[116] K. Li, M. Li, and U. Hanebeck. Towards High-Performance Solid-State-LiDAR-Inertial Odometry and Mapping. *IEEE Robotics and Automation Letters (RA-L)*, 6(3):5167–5174, 2021.

[117] X. Li, H. Zhang, and W. Chen. 4D Radar-Based Pose Graph SLAM With Ego-Velocity Pre-Integration Factor. *IEEE Robotics and Automation Letters (RA-L)*, 8(8):5124–5131, 2023.

[118] H. Lim, K. Han, G. Shin, G. Kim, S. Hong, and H. Myung. ORORA: Outlier-Robust Radar Odometry. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[119] H. Lim, S. Hwang, and H. Myung. ERASOR: Egocentric Ratio of Pseudo Occupancy-Based Dynamic Object Removal for Static 3D Point Cloud Map Building. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):2272–2279, 2021.

[120] H. Lim, L. Nunes, B. Mersch, X. Chen, J. Behley, H. Myung, and C. Stachniss. ERASOR2: Instance-Aware Robust 3D Mapping of the Static World in Dynamic Scenes. In *Proc. of Robotics: Science and Systems (RSS)*, 2023.

[121] H. Lim, M. Oh, and H. Myung. Patchwork: Concentric Zone-Based Region-Wise Ground Segmentation With Ground Likelihood Estimation Using a 3D LiDAR Sensor. *IEEE Robotics and Automation Letters (RA-L)*, 6(4):6458–6465, 2021.

[122] D. Lisus, K. Burnett, D.J. Yoon, R. Poulton, J. Marshall, and T.D. Barfoot. Are Doppler Velocity Measurements Useful for Spinning Radar Odometry? *IEEE Robotics and Automation Letters (RA-L)*, 10(1):224–231, 2025.

[123] Z. Liu and F. Zhang. Balm: Bundle Adjustment for LiDAR Mapping. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):3184–3191, 2021.

[124] K.L. Low. Linear Least-Squares Optimization for Point-To-Plane ICP Surface Registration. Technical Report TR04-004, Department of Computer Science, University of North Carolina at Chapel Hill, 2004.

[125] S. Lupfer, M. Rapp, K. Dietmayer, P. Brosseit, J. Lombacher, M. Hahn, and J. Dickmann. Increasing FastSLAM Accuracy for Radar Data by Integrating the Doppler Information. In *Proc. of the Intl. Conf. on Microwaves for Intelligent Mobility (ICMIM)*, 2017.

[126] W. Lyu, W. Ke, H. Sheng, X. Ma, and H. Zhang. Dynamic Downsampling Algorithm for 3D Point Cloud Map Based on Voxel Filtering. *Applied Sciences*, 14(8):3160, 2024.

[127] L. Ma, C. Kerl, J. Stückler, and D. Cremers. CPA SLAM: Consistent Plane-Model Alignment for Direct RGB-D SLAM. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

[128] Y. Ma, X. Zhao, H. Li, Y. Gu, X. Lang, and Y. Liu. RoLM: Radar on LiDAR Map Localization. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[129] M. Malladi, T. Guadagnino, L. Lobefaro, and C. Stachniss. A Robust Approach for LiDAR-Inertial Odometry Without Sensor-Specific Modeling. *arXiv preprint*, arXiv:2509.06593, 2025.

[130] M. Milford and G. Wyeth. SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2012.

[131] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[132] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proc. of the Conf. on Advancements of Artificial Intelligence (AAAI)*, 2002.

[133] R. Mur-Artal, J. Montiel, and J.D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. on Robotics (TRO)*, 31(5):1147–1163, 2015.

[134] P. Narksri, E. Takeuchi, Y. Ninomiya, Y. Morales, N. Akai, and N. Kawaguchi. A Slope-Robust Cascaded Ground Segmentation in 3D Point Cloud for Autonomous Vehicles. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2018.

[135] A. Nayak, D. Cattaneo, and A. Valada. RaLF: Flow-Based Global and Metric Radar Localization in LiDAR Maps. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[136] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proc. of the Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, 2011.

[137] Y.Z. Ng, B. Choi, R. Tan, and L. Heng. Continuous-Time Radar-Inertial Odometry for Automotive Radars. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[138] A. Ošep, T. Meinhardt, F. Ferroni, N. Peri, D. Ramanan, and L. Leal-Taixé. Better Call Sal: Towards Learning to Segment Anything in LiDAR. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2024.

[139] A. Palffy, E. Pool, S. Baratam, J.F. Kooij, and D.M. Gavrila. Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset. *IEEE Robotics and Automation Letters (RA-L)*, 7(2):4961–4968, 2022.

[140] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li. MULLS: Versatile LiDAR SLAM Via Multi-Metric Linear Least Square. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[141] Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, and C. Stachniss. PIN-SLAM: LiDAR SLAM Using a Point-Based Implicit Neural Representation for Achieving Global Map Consistency. *IEEE Trans. on Robotics (TRO)*, 40:4045–4064, 2024.

[142] Y.S. Park, Y.S. Shin, and A. Kim. Pharao: Direct Radar Odometry Using Phase Correlation. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[143] Y.S. Park, Y.S. Shin, J. Kim, and A. Kim. 3D Ego-Motion Estimation Using Low-Cost mmWave Radars via Radar Velocity Factor. *IEEE Robotics and Automation Letters (RA-L)*, 6(4):7691–7698, 2021.

[144] B. Peng, H. Xie, and W. Chen. ROLL: Long-Term Robust Lidar-Based Localization With Temporary Mapping in Changing Environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.

[145] G. Peng, H. Li, Y. Zhao, J. Zhang, Z. Wu, P. Zheng, and D. Wang. TransLoc4D: Transformer-based 4D Radar Place Recognition. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[146] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, et al. Openscene: 3D Scene Understanding With Open Vocabularies. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[147] R. Prophet, J. Martinez, J.C.F. Michel, R. Ebelt, I. Weber, and M. Vossiek. Instantaneous Ghost Detection Identification in Automotive Scenarios. In *Proc. of the IEEE Radar Conf. (RadarConf)*, 2019.

[148] C. Qian, Z. Xiang, Z. Wu, and H. Sun. RF-LIO: Removal-First Tightly-coupled LiDAR Inertial Odometry in High Dynamic Environments. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[149] M. Rapp, M. Barjenbruch, K. Dietmayer, M. Hahn, and J. Dickmann. A Fast Probabilistic Ego-Motion Estimation Framework for Radar. In *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*, 2015.

[150] M. Rapp, M. Barjenbruch, M. Hahn, J. Dickmann, and K. Dietmayer. Probabilistic Ego-Motion Estimation Using Multiple Automotive Radar Sensors. *Journal on Robotics and Autonomous Systems (RAS)*, 89:136–146, 2017.

[151] K. Retan, F. Loshaj, and M. Heizmann. Radar Odometry on $SE(3)$ With Constant Velocity Motion Prior. *IEEE Robotics and Automation Letters (RA-L)*, 6(4):6386–6393, 2021.

[152] H. Rohling. Radar CFAR Thresholding in Clutter and Multiple Target Situations. *IEEE Trans. on Aerospace and Electronic Systems*, AES-19(4):608–621, 1983.

[153] I. Roldan, A. Palffy, J.F. Kooij, D.M. Gavrila, F. Fioranelli, and A. Yarovoy. See Further Than CFAR: A Data-Driven Radar Detector Trained by LiDAR. In *Proc. of the IEEE Radar Conf. (RadarConf)*, 2024.

[154] F. Roos, M. Sadeghi, J. Bechter, N. Appenrodt, J. Dickmann, and C. Waldschmidt. Ghost Target Identification by Analysis of the Doppler Distribution in Automotive Scenarios. In *Proc. of the Intl. Radar Symposium (IRS)*, 2017.

[155] Ş. Săftescu, M. Gadd, D. De Martini, D. Barnes, and P. Newman. Kidnapped Radar: Topological Radar Localisation Using Rotationally-Invariant Metric Learning. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[156] J. Schlichenmaier, L. Yan, M. Stolz, and C. Waldschmidt. Instantaneous Actual Motion Estimation With a Single High-Resolution Radar Sensor. In *Proc. of the Intl. Conf. on Microwaves for Intelligent Mobility (ICMIM)*, 2018.

[157] L. Schwarzer, M. Zeller, D. Casado Herraez, D. Simon, M. Heidingsfeld, and C. Stachniss. Self-Supervised Moving Object Segmentation of Sparse and Noisy Radar Point Clouds. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2025.

[158] D.W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2015.

[159] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proc. of Robotics: Science and Systems (RSS)*, 2009.

[160] N. Serre. Porsche 911 GT3 RS – Track Day ASA + GTRO. `http://lotus.ze-forum.com/`, 2014. Photograph taken at Circuit Pau Arnos on 18 January 2014. Licensed under CC BY 2.0.

[161] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[162] T. Shan and B. Englot. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[163] S. Siva, Z. Nahman, and H. Zhang. Voxel-based Representation Learning for Place Recognition Based on 3D Point Clouds. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[164] J. Sola, J. Deray, and D. Atchuthan. A Micro Lie Theory for State Estimation in Robotics. *arXiv preprint*, arXiv:1812.01537, 2018.

[165] D. Steinhauser, O. Ruepp, and D. Burschka. Motion Segmentation and Scene Classification from 3D LiDAR Data. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2008.

[166] T.Y. Tang, D. De Martini, S. Wu, and P. Newman. Self-Supervised Learning for Using Overhead Imagery as Maps in Outdoor Range Sensor Localization. *Intl. Journal of Robotics Research (IJRR)*, 40(12-14):1488–1509, 2021.

[167] H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[168] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[169] T. Tuna, J. Nubert, Y. Nava, S. Khattak, and M. Hutter. X-ICP: Localizability-Aware LiDAR Registration for Robust Localization in Extreme Environments. *IEEE Trans. on Robotics (TRO)*, 40:452–471, 2024.

[170] M. Usuelli, M. Frosi, P. Cudrano, S. Mentasti, and M. Matteucci. RadarLCD: Learnable Radar-based Loop Closure Detection Pipeline. In *Proc. of the Intl. Joint Conf. on Neural Networks (IJCNN)*, 2024.

[171] M.A. Uy and G.H. Lee. Pointnetvlad: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[172] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss. KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters (RA-L)*, 8(2):1029–1036, 2023.

[173] O. Vysotska, T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Efficient and Effective Matching of Image Sequences Under Substantial Appearance Changes Exploiting GPS Prior. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[174] O. Vysotska and C. Stachniss. Lazy Data Association for Image Sequences Matching Under Substantial Appearance Changes. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):213–220, 2016.

[175] O. Vysotska and C. Stachniss. Effective Visual Place Recognition Using Multi-Sequence Maps. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1730–1736, 2019.

[176] D. Wang, Y. Duan, X. Fan, C. Meng, J. Ji, and Y. Zhang. Maroam: Map-Based Radar SLAM Through Two-Step Feature Selection. *arXiv preprint*, arXiv:2210.13797, 2022.

[177] D. Wang, H. Haag, D. Casado Herraez, S. May, C. Stachniss, and A. Nuechter. Doppler-SLAM: Doppler-Aided Radar-Inertial and LiDAR-Inertial Simultaneous Localization and Mapping. *IEEE Robotics and Automation Letters (RA-L)*, 10(9):9438–9445, 2025.

[178] D. Wang, S. May, and A. Nuechter. RIV-SLAM: Radar-Inertial-Velocity Optimization Based Graph SLAM. In *Proc. of the Intl. Conf. on Automation Science and Engineering (CASE)*, 2024.

[179] H. Wang, C. Wang, and L. Xie. Intensity Scan Context: Coding Intensity and Geometry Relations for Loop Closure Detection. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[180] Y. Wang, N. Funk, M. Ramezani, S. Papatheodorou, M. Popovic, M. Camurri, S. Leutenegger, and M. Fallon. Elastic and Efficient LiDAR Reconstruction for Large-Scale Exploration Tasks. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[181] X. Wei, I.A. Bârsan, S. Wang, J. Martinez, and R. Urtasun. Learning to Localize Through Compressed Binary Maps. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[182] J. Wen, J. Tang, H. Liu, C. Qian, and X. Fan. Real-time Scan-to-Map Matching Localization System Based on Lightweight Pre-built Occupancy High-Definition Map. *Remote Sensing*, 15(3):595, 2023.

[183] R. Weston, S. Cen, P. Newman, and I. Posner. Probably Unknown: Deep Inverse Sensor Modelling Radar. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[184] R. Weston, M. Gadd, D. De Martini, P. Newman, and I. Posner. Fast-MBYM: Leveraging Translational Invariance of the Fourier Transform for Efficient and Accurate Radar Odometry. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2022.

[185] L. Wiesmann, L. Nunes, J. Behley, and C. Stachniss. KPPR: Exploiting Momentum Contrast for Point Cloud-Based Place Recognition. *IEEE Robotics and Automation Letters (RA-L)*, 8(2):592–599, 2022.

[186] World Health Organization. Road Traffic Injuries. `https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries`, December 2023.

[187] T.X. Xu, Y.C. Guo, Z. Li, G. Yu, Y.K. Lai, and S.H. Zhang. TransLoc3D: Point Cloud Based Large-Scale Place Recognition Using Adaptive Receptive Fields. *Communications in Information and Systems*, 23(1):57–83, 2023.

[188] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Trans. on Robotics (TRO)*, 38(4):2053–2073, 2022.

[189] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, and R. Xiong. DiSCO: Differentiable Scan Context With Orientation. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):2791–2798, 2021.

[190] J. Yang, R. Ding, W. Deng, Z. Wang, and X. Qi. RegionPLC: Regional Point-Language Contrastive Learning for Open-World 3D Scene Understanding. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[191] L. Yang, S.M. Prakhya, S. Zhu, and Z. Liu. Lifelong 3D Mapping Framework for Hand-Held & Robot-Mounted LiDAR Mapping. *IEEE Robotics and Automation Letters (RA-L)*, 9(11):9446–9453, 2024.

[192] H. Yin, X. Xu, S. Lu, X. Chen, R. Xiong, S. Shen, C. Stachniss, and Y. Wang. A Survey on Global LiDAR Localization: Challenges, Advances and Open Problems. *Intl. Journal of Computer Vision (IJCV)*, 132(7):3139–3171, 2024.

[193] H. Yin, X. Xu, Y. Wang, and R. Xiong. Radar-to-LiDAR: Heterogeneous Place Recognition via Joint Learning. *Frontiers in Robotics and AI*, 8(661199):1–12, 2021.

[194] K. Yoneda, N. Suganuma, R. Yanase, and M. Aldibaja. Automated Driving Recognition Technologies for Adverse Weather Conditions. *Journal of the Intl. Association of Traffic and Safety Sciences (IATSS)*, 43(4):253–262, 2019.

[195] K. Yoneda, H. Tehrani, T. Ogawa, N. Hukuyama, and S. Mita. LiDAR Scan Feature for Localization With Highly Precise 3-D Map. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2014.

[196] Z. Yu, Z. Qiao, W. Liu, H. Yin, and S. Shen. SLIM: Scalable and Lightweight LiDAR Mapping in Urban Environments. *IEEE Trans. on Robotics (TRO)*, 41:2569–2588, 2025.

[197] J. Yuan, P. Newman, and M. Gadd. Off the Radar: Uncertainty-Aware Radar Place Recognition With Introspective Querying and Map Maintenance. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[198] M. Zeller, J. Behley, M. Heidingsfeld, and C. Stachniss. Gaussian Radar Transformer for Semantic Segmentation in Noisy Radar Data. *IEEE Robotics and Automation Letters (RA-L)*, 8(1):344–351, 2022.

[199] M. Zeller, D. Casado Herraez, B. Ayan, J. Behley, M. Heidingsfeld, and C. Stachniss. SemRaFiner: Panoptic Segmentation in Sparse and Noisy Radar Point Clouds. *IEEE Robotics and Automation Letters (RA-L)*, 10(2):923–930, 2024.

[200] M. Zeller, D. Casado Herraez, J. Behley, M. Heidingsfeld, and C. Stachniss. Radar Tracker: Moving Instance Tracking in Sparse and Noisy Radar Point Clouds. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[201] D. Zermas, I. Izzat, and N. Papanikolopoulos. Fast Segmentation of 3D Point Clouds: A Paradigm on Lidar Data for Autonomous Vehicle Applications. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[202] J. Zhang and S. Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Proc. of Robotics: Science and Systems (RSS)*, 2014.

[203] J. Zhang, H. Zhuge, Y. Liu, G. Peng, Z. Wu, H. Zhang, Q. Lyu, H. Li, C. Zhao, D. Kircali, S. Mharolkar, X. Yang, S. Yi, Y. Wang, and D. Wang. Ntu4dradlm: 4D Radar-Centric Multi-Modal Dataset for Localization and Mapping. In *Proc. of the IEEE Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2023.

[204] J. Zhang, H. Zhuge, Z. Wu, G. Peng, M. Wen, Y. Liu, and D. Wang. 4DRadarSLAM: A 4D Imaging Radar SLAM System for Large-Scale Environments Based on Pose Graph Optimization. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[205] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue. Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges. *Sensors*, 22(11):4208, 2022.

[206] Y. Zhuang, B. Wang, J. Huai, and M. Li. 4D iRIOM: 4D Imaging Radar Inertial Odometry and Mapping. *IEEE Robotics and Automation Letters (RA-L)*, 8(6):3246–3253, 2023.

[207] G. Zhuo, S. Lu, H. Zhou, L. Zheng, and L. Xiong. 4DRVO-Net: Deep 4D Radar-Visual Odometry Using Multi-Modal and Multi-Scale Adaptive Fusion. *IEEE Trans. on Intelligent Vehicles (TIV)*, 9(6):5065–5079, 2024.

List of Figures

List of Tables