Learning from Maps: Scalable Ground Truth Generation in Autonomous Driving

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

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

angenommene

Dissertation

von

M.Sc. Frank Joachim Bieder

aus Ostfildern

Tag der mündlichen Prüfung:	10.03.2026
Hauptreferent:	Prof. Dr.-Ing. Christoph Stiller
Korreferent:	Univ.-Prof. Dr.-Ing. Mirko Mählisch

Abstract

A reliable and robust map perception is a key enabler for scalable autonomous driving by supporting multiple essential functions: Detecting map elements from on-board sensors enables localization within a high-definition (HD) map, verification that a map is up-to-date, and on-the-fly perception of map information towards map-less driving paradigms. Traditional methods for training perception systems rely heavily on the costly, time-consuming, and often sensor-specific process of manually annotating sensor data. In contrast, this work presents a scalable approach for the generation of training data for machine learning models by leveraging HD maps as a supervision source across different sensor modalities and perception tasks.

The first part of the thesis details the specification, recording and processing of a multi-drive dataset that serves as a foundation for learning perception models directly from HD maps. Complementary mapping approaches are applied to create a comprehensive set of map layers. The contained map elements differ in geometric representation, abstraction level, and functional role and offer a diverse set of training targets for various perception tasks.

Based on this foundation, the second part of the thesis trains and evaluates multiple perception models across different sensor modalities and learning representations. Towards this goal, a bird's-eye view (BEV) perception model for online HD map construction from surround-view cameras is trained, while analyzing and optimizing the training regime across multiple dimensions. Complementary to the BEV representation, multiple perspective view map perception models are trained for different sensor modalities and tasks. One line of experiments generates pixel-accurate annotations for front-view cameras, where trained models robustly predict diverse map element classes, including fine-grained map elements such as dashed road markings, traffic lights, and traffic

signs. The effectiveness of the proposed approach is also demonstrated in a fully self-supervised cross-modal domain adaptation setting, transferring knowledge from a richly annotated image dataset to the Light Detection and Ranging (LiDAR) domain via automatically generated HD maps.

Overall, this work positions HD maps as a versatile source of supervision for training perception models, enabling automatic, scalable, and cost-efficient training data generation in autonomous driving.

Usage of Large Language Models To enhance the written presentation of this thesis, I used large language models (LLMs), including ChatGPT 4o/5, and GitHub Copilot. I used these models to linguistically and grammatically enhance and restructure individual sentences and text passages. All LLM-generated text was manually checked, and often revised further. The models were not used to generate new content. All methods, experiments, and results described in this thesis were developed independently or by the co-authors indicated in each case.

Kurzfassung

Eine zuverlässige Kartenwahrnehmung ist eine zentrale Voraussetzung für die Skalierbarkeit des autonomen Fahrens. Sie unterstützt hierbei eine Vielzahl von Aufgaben: Die Erkennung von Kartenelementen durch bordeigene Sensoren ermöglicht die Lokalisierung innerhalb hochauflösender (HD-)Karten, die Überprüfung ihrer Aktualität sowie die Online-Wahrnehmung von Kartenmerkmalen für das kartenlose Fahren. Herkömmliche Methoden zum Trainieren von Wahrnehmungssystemen stützen sich stark auf den kostspieligen, zeitaufwändigen und oft sensorspezifischen Prozess der manuellen Annotation von Sensordaten. Im Gegensatz dazu präsentiert diese Arbeit einen skalierbaren Ansatz zur Generierung von Trainingsdaten für das maschinelle Lernen, indem HD-Karten als Annotationsgrundlage für verschiedene Sensormodalitäten und Wahrnehmungsaufgaben genutzt werden.

Der erste Teil der Arbeit beschreibt die Spezifikation, Aufzeichnung und Verarbeitung eines Datensatzes mit Mehrfachbefahrungen, der als Grundlage für das Lernen von Wahrnehmungsmodellen aus HD-Karten dient. Dabei werden durch verschiedene Kartierungsansätze Kartenelemente erzeugt, die sich grundlegend in ihrem Erstellungsprozess und den enthaltenen Informationen unterscheiden. Die Kartenelemente variieren hinsichtlich ihrer geometrischen Darstellung, ihres Abstraktionsgrades und ihrer funktionalen Rolle für das autonome Fahren. Dadurch bieten sie eine vielseitige Grundlage für das Training unterschiedlicher Wahrnehmungsmodelle.

Im zweiten Teil der Arbeit wird dieser Datensatz genutzt, um mehrere Wahrnehmungsmodelle über verschiedene Sensormodalitäten und Lernrepräsentationen hinweg zu trainieren und zu evaluieren. Zunächst wird ausgehend von den Sensordaten eines Multi-Kamerasystems ein Wahrnehmungsmodell

trainiert, welches Kartenelemente in der Vogelperspektive schätzt. Ergänzend hierzu werden perspektivische Wahrnehmungsmodelle für unterschiedliche Sensormodalitäten und Aufgaben trainiert. Ein Beispiel hierfür ist die Erzeugung von pixelgenauen Annotationen für Frontkameras, sodass es den trainierten Modellen gelingt, Kartenelemente verschiedener Kategorien robust zu klassifizieren. Darunter zählen auch feingranulare Kartenelemente wie gestrichelte Straßenmarkierungen, Ampeln und Schilder. Die Wirksamkeit des vorgeschlagenen Ansatzes wird zudem in einem vollständig selbstüberwachten Domain-Adaptation-Setting demonstriert, bei dem Wissen aus einem umfangreich annotierten Bilddatensatz über automatisch generierte HD-Karten in die LiDAR-Domäne und damit einer anderen unterrepräsentierten Sensormodalität übertragen wird.

Insgesamt präsentiert diese Arbeit HD-Karten als vielseitige Informationsquelle für die automatisierte, skalierbare und kosteneffiziente Generierung von Trainingsdaten im Kontext des autonomen Fahrens.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Forschungszentrum Informatik (FZI) in Zusammenarbeit mit dem Institut für Mess- und Regelungstechnik (MRT) des Karlsruher Institut für Technologie (KIT).

Mein besonderer Dank gilt meinem Doktorvater Prof. Dr.-Ing. Christoph Stiller für die Betreuung dieser Arbeit sowie für die hervorragenden wissenschaftlichen Rahmenbedingungen, die mir das freie Forschen in einem außerordentlich spannenden Umfeld ermöglicht haben. Ebenso danke ich Univ.-Prof. Dr.-Ing. Mirko Mählisch für sein Interesse an meiner Arbeit und die Übernahme des Korreferats.

Meinen aktuellen und ehemaligen Kolleginnen und Kollegen am MRT und am FZI danke ich herzlich für die stets angenehme Arbeitsatmosphäre und die hervorragende Zusammenarbeit. Besonders bedanken möchte ich mich bei den Mitgliedern meiner Forschungsgruppe: Fabi P., Jan, Annika, Janosch, Hao, Richard, Fabi I., Nils, Alex und Jonas. Ebenso danke ich Sascha, Sven, Pio, Chris, Jannik, Jonny, Flo, Kevin und Hendrik. Sie haben mich in den letzten Jahren mit wertvollem Feedback, spannenden Diskussionen und konstruktiver Kritik auf dem Weg zu dieser Arbeit begleitet und die Zeit durch unvergessliche Erlebnisse und Freundschaften stark geprägt.

Im Verlauf meiner Promotion durfte ich verschiedene Studierende betreuen. Im Zusammenhang mit dieser Arbeit möchte ich insbesondere Johannes, Oguzhan, Jian, Marc und Paras hervorheben. Mein Dank gilt außerdem dem Sekretariat, den Werkstätten und Werner, die mit großem Einsatz im Hintergrund dazu beigetragen haben, dass wir Doktorandinnen und Doktoranden uns auf unsere Forschung konzentrieren konnten.

Für die Möglichkeit, im Rahmen des MBA Fundamentals Program an der Hector School einen Blick über den fachlichen Tellerrand hinaus in die Welt der Unternehmensführung zu werfen, danke ich der Karlsruhe School of Optics and Photonics (KSOP) und der Robert Bosch GmbH für ihre finanzielle Unterstützung. Ein weiterer Dank gilt dem Karlsruhe House of Young Scientists (KHYS) sowie Professor Masayoshi Tomizuka für die Unterstützung meines Forschungsaufenthalts an der University of California, Berkeley. Auch der fachliche Austausch mit Forschern der Mercedes-Benz AG im Rahmen verschiedener Kooperationsprojekte war für mich äußerst wertvoll und bereichernd.

Abschließend danke ich allen, die mich während der Promotion mit Freundschaft, Verständnis, einem offenen Ohr und einer großen Portion Verrücktheit begleitet und diese Zeit für mich zu etwas ganz Besonderem gemacht haben. Mein größter Dank gilt jedoch meinen Eltern, Gudrun und Hubert, die mich stets unterstützt und mir in allen Phasen meines Lebens den Rücken gestärkt haben. Es ist von unschätzbarem Wert, einen Ort auf der Welt zu haben, an dem einen stets guter Rat, Geborgenheit und bedingungslose Liebe erwarten - und diesen Ort habt ihr mir gegeben. Ohne euch wäre ich heute nicht hier und nicht der Mensch, der ich bin. Euch sei diese Arbeit gewidmet.

Karlsruhe, den 30. Oktober 2025

Frank Joachim Bieder

Contents

Notation

This chapter introduces the notation and symbols which are used in this thesis.

Coordinate Systems and Poses

$\mathcal{F}_{\mathrm{A}}$	Coordinate system A
$\mathcal{F}_{\mathrm{R}}$	World coordinate system
$\mathcal{F}_{\mathrm{V}}$	Vehicle coordinate system
$\mathcal{F}_{\mathrm{S}}$	Sensor coordinate system
$\mathbf{R} \in \mathbb{R}^{3\times3}$	Rotation matrix
$\mathbf{t} \in \mathbb{R}^{3}$	Translation vector
$\mathrm{T} \in SE(3)$	Isometric transformation or pose
${}^{\mathrm{B}}\mathrm{T}_{\mathrm{A}}$	Transformation from coordinate system $\mathcal{F}_{\mathrm{A}}$ to $\mathcal{F}_{\mathrm{B}}$
T_i	Vehicle pose at timestamp i in coordinate system $\mathcal{F}_{\mathrm{R}}$
$\mathrm{T}_{\Delta ij}$	Vehicle pose delta between timestamp i and j in $\mathcal{F}_{\mathrm{R}}$
$\mathrm{T}_s^{\mathrm{E}}$	Pose of sensor s in $\mathcal{F}_{\mathrm{V}}$, extrinsics
$\mathbb{T}$	Set of poses
$\mathcal{T}$	Track, consisting of set of consecutive poses

Sensor Data and Models

$\mathrm{u} = (u, v)^{\mathrm{T}}$	Pixel on 2D image plane at pixel coordinates u, v
$\mathrm{c} = (x, y)^{\mathrm{T}}$	Cell on 2D grid map at cell coordinates x, y

$\mathrm{p} = (x, y, z)^{\mathrm{T}}$	Point in 3D space at point coordinates x, y, z
$\mathrm{u}_c = (u_c, v_c)^{\mathrm{T}}$	Principal point of camera model
$\mathtt{I}$	Image
I	Set of images
$\mathtt{P}$	Point cloud, set of points
P	Set of point clouds, set of set of points
f	focal length
$\mathscr{D}_{\mathrm{F}}$	Forward camera model
$\mathscr{D}_{\mathrm{B}}$	Backward camera model

Sensors and Maps

$\mathrm{C}^{\mathrm{c}}_{\mathrm{FV}}$	Front-view color camera, part of surround-view camera setup
$\mathrm{C}^{\mathrm{g}}_{\mathrm{LV}}$	Left-view greyscale camera, part of surround-view camera setup
$\mathrm{C}^{\mathrm{g}}_{\mathrm{RV}}$	Right-view greyscale camera, part of surround-view camera setup
$\mathrm{C}^{\mathrm{g}}_{\mathrm{BV}}$	Back-view greyscale camera, part of surround-view camera setup
$\mathrm{C}^{\mathrm{g}}_{\mathrm{SFV}}$	Front-view greyscale stereo camera setup
$\mathrm{C}^{\mathrm{g}}_{\mathrm{TFV}}$	Roof-top front-view greyscale camera
$\mathrm{C}^{\mathrm{g}}_{\mathrm{TBV}}$	Roof-top back-view greyscale camera
$\mathrm{L}^{128}_{\mathrm{AP}}$	Velodyne Alpha Prime LiDAR sensor with 128 beams

Maps

ℓ	Map element
$\mathscr{L}$	Set of map elements
M_{DS}	Physical map layer of dense map elements, representing ground surface and road markings
M_{LT}	Lane-level topological map layer, representing lanes and their connectivity
M_{EP}	Physical map layer of sparse, elevated map elements, representing poles, traffic signs, and traffic lights

Metrics and Evaluation

$d_{\mathrm{CD}}(S_1, S_2)$	Chamfer Distance between point sets S_1 and S_2
PR	Precision
RE	Recall
AP	Average Precision
$N_{\mathrm{TP},c}$	Number of true positives for class c
$N_{\mathrm{FP},c}$	Number of false positives for class c
$N_{\mathrm{FN},c}$	Number of false negatives for class c
IoU_c	Intersection over Union for class c
mIoU	Mean Intersection over Union
PQ	Panoptic Quality
RQ	Recognition Quality
SQ	Segmentation Quality
$^{\mathrm{th}}$	Indicator for *thing* classes
$^{\mathrm{st}}$	Indicator for *stuff* classes

Identification of Authorship

$[\cdot]^*$	The author of this work contributed as first author
$[\cdot]^\dagger$	The author of this work contributed as co-author

Others

$\mathbb{I}$	Identity matrix

Acronyms

2D	2-dimensional
3D	3-dimensional
ADAS	Advanced Driver Assistance System
AP	Average Precision
BEV	Bird's Eye View
DIRD	DIRD is an Illumination Robust Descriptor
FCN	Fully Convolutional Network
FOV	Field of view
FZI	Forschungszentrum Informatik
GNSS	Global Navigation Satellite System
GT	ground truth
IMU	Inertial Measurement Unit
IoU	Intersection over Union
JOSM	Java OpenStreetMap Editor

KHYS	Karlsruhe House of Young Scientists
KIT	Karlsruher Institut für Technologie
KSOP	Karlsruhe School of Optics and Photonics
LiDAR	Light Detection and Ranging
LL2MLconv	Map-to-Label Converter
mIoU	mean Intersection over Union
MRT	Institut für Mess- und Regelungstechnik
OSGeo	Open Source Geospatial Foundation
PQ	Panoptic Quality
RaDAR	Radio Detection and Ranging
SAE	Society of Automotive Engineers
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded Up Robust Features
SVP	single viewpoint
TAF	Testfeld Autonomes Fahren
TMS	Tile Map Service

1 Introduction

1.1 Towards Fully Automated Driving

The advent of fully automated driving promises substantial improvements in road traffic efficiency, comfort and safety: It increases efficiency by reducing traffic congestions, optimizing traffic flow and facilitating shared mobility, which ultimately reduces the number of vehicles on the road and CO2 emission [Pet18]. Besides eliminating the need for a steadily alert driver, prescient decision-making can improve passenger comfort through smooth and consistent driving behavior. Alongside this, safety is increased by minimizing human error and reaction times. In 2022, an estimated number of 5.93 million police-reported crashes occurred in the United States alone, resulting in more than 43 000 fatalities. This translates to 1.33 fatalities per 100 million miles traveled [NHT24] and makes the improvement of road safety a crucial societal goal.

The economic potential of autonomous driving is also substantial: recent estimates project a global market volume of $300 to $400 billion annually by 2035 [DEH23], with major contributions expected from Level 4 automation as defined by Society of Automotive Engineers (SAE) [SAE21], i.e., highly automated driving in specific conditions. Consequently, a growing number of companies are entering the field, ranging from established automotive manufacturers to big tech companies and startups. For example, Waymo [Way24b] and Cruise [Cru24] have launched their corporate robotaxi fleets in San Francisco in 2021 and 2022, respectively. Since then, Waymo has expanded its services to many US cities, i.e., Phoenix, Los Angeles, Austin, and soon to Miami [Way24a], reaching over 150 000 rides per week. At the same time, there is an increasing number of research-oriented robo-shuttle pilot projects, e.g., FABULOS [Fab21] or Testfeld Autonomes Fahren (TAF) [OFO23]. However,

these projects are still constrained to restricted service areas or specialized use cases, such as low-speed passenger shuttles or last-mile cargo transport.

1.2 Bridging the Gap between Map-based and Map-less Driving

In contrast, for a safe and scalable deployment of fully autonomous driving in a close-to-open world, each component of its hardware and software stack must operate with high reliability and robustness. These components range from the vehicle's sensor suite and perception system to its behavior generation modules and actuation system, as illustrated in Figure 1.1. Recently, the performance of perception systems has been evolving rapidly, with continuous advancements leading to significant improvements in accuracy and range of information for downstream modules [JGB20]. Besides the long-tail distribution of rare events and the awareness of unknowns, remaining challenges of the perception system include uncertainty caused by occlusions or sensor limitations like sensor range and measurement accuracy.

Here, HD maps may serve as an advanced and unique virtual sensor, offering insights into the static environment that exceed the capabilities of onboard sensors and causal processing systems with real-time constraints. The information they provide far exceeds physical road infrastructure or explicit road geometry: They include complex features such as lane topology or associations between traffic elements and lanes for which to infer a deep understanding of the road topology is required. Yet, the reliance on HD maps comes with high costs due to their cost-intensive creation and tedious maintenance caused by frequent changes in the so-called *static* world.

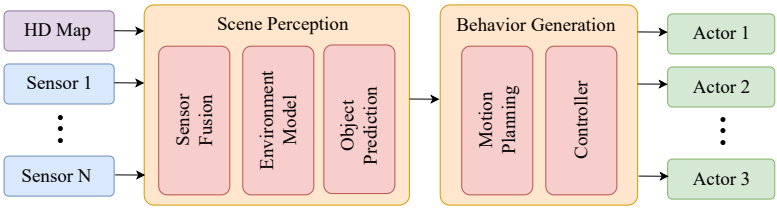


Figure 1.1: A schematic overview of an autonomous driving software stack. For a more detailed description, see [MM15].

Even with substantial investments in continuous updates, maps inevitably represent a snapshot of the past and may contain outdated information. Thus, many global players are doubting the feasibility of verified HD maps as a permanent prerequisite for the application of autonomous driving [Kar20]. These considerations paired with the recent advancements in perception systems have sparked a paradigm shift from map-dependent approaches towards map-less or less-map driving by perceiving map elements from on-board sensor data. Instead of relying on either a pure map-dependent or map-less solution, future approaches might combine the unique strengths of both worlds using HD maps as a prior or backup system to increase reliability of the overall system.

1.3 Map Perception and Its Applications

Perceiving map elements from on-board sensor data is referred to as *map perception* and consists of many sub-tasks depending on sensor modality, task representation and requirements set by downstream applications. This ability serves not only as a foundation for map-less driving, but also offers several other valuable applications. In the following, four key applications are identified and briefly described.

Semantic localization Semantic features of various representations can be used as landmarks for localization in semantic HD maps. For instance, [PSH18] localize with monocular semantic segmentation in a planning

map, while [KSP19] uses geometric primitives like poles or facades to localize in a map with corresponding features.

Map verification, validation and change detection Real-world features captured in an HD map are often described as static, whereas in reality, they should be considered only temporarily stable and in a state of constant change [LH21]. Thus, a map perception system to detect map changes is essential to be able to verify existing HD maps or render parts of the map as invalid [PSH18].

Automatic HD map generation and map update In contrast, map update, e.g., [PLH20], is the task of partially updating outdated map elements with new detections. For further reference, [WJY24] and [BHL23] have conducted extensive reviews for automatic mapping and map updates.

Less-map and map-less driving A central objective of map perception is to eliminate the necessity of a well-maintained and comprehensive HD map. The author of this work differentiates between less-map and map-less driving. The former refers to a transitional state in which the HD map can either be reduced to a simpler SD map or the continuous availability of a verified HD map is not essential at all times [WLL23, LCW23, WNS24].

Both the second and third applications can, to some extent, be conducted with aerial images. However, the coverage, fidelity and frequency of a respective application does tremendously benefit from using on-board sensor data of a vehicle fleet.

1.4 Goal and Contributions

The overarching goal of this thesis is to investigate the potential of HD maps as a supervision source for training perception models across various sensor modalities and tasks. Conventional approaches to training perception models typically depend on vast amounts of manually annotated sensor data, tightly coupled to a specific sensor modality and task. This costly and time-consuming dependence

not only limits scalability but also makes models susceptible to domain gaps across sensors. Given that HD maps encode rich and structured semantic information, the generation of training data directly from maps, here referred to as *learning from maps*, offers a promising alternative for map-related perception tasks and beyond. While the general idea is supported or already applied by various research directions, there is not much research which specifically and comprehensively targets this topic. This thesis leverages different map representations to train models for HD map construction as well as for general perception and domain adaptation tasks. It covers the entire process from sensor data collection over HD map generation and training regimes to model evaluation. The main contributions are summarized as follows:

- A novel multi-drive dataset is specified, recorded, and processed according to well-defined requirements that enable the exploration of HD maps as a supervision source. It features a comprehensive, well-calibrated sensor suite mounted in a close-to-production configuration, with at least four drives per sequence and a highly accurate multi-drive SLAM backbone. This dataset not only serves as a foundation for the conducted experiments in this work but also as a reference for future dataset designs.
- Three complementary HD mapping methods are employed to create diverse, high-quality map layers with centimeter-level precision. These layers differ fundamentally in their geometric representation, abstraction level, and functional role, as well as in their generation process. Combined, they yield a representative set of map elements for different learning tasks, covering a spectrum from physical road-surface representations to geometric models of traffic infrastructure, and topological, planning-grade map elements. This enables *learning from maps* across heterogeneous feature representations and learning tasks.
- Perception models are trained for multiple tasks and paradigms to evaluate the effectiveness of map-based supervision. This includes a BEV map perception model for online HD map construction from surround-view cameras. The in-depth study investigates the benefit of different training regimes, such as multi-drive supervision and heterogeneous camera

setups, along with the integration of recent advancements in label generation and definition to achieve state-of-the-art label quality.

- Lastly, *learning from maps* is explored in the context of perspective view map perception within two settings: First, the proposed XD-MAP framework performs fully self-supervised domain adaptation that transfers sensor-specific knowledge between sensor modalities using a semantic parametric map as a bridge between the domains. Unlike prior approaches, it does not require similar sensor characteristics or overlapping sensor fields of view and its effectiveness is demonstrated across multiple perception tasks. Second, pixel-accurate annotations for front-view cameras are generated from HD maps, enabling training of models for a wide range of map elements, from large road surfaces to fine-grained traffic light structures.

1.5 Outline

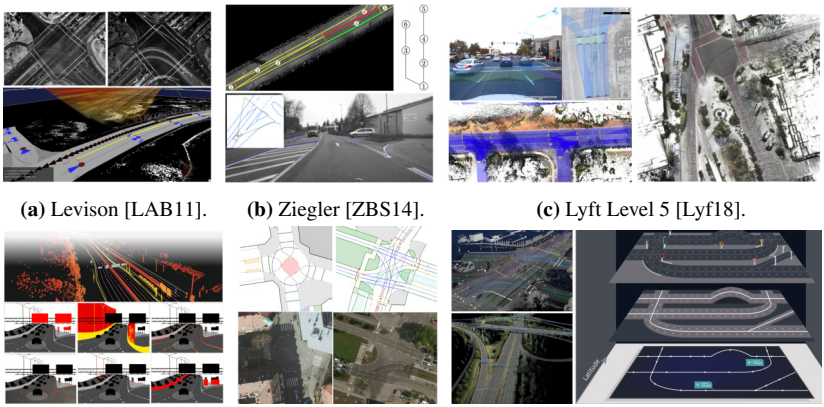
The remainder of this work is structured as follows: First, Chapter 2 and Chapter 3 review the corresponding related work and fundamentals. Chapter 4 guides through the creation of a multi-drive dataset to train perception models from HD maps, followed by the description and application of multiple HD mapping pipelines in Chapter 5. Chapter 6 and Chapter 7 comprise the training and evaluation of models for different perception tasks and sensor modalities by leveraging the created dataset. Finally, Chapter 8 summarizes the main contributions and discusses future research directions.

2 Related Work

This chapter provides an overview of related work in the context of using HD maps as a supervision source for training perception models across various sensor modalities and tasks. The following research components are relevant or related to this work and covered in this chapter: First, it reviews the design, format and generation of HD maps in general as well as their availability in public autonomous driving datasets. Second, efficient annotation methods for autonomous driving datasets are discussed, followed by related work to map perception. Finally, it introduces Simultaneous Localization and Mapping (SLAM) techniques that form the algorithmic foundation for creating consistent HD maps.

2.1 HD Map Design, Format and Generation

HD maps serve as a strong prior for understanding the static environment and complementing uncertainty-affected on-board perception, making them a crucial component in many autonomous driving stacks. One perspective is to view HD maps as a precomputed database able to combine data from multiple sources, viewpoints, and sensing modalities [Lyf18], effectively solving a subset of the autonomy problem by leveraging exhaustive computational resources and sensor data. Others [Atl20] consider HD maps as a digital twin, implying that it is not only a replica of the physical environment but also in direct communication with its physical counterpart [IBM22]. In autonomous driving, this might be presented by a permanently updated environment model, so that both map and model receive updates of the static environment and involved traffic participants [SW22].



(a) Levison [LAB11]. **(b)** Ziegler [ZBS14]. **(c)** Lyft Level 5 [Lyf18].

(d) TomTom [Tom18]. **(e)** Poggenhans [PPJ18]. **(f)** HERE Technologies [HER].

Figure 2.1: Examples of different HD map designs. Some layer architectures were used in autonomous driving projects, others were suggested in conjunction with a corresponding map format or are independently presented in research papers. A detailed overview of different layer compositions is given in Table A.2.

Due to the multiplicity of HD map applications, continuously changing requirements and an ecosystem of competing map providers, there is not yet one established standard or format for HD maps. Instead, various providers initially developed maps for Advanced Driver Assistance System (ADAS) systems and incrementally extended them to meet the growing demands of autonomous driving. This has led to a diverse set of map designs, layer structures, and formats. Prominent examples from academia and industry are depicted in Figure 2.1 alongside with a more detailed comparison of layer compositions in Table A.2. The research, conducted within this work, builds on the Lanelet2 map framework [PPJ18] and also contributed to a novel extension of Lanelet2, Map-to-Label Converter (LL2MLconv) [IFB25b][†], which bridges the gap between HD maps and training map perception models.

[†] To which the author contributed as co-author.

From map format to map framework Rather than focusing solely on map formats, there is a growing need for comprehensive and integratable map frameworks that not only define how to store spatial data but also provide a rich toolset to interact with HD maps. Lanelet2 [PPJ18, PJ20] is such a map framework, aiming to serve as a unified map format for a great variety of autonomous driving tasks and providing an unprecedented set of tools to access, manipulate and extend HD maps. Continuing the bottom-up modeling philosophy of its predecessor, LibLanelet [BZS14], Lanelet2 structures road networks using atomic segments called lanelets. This representation allows for precise modeling of complex road geometries and intersections. Consequently, it has become the de-facto standard for map datasets addressing prediction and planning tasks [ZSW19, BKM20, KMB20] and is also used in modern automated driving stacks [Aut23] due to its ROS [MFG22] integration paired with an efficient implementation. OpenDrive [HSH14], originally developed for road network description in driving simulations, is a coexisting popular HD map format in autonomous driving stacks. Similar to many other formats, it is modeled top-down, which makes the representation of complex road networks challenging and has no official map validation or certification process.

Map generation and mapping of geometric primitives Creating reliable, high-fidelity maps is often a labor-intensive process that involves manual annotation and quality control, increasing the interest in (semi-)automated map generation methods. The qualitative term high-fidelity is used to describe how the level of detail and the centimeter accuracy reflect the real-world, distinguishing global accuracy, i.e., positioning of features on the earth-scope, and local accuracy, i.e., positioning of features relative to its feature-neighbors [Ian18, Atl20]. Automated map generation approaches typically rely on sensor data collected either by commercial fleets or via crowdsourced individual vehicles. For example, basic SD map features like the road network are inferred from ego trajectories, for which [BE12] provides a survey. More recent approaches target the generation of HD maps, or a map element subset, by processing collected on-board sensor data or receiving semantic feature detections from an on-board processing unit [DDG17, IFG23]. Some methods also incorporate

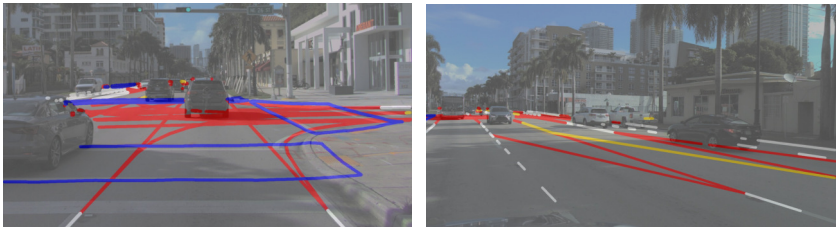
aerial imagery to extract physical features [PS10] or to infer high-level map elements like road topology [HB22]. Finally, hybrid approaches combine aerial and on-board data sources to enhance HD map generation [WMB22, MWF16], benefiting from the complementary strengths of both sources.

The choice of geometric primitives for map elements in automatic mapping approaches has become a common practice. Its suitability for a specific map element depends on many factors, ranging from its semantic class to its functional role in the automated driving stack. While a compact representation with fewer parameters is efficient to store and robust to estimate, many tasks require a minimum level of detail and benefit from geometric richness. Next to elements representing the road layout, traffic signs have been a typical class to be automatically detected and mapped [SPV13, VYF13], commonly modelled as planar structures in 3D space. In contrast, traffic lights are often mapped as points equipped with corresponding height and number of lights [FU11, LAB11]. Geometric primitives such as edges [ZS14], flat surfaces [ZS14, KSP19], and poles [SDS17, KSP19] have been proposed as compact and robust semantic landmarks, typically in the context of localization and mapping. A generic approach by [PSS21] proposes to detect semantic objects in camera images as instance masks and fuse them with LiDAR depth to estimate parametric representations. These parametric representations are semantically tailored to the object class, i.e., modelling poles and traffic lights as cylinders and traffic signs as planes.

2.2 HD Maps in Autonomous Driving Datasets

In the context of online map perception, three datasets, nuScenes [CBL20], Argoverse 2 [WQA21], and Waymo Open Motion dataset [Way24c], are particularly relevant as they provide a comprehensive sensor suite paired with meaningful HD maps. The first two are widely used for benchmarking recent research in vectorized online map construction, e.g., [LCW22, LCZ24, CWT24]. They are briefly characterized and compared in the following.

(a) Reprojection examples from nuScenes using the most current map update, version 1.3. The semantic map elements are annotated as 2D surfaces and distinguish eleven classes.



(b) Reprojection examples from Argoverse 2. The vectorized map elements are annotated in 3D and include lane boundaries, road markings, traffic directions, crosswalks and intersections.

Figure 2.2: Qualitative comparison of reprojected map elements in the sensor space. It underlines the importance of 3D map features in the context of maps as supervision signal.

NuScenes The nuScenes dataset [CBL20], recorded in Boston and Singapore, consists of 1000 20-second scenarios, aiming for a diverse set of locations, times and weather conditions. In contrast to many previous datasets, it strives to provide a comprehensive autonomous driving sensor suite including a camera ring with six 1600 px × 900 px cameras, a 32 beam spinning LiDAR, operating at 20 Hz, five Radio Detection and Ranging (RaDAR), and a GPS-IMU unit. Keyframes, selected with 2 Hz, are humanly annotated with 3D bounding boxes in 23 semantic classes. Each scenario is also paired with an HD map, distinguishing eleven semantic surface categories. However, the map is still limited to 2-dimensional (2D) features and does solely provide semantic information as flat polygons. Recently, Naumann et al. [NHG23] converted nuScenes maps into the Lanelet2 format, allowing explicit access to road geometry and topology.

Argoverse 2 The Argoverse 2 dataset [WQA21] was recorded in six US cities. Its synchronized sensor suite includes a ring with seven 2048 px × 1550 px RGB cameras, two stacked VLP-32 LiDAR sensors with a total of 64 beams and a front facing stereo camera. The corresponding *Sensor Dataset* includes 1000 15-second scenarios, annotated at 10 Hz, and a 3-dimensional (3D) lane-level annotated semantic vector map containing lane boundaries, marking types, traffic directions, crosswalks and more. Additional ground truth annotation for 30 object classes ranging from different types of vehicles to pedestrians and even wheelchairs are labeled amodal 3D bounding cuboids, which are labeled with a fixed size over time of one scenario. With the release of OpenLane-V2 [WLL23], Wang et al. build on Argoverse 2 and nuScenes by extending the lane-level map with lane connectivity and topology, in conjunction with lane-associated traffic elements and signals.

While, due to historic reasons, nuScenes is favored in many research papers, Argoverse 2 is the superior dataset in the author's view due to its spatial diversity, sensor suite, and vectorized rich 3D map features. In contrast to a sample size of 40 000, covering 5 km^2 in case of nuScenes, Argoverse 2 covers 17 km^2 with more than 150 000 annotated frames. The pioneering work of Lilja et al. [LFS24], in which the authors emphasize the importance of a strict geographic split of training and test set for benchmarking map perception, supports this view, as nuScenes suffers from a significantly more pronounced accuracy drop when evaluated with a strict geographic split compared to Argoverse 2. The study demonstrates that a map perception model trained on nuScenes struggles to generalize in non-visited scenarios, which might be due to its limited geographic coverage. The importance of 3D annotation of map elements is shown in Figure 2.2.

There are more datasets containing rich HD maps paired with drone recordings but these do not include sensor data from a vehicle's perspective [ZSW19, BKM20, KMB20]. Often these datasets are designed for tasks like behavior prediction or motion planning and are not further discussed in this work as they are not suitable for learning a perception from on-board sensor data. Other

datasets such as Lyft [Lyf18], TuSimple [TuS23] or Argoverse 1 [CLS19] provide both an HD map and on-board sensors but are inferior to the aforementioned datasets, hence less prominent in recent research and also not further discussed in this work.

2.3 Efficient Annotation of Autonomous Driving Datasets

Manually annotating a dataset is a tedious and time-intensive procedure, especially if each training instance is annotated manually for a specific task. This high effort increases with the growing complexity of features to be annotated, and the larger data volumes required for modern machine learning models. In the following, a selection of related work is presented that describes the creation of datasets for automated driving applications with a focus on reducing the required labor by automatic or semi-automatic methods.

For the pixel-wise semantic instance annotation of an image in the Cityscapes dataset [COR16], the authors reported an annotation time of 1.5 h per image. The reported time included the quality control of the ground truth (GT) and it is worth noting that Cityscapes set new standards in annotation quality at the time, acting as a widely used benchmark for semantic and instance segmentation still today. Similarly, [XKS16] reported a manual annotation time of 1 h and proposed a semi-automated 3D to 2D label transfer to speed up the process: After a static scene reconstruction from stereo and LiDAR, rough 3D primitives are annotated in a local 3D world. From this representation, 2D instance polygons are derived in the image domain by a back-projection. This work is extended by [LXG23], which also proposes to label 3D primitives in a local batch of accumulated point clouds. In comparison to [XKS16], the novel 3D to 2D inference model is updated and allows to include the label propagation of dynamic objects. Similarly, the previously mentioned datasets nuScenes [CBL20] and Argoverse 2 [WQA21] also provide semantic information in an HD map or, to be more specific, a format which is a map-like representation.

Further approaches have been developed to facilitate the annotation process of training data by using machine learning techniques to assist in the annotation process. As an example, [LGK19] outlines a method in which a human annotator sets initial conditions such as drawing bounding boxes or adjusting polygon points. Along with the data to be annotated, these inputs are repeatedly re-interpreted by a graph neural network, which incrementally refines the annotations while keeping the human annotator in the loop. This process continues until the human annotator approves the final annotation proposal.

A cost-effective way to generate vast amounts of training data is to derive it from a synthetic simulation. [RSM16] and [RVR16] are examples of datasets for urban semantic scene understanding, which aim to provide close-to photo-realistic camera images from a virtual city. In contrast, CARLA [DRC17] offers a highly customizable simulation environment with interactive, intelligent models for traffic participants and various virtual sensor models. This allows to simulate tailored traffic scenes fulfilling specific requirements regarding scene complexity, weather conditions, and GT definition. However, the domain gap between synthetic and real-world data remains a challenge and limits the direct applicability of models trained on synthetic data in the real world.

2.4 HD Map Perception

In the context of autonomous driving, perception refers to the process of interpreting on-board sensor data to understand the surrounding environment and the traffic scene. Perception systems vary widely depending on the sensor modality and data representation as well as the target task and inference approach. In this work, HD map perception includes all perception tasks that can be applied to perceive map elements or information to derive map-like features. Common perception tasks range from object detection, classification, and tracking to online HD map construction or completion. Recent breakthroughs in deep learning and computer vision, combined with the release of large-scale datasets and benchmark definitions, have fundamentally extended the boundaries of perception systems, increasing robustness, accuracy, and task complexity. While the landscape of perception tasks in autonomous driving is vast and a significant

share can be applied to online perception of HD maps, this section focuses on tasks used to evaluate the presented work in Chapter 6 and Chapter 7, namely semantic and panoptic segmentation and online HD map construction.

Pixel-wise and point-wise segmentation The first of three well-studied pixel-wise image segmentation tasks is *semantic segmentation*, which assigns a semantic class to each pixel in an image. Fully Convolutional Networks (FCNs) [LSD15, RFB15, YŞU19] long dominated this field, usually designed in an encoder-decoder architecture with convolutional network layers. Following the success in language and vision [KNH22], many recent approaches [CSK21, SGL21, ZLZ21] have shifted towards transformer-based architectures. The second task *instance segmentation* not only predicts the semantic class of objects but also distinguishes between different instances of the same class. However, solely pixels of instantiable classes, called *thing* classes, are considered, in contrast to *stuff* classes representing the background scene. Common approaches [HGD17, CV18] formulate it as a mask classification problem by predicting a binary mask along with a semantic class for each instance. Finally, *panoptic segmentation*, introduced by Kirillov et al. [KHG19], reformulates the joint semantic and instance segmentation of an image as a unified task, targeted by both FCN-based [KGH19] and transformer-based [CSK21, WZA21] approaches. While a semantic class is assigned to each pixel, objects representing a countable *thing* class are additionally distinguished as instances. In contrast to these specialized architectures, Mask2Former [CMS22] and OneFormer [JLC23] unify the segmentation tasks by achieving state-of-the-art results with one model architecture or even a single training in all three disciplines.

In contrast to images, LiDAR point clouds have an unordered structure with varying point density, making it crucial to design architectures to mitigate these challenges. While many approaches apply spherical [MVB19, KLC23, XWW20] or Bird's Eye View (BEV) projections [BWJ20, BLR21], others explore sparse attention or local feature strategies to directly process the point

cloud [QYS17, MXN21]. Further approaches partition the 3D space into regional groups [FPZ22] or use cylindrical partition for voxels [ZZW21] paired with asymmetrical 3D convolution to better fit point distributions.

Online HD map construction Originally introduced along with the HDMapNet [LWW22] architecture, the task of online HD map construction involves predicting a set of polygons and polylines that represent map elements in a 2D BEV grid, using onboard sensor data. The onboard sensor data typically consists of surround-view camera images, LiDAR point clouds, or a combination of both. While the original work adopts a multi-stage pipeline, i.e., first predicting a semantic map followed by a heuristic post-processing to vectorized instances, many follow-up works have reformulated the task as a single-stage problem, directly predicting vectorized map instances using a DETR-based transformer architecture [CMS20]. Prominent examples include VectorMapNet [LYW23], which detects vertices of each curve with an autoregressive decoding strategy and MapTR [LCW22], which introduces a hierarchical bipartite matching mechanism and concurrently infers a fixed number of points per map instance. The latter design is extended in [LCZ24] to MapTRv2 by incorporating a one-to-many query design, adding perspective semantic segmentation of map elements as an auxiliary supervision signal, and introducing a novel point-to-point matching strategy. Others explore further training strategies [ZLW23, CDF23] or curve representations [QDQ23, ZZD24]. Whereas earlier approaches constrained the standard perception range, both for prediction and evaluation, to 30 m longitudinally and 15 m laterally, more recent works have extended this near range setting to larger grid maps, e.g., 100 m $\times$ 50 m [YLW24] or even 240 m $\times$ 60 m [JZL24], particularly when incorporating far-seeing priors along with the single-shot sensor input. Examples for such far-seeing priors include pre-existing maps [SYL25, JZL24, IFB25a, IPF25], as well as accumulated sensor data from previous recordings in suitable representations [CWT24, XLY23, SCC24].

Online HD map construction has become one of the most prominent representatives of map perception and in the context of using HD maps as a supervision signal for training perception models. This development has mainly been driven

by the release of datasets which offer both a rich HD map and a comprehensive sensor suite, namely nuScenes and Argoverse 2 datasets. In addition to online HD map construction, BEV semantic segmentation is also performed on these datasets, often jointly with vehicle detection [ZK22, CCW22] in a coarse cell resolution of 25 cm to 50 cm. Other approaches such as [PCF23] also perform lane segmentation in a finer cell resolution of 15 cm.

2.5 Localization and Mapping in Autonomous Driving

Simultaneous Localization and Mapping (SLAM) [TBF05] is a fundamental task for mobile robotic systems, enabling an agent to localize itself within a map while concurrently enlarging, updating or refining it. Usually, a localization map layer consists of a set of landmarks, i.e., feature vectors, or other representations of encoded geographically distinctive information. Since there is a vast amount of literature on SLAM, the reader is referred to [CPA23] for a comprehensive survey on state-of-the-art techniques. The following briefly overviews visual localization methods, including feature-based and direct methods, as this is the focus of the presented work.

Feature-based methods, e.g., [LS14], rely on encoding distinctive local image information into feature vectors in order to match them across different frames. Depending on the application, the feature descriptor should be invariant to certain transformations like scale, rotation, and illumination changes. Prominent examples of feature descriptors are Scale-Invariant Feature Transform (SIFT) [Low04], introducing a scale and rotation invariant descriptor, Speeded Up Robust Features (SURF) [BTV06], adopting inspirations of SIFT while being notably faster, and ORB [RRK11], a quite fast binary descriptor performing well under transformations making it popular for real-time applications [MT17, CER21].

In contrast, direct methods are optimizing the pose graph by directly minimizing the photometric error between warped image pixels of consecutive frames. They are generally considered more robust in low-texture environments where

few distinctive landmarks are present. However, their high computational cost makes them less suitable for real-time applications. Consequently, many real-time SLAM systems, including the approach presented in this work, rely on feature-based methods. Regarding further literature on SLAM itself, the reader is referred to [ESC14, EKC18].

3 Fundamentals

Basic fundamentals essential for understanding the presented work are introduced in this chapter. It begins with an overview of the fundamentals required for sensor data capturing, processing and mapping in robotics, followed by the applied evaluation metrics in the evaluation chapter.

For further reading, Beyerer et al. [BPF16] offer an extensive overview of coordinate systems, transformations, and different aspects of image acquisition. An in-depth exploration of SLAM and place recognition algorithms is offered by Thrun et al. [TBF05] and Lategahn and Stiller [LS14] for visual SLAM in particular. For literature on machine learning and deep learning, the reader is referred to Bishop [Bis06], Goodfellow et al. [GBC16], and LeCun et al. [LBH15] for comprehensive overviews.

3.1 Isometric Transformation and Coordinate Systems

A 3D isometric transformation $\mathtt{T} \in SE(3)$ is a rigid body transformation between two coordinate systems that preserves Euclidean distances and angles between all transformed points [Ced04]. It is defined by a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$, which can be combined into

$$ {}^{\mathrm{A}}\mathtt{T}_{\mathrm{B}} = \begin{bmatrix} & \mathbf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1} $$

to form a single 4×4 matrix representing the isometric transformation from coordinate system B to A, denoted as $\mathscr{F}_{\mathrm{B}}$ and $\mathscr{F}_{\mathrm{A}}$ respectively. Given a point

$\mathrm{p}_\mathrm{B} \in \mathbb{R}^3$ in coordinate system $\mathscr{F}_\mathrm{B}$, it is transformed to $\mathscr{F}_\mathrm{A}$ by

$$\begin{bmatrix} \mathrm{p}_\mathrm{A} \\ 1 \end{bmatrix} = {}^{\mathrm{A}}\mathrm{T}_\mathrm{B} \begin{bmatrix} \mathrm{p}_\mathrm{B} \\ 1 \end{bmatrix}, \tag{3.2}$$

where the points p_A and $\mathrm{p}_\mathrm{B} \in \mathbb{R}^3$ are extended to homogeneous coordinates. Furthermore, multiple isometric transformations can be concatenated by matrix multiplication, i.e., ${}^{\mathrm{A}}\mathrm{T}_\mathrm{C} = {}^{\mathrm{A}}\mathrm{T}_\mathrm{B}\,{}^{\mathrm{B}}\mathrm{T}_\mathrm{C}$, and the inverse of an isometric transformation is given by

$$\left({}^{\mathrm{B}}\mathrm{T}_\mathrm{A}\right)^{-1} = {}^{\mathrm{A}}\mathrm{T}_\mathrm{B}, \quad \text{with} \quad {}^{\mathrm{A}}\mathrm{T}_\mathrm{B}\,{}^{\mathrm{B}}\mathrm{T}_\mathrm{A} = \mathbb{I} \in \mathbb{R}^{4\times4}. \tag{3.3}$$

In this work, multiple coordinate systems are defined: First, the *world coordinate system* $\mathscr{F}_\mathrm{R}$ serves as a fixed point of origin for HD maps and as the reference frame for ego-vehicle poses. Second, the *vehicle coordinate system* $\mathscr{F}_\mathrm{V}$ is a local coordinate system at a fixed point on the vehicle's body, typically located directly beneath the center of the rear axle on the ground level. As defined in `ISO8855` [Int11], its x-axis is directed to the front of the vehicle, the y-axis to the left and the z-axis to the top. In addition, each sensor has its own *sensor coordinate system* $\mathscr{F}_\mathrm{S}$, characterized by its position, orientation and type of sensor, for which this work follows the convention in [GLU12].

To improve readability, this work simplifies the definition of some reoccurring transformations. For instance, the vehicle pose at timestamp i in the world coordinate system $\mathscr{F}_\mathrm{R}$ is denoted as T_i, while the delta of two vehicle poses in $\mathscr{F}_\mathrm{R}$ from timestamp i to j is denoted as $\mathrm{T}_{\Delta ij}$. The pose of a mounted sensor s in the vehicle coordinate system, also referred to as the sensor's *extrinsics*, is denoted as T_s^E.

A set of poses $\{\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_n\}$, which have a consecutive order in time and represent a discrete sampling of a driven trajectory, are referred to as *track* or *drive* $\mathscr{T}$.

3.2 Camera Models

A camera model is a mathematical framework that describes the correspondence of a 3D point in the real world to a 2D point on the image plane. It is fully defined by either a forward model $\mathscr{P}_{\mathrm{F}} : \mathbb{R}^3 \to \mathbb{R}^2$, which map a 3D point to a 2D pixel on the image plane, or a backward model $\mathscr{P}_{\mathrm{B}} : \mathbb{R}^2 \to \mathbb{R}^3$, which maps a 2D pixel on the image plane to a viewing ray in 3D space.

In this work, two single viewpoint (SVP) camera models are primarily used, a *pinhole model* and a *spherical model*, each with individual strengths in different contexts. In general, SVP camera models are based on the assumption that all rays of light pass through a single projection center. For non-SVP models or a more comprehensive overview, the reader is referred to [SRG11, Bec21]. The parameters required for the definition of a camera model are called its *intrinsic* parameters. Both pinhole and spherical camera model are fully defined by three intrinsic parameters: The focal length $f \in \mathbb{R}^+$, i.e. the distance between the image plane and the focal plane, and the principal point $\mathrm{u}_c = (u_c, v_c) \in \mathbb{R}^2$, representing the image plane's shift, orthogonal to the optical axis in pixels.

Pinhole camera model Next to its single projection center, the ideal pinhole model assumes no lens distortions and a flat image plane. Due to its simplicity and linear forward model, it is widely used in computer vision and robotics. The closed-form forward model of an ideal pinhole model is given by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f\frac{x}{z} + u_c \\ f\frac{y}{z} + v_c \end{pmatrix}, \tag{3.4}$$

projecting a point $\mathrm{p} = (x,y,z)$ to a pixel $\mathrm{u} = (u, v)$ on the image plane. The backward camera model returns a viewing ray $\mathrm{p}(\lambda) \in \mathbb{R}^3$ for a given pixel $\mathrm{u} = (u, v)$ on the image plane by

$$\mathrm{p}(\lambda) = \begin{pmatrix} u - u_c \\ v - v_c \\ f \end{pmatrix} \lambda, \tag{3.5}$$

where $\lambda \in \mathbb{R}^+$ is the distance of the point to the camera center.

Spherical camera model In contrast to the pinhole model, the sphere camera model projects the scene onto a spherical surface. This makes it a more accurate model for wide-angle cameras, as it is better suited to model the radial distortion of lenses. In addition, it can represent sensors with a full 360° Field of view (FOV), which enables the model to be used for surround-view sensors, while a pinhole model is limited to strict FOV < 180°. Its forward model $\mathscr{P}_{\mathrm{F}}$ can be divided into two calculation steps: First, the spherical coordinates, i.e., azimuth angle ϕ and polar angle θ, are obtained by

$$\begin{pmatrix} \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{\sqrt{x^2+y^2}}{z}\right) \\ \arctan\left(\frac{y}{x}\right) \end{pmatrix} \tag{3.6}$$

to subsequently identify the pixel coordinates (u, v) by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_c \\ v_c \end{pmatrix} + f \begin{pmatrix} -\phi \\ \theta - \frac{\pi}{2} \end{pmatrix}. \tag{3.7}$$

After first determining the spherical coordinates

$$\begin{pmatrix} \theta \\ \phi \end{pmatrix} = \frac{1}{f}(c - u) + \begin{pmatrix} 0 \\ \frac{\pi}{2} \end{pmatrix}, \tag{3.8}$$

the viewing ray $\mathrm{p}(\lambda) \in \mathbb{R}^3$ can be derived by applying the backward model

$$\mathrm{p}(\lambda) = \begin{pmatrix} \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\theta) \end{pmatrix} \lambda. \tag{3.9}$$

3.3 Evaluation Metrics

This section defines commonly used evaluation metrics that are directly or indirectly applied in Chapter 7.

3.3.1 Chamfer Distance

The Chamfer Distance is a distance measure to quantify the similarity between two sets of points. While an initial formulation as Chamfer Matching method originates from [BTB77], intended to match segmented features in images, Fan et al. [FSG17] have reformulated the Chamfer Distance as a loss function to compare a set of predicted points with a corresponding ground truth. Given two point sets $S_1, S_2 \subseteq \mathbb{R}^3$, the Chamfer Distance is defined as

$$d_{\mathrm{CD}}(S_1, S_2) = \sum_{\mathrm{p}_i \in S_1} \min_{\mathrm{p}_j \in S_2} \|\mathrm{p}_i - \mathrm{p}_j\|^2 + \sum_{\mathrm{p}_j \in S_2} \min_{\mathrm{p}_i \in S_1} \|\mathrm{p}_i - \mathrm{p}_j\|^2. \tag{3.10}$$

The algorithm sums the squared distances from each point in S_1 to its nearest neighbor in S_2 and vice versa. It should be noted that the Chamfer Distance is not a distance metric in the mathematical sense, as it does not satisfy the triangle inequality.

3.3.2 Precision, Recall and Average Precision

The fraction of predictions which are correctly classified is referred to as *precision* PR and is calculated using the number of true positives N_{TP} and false positives N_{FP} by

$$\mathrm{PR} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FP}}}. \tag{3.11}$$

Similarly, the *recall* RE represents the fraction of ground truth examples that are correctly classified and is defined as follows

$$\mathrm{RE} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FN}}}, \tag{3.12}$$

where N_{FN} is the number of false negatives.

Usually, precision and recall exhibit an inverse relationship, i.e., increasing recall by including lower confident predictions reduces precision, while the opposite is true for solely including the most confident predictions. This trade-off is illustrated in the *precision-recall curve*, which plots precision against recall for different confidence thresholds. Since the quality of a prediction is characterized by both high precision and recall, the *average precision* AP is a performance indicator combining both measures by integrating the area under the curve with

$$\mathrm{AP} = \int_0^1 \mathrm{PR}_{\mathrm{RE}}(\mathrm{RE})\, d\mathrm{RE}, \tag{3.13}$$

where $\mathrm{PR}_{\mathrm{R}}(\mathrm{RE})$ is the precision as a function of the recall.

3.3.3 Pixel-Wise Segmentation Tasks

Semantic Segmentation A commonly used measure for semantic segmentation performance is the *Intersection over Union (IoU)* or *Jaccard similarity coefficient* [EEG14]. Given a set of C semantic classes, the per-class IoU for $c \in C$ is defined as

$$\mathrm{IoU}_c = \frac{N_{\mathrm{TP},c}}{N_{\mathrm{TP},c} + N_{\mathrm{FP},c} + N_{\mathrm{FN},c}}, \tag{3.14}$$

where $N_{\mathrm{TP},c}$, $N_{\mathrm{FP},c}$ and $N_{\mathrm{FN},c}$ are the number of true positives, false positives and false negatives for class c, respectively. Subsequently, the mean Intersection over Union (mIoU) is then calculated by

$$\mathrm{mIoU} = \frac{1}{|C|} \sum_{c \in C} \mathrm{IoU}_c. \tag{3.15}$$

Panoptic Segmentation For the unified task of semantic and instance segmentation, Kirillov et al. [KHG19] define

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in \mathrm{TP}} \mathrm{IoU}(p,g)}{|\mathrm{TP}|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|\mathrm{TP}|}{|\mathrm{TP}| + \frac{1}{2}|\mathrm{FP}| + \frac{1}{2}|\mathrm{FN}|}}_{\text{recognition quality (RQ)}} \tag{3.16}$$

as the performance metric Panoptic Quality (PQ) for each *thing* and *stuff* class c. It is the product of *segmentation quality* and *recognition quality*, two intermediate scores representing the F1 score and the average IoU of marched segments, respectively. While TP represents the matched pair of segments between prediction p and ground truth g, FN and FP are the pair of unmatched segments in the ground truth and prediction, respectively. One characteristic of this metric is that instances of arbitrary size still contribute equally to the score, making it sensitive to false positives with a small instance area.

4 A Multi-Drive Dataset for Learning Perception from HD Maps

This chapter aims to create a multi-drive dataset that serves as a foundation for learning diverse perception systems directly from HD maps. Towards this goal, key requirements are identified on the sensor setup, the recorded sequences, the processing pipeline, and map diversity, which fall short in at least one dimension in existing public datasets. The following key requirements are derived:

Scalable mapping via multi-drive SLAM A robust and accurate multi-drive SLAM framework enables the generation of training data at scale by acquiring drives under varying environmental conditions.

High-resolution sensing for high precision and fine-grained map features To not only perceive fine-grained map features, such as traffic lights and dashed lines, but also learn them from maps, imposes the requirements of high resolution sensors for the processing pipeline and reprojection.

Multi-sensor coverage for cross-modal domain adaptation and learning Given accurate multi-sensor calibration, the dataset supports the learning of perception models across different sensor modalities including cross-modal domain adaptation via HD maps.

Realistic, close-to-production sensor mounting By placing cameras behind the windshield and below side mirrors, this dataset uses sensor mounting positions common in production vehicles. This introduces real-world challenges such as reflections, distortion, and occlusions.

Heterogeneous HD map layers To enable research on a wide variety of training tasks, the dataset includes map annotations across several semantic

and geometric layers, ranging from planar road markings to elevated 3D models of traffic elements and relational topological networks.

Although several public datasets provide HD maps, see Section 2.2, none meet all above requirements simultaneously. Common sensor suite limitations range from no raw sensor data at all to low-resolution sensors or idealized sensor setups not reflecting close-to production setups. Further limitations include single-drive mappings and incomplete or simplified map layers.

This work presents the complete process of creating a multi-drive dataset for learning perception models from HD maps that meets all of the above criteria. First, the sensor setup, calibration process, and sequence recording are discussed. Second, the sensor data processing is detailed. Finally, the generation of the HD maps is covered in the next chapter.

4.1 Experimental Vehicle and Sensor Setup

For the recording, a Mercedes-Benz E class limousine, named *BerthaOne*, is chosen as the experimental vehicle. It already served successfully in previous research projects [ZBS14, TSP18], equipped with a comprehensive sensor suite and processing unit. As the computing unit, a general-purpose computer with an AMD EPYC 7702P 64-Core processor, 256 GB RAM and two NVIDIA GPUs, a Titan V and a Titan X was integrated into the trunk of the vehicle.

The applicability of the presented approach is demonstrated across different sensor modalities, i.e., LiDAR and camera. However, other sensor modalities like GNSS, IMU and wheel odometry are also necessary in order to build an accurate pose graph and post-processing of the sensor data. This section briefly introduces sensors in general and, subsequently, the specific sensor characteristics of the setup.

Table 4.1: Specifications of the sensor setup. All four cameras mounted behind windshields are BlackFly S [FLI20], equipped with a Meike fisheye lens [Mei17]. While the roof and left side cameras are BlackFly PGE-50S5M/C [Tel17], the right side camera is a Flir Flea3 GigE [FLI17], all of them paired with a Lensation Bm40 lens [Len20]. While c indicates RGB cameras and m grayscale cameras, w and h denote image width and height in pixel, θ_w the horizontal field of view in degree and θ_{res} the angular resolution.

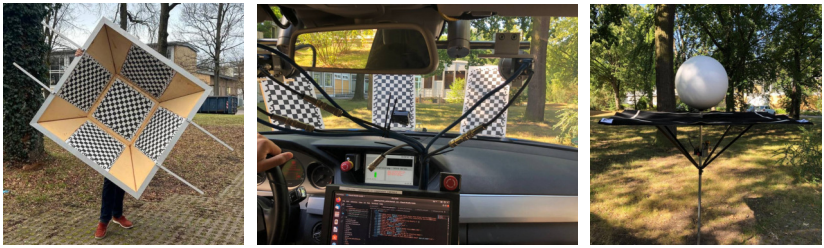
	surround-view vision				roof sensors		depth sensors	
	C^{c}_{FV}	C^{g}_{LV}	C^{g}_{RV}	C^{g}_{BV}	C^{g}_{TFV}	C^{g}_{TBV}	L^{128}_{AP}	C^{g}_{SFV}
channel	c	m	m	m	c	m	-	m
w [px]	4096	2448	1928	4096	2448	2448	1810	4096
h [px]	1536	1536	1448	1536	1360	1360	128	1536
θ_w [°]	130	101	101	130	120	120	360	130
θ_{res} [$\frac{\text{px}}{\circ}$]	32	24	19	32	20	20	5	32

Camera and LiDAR setup A wide-angle color camera and a stereo camera setup are mounted behind the front windshield, while a grayscale camera faces the rear window. Denoted as C^{c}_{FV}, C^{g}_{SFV}, and C^{g}_{BV}, respectively, these four cameras record a resolution of 4096 px × 1536 px with global shutter and a 130° field of view. Two additional grayscale cameras, C^{g}_{LV} and C^{g}_{RV}, are mounted beneath the left and right side mirror with solely their lenses exposed to the outside of the vehicle. Finally, the roof is equipped with two cameras, C^{g}_{TFV} and C^{g}_{TBV}, facing forward and backward.

The first four monocular cameras are considered as the default surround-view setup as all cameras are installed in a close-to-production position. The additional roof cameras serve as a reference and alternative mounting strategy with an elevated less-occluded view. All camera types and specifications are summarized in Table 4.1.

A Velodyne Alpha Prime L^{128}_{AP} [Vel20] is mounted in the center of the roof, featuring a 360° horizontal and a 40° vertical field of view. It continuously spins at 10 Hz and with 128 channels. Figure 4.1 shows a top-down view of the research vehicle with the mounting positions of the cameras and Velodyne Alpha Prime. As depicted in the top pictures, four additional Velodyne VLP-16 [Vel19] are mounted on the front and rear roof corners, yet are not further used in this work due to the superiority of the Alpha Prime.

Figure 4.1: Research Vehicle *BerthaOne* [TSP18] with corresponding sensor suite. The specifications of each sensor are further listed in Table 4.1.



(a) Frustum target. **(b)** Flat board targets. **(c)** Spherical target.

Figure 4.2: Calibration process of research vehicle *BerthaOne*, using three calibration targets with different geometries and purpose. The frustum target and the flat board targets are used to estimate both camera intrinsics and extrinsics. The spherical target is utilised for the camera-to-lidar calibration.

GNSS, IMU and inertial-wheel odometry A Global Navigation Satellite System (GNSS) sensor receives signals from multiple satellites and uses the time difference to estimate its global position on the earth's surface. The accuracy may vary depending on many factors such as the number of satellites in view, causing multipath effects caused by reflected signals and the system's inherent design. In contrast, an Inertial Measurement Unit (IMU) measures its linear acceleration and the angular velocity using multiple accelerometers and gyroscopes, typically aligned along three orthogonal axes. Many sensor data fusion approaches exploit the complementary strengths of the GNSS global positioning capabilities and the fine-granular relative motion estimation by an IMU [CDP06, SKH16]. Alternatively, an inertial wheel odometry sensor measuring a wheel's rotation and steering angle can serve to estimate the vehicle's motion [HWS22].

For this dataset, a Ublox C94-M8P GNSS [U-B22] receiver is used for global positioning, while a Xsens MTi-300 IMU [Mov22] mounted beneath the Alpha Prime on the roof and the wheel odometry are used jointly to estimate the vehicle's motion.

4.2 Sensor Calibration

In this section, the calibration process for the entire sensor setup, conducted prior to the recording, is discussed. Its quality has a crucial impact on consecutive computer vision tasks including object detection, pose estimation and mapping. Calibration of this setup is particularly challenging as the camera views do only partially overlap. In addition, three cameras are behind the windshield and one behind the rear window. It can be divided into three distinctive steps: First, the intrinsics of each camera and their extrinsics towards one particular camera, serving as temporary origin, are estimated. Second, the extrinsics of the LiDAR sensor towards the cameras are calculated. Lastly, the transformation from the sensor setup to the vehicle's rear axle is determined.

All camera and LiDAR sensors are synchronized by an external trigger signal while the cameras are activated at the same point in time as the LiDAR scanner

is directed forward. This ensures that sensor data from different sensors can be associated with a certain timestamp and a discrete vehicle pose. While the GPS and wheel odometry cannot be synchronized this way, they provide a high frequency stream of data with up to 100 Hz. After the calibration process is complete, an initial drive is conducted to review the data and adapt the sensor's meta parameters for the recording. For instance, this includes exposure time, gain factor, ROI, color depth, and image format for each camera.

Camera calibration For calibration of intrinsic and extrinsic parameters of a camera rig, this work follows Strauß et al. [SZB14] who employ a rigid 3D pyramidal calibration target with unique identification codes in every edge for global association of detected corners, see Figure 4.2a. A recording sequence where the frustum target is moved in front of each camera is complemented by a driven sequence of the vehicle in which a static arrangement of flat board targets in slightly different orientations are simultaneously recorded by multiple cameras, see Figure 4.2b. A full bundle adjustment and optimization follows in which poses of each target and parameter of each camera are jointly estimated by minimizing the reprojection error of all observed corners N_k in the full calibration sequence using a projection function $\mathscr{P}_{\mathrm{F}}$. Let j be the camera which observes the i-th corner at coordinate u_i, the sum of reprojection errors

$$E = \sum_{i=1}^{N_k} \left\| \mathscr{P}_{\mathrm{F}}(\mathrm{I}_j, \mathrm{p}_i) - \mathrm{u}_i \right\|^2 \tag{7}$$

is minimized, with I_j the estimated intrinsics of camera j, p_i the position of the i-th observed corner in the coordinate system of the respective camera w.r.t. estimated target poses and extrinsics. A tailored B-spline camera model [BS18, Bec21] is integrated in the calibration process to mitigate the effects of lens distortion by the windshield and fisheye lens. For each camera, both a pinhole and spherical camera model is derived as they offer unique advantages: while the spherical camera model offers a more accurate representation of the fisheye lens, the pinhole camera model is the default model in multiple frameworks, especially for vision-based BEV perception [CCW22, ZK22, LCW22].

Lidar-to-camera and sensor-to-rear-axle calibration While the previous approach estimated the relative camera poses towards a reference camera, the next steps are to estimate the pose of the LiDAR towards the reference camera and finally, find the transformation between sensor setup and the *vehicle coordinate system* $\mathcal{F}_{\mathrm{V}}$, as defined in Section 3.1. In order to achieve the LiDAR-to-camera calibration, this work follows [KKL18, Küm20] by applying a white spherical target, depicted in Figure 4.2c. The spherical target can be observed by both the LiDAR and camera sensors, where its edges can be detected with sub-pixel accuracy. As recommended by [KK20], a black cardboard is integrated to the target mount and placed behind the sphere to increase the contrast for edge detection in camera images. For the sensor-to-rear-axle calibration, [KWL19] extends the LiDAR-to-camera calibration approach of [KKL18] by adding additional external cameras to the scene, which observe the spherical calibration target and the rear wheel centers of the vehicle. By including the external cameras in the bundle adjustment optimization, the pose of the rear wheel towards the sensor rig can be estimated.

Online calibration While offline calibration can result in a high accuracy, it often requires a controlled environment, specific targets or conditions. Since calibration parameters change over time due to mechanical wear, vibrations or temperature variations, an online calibration approach can continuously monitor and adjust the calibration parameter during operation of a vehicle. Hu et al. propose TEScalib [HHB22]†, a targetless online calibration approach for the extrinsics of a stereo camera and a LiDAR sensor. It is a co-calibration approach which iteratively alternates between optimizing the extrinsics of the stereo camera itself and between stereo camera and LiDAR sensor using surface normal information and photometric objective functions. The approach is not further applied in this work, since the recordings were conducted in a short time frame for which the precise initial offline calibration results were sufficient.

4.3 Data Collection and Sequence Recording

As introduced in [BHS23]*, the dataset was recorded in Karlsruhe and Sindelfingen, Germany, containing eight unique sequences. This work focuses on a subset of five sequences in Karlsruhe all of which contain four drives each. All sequences form a loop in order to allow for loop closure in the mapping process and cover a wide range of traffic scenarios, as shown in Figure 4.3. These include single-lane and multi-lane roads in urban environments, often featuring bicycle lanes, pedestrian crossings, and intersections. In addition, the dataset contains roads situated on the outskirts, similar to highways.

After obtaining the multi-drive pose graph in Section 4.4.4, the initial drives are filtered to remove highly redundant frames permanently from the dataset. This allows for a consequent division between single- and multi-drive experiments and is conducted following two strategies: First, the drives are cut after the loop closure so that each drive only contains one stream of data of a unique scene. Second, the trajectory is filtered to keep a minimum distance τ_{dist} between two consecutive poses since the same scenery is occasionally captured by a vast number of frames due to traffic jams, stop lines, and red lights. The resulting dataset contains 22 km of unique road, 62 km of driven trajectory and a total of 65 000 frames, for which Table A.1 provides an overview.

Figure 4.3: Spatial distribution of the sequences in Karlsruhe, Germany. Imagery: [Esr23].

* To which the author contributed as first author.

4.4 Preprocessing of Sensor Data

For multiple steps in the map generation pipeline, the recorded sensor data needs to be preprocessed. This section summarizes some preprocessing steps, applied to the data base. A prerequisite of the following is the calibration and synchronization of the sensors as described in Section 4.1 and Section 4.2.

4.4.1 Stereo Vision

The goal of stereo vision is to estimate the 3D position of each pixel from two simultaneously captured images of the same scene. By identifying corresponding pixels in both images and their displacement, a disparity map is derived. A combination of the disparity map, the known relative poses of the cameras and the camera intrinsics yield a depth image of the scene. Typically, local stereo matching algorithms conduct a block-wise comparison and are prone to ambiguity caused by non-distinctive texture of surfaces. Before pixel pairs are matched, a rectification step [FTV00] is applied to project the stereo images into one common image plane. This transformation aligns the epipolar lines and reduces the search space for the disparity to a one-dimensional line.

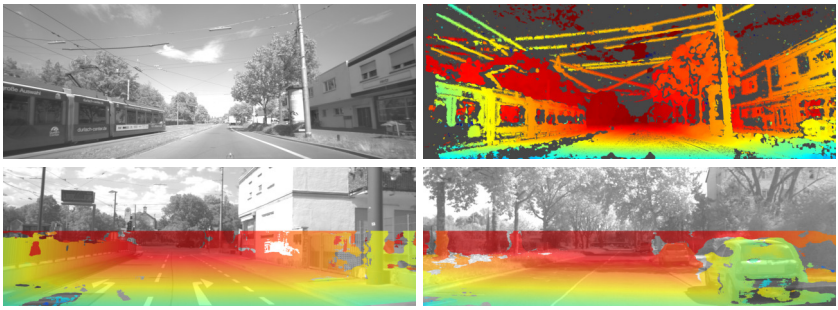


Figure 4.4: Stereo vision results on C^{g}_{SFV}. Compared to stereo vision results based on public datasets such as KITTI [GLU12], the presented setup is more challenging for disparity estimation, showcased in the right upper image with an open source solution [Bra00]. The mounting position behind the windshield causes reflections and distortions, and results in a shallow camera angle towards the road surface. However, even in this challenging scenario, [RS14] is capable of robustly estimating a dense disparity map of the road surface, as depicted in the lower images.

In this work, it is of particular interest when it comes to reconstructing the driveable road surface in 3D and, hence, to robustly estimate a dense disparity map of the road. As the road surface is viewed at a shallow angle and has very little distinctive texture, except for a few isolated road markings, conventional local stereo matching algorithms might fail to robustly produce a dense disparity map. In order to robustly estimate disparities in this challenging scenario, a real-time capable stereo matching algorithm modelling arbitrarily oriented slanted planes is employed [RD12, RS14]. It differs from other approaches [EE14, EE15] by not predefining the orientation of plane models. As depicted in Figure 4.4, the algorithm is able to robustly estimate the disparity of the road surface for the presented setup of stereo camera C^{g}_{SFV}.

4.4.2 Semantic and Instance Segmentation

For multiple processing steps, a semantic or instance segmentation of camera images is required. The performance of state-of-the-art DNNs depends on the model architecture, training regime and the underlying dataset. In this work, the models are employed on a custom and challenging sensor setup, imposing requirements to robustness and generalization capabilities. With the recent advances in model architecture and training techniques, a wide range of models exists which yield competitive yet similar performances, making the choice of the right dataset crucial for custom applications. Unlike many autonomous

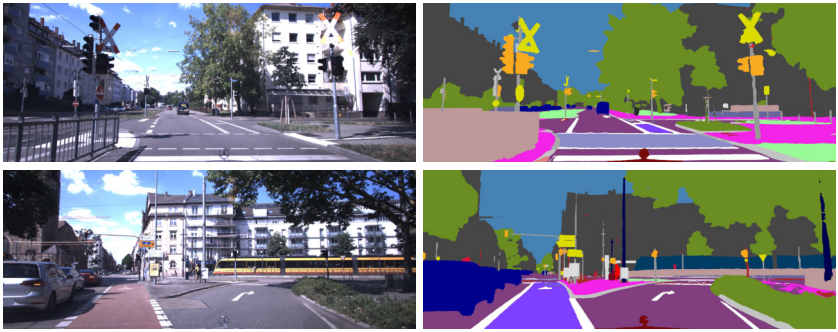


Figure 4.5: Semantic segmentation results using a Mask2former [CMS22] architecture trained on the Mapillary Vistas 2.0 dataset [KL21].

driving datasets, recorded using a single camera or a fleet with similar camera setups [COR16, CBL20, LXG23], the Mapillary Vistas dataset [NOB17] contains images captured by a wide range of camera setups, including different mounting positions, focal lengths and resolutions. Initially, distinguishing 66 categories, its upgrade [KL21] introduced a more refined taxonomy of 124 semantic categories with 70 instance-aware annotated.

A direct comparison with models trained on Cityscapes [COR16] confirmed Mapillary Vistas superiority and its suitability for this work. Regarding the model architecture, Mask2former [CMS22] was employed for most tasks due to its excellent general segmentation performance, see Figure 4.5, while Seamseg [PBC19] yields better results for small infrastructure objects making it particularly suitable for the automatic semantic mapping pipeline in Section 5.6.

4.4.3 LiDAR Preprocessing

A LiDAR point cloud is stored as a set of 3D points. For further processing steps, the LiDAR point cloud is compensated for motion artifacts and a spherical range image projection is applied.

Motion compensation The Velodyne Alpha Prime suffers from a rolling shutter effect as it is constantly spinning on a moving platform while taking depth measurements. It causes the point cloud to be distorted and cannot be trivially referenced to a single pose. This effect can be compensated for static objects by desweking the points linearly and considering the sensor's motion. By incorporating its relative pose $\mathrm{T}_l^{\mathrm{E}}$ in the vehicle coordinate system $\mathcal{F}_{\mathrm{V}}$, its traveled trajectory

$$\mathcal{T}^{\mathrm{l}} = \{\mathrm{T}_1^{\mathrm{l}}, \mathrm{T}_2^{\mathrm{l}}, \ldots, \mathrm{T}_n^{\mathrm{l}}\} \text{ with } \mathrm{T}_i^{\mathrm{l}} = \mathrm{T}_l^{\mathrm{E}} \mathrm{T}_i$$

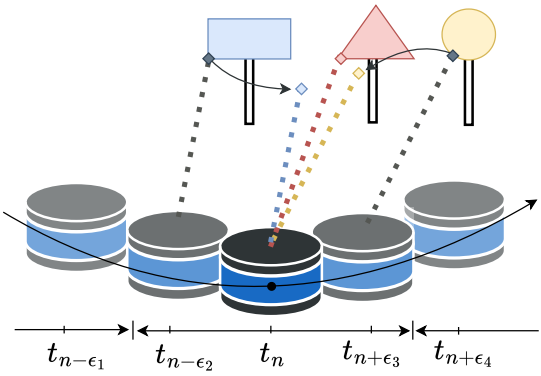


Figure 4.6: Illustration of motion compensation of LiDAR point cloud. All points measured in range $[t_{n-0.5\tau}, t_{n+0.5\tau}]$ with τ being 0.1 second are associated with the pose at time t_n and compensated for the motion of the sensor. In the illustration, the points recorded at time $t_{n-\epsilon_2}$ and $t_{n+\epsilon_3}$ with $\epsilon_2, \epsilon_3 < \tau$ are associated to t_n. While their uncompensated point positions of the blue and yellow sign are depicted in blue and yellow, respectively, the corresponding compensated points are colored gray.

can be derived from the ego trajectory. Next, the position of each point can be compensated for the motion between the respective time of the point measurement and the point clouds timestamp by interpolating between poses and applying a constant velocity model. Figure 4.6 illustrates the motion compensation of a LiDAR point cloud. While this work uses the pose graph obtained in Section 4.4.4, a prominent framework for LiDAR odometry estimation and motion compensation is KISS-ICP [VGM23].

Spherical camera model for range image projection Depending on the application, a LiDAR point cloud can also be represented as a 2D grid, usually referred to a range image where each pixel contains the distance and intensity of a point. This $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ projection of points to pixels is commonly achieved by a bijective geometric mapping strategy allowing to back-project the range image or information associated with the pixels into a 3D point cloud. In its simplest form, the projection incorporates the sensor's ray geometry, efficiently storing each vertically-stacked laser channels in a dedicated row and each different rotation position in a column. However, this representation does not account

for the strong inequality of the distribution of vertical channels and is prone to positional changes of LiDAR points due to motion compensation. In order to mitigate these limitations and achieve a more geometrically consistent scene representation, a spherical projection, similar to [MVB19], is applied to map LiDAR points onto a grid.

In this work, the basic spherical camera model, introduced in Section 3.2, is extended by incorporating independent horizontal and vertical resolutions, aligned with the respective minimum angular resolutions of the LiDAR sensor and realized by the focal lengths f_{u} and f_{v}, respectively. After integrating the different focal lengths, Equation (3.6) and Equation (3.7) can be rearranged to yield the projection of a LiDAR point (x, y, z) onto a pixel (u, v) on a spherical image with

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_c \\ v_c \end{pmatrix} + \begin{pmatrix} -\arctan\left(\frac{y}{x}\right) \cdot f_{\mathrm{u}} \\ \arctan\left(\frac{\sqrt{x^2+y^2}}{z}\right) \cdot f_{\mathrm{v}} - \frac{\pi}{2} \cdot f_{\mathrm{v}} \end{pmatrix}, \tag{4.1}$$

where u_c and v_c are the center pixel coordinates of the spherical projection.

In this case, multiple LiDAR points are projected onto the same pixel and the point with the smallest distance to the LiDAR sensor is selected and stored in the range image. The consequent information loss and the grid discretization of the spherical projection impede the artifact-free back-projection of the spherical range image into the 3D point cloud and can be mitigated by adjusting the angular resolutions depending on the application.

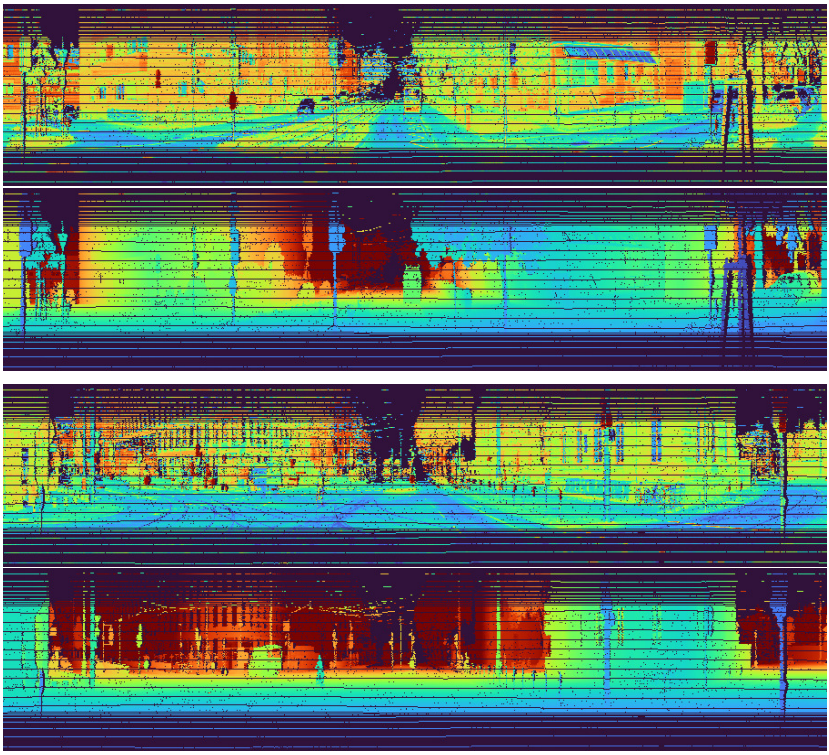


Figure 4.7: Spherical range image projection separating intensity and range layers. It depicts the front 180° of two scenes with intensity layer (top) and range layer (bottom) for each scene. The back-view is omitted for better visualization.

The spherical projection cannot only be used for processing LiDAR point clouds in a structured representation but also to fuse LiDAR and camera information by projecting LiDAR depth or intensity into the camera image. Lifting 2D vision-based detections with LiDAR depth information is a key component in different parts of this work like the parametric semantic mapping in Section 5.6 or the baseline approach for cross-modal domain adaptation in Section 7.1.1. Furthermore, the 2D spherical projected range images are used as 2D input for deep learning models in Section 7.1.2.

An example of the spherical range image projection is shown in Figure 4.7, separating intensity layer, and range layer and omitting back-view for illustration purposes.

4.4.4 Visual Multi-Drive SLAM

The SLAM system used in this work is required to meet several key criteria outlined at the beginning of this chapter, including general accuracy, robustness towards varying environment conditions such as illumination changes, and the support for multi-drive mapping. To support multi-drive mapping, the system must be capable of re-localizing an arbitrary drive to the localization layer of a reference drive. Computing an independent SLAM for each individual drive is not sufficient, as semantic map layers annotated relative to the pose graph of the reference drive would not be aligned on a centimeter or sub-centimeter scale with the independently mapped pose graph of other drives. Moreover, when the system is intended to be used over a longer period of time or under frequently changing environment conditions, it becomes essential to employ a robust descriptor and incorporate a map update mechanism for life-long mapping, which ensures that the map adapts to changes while keeping its complexity constant.

In order to fulfil these requirements, a comprehensive visual feature-based SLAM framework is employed, initially proposed by Lategahn and Stiller [LS14] and further extended by Sons et al. [SLK15, SLK17, SS18]. It

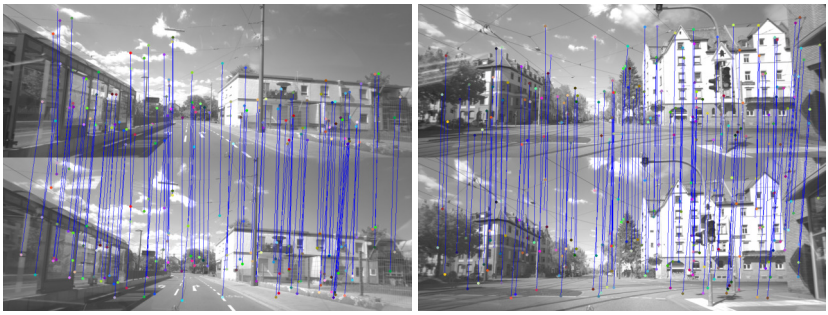


Figure 4.8: Feature point matching of DIRD features [LBS14] between left and right image of the stereo camera C^{g}_{SFV}. Non-maxima suppression is applied in a region of 6×6 pixels. For illustration purposes, solely every 20^{th} feature point is shown.

consists of two main components: First, a localization layer using a coarse feature-based place recognition approach, and second, a precise pose estimation module based on feature correspondences. Lategahn et al. [LBK13] presented a system to identify and learn illumination robust image features. The resulting Haar feature-based descriptor, denoted DIRD is an Illumination Robust Descriptor (DIRD) and further optimized regarding computational effort in [LBS14], is employed in the SLAM framework [LS14]. Later extensions of the framework incorporate surround view [SLK17] and an efficient multi-drive support with map optimization [SLK15, SS18].

Qualitative results of the DIRD-based feature matching are shown in Figure 4.8 for two stereo image pairs of the dataset. A further qualitative evaluation of the multi-drive mapping performance is provided in Section 5.7 by the backprojection of map elements in three different drives. While the backprojection reveals a centimeter-accurate localization of different drives in most scenarios, it also shows that the localization quality suffers in some rare scenarios such as high turning rates in crossings.

4.4.5 Multi-Modal, Continuous-Time Trajectory SLAM

For most scenarios, the visual SLAM framework described above provides a highly accurate and consistent pose graph serving as the foundation for most of this work. However, in rare occasions, such as the turning maneuver shown in Figure A.3, the visual SLAM drifts for a short period of time, which is particularly detrimental for the XD-Map pipeline, presented in Section 7.1. It requires a particularly precise pose graph as the automatic semantic mapping of tailored landmarks is prone to very small pose errors which are propagated in the cross-modal domain adaptation process. Therefore, a continuous-time trajectory SLAM framework, proposed by Hu [Hu26], is used for the XD-Map pipeline. It utilizes features from camera and LiDAR data and can integrate measurements with varying frequencies to estimate trajectories based on uniform B-splines [HBL20]. While this approach offers a higher accuracy and robustness, its current implementation does not support lifelong mapping.

4.5 Conclusion

This chapter outlines the complete process of creating a multi-drive dataset for learning perception models from HD maps. It provides a detailed description of the sensor setup, the calibration process, the data recording, the preprocessing, and the multi-drive SLAM. In order to meet the dataset requirements, a comprehensive sensor suite, comprising high-resolution camera and LiDAR sensors, was installed in a close-to-production configuration and carefully calibrated to ensure accurate extrinsic and intrinsic parameters. A SLAM framework capable of mapping multi-drive sequences by incorporating surround-view vision was employed and, for specific applications, partially replaced by a multi-modal, continuous-time trajectory SLAM.

The interdependence between sensor resolution, calibration accuracy, SLAM quality, and HD map fidelity is evaluated in the next chapter by assessing the precision of back-projected map elements into the camera images across different drives.

5 HD Map Generation

In order to satisfy requirements set by this work, three complementary HD mapping pipelines were applied to the dataset. The first pipeline is a manual annotation of dense 2D road surface features in bird's eye view. A surface reconstruction and texturing approach is developed to serve twofold: The textured surface serves as a context basis for annotating road surface features relative to the driven trajectory and it provides a high-quality 3D reconstruction of the road surface to lift 2D annotations into 3D space. By making use of the surface reconstruction and texturing, the semi-automatic second pipeline infers a lane-level road network from structured cues such as road borders and lane directions. A framework which incorporates the StVO rule set and translates low-level annotations into relational and topological map elements is applied and yields a rich planning-level HD map layer. The third pipeline is a fully automatic mapping of semantically tailored landmarks. It makes use of semantic instance detections and geometric priors to estimate parametric 3D models of traffic signs, poles, and traffic lights.

The resulting map layers differ essentially in their geometric representation, abstraction level and functional role. Ranging from large surface features like solid line markings to higher-level features such as lane centerlines and fine-grained cylindrical traffic lights, the included map elements offer a comprehensive set of training targets for learning perception models from HD maps.

In this chapter, the process of HD map annotation and generation is discussed along with the definition of map elements, training samples, and training instances.

5.1 Map Elements, Training Samples and Training Instances

This section depicts recurring terms throughout this work and their relationships in the context of HD map information in different stages. *Map elements* are the fundamental entities of an HD map, defined by geometric, semantic, and relational properties. They can represent a wide range of information and their heterogeneity reflects their utility across different autonomous driving tasks. A *training sample* refers to a pair of input sensor data and their corresponding ground truth, associated with a unique timestamp and an ego-vehicle pose in an HD map. A training sample has a certain learning representation and projection space and typically comprises multiple training instances. A *training instance* is usually derived from one map element and represents its information or at least a subset of it. This may correspond to only a subset of the map element's information because the training sample geographically bounded, the map element is occluded, or the training task simplifies the information of the map element, e.g., by reducing its geometric complexity to a bounding box. In rare cases, a training instance can also be derived from multiple map elements, e.g., if information of multiple map elements is required to infer an instance or if a simplification of the task definition condenses multiple map elements into one training instance. An example of the latter would be dashed lines which overlap in the image plane and are represented by one polygon in case of semantic segmentation.

Characteristics of map elements Elements in HD maps are quite heterogeneous and can be categorized along multiple dimensions. The following taxonomy outlines some principal aspects that determine their structure and utility in autonomous driving applications:

Functional role HD map information can be used for diverse tasks in an autonomous driving stack including localization, navigation or planning tasks. This characteristic describes which tasks benefit from the information an element provides.

Level of abstraction It can range from physical features, like a geometric cylinder representing a traffic light, to the influence it has on its surrounding environment. A good example is given in [PPJ18] by defining three layers of abstraction for the Lanelet2 framework: The physical, the relational, and the topological layer.

Geometric richness A map element can be represented by different levels of detail, ranging from a simple point over polygon to complex parametric 3D models.

Confineness While confined map elements impose strict spatial constraints, e.g., dashed lines or traffic signs, others like road topology or road boundaries do not possess similar restricted spatial boundaries. Their continuity and geometry have implications for the representation in training samples, which usually bounds them to a certain local map.

Characteristics of training samples from maps With a focus on the reprojected training data rather than the input data, this section distinguishes between two types of characteristics. First, it starts with aspects that define the ground truth representation itself:

Learning representation It depends on the machine learning task for which the training data is generated and defines which semantic and geometric information is encoded. Prominent examples are 2D bounding box detection, panoptic segmentation or vectorized map construction.

Projection space It specifies the coordinate frame in which ground truth is projected. Commonly, it is connected to the projection space of the input data, e.g., perspective view of camera. Alternatively, the two spaces can be decoupled and the machine learning model performs a mapping between the two spaces, e.g., vectorized map construction from camera images in bird's eye view.

Range and resolution The range and resolution of the training data define the spatial extent and granularity of the training samples.

The second set of type characteristics describe the quality of the reprojection framework regarding consistent and meaningful training data generation.

Independence of annotation artifacts Manual and automatic mapping can yield diverse annotations of the same real-world elements, differing in polyline point placement, breaks between polylines, and also lane-to-lanelets division. To ensure consistency, these variations should be normalized and ideally be resolved to an unambiguous instance label representation in the training data.

Perceptability handling Mechanisms preventing the projection of training instances which are not perceptual in the sensor data or implicitly inferable increase the data quality. This includes handling of static or dynamic occlusions, map verification or field of view constraints.

Backprojection precision It is dependent on multiple factors of the annotation pipeline, e.g., dimensionality of the ego pose, 2D or 3D map elements and calibration quality.

In [IFB25b]†, some of the above characteristics are discussed as requirements for a meaningful data generation framework for map perception. This list can be seen as an extension adding more view-points. Guided by these principles, Immel et al. [IFB25b]† also released a new software module within the Lanelet2 framework, further denoted LL2MLconv, resulting in the first unified HD map framework for map-based driving and map perception tasks.

5.2 Scene Context for HD Mapping

There are different sources of information and data representations which provide the scene context in order to build meaningful maps. This holds true for human annotation as well as for automated mapping approaches. Sources for scene context range from aerial imagery to on-board sensor data and respective scene reconstruction approaches. While aerial imagery provides a high-level overview with little occlusions and geospatial correctness, they sometimes lack the resolution and detail which can be obtained from on-board sensor data. A

crucial drawback of relying solely on aerial imagery is that it imposes the need of a high precision geo-referencing between imagery and vehicle localization, which is challenging to obtain in satisfying quality. In contrast, approaches which provide scene context based on on-board sensor data are dependent on sensor calibration and localization quality. However, they not only provide a high level of detail, but also allow to locally adjust aerial imagery.

In this work both aerial imagery and a ground surface reconstruction based on LiDAR and camera data are employed simultaneously. While the ground surface reconstruction provides a fine-grained resolution of 2 cm within a radius of 5 m around the driven trajectory, the aerial imagery serves as a source for the wider context essential for distant opposite lanes or huge intersections. It is provided by ©Stadt Karlsruhe | Liegenschaftsamt and Esri World Imagery [Esr23]. The former offers a higher resolution, yet, it is limited to the city of Karlsruhe. For alignment of aerial imagery and road surface reconstruction, the tile-based surface reconstruction results, obtained in 5.3, are converted into a geo-referenced tile map pyramid, consisting of a tile hierarchy of different zoom levels. This structure, defined by Open Source Geospatial Foundation (OSGeo) [Ope] as Tile Map Service (TMS), allows to access the stored tile information on different zoom levels alongside other geospatial data via applications such as Java OpenStreetMap Editor (JOSM) [JOSM].

5.3 Tile-Based Reconstruction of Road Surfaces

This section covers the surface reconstruction approaches used in this work and an outlook on a successive publication. The presented approach aims to reconstruct and texture the road surface in a tile-based manner. It builds on a framework for stereo-based ground surface mapping by Poggenhans et al. [PSS15] and a LiDAR-based ground surface modeling by Hu [Hu26].

Tiles are assigned to key frames selected to have a distance of 30 m to each other. By dividing the environment into independently processed tiles of 30 m×30 m, the reconstruction process can be scaled to areas of arbitrary size. The reconstruction achieves a resolution of 2 cm×2 cm with a consistent cell coverage

which is significantly finer than typical aerial imagery. This enables the precise annotation of small-scale road features with centimeter-level local accuracy.

In the initial stereo-based ground surface mapping, the freespace and drivable regions are segmented in the stereo disparity maps using [PF10]. Subsequently, all pixels classified as drivable regions are projected onto a Cartesian grid in top view representation. Next, the vehicle motion is used to aggregate the projected pixels associated to one key frame and tile. In case of multiple pixels projected onto the same grid cell, the pixel with the largest disparity to account for the quadratic increase of stereo vision error with distance and the corresponding confidence. The photometric and height information of each cell's pivot pixel are stored in two separate grid layers. For the purpose of this approach, stereo disparity maps are computed as described in Section 4.4.1 and the pose graph obtained in Section 4.4.4 is employed for the vehicle motion.

Aiming to improve the accuracy, the initial approach was extended by incorporating LiDAR measurements to benefit from the geometric accuracy of their depth measurements. In a first attempt, the 3D stereo points were simply exchanged by LiDAR points and, towards this goal, a more advanced recursive fusion approach of LiDAR measurements in grid maps was developed within [Fen22], a master thesis supervised by the author. However, LiDAR-only approaches, although aggregated over time, yield significantly sparser point clouds and thus fail to consistently cover the grid cells in the desired resolution. Also, the LiDAR-only results lack the textural richness of the stereo-based approach. To address these limitations, a novel LiDAR-camera fusion approach was developed, leveraging the strengths of both modalities. This approach aims to combine the geometric accuracy of a LiDAR-based ground plane model with the rich texture information provided by camera images.

Fusion of monocular camera and LiDAR-based ground plane model In order to lift 2D image pixels to a 3D point cloud representing the ground surface, the viewing ray of each image pixel is intersected with the B-Spline plane model of the ground.

From the backward camera models in Section 3.2, a viewing ray

$$\mathrm{p}(\lambda) = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \lambda + \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} \tag{5.1}$$

can be derived for each image pixel (u, v), defined by a support point (c_x, c_y, c_z), a ray direction (r_x, r_y, r_z) and a scalar parameter λ, representing the position of the 3D point $(x_\lambda, y_\lambda, z_\lambda)$ along the viewing ray. The application of a semantic mask, as inferred in 4.4.2 to filter the image, is recommended to exclude pixels that belong to a dynamic class or even all non-ground pixels.

To obtain a semi-global B-Spline plane modelling the ground surface from aggregated LiDAR points, this work applies a two-stage approach proposed by [Hu26]: First, a ground plane is estimated for each single-shot LiDAR scan in order to identify and remove non-ground points. This is achieved by following the approach of Wirges et al. [WRB21]†. Here, ground planes are estimated based on range measurements of single LiDAR scans and represented by splines, utilizing the uniform B-splines framework by Beck [Bec21]. Its formulation imposes smoothness constraints on B-splines to restrain overfitting, in particular in areas with sparse detections. A penalization term on the second spline derivate leads to a constant incline extrapolation in areas with no detections. Points with a distance larger than a certain threshold to the estimated ground plane, set to 20 cm within this work, are considered non-ground and removed for further processing. In the second stage, solely ground points are processed and aggregated for batches of 300 LiDAR scans to estimate a refined, semi-global continuous B-Spline ground plane. This step is repeated each 300 timestamps, yielding multiple semi-global ground surface models. The B-Spline model can be defined as a function that maps the Cartesian coordinates (x, y) to a corresponding surface height $z_{\mathrm{bs}} = \mathrm{G}_{\mathrm{bs}}(x, y)$.

Given the B-Spline plane model and the viewing ray of each image pixel, the optimal depth parameter $\hat{\lambda}$ minimizing the distance between ground surface

model and viewing ray can be iteratively estimated by solving

$$\hat{\lambda} = \arg\min_{\lambda} \left\| G_{bs}(x_\lambda, y_\lambda) - z_\lambda \right\|^2 \tag{5.2}$$

for the viewing ray of each pixel. A solution $p(\hat{\lambda})$ is considered valid if the respective distance is below a certain threshold, set to 10 cm for this work.

Similar to the stereo-based approach, the lifted 3D points are aggregated into a tile-based point cloud representation. After lifting the camera pixels to 3D points, the further processing is equivalent to the stereo-based approach.

Results and discussion In the upcoming chapters, both stereo-based and fusion-based approaches are applied and evaluated. Qualitative results are shown in Figure 5.1 for texturing and Figure 5.2 for height estimation. While the left column depicts stereo-based results, both the middle and right columns show fusion-based results with different support point spacing, being 0.5 m and 0.2 m, respectively.

In general, the fusion-based approach yields a more consistent and continuous ground surface reconstruction. The approach with 0.2 m spacing is discarded in the following chapters in favor of 0.5 m spacing, as smaller support point spacing leads to a less smooth ground surface model and, hence, fulfill the minimum distance requirement set for the results of equation (5.2) less reliably. This yields a slightly sparser ground surface reconstruction. A further difference between stereo-based and fusion-based results are the wider mapped spatial areas. This is not a strict limitation of the stereo-approach, but a design parameter choice as the depth estimation based on the stereo-setup behind the windshield gets less reliable at the sides with an increasing viewing angle.

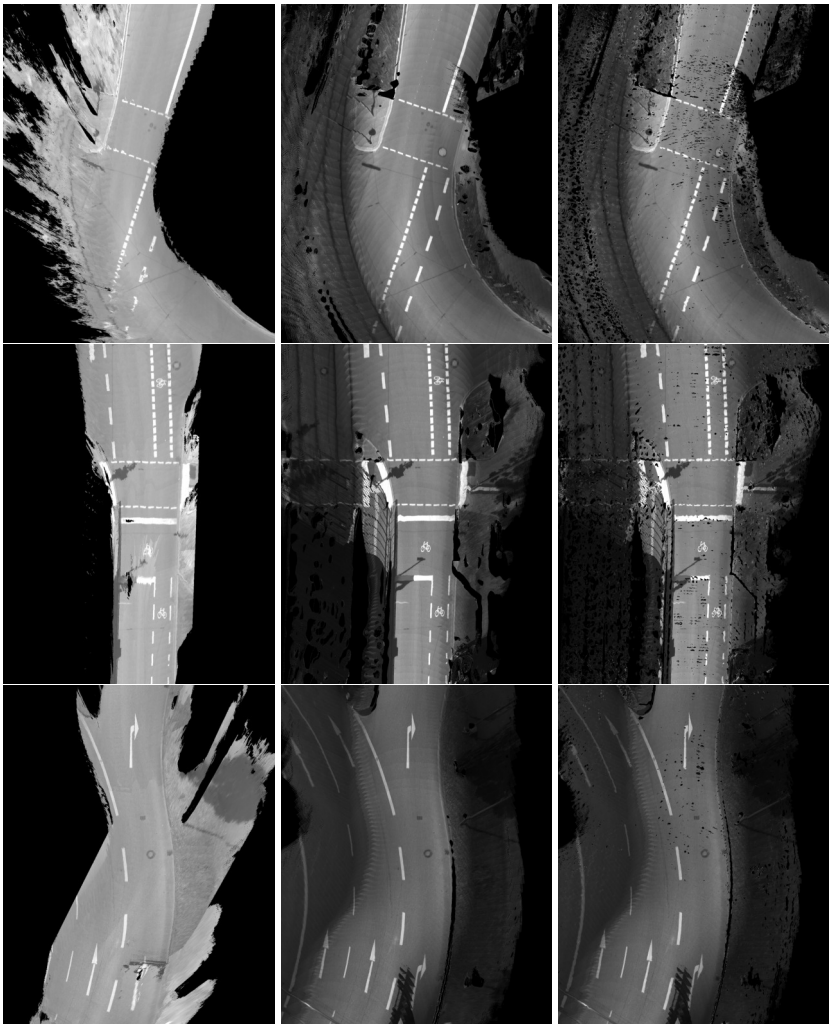


Figure 5.1: Visual comparison of road surface texture across different reconstruction pipelines. Each row represents an individual scene, with the left column displaying results from the stereo-based approach, and the middle and right columns showcasing fusion-based approaches with support point spacing of 0.5 m and 0.2 m, respectively.

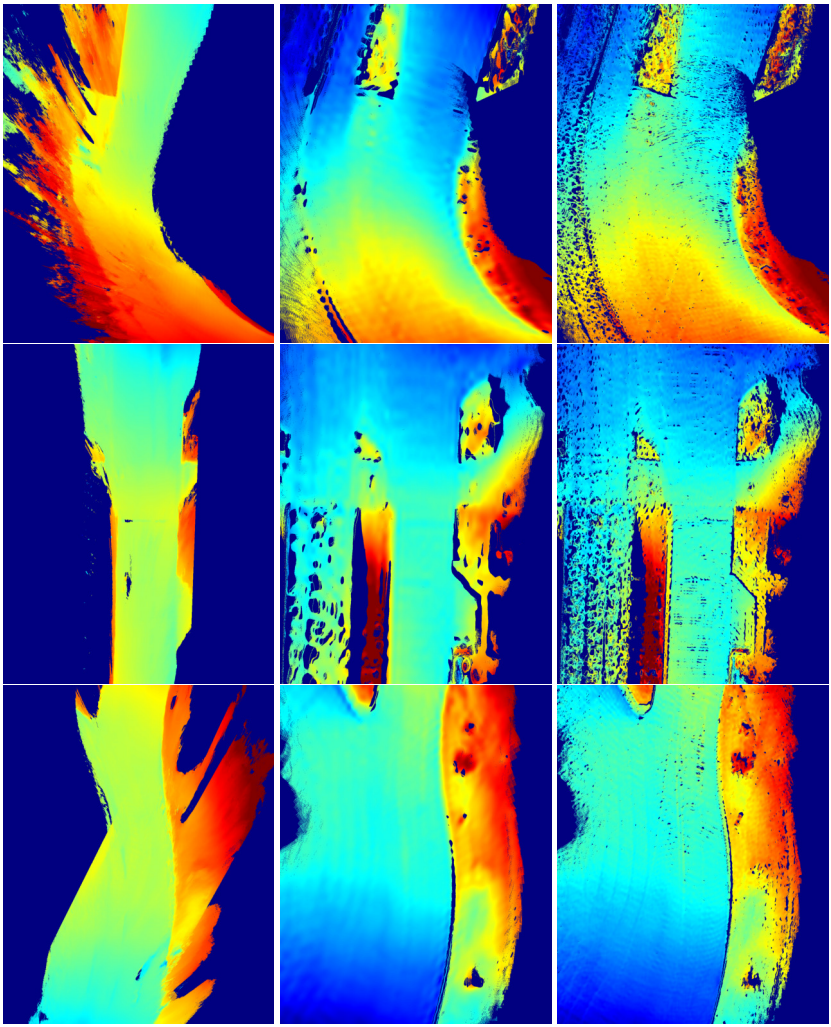


Figure 5.2: Visual comparison of road surface height across different reconstruction pipelines: Each row represents an individual scene, with the left column displaying results from the stereo-based approach, and the middle and right columns showcasing fusion-based approaches with support point spacings of 0.5 m and 0.2 m, respectively. Red areas indicate higher elevations, while blue areas indicate lower elevations or no measurement data.

From ground surface to scene reconstruction In a successive work, Hu et al. [HYW23]† propose a large-scale 3D reconstruction, texturing and semantic mapping pipeline based on LiDAR and camera data. They model surfaces implicitly using an Adaptive Truncated Signed Distance Function to handle varying point densities. The subsequently extracted mesh is then textured and semantically annotated. In contrast to the previously described method, this approach is not limited to the ground surface but aims to reconstruct the entire scene. Results are depicted in Figure 5.3. Notably, these results are based on the dataset presented in Chapter 4 and a semantic segmentation model developed in Chapter 7. The following chapters are based on the surface reconstruction outlined in the prior paragraph, yet, it could be exchanged in future works by this more advanced approach.

(a) 3D reconstruction. **(b)** Texturing. **(c)** Semantic mapping.

Figure 5.3: Application of the dataset presented in Chapter 4 in the context of 3D reconstruction, texturing and semantic mapping by Hu et al. [HYW23]†. It is also the first application of a model developed within this thesis and presented in Chapter 7, in particular for detection of road features.

5.4 Manual Mapping of Geometrically Lifted Dense Road Surface Features

The dense road surface map is annotated manually based on the reconstructed road surface texture and aerial imagery. In a post-processing step, the 2D map annotations are lifted to 3D map elements using the height information of the road surface reconstruction.

Regarding the surface annotation, the main goal is a clear distinction between drivable, i.e., roadway, and non-drivable areas. The latter is further divided

into sidewalks, intended for pedestrians and cyclists, and terrain. On top of the dense surface annotation, three different types of road markings are annotated: dashed lines, solid lines, and other markings. While dashed lines and solid lines primarily represent lane dividers, other markings comprise markings with various meanings, such as arrows or stop lines, each of which is crucial for understanding the road network. This simplification is done to mitigate limitations due to annotation complexity and dataset scaling in the course of this work. A detailed definition of the semantic surface classes and their annotation hierarchy is given in Table A.3.

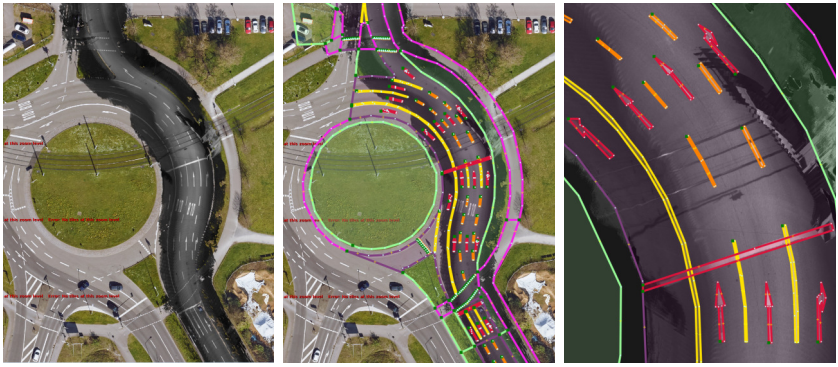


Figure 5.4: Annotation process of dense road surface features. Left: overlay of reconstructed road surface and aerial imagery, middle shows the annotated road surface features and right shows the same features in a zoomed-in view. The color coding is explained in Table A.3. Imagery: ©Stadt Karlsruhe | Liegenschaftsamt.

All three surface layers, *road*, *sidewalk*, and *terrain*, share the same borders. This prevents the need for a strict label hierarchy since areas do not overlap. While preventing holes in the map, it requires a complete assignment of surfaces in the region of interest to either one of the three classes resulting in one class to also cover not previously defined areas. The unlabeled area class is automatically assigned to the outskirts of the region of interest. By design, this annotation scheme not only allows to extract area labels but also explicitly defines the border between different surfaces such as road and terrain or road and sidewalk. This is further explored in the student thesis [Sch21], which not

only extracts the boundaries of surface areas but also derives the middle line of road features in order to train a generic line segment detector network called *Yolino* [MSP21].

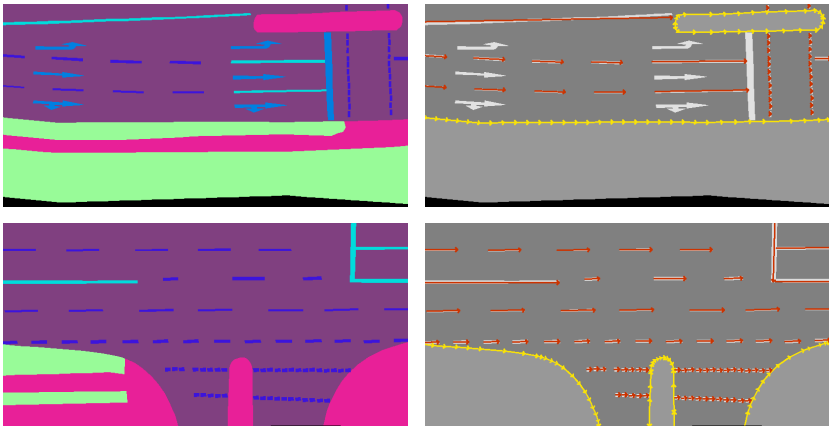


Figure 5.5: GT representations for learning a BEV perception [BHS23]* from the dense road surface annotations. Two example scenes are shown in the top-view perspective. Left: semantic segmentation, right: line segment detection with Yolino [MSP21, Sch21].

Derived from the dense surface annotation, two exemplary learning representations in BEV space are shown in Figure 5.5: A semantic segmentation using all six classes and the line segment detection representation of Yolino for road boundaries and estimated centerlines of road features.

5.5 Semi-Automatic Mapping of Lane-Level Planning Maps

A goal of this work is to create a consistent and well-defined map base for training state-of-the-art vectorized map construction models [LWW22, LYW23, LCW22], which include classes like road borders, lane divider types, and centerlines. For this purpose, the semi-automatic mapping framework of [Pog19]

is employed, yet slightly adapted and simplified to focus solely on the lane network and topology, instead of including traffic rules and associations of traffic signs or lights to lanelets.

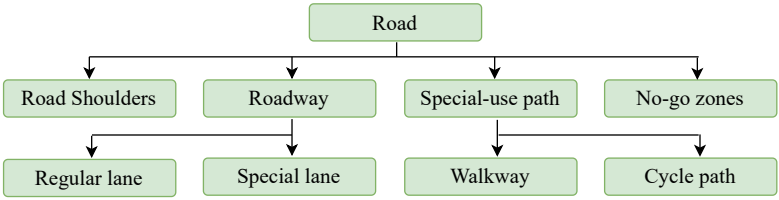


Figure 5.6: Road outline definition. Schematic overview of the road layout, similar yet simplified as defined in the German road traffic regulations [Bun13].

In [Pog19], Poggenhans presented a semi-automatic mapping framework for generating detailed, lane-level planning maps in the Lanelet2 format. By incorporating the German StVO rule set [Bun13], geometric reasoning, and domain-specific assumptions, the framework aims to infer relational and topological map elements from minimal physical cues. These physical cues can be provided by manual annotation or by semantic detections from on-board sensors.

Before inferring a road network, its structure must be well-defined. In general, a road consists of various sub areas such as road ways, road shoulders, no-go-zones and special-use paths, e.g., cycle paths and sidewalks. Road ways, representing the part of the road intended for vehicles, can be further divided into regular lanes and special lanes which have certain restrictions like bus, taxi or bike lanes. A schematic overview of the road layout is shown in Figure 5.6.

As stated earlier, this work focuses on the lane topology and network, i.e., parts of the road network intended for vehicles. Usually, this network can be modeled largely independently of most other subcategories of the road or even of most special lanes. One exception in urban German cities are bike lanes, which are quite common and interfere heavily with the road network as they change the lane topology. In the following, both the concept of the adapted semi-automatic mapping framework and challenges imposed by implicit classes, such as bike lanes, are further discussed.

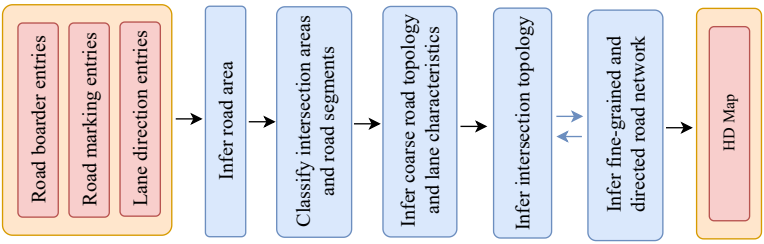
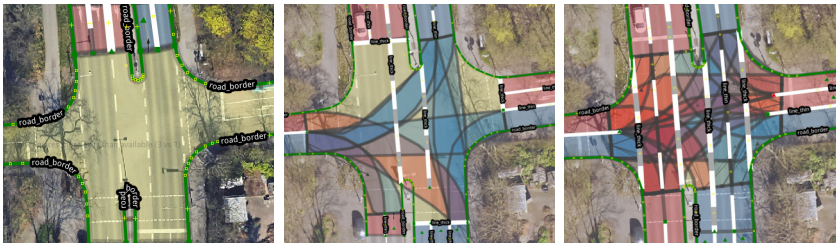


Figure 5.7: Overview of the semi-automatic mapping pipeline for creating Lanelet2 maps. This pipeline is a simplified version of the one presented in [Pog19] as it does not aim to create a complete and functional planning map for autonomous driving but rather a consistent and well-defined base for training a map inference model. So traffic rules and associations of traffic signs or lights to lanelets are not considered. Left in an orange box are the physical cues which are manually annotated in a BEV perspective and provided as input of the pipeline.

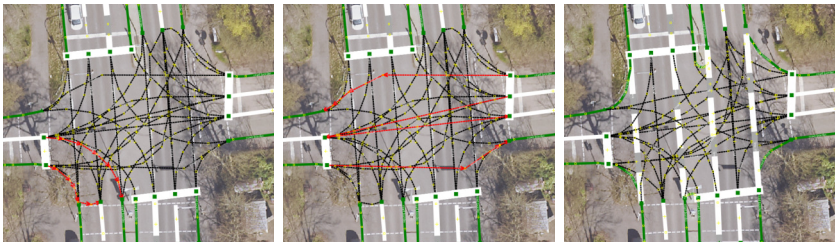
Semi-automatic mapping pipeline A conceptual overview of the adapted and simplified semi-automatic mapping pipeline is provided in Figure 5.7. For this work, the physical cues are solely provided by manual annotations and consist of road border, lane direction, and road marking prompts in the BEV perspective. In a first step, the road area is identified and classified into road segments and intersections. This is done by constructing a Voronoi diagram, see A.4, from annotated road markings and borders, which allows to partition the road network into atomic/manageable parts. In combination with the lane direction prompts, the Voronoi cells are the basis to infer an initial lane topology, in particular for the road segments. Subsequently, the topology of intersections is estimated based on the number and direction of incoming and outgoing lanes, and a combination of formal traffic rules and heuristic assumptions. Since the entire map is not usually resolved in a single step, the inference of road segments and intersections is inherently iterative, progressively refining both intersections and road segments. The process terminates when either all segments are successfully resolved or the number of unresolved segments stays constant, typically due to insufficient or inconsistent input prompts.

In Figure 5.8, intermediate and final results of the mapping pipeline are shown. First, Figure 5.8a illustrates optimization results, ranging from non-resolvable intersection areas, over partially resolved intersection areas with the addition

of visible road markings and entry hints, to a resolved intersection area after the completion of entry information by adding also non-visual road markings, which can be derived from context. Second, Figure 5.8b depicts manual post-processing of common optimization artifacts such as ill-formed lanes for right turns or straight lanes in the absence of road marking cues. Finally, the resulting solution after post-processing is shown in the image on the right.



(a) Intermediate optimization outputs. Left: annotation of road border yields a non-resolvable intersection area. Middle: After adding visible road markings and multiple entry hints some lanes become resolvable. Right: Completion of entry information by adding also non-visual road markings, which can be derived from context, yielding a resolved intersection area. Non-resolved areas are depicted in yellow. Resolved lanes are shown in red and blue with respect to their heading direction.



(b) Manual postprocessing of common optimization defects and final map result. Left: Inference of misshaped lane for right turn. Middle: Inference of mishaped straight lanes in the absence of road marking hints. In both cases a manual adjustment of the affected linestrings is done. Right: final results. The misshaped lanes are selected and depicted in red.

Figure 5.8: Visualization of intermediate and final results of the semi-automatic mapping pipeline for creating lanelet2 maps by [Pog19]. Imagery: ©Stadt Karlsruhe | Liegenschaftsamt.

Consistent representation of implicit classes The presented map definition is designed to allow the generation of consistent training data for vectorized

instances of road borders, lane dividers and also the lane topology, i.e., centerlines. Some classes, such as road borders in non-intersection areas or visible lane dividers are explicitly annotated, while others, such as centerlines, are derived from context. On an annotation level, this requires a label strategy so that these classes can be inferred in a meaningful and consistent way. This becomes especially challenging in the presence of components of the road layout which are not part of the derived label set, but still significantly influence the inference of other map classes such as centerlines. For instance, bike lanes are not part of the standard map perception task, yet they are common in urban areas and influence the lane topology. Figure 5.9 illustrates this issue by comparing three different annotation strategies in an intersection which includes bike lanes, highlighting the importance of context-aware labeling for classes that are not present in downstream perception tasks or even in the map definition. Either by annotation definition or in a later inference step, the remaining target classes must remain consistent and meaningful.

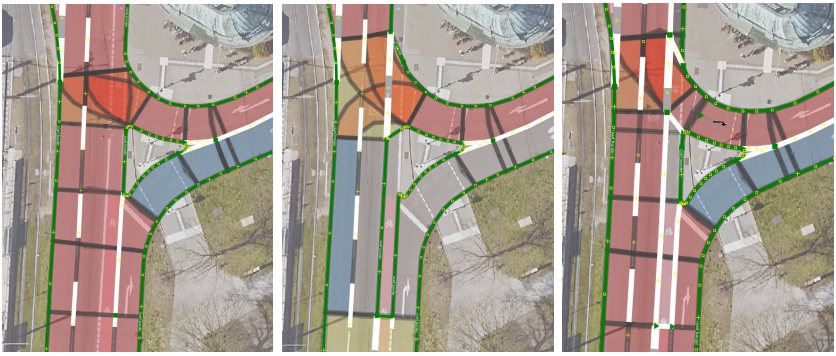
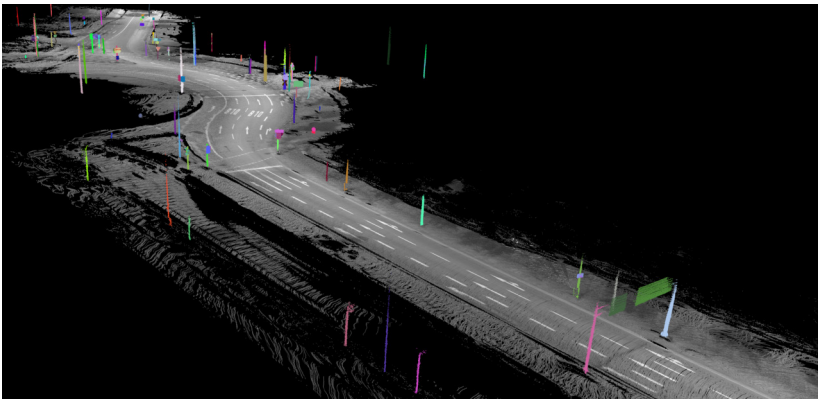
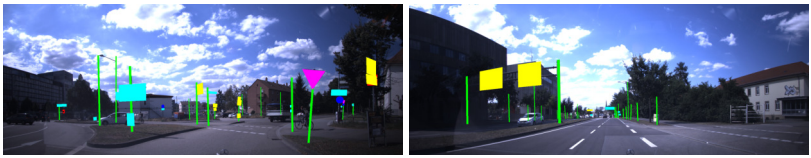


Figure 5.9: Example approaches on how to handle bike lanes. Bike lanes have no explicit class in the standard map perception task. Hence, an implicit label strategy is to be designed to be consistent for all possible scenarios in regard of all explicit classes. Left: Include bike lanes to the driveable area which neglects associated lane dividers and traffic rules. Middle: Consider only regular lanes as driveable area, resulting in road border on dashed line and, as depicted in the example, non-resolvable intersections. Right: Closing off and divide bike lanes from regular lanes with solid lines. This method is chosen as it represents the most consistent approach regarding lane topology. Imagery: ©Stadt Karlsruhe | Liegenschaftsamt.

5.6 Automatic Mapping of Semantically Tailored Landmarks



(a) Associations and tracking in the global map frame with a color per instance.



(b) Exemplary rendering of geometric primitives.

Figure 5.10: Example results of the automatic mapping of semantically tailored landmarks, visualized in a global map frame and as a reprojection in camera images.

Inspired by [PSS21], yet further developed and refined by [Hu26], the parametric mapping pipeline applied in this work, is designed to automatically generate a map of semantically tailored landmarks from on-board sensor data and detections. More specifically, the system models static objects from three exemplary semantic classes with geometric primitives, i.e., poles and traffic lights as cylinders, and traffic signs as upright planes of various shapes. The parametric mapping pipeline consists of three main processing parts: Semantic primitive detection, primitive association and tracking, and parametric modeling and estimation.

Its input consists of camera images with corresponding instance segmentations, LiDAR point clouds and an accurate 6D ego motion. The instance segmentations are computed with [PBC19], due to its high generalization capabilities and strength in detecting small infrastructure objects, formerly discussed in Section 4.4.2. Motion compensated LiDAR points, see Section 4.4.3, are projected into the masked camera image to determine a map element-specific point cloud for each instance mask. Sequentially, the map element-specific point clouds are associated and tracked in the global map frame considering semantic and geometric constraints. In contrast to [PSS21], the pipeline of [Hu26] exploits all instance masks and element point clouds representing a single map element to optimize its model parameters. Using a non-linear optimizer [AMT23], the model parameters are estimated by minimizing deviations between the map element-specific point clouds and viewing rays on the instance mask's contour pixels and the object shape's hull. While rectangles, circles, and triangles are used as specific primitives to model traffic signs if their shape matches corresponding detections, detections with unspecified or unknown shapes are also modeled as rectangles. This results in an accurate and complete, yet fully automatically generated map covering all regions simultaneously observed by camera and LiDAR sensor.

Example results of the parametric mapping are shown in Figure 5.10. Figure 5.10a depict the detected and tracked primitives in the global map frame. Given the ego pose and camera calibration, the 3D shapes can be projected back into camera images, yielding pixel-accurate instance masks as depicted in Figure 5.10b.

A review of the resulting maps demonstrates that the parametric mapping pipeline is capable of generating complete, consistent and accurate parametric landmarks considering position, geometry and semantic class. An exception are exceptionally high mounted traffic signs or lights which are not reliably observed by the LiDAR sensor and, thus, are occasionally not included in the map. Also, a qualitative map evaluation and initial experiments, conducted using the XD-map pipeline of Section 7.1, have revealed that the parametric mapping pipeline is prone to pose inaccuracies, e.g., present for the visual SLAM in turning maneuvers. For example, slight inaccuracies in estimating the heading

angle can provoke failed associations and tracking, resulting in duplicated, missing or malformed map elements. As a consequence, a more precise multi-modal continuous-time trajectory SLAM, described in Section 4.4.5, replaces the visual SLAM framework for experiments regarding cross-modal domain adaptation via automatic parametric mapping, see Section 7.1.

5.7 Map Analysis and Qualitative Review

This section reviews the quality of the generated HD maps and the interplay between different components of the overall pipeline. For simplicity, the three map layers are abbreviated as follows: M_{DS} for the dense surface feature map, M_{LT} for the lane-level topological map and M_{EP} for the semantically tailored traffic elements. Important to note is that both M_{DS} and M_{LT} are mapped relative to the initial multi-drive pose graph, described in Section 4.4.4, and thus are subject to its accuracy. In contrast, the fully automatic pipeline for mapping M_{EP} is even prone to tiny localization errors, which require a new pose graph optimization, described in Section 4.4.5, to achieve satisfying results.

Table 5.1 summarizes statistics of map M_{DS} and M_{EP} for all recorded sequences in Karlsruhe, Germany. In total, more than 21 000 road markings are mapped in M_{DS}, including dashed lines, solid lines, and other road markings, such as pedestrian crossings and stop lines. The automatic mapping process yielded 2353 traffic signs, 1124 traffic lights and 3454 poles along the close to 22 km of unique road.

Table 5.1: Statistics of annotated map elements in M_{DS} and M_{EP}.

Sequence		M_{DS}**: Road surface features**				M_{EP}**: Parametric elements**		
#	l_{SQ} [km]	l_R [km]	n_{DL}	n_{SL}	n_{OL}	n_{TS}	n_{TL}	n_{PO}
SQ_A	5.35	20.4	6090	256	380	897	367	640
SQ_B	2.97	8.7	3729	175	177	479	170	369
SQ_C	2.98	7.1	2273	63	150	449	163	323
SQ_D	5.33	14.5	2810	158	370	800	159	547
SQ_E	4.95	12.7	4028	78	289	829	265	474
Sum	21.6	63.4	18930	730	1366	2353	1124	3454

l_{SQ}: sequence length; l_R: mapped road border length; n_{DL}: dashed lines; n_{SL}: solid lines; n_{OL}: other line markings; n_{TS}: traffic signs; n_{TL}: traffic lights; n_{PO}: poles.

Figure 5.11: Qualitative evaluation of the dense surface feature map M_{DS} in three scenarios, comparing different reconstruction approaches. In each pair, the top image shows the stereo-based result, and the bottom image shows the fusion-based result. Areas in red are identified as occluded by dynamic objects, further described in Section 7.2.1. While both methods produce comparable reconstructions in the central region, the fusion-based approach provides greater accuracy near the boundaries of the field of view.

Quality of road surface reconstruction The visual assessment of road surface reconstruction and texturing, as well as the resulting quality of M_{DS}, is conducted by rendering HD map elements into the image plane using the forward camera model and the estimated 6D vehicle pose. Figure 5.11 shows three exemplary scenarios with two different reconstruction methods. The top image of each pair corresponds to the stereo-based reconstruction, while the bottom image corresponds to the fusion-based approach, both described in Section 5.3. For better visualization, the semantic occlusion handling for dynamic objects, further described in Section 7.2.1, is applied. Consequently, map elements are solely rendered if not occluded by dynamic objects, which are depicted in red. The image overlays demonstrate the overall high accuracy of the reconstruction and mapping process, including calibration and SLAM, as the projected road features are well aligned with the camera images. A comparison of the two methods indicates that both achieve similar quality in the central field of view. However, the fusion-based approach produces fewer artifacts near the image boundaries. For example, in the second and third scenarios, the stereo-based method lifts features at the far left above the road surface, clearly visible for the solid line in the second case and the leftmost line in the third case.

Several factors may explain this observation. LiDAR depth measurements are typically more accurate, and explicitly modeling the ground surface with a B-spline can improve the robustness towards outliers. Second, even minor distortions caused by the windshield, behind which the stereo camera is mounted, can degrade disparity estimation, with stronger effects toward the image periphery. Finally, the stereo-based method inherently provides fewer depth measurements near the borders of the field of view. This can increase the distance between map elements and their nearest reconstructed surface points, amplifying errors during the lifting process. Nevertheless, the advantage of the fusion-based approach cannot be generalized, as it strongly depends on the specific sensor configuration and calibration quality.

Quality review of multi-drive mapping In the previous example, the map elements of M_{DS} were back-projected into camera images from the primary drive of a sequence. This primary drive is both used for the loop closure in

the SLAM process and as the basis for surface reconstruction and map annotation. An accurate multi-drive mapping enables the generation of training data at scale by acquiring drives with varying environmental conditions as long as the mapping remains consistent. In Figure 5.12, the quality of the multi-drive mapping is assessed by back-projecting map features of M_{DS} and M_{EP} into the camera images of different drives. In all drives, the backprojection quality is high enough that tiny map features, such as 12 cm wide dashed lane markings or small road signs, are well aligned with the sensor data. Depending on the location of the feature the back-projecting error is on a centimeter level, especially for close road markings. It demonstrates that the back-projection quality is consistent across multiple drives of the same sequence, confirming the robustness of the multi-drive mapping.

Figure 5.13 presents additional qualitative results for three traffic elements, mapped in M_{EP} and reprojected into consecutive image sequences of three different drives. For each re-projection, an image patch centered on the respective element is cropped and all patches belonging to the same passing are arranged from left to right in a row. The figure highlights the accuracy of the calibration, semantic mapping, and the multi-drive pose graph alignment. It should be noted that all drives in this dataset were collected under similar weather and lighting conditions. The scaling with additional drives become particularly beneficial when the drives differ significantly in environmental conditions, such as heavy rain, snow or night driving. These challenging conditions are however not part of the current dataset and can thus not be evaluated here.

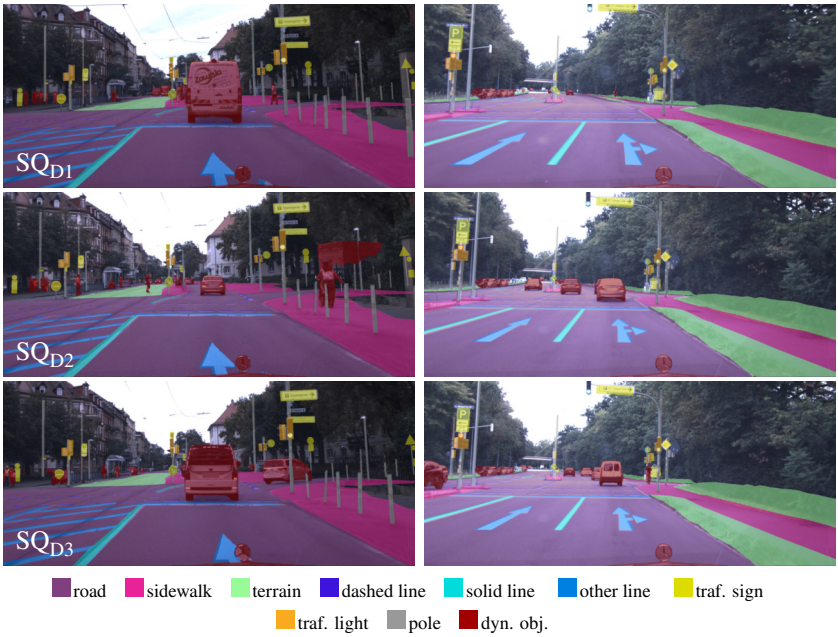


Figure 5.12: Qualitative evaluation of multi-drive mapping precision by assessing rendered map features in three drives of the same sequence. Left and right show two different scenarios and top to bottom depicts the same scenario in three different drives.

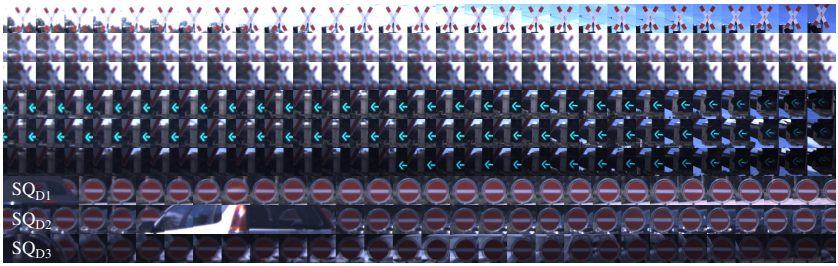


Figure 5.13: Qualitative results of multi-drive mapping. Three traffic elements are mapped in M_{EP} and back-projected into camera images from three different drives. Each row corresponds to one drive, or passing, showing a sequence of cropped image patches for the respective traffic element. The consistent alignment across rows demonstrates the precision of calibration, semantic mapping, and multi-drive registration.

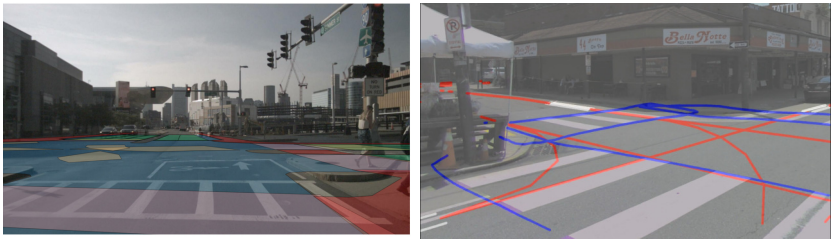


Figure 5.14: Qualitative examples of reprojected map features from public datasets. Left: nuScenes [CBL20], right: Argoverse 2 [WQA21]. More examples in Figure 2.2.

Comparison to other state-of-the-art datasets To put the quality of the generated HD maps into perspective, Figure 5.14 presents qualitative examples of reprojected map features from nuScenes and Argoverse 2, the two state-of-the-art datasets that combine on-board sensor data with HD map annotations. It showcases that the presented multi-layer HD map offers a higher degree of geometric fidelity, completeness and diversity of map elements across different categories. An example of the improved high-fidelity is the centimeter-level alignment of the presented map features with the corresponding physical structures in the environment. Differences in map element diversity include that public datasets are limited to planar, ground-level map elements, while rather small, volumetric or even elevated features are completely missing. Both high-fidelity and diversity of map elements are crucial factors for the range and quality of perception models that can be trained with maps as supervision signal.

Due to the lack of a comprehensive map format and framework for the public datasets, it is challenging to validate planning-grade features for inconsistencies or check for completeness. This also increases the difficulty to infer missing information consistently from the available map data, e.g., as conducted by [LCJ24] for centerlines in nuScenes. An example for incompleteness is that nuScenes does not provide consistent divider annotations, i.e., dividers are fully missing in intersections.

5.8 Conclusion

Three mapping pipelines were applied to build a complementary set of map layers combining manual, semi-automatic, and fully automatic mapping. The process is tailored to create map features of different abstraction levels, for different functional roles and of different geometric representations. Together the multi-layer map elements span from dense physical road-surface features to topological lane networks and elevated traffic elements. A comparison to other state-of-the-art datasets highlights that the accuracy and richness of map elements is on par or exceeds those of other datasets. By back-projecting the generated map elements into the sensor space a centimeter level of precision is achieved, allowing to match even small map features like dashed line markings or small traffic signs.

6 Bird's Eye View Map Perception

Bird's Eye View (BEV) perception has attracted significant interest within the robotics and automated driving community. It denotes the inference and storage of scene information from on-board sensor data in a top-down view. This representation is particularly advantageous as it allows the efficient storage of scene information in absolute scale and with precise localization relative to an agent's coordinate system. This spatially consistent representation and alignment with the geometry of traffic scenes make it well-suited for various downstream tasks, including sensor data fusion, object tracking, and motion planning.

This chapter focuses on the task of learning an online HD map construction from surround-view camera data in a BEV representation by exploiting HD maps as a supervision signal. The emerging task of online HD map construction is especially promising for future autonomous driving systems, as its vectorized output of the static environment aligns well with the needs of many downstream components that traditionally depend on HD maps.

The goals of this chapter are twofold: First, to define a label set and data generation pipeline that produce high-quality, geometrically rich annotations. These should consist of semantic classes valuable for downstream tasks and are typically only available through HD maps. Second, to identify a training regime that effectively exploits and employs the characteristics of the presented sensor setup and dataset in the context of real-world applicability. This includes studying trade-offs between inference time and model performance, as well as analyzing the impact of individual sensors and localization noise on overall system performance.

The first goal is addressed in Section 6.2 and Section 6.3, which outline a strategy for generating high-quality, high-value labels from HD maps. Extensive

ablation studies are then presented in Section 6.4 and Section 6.5, evaluating the effectiveness and deployability of different training configurations.

6.1 Online HD Map Construction

The task of *Vision-Centric* BEV Perception is a concept of transforming a batch of V perceptive-view images with height H, width W and C channels $\mathrm{I} \in \mathbb{R}^{V \times H \times W \times C}$ into BEV features $F_{\mathrm{BEV}} \in \mathbb{R}^{X \times Y \times D}$, where X and Y denote spatial dimensions and D the number of BEV feature channels. Subsequently, various inference tasks, such as object detection, semantic segmentation or instance segmentation, can be performed on the BEV features. A comprehensive overview of existing methods is provided by [MWB24].

One of the most prominent tasks in this context is online HD map construction, as discussed in Section 2.4, i.e., to predict a set of vectorized map instances represented by polygons and polylines in BEV representation. Typical methods consist of at least three components: First, a backbone network extracts features from the input sensor data. For image data, common choices include a CNN-based ResNet [HZR16] or a transformer-based Swin [LLC21] architecture. Second, a PV-2-BEV transformation module converts feature embeddings from the perspective view into the BEV representation. Ma et al. [MWB24] survey a wide range of different PV-2-BEV modules including geometry-based methods, which utilize depth estimations [RKC18, PF20] or homographies [HZG20], or learning-based methods which employ multi-layer perceptrons [LWW22] or attention-based transformers [LWL22, ZK22]. Third, a task-specific head, often referred to as a map decoder module in online HD map construction, predicts the target map elements from the BEV features. In many recent works, this module is a transformer-based architecture [LYW23, LCW22, LCZ24], similar to DETR [CMS20].

This work employs the MapTRv2 architecture [LCZ24], which combines several recent advancements in the field and has established itself as a well-studied baseline for online HD map construction. Its map decoder employs a hierarchical bipartite matching strategy with a fixed number of points per predicted

map instance along with multiple auxiliary supervision signals and a one-to-many query design to improve convergence and prediction quality. A ResNet-50 backbone, pre-trained on ImageNet [DDS09], was selected over the more powerful Swin backbone due to its favorable trade-off between accuracy and computational efficiency. The PV-2-BEV transformation module is based on the Lift-Splat-Shoot (LSS) method [PF20], which predicts a depth distribution for each pixel ray and uses it, together with a context vector, to determine the corresponding BEV features along the line of sight.

6.2 Label Definition and Generation

In the original task definition of Li et al. [LWW22] and many following works, online HD map construction is evaluated on three types of map elements: Pedestrian crossings, lane dividers, and road boundaries. All three map instances were represented by a fixed number of 2D points, either as polygons for pedestrian crossings or as polylines for lane dividers and road boundaries. These labels all represent physical map elements from which the derivation of higher-level topological information, essential for downstream behavior and motion planning tasks, is challenging. To address this limitation, MapTRv2 has proposed to incorporate lane centerlines as a fourth map element class. Unlike other map instances, which are undirected, centerlines have a defined direction, requiring a distinct matching strategy between predicted and ground truth instances during training and evaluation. This was a significant step towards alignment of map perception output with the actual requirements of downstream tasks. In contrast to prior works which solely predicted 2D map instances, MapTRv2 also extended the geometric representation of predicted map instances into 3D space. The next essential label improvement was presented by Immel et al. [IFB25b, IFB25a][†] proposing the distinction between solid and dashed lane dividers as two separate classes. In real-world autonomous driving, it is crucial to be able to identify the type of a divider as it indicates whether a lane change maneuver is permitted. Additionally, the authors corrected several annotation inconsistencies and advocated to utilize a comprehensive map framework, e.g., Lanelet2 [PPJ18], as a basis

to generate more valid and consistent HD map labels instead of custom map representations. To this end, a novel module denoted LL2MLconv, which specifically targets the generation of labels for online HD map construction was integrated in the Lanelet2 framework and was released as open-source software along with the publication. The label generation pipeline for online HD map construction is illustrated in Figure 6.1.

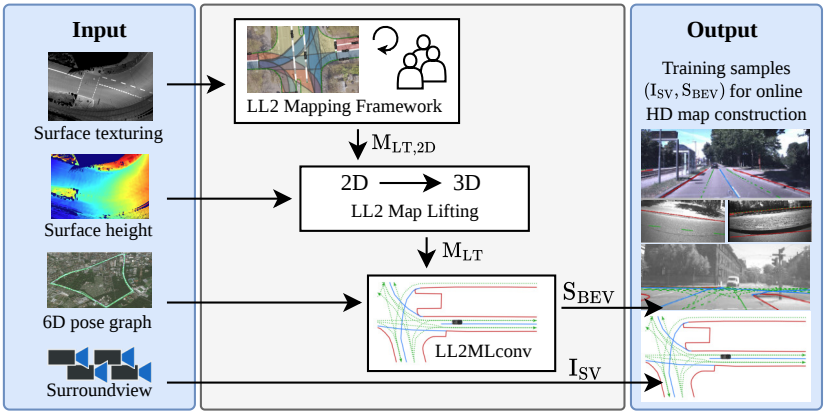


Figure 6.1: Overview of the label generation pipeline for online HD map construction. First, preprocessing estimates a high-accuracy 6D pose graph and a surface reconstruction, detailed in Sections 4.4.4 and 5.3, respectively. Next, the reconstructed surface texture is manually annotated with structured cues and converted by the Lanelet2 mapping framework into a topological 2D planning map layer as described in Section 5.5. This map is then lifted into 3D using the reconstructed ground surface, yielding M_{LT}. Finally, LL2MLConv extracts local map elements for each 6D pose, converts them into a set of fixed-size polylines and polygons, and assigns semantic class labels. Together with synchronized surround-view camera images I_{SV}, the labels S_{BEV} are used to train and evaluate the online HD map construction model.

This work builds upon these recent advancements in label definition and generation for online HD map construction. The labels are extracted from the topological HD planning map M_{LT}, further described in Section 5.5, using the LL2MLconv module. All map elements are lifted into the 3D space by incorporating the ground surface reconstruction detailed in Section 5.3, resulting in

a high-fidelity 3D representation of the road geometry, as demonstrated in Section 5.7. Accordingly, the generated map instances are represented in 3D and include directed lane centerlines, along with a distinction between solid and dashed lane dividers. Pedestrian crossings were not considered in this work as they are relatively rare in the limited range of the custom dataset to provide sufficient samples for training and evaluation. By incorporating these improvements and executing them with high-precision mapping and label generation pipelines, i.e., the Lanelet2 mapping framework [Pog19], the presented ground surface reconstruction and the LL2MLconv module [IFB25b]†, this work achieves state-of-the-art label quality for online HD map construction. Figure 6.2 shows example ground truth labels extracted from M_{LT}.

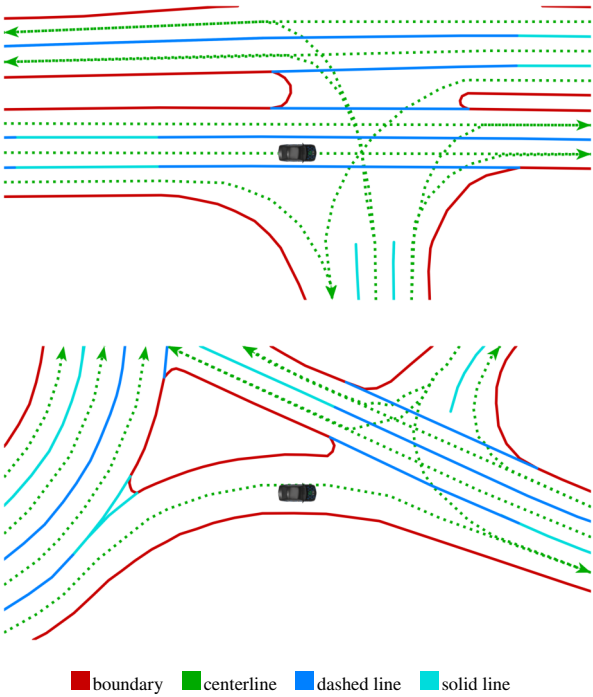


Figure 6.2: Example ground truth for online HD map construction, generated from the topological HD planning map M_{LT} .

6.3 Data Set Definition and Geographic Split

A recent study by Lilja et al. [LFS24] demonstrated that the performance of map perception models on evaluation benchmarks is significantly affected by geographic overlap between the training and test sets. Until then, this aspect had been largely overlooked in established map perception benchmarks and reported evaluation results were based on a mixture of geographically overlapping and non-overlapping splits. This made a fair comparison of generalization performance difficult.

Building on this insight, all experiments in this work use a strict geographic separation between training and test sets, ensuring a minimum distance of 100 m between them. For the geo-split test set, two distinct urban areas are selected to increase diversity, covering both a rather rural region and a densely populated area close to the center of the city. Approximately 2 km of the total 22 km are allocated to the test set, with the remainder used for training. The dataset comprises four drives across five sequences, allowing for specific multi-drive related analysis and research. To this end, a multi-drive dataset is defined, consisting of three drives for training. Additionally, a pure overlap evaluation split, consisting solely of the fourth drives, is introduced to particularly analyze performance gains when the same scenes are already observed during training. In summary, the single-drive training split contains ∼19 000 frames, the multi-drive training split ∼42 000 frames, the geo-split test set ∼1800 frames, and the overlap test set ∼13 000 frames. For faster evaluation during training, every fourth frame is selected in case of the geo-split test set and every tenth frame in case of the overlap test set.

Table 6.1 compares the characteristics of the labels and data splits used in this work with those of related works in online HD map construction. It highlights that most prior works do not include crucial map element categories such as lane divider types and centerlines. Furthermore, many studies do not utilize 3D instance representation. However, most recent works have adapted the geographic split. This work incorporates all previous improvements and also considers the separate consideration of single-drive and multi-drive settings.

Table 6.1: Comparison of label and data split characteristics of related works. (✓) indicate that 3D instance representation was applied if possible. S/M-dri. set. denotes the separate consideration of single-drive and multi-drive training settings.

Method	Data set	Divider types	Lane centerl.	3D instances	Geo. split	S/M-dri. set.
MapTR [LCW22]	nu	-	-	-	-	-
VectorMapNet [LYW23]	nu/av2	-	-	-	-	-
MapTRv2 [LCZ24]	nu/av2	-	✓	(✓)	-	-
MapTracker [CWT24]	nu/av2	-	-	-	✓	-
StreamMapNet [YLW24]	nu/av2	-	-	-	✓	-
M3TR [IFB25a]	nu/av2	✓	✓	(✓)	✓	-
This work	ours	✓	✓	✓	✓	✓

6.4 Experimental Evaluation

This section outlines the experimental evaluation of online HD map construction from surround-view camera data trained on the custom dataset developed in this work. The unique sensor setup and dataset characteristics allow for a series of ablation studies to analyze the influence of various factors on model performance. These aspects include the effect of input resolution, camera setup, model initialization, and sensor data frequency. Additionally, the robustness of the training process is examined under varying localization noise patterns.

Experiment layout Input resolution is a critical hyperparameter for the real-time application of perception models, particularly those operating on multiple camera views and large network architectures. In previous works, MapTRv2 drastically reduced the input resolution of the multi-camera input images. A scale factor of 0.5 for nuScenes images and 0.3 for Argoverse images is applied, yielding input resolutions of 800 px × 450 px and 614 px × 614 px, respectively. In this work, the high-resolution surround-view camera images are significantly larger, exceeding 6.5 megapixels per image. After cropping a region of interest of 3000 px × 1500 px, different scale factors are applied ranging from 0.15 to

0.8 in order to find a good trade-off between performance and efficiency. Selected models are also evaluated on the overlapping split test set to assess their performance in geographic regions familiar from training.

Next, the benefit of different camera setups is investigated. This includes removing individual cameras from the surround-view setup, such as the rear or low-mounted side cameras, to assess their contribution to the overall performance. Also, the addition of two roof-mounted cameras is evaluated to analyze the impact of their superior mounting position. Furthermore, experiments are conducted using different training sample frequencies, i.e., varying the selection rate of sequential sensor frames and incorporating multi-drive setups. Data sample frequency directly correlates with the dataset size, which strongly motivates keeping the sampling frequency as low as possible without degrading model performance. The multi-drive setup can increase the effective observation frequency of static scenes beyond the sensor rate while also introducing new traffic scenarios and environmental conditions. Furthermore, the influence of different pretraining strategies is studied. Pretraining and representation learning are crucial to overcome data scarcity, especially as model capacity and complexity continue to grow. Although the open-source datasets differ in sensor setup, geographic region, and annotation strategies, they can help to build a strong initial feature representation. With the custom dataset being relatively small for a complex task like online HD map construction, several pretraining strategies are compared, including training from scratch and pretraining on different datasets, label sets, and training regimes. These experiments offer important insights into the effective deployment of real-world online HD map construction in new domains with limited data availability.

Finally, a study is conducted to examine the influence of pose degradation on ground truth generated from HD maps. This follows an in-depth study of Blumberg et al. [BMF25]†. They are the first to model various real-world localization noise patterns and apply them on the Argoverse 2 dataset before ground truth generation. In this work, the same noise types, Gaussian, Ramp, and Perlin noise, are applied to the custom dataset to study which noise levels lead to significant performance degradation given the specific sensor setup and dataset size.

Evaluation range and metric The evaluated perception range is defined over a range of $-30\,\text{m}$ to $30\,\text{m}$ in longitudinal and $-15\,\text{m}$ to $15\,\text{m}$ in lateral direction. While this is de-facto the standard for the perception range in online HD map construction, more recent works, in particular when incorporating prior map or temporal information, also consider more extended ranges of $100\,\text{m} \times 50\,\text{m}$ [YLW24] or even $240\,\text{m} \times 60\,\text{m}$ [JZL24]. Analogous to [LYW23, LCW22], the Chamfer Distance $d_{\text{CD}}(S_1, S_2)$ is used as the evaluation metric to determine the prediction quality. It is a similarity measure between two sets of points S_1 and S_2 and is further discussed in Section 3.3.1. Based on the d_{CD}, an Average Precision (AP) is computed for three different modes: Easy, moderate, and hard, corresponding to three thresholds $\tau \in T$, $T = \{0.5, 1.0, 1.5\}$, respectively. As long as $d_{\text{CD}} < \tau$, the predicted instance is considered as a true positive. The average across all thresholds, denoted as

$$\text{mAP} = \frac{1}{|T|} \sum_{\tau \in T} \text{AP}_\tau, \quad \text{with} \quad \tau \in T,\, T = \{0.5, 1.0, 1.5\}, \tag{6.1}$$

represents the overall performance measure. Additionally, the AP is reported for each map element class separately, in order to provide a fine-grained performance review, denoted AP_{bou}, AP_{cen}, AP_{dsh}, and AP_{sol} for road boundaries, lane centerlines, dashed lane dividers, and solid lane dividers, respectively.

Training hyperparameters and base model The hyperparameters of the MapTRv2 model are mostly kept identical to those proposed in [LCZ24] for training with the Argoverse 2 dataset. All experiments are trained for 8000 iterations with a batch size of 32, using the AdamW optimizer [LH19] with a learning rate of 6×10^{-4} and a weight decay of 1×10^{-2}. When initializing with pretrained weights, the learning rate multiplier of the backbone is set from 1 to 6 to account for the strong domain shift in input images. While the depth prediction loss is turned off, the two other dense auxiliary losses are employed, namely the BEV segmentation loss and the PV segmentation loss.

6.5 Results

The base model for all experiments is trained with an input scale factor of 0.5 on the full surround-view camera setup, resulting in an input resolution of 1500 px × 750 px per image. The input image resolution of the base model was defined with respect to Figure 6.3, which shows a study of different input image resolutions ranging from 450 px × 225 px to 2400 px × 1200 px comparing their performance and inference time. Here, the base model offers a good trade-off between performance and efficiency. Higher resolutions do not increase the performance significantly, while having much higher inference times. The model is initialized with pretrained weights from an Argoverse 2 training with four classes, denoted as $\mathrm{av}_{\mathrm{c4}}$. It is trained with the training sample frequency $\mathrm{f}_{\mathrm{S/1}}$, i.e., using the single-drive training samples with the full 10 Hz frequency. This base model achieves a mAP of 28.9% on the geo.-split test set.

Figure 6.4 and Figure 6.5 present qualitative results of the base model on examples from the geo.-split test set. The figures demonstrate that the model can accurately predict not only simple, straight road geometry but also more complex topologies including intersections.

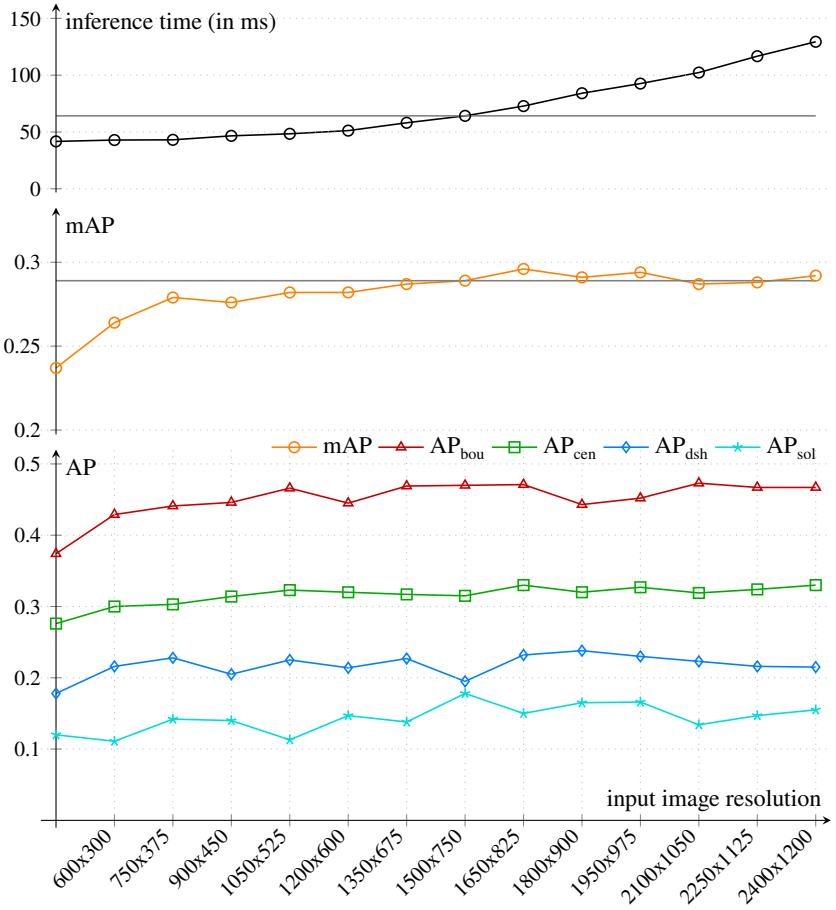


Figure 6.3: Quantitative results of HD map construction performance for different resolutions of the surround-view image input. On the original resolution of 3000 px × 1500 px scale factors ranging from 0.15 to 0.8 are applied, resulting in image resolutions from 450 px × 225 px to 2400 px × 1200 px. The top plot shows the inference time in milliseconds, the middle plot the mean Average Precision (mAP) over all classes and the bottom plot the Average Precision (AP) for all individual map element classes. The horizontal gray lines indicate the performance and inference time of the base model trained and evaluated on an image input of 1500 px × 750 px.

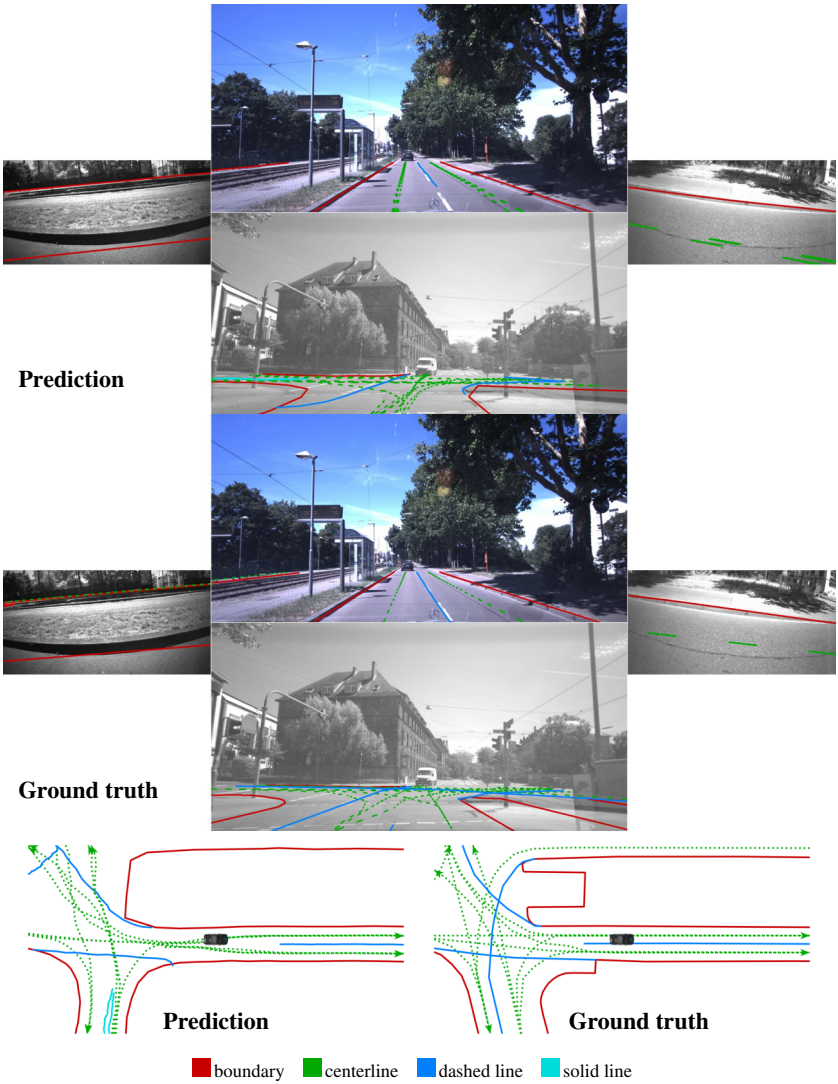


Figure 6.4: Example of HD map construction performance on the geo split. Top: Predicted map instances overlaid on the surround-view camera images. Middle: Ground truth map instances overlaid on the surround-view camera images. Bottom left: Predicted map instances in BEV. Bottom right: Ground truth map instances in BEV.

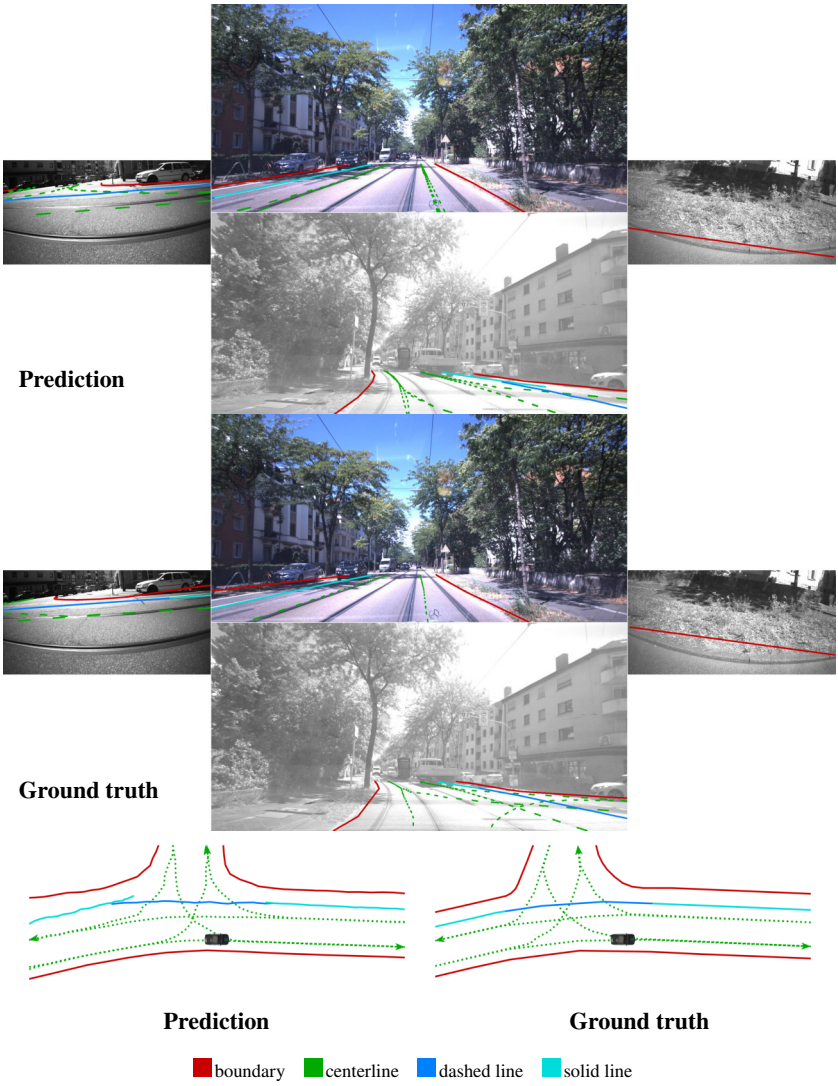


Figure 6.5: Example of HD map construction performance on the geo split. Top: Predicted map instances overlaid on the surround-view camera images. Middle: Ground truth map instances overlaid on the surround-view camera images. Bottom left: Predicted map instances in BEV. Bottom right: Ground truth map instances in BEV.

Table 6.2 summarizes the quantitative results of three ablation studies on the geo-split test set. For the sensor setup four different configurations are evaluated, including the full surround-view setup sv_4, removing the rear cameras sv_3, removing the low-mounted side cameras sv_2 and adding the roof cameras sv_{+r} to the general surround-view setup. It shows that, in the given setup, the removal of the rear camera from the surround-view setup has a significantly stronger negative impact on the performance than removing the low-mounted side cameras. The results indicate that a two-camera setup with wide opening angles is already capable of capturing most of the relevant information. The benefit which comes from adding the roof cameras is limited. This observation may be due to the lower quality of the roof cameras compared to the respective surround-view camera. In a fair comparison to assess the benefit these should have been of similar quality. This can be seen as a limitation of the presented dataset and, thus, the roof cameras are not considered in the following experiments.

The results with different pretraining strategies show that initializing the weights from scratch yields better results than an initialization on nuScenes, both in a pretraining with centerline nu_{c4} and without centerline nu_{c3}. This indicates that pretraining on less qualitative data can also have a negative impact on the final performance. However, pretraining on Argoverse 2 improves the performance significantly in all four assessed pretraining cases: av_{c3} and av_{c4} are standard trainings with a label set of three and four classes, respectively. av_{m3tr} denotes a standard training setting in which the ground truth enhancements of M3TR [IFB25a] are applied on the Argoverse 2 training data. Each label improvement making the pretraining more beneficial for the fine-tuning on the custom dataset. av_{m3tr+} further increases the performance by applying the advanced training regime of the M3TR Generalist model.

Experiments on different training sample frequencies are conducted on the single-drive setup with 10 Hz, 4 Hz, 0.5 Hz and 0.25 Hz denoted as $f_{S/1}$, $f_{S/5}$, $f_{S/20}$ and $f_{S/40}$, respectively. Furthermore, two multi-drive setups are evaluated, including the standard multi-drive setup $f_{M/1}$ and a setup in which the training samples are geographically equally sampled f_{Mges}. The separate mode f_{Mges} is necessary since not all drives cover the full sequence and, thus, some

regions are over-represented in the multi-drive setup. A trend of decreasing performance with decreasing training sample frequency is observed. Still, even when taking only every 20th sample, the performance loss amounts to only 1.5%, which is a relatively small drop considering the significant reduction in dataset size. When making sure that the multi-drive setup is geographically equally sampled, this yields a notable performance boost compared to the standard multi-drive setup. Finally, the results of the robustness study regarding localization noise mimic the observations made by Blumberg et al. [BMF25] and are further discussed in Appendix A.6.

Table 6.2: Quantitative results along three experimental dimensions: sensor setup, weights initialization and dataset size w.r.t., frequency of training samples. All models are trained with an image input 1500 px × 750 px on the standard range of 60 m × 30 m and evaluated using mean Average Precision (mAP) on the geographical split evaluation set. The performance difference to the base model *vs.ba.* is depicted in the last column.

	Experimental Setting			Average Precision					
	Setu.	Pret.	Freq.	$\mathbf{AP_{bou}}$	$\mathbf{AP_{cen}}$	$\mathbf{AP_{dsh}}$	$\mathbf{AP_{sol}}$	**mAP**	vs.ba.
Setup	sv_2	av_{c4}	$f_{S/1}$	43.7	30.6	21.4	14.7	27.6	-2.3
	sv_3	av_{c4}	$f_{S/1}$	31.8	26.2	14.5	5.4	19.5	-9.4
	sv_4	av_{c4}	$f_{S/1}$	47.0	31.5	19.5	17.8	28.9	base
	sv_{+r}	av_{c4}	$f_{S/1}$	46.2	33.1	20.5	16.4	29.1	+0.2
Pretraining	sv_4	w_0	$f_{S/1}$	38.6	24.3	16.8	9.4	22.3	-6.6
	sv_4	nu_{c3}	$f_{S/1}$	19.7	19.7	7.9	5.4	13.2	-15.7
	sv_4	nu_{c4}	$f_{S/1}$	24.7	18.8	10.3	7.4	15.3	-13.6
	sv_4	av_{c3}	$f_{S/1}$	45.6	30.1	20.5	12.6	27.2	-1.7
	sv_4	av_{c4}	$f_{S/1}$	47.0	31.5	19.5	17.8	28.9	base
	sv_4	av_{m3tr}	$f_{S/1}$	43.8	33.8	23.2	18.0	29.7	+0.8
	sv_4	av_{m3tr+}	$f_{S/1}$	46.8	33.4	25.6	16.7	30.2	+1.3
Frequency	sv_4	av_{c4}	$f_{S/40}$	43.4	28.9	16.8	13.5	25.7	-3.2
	sv_4	av_{c4}	$f_{S/20}$	45.1	30.9	21.2	12.6	27.4	-1.5
	sv_4	av_{c4}	$f_{S/5}$	46.4	31.6	23.1	13.1	28.6	-0.3
	sv_4	av_{c4}	$f_{S/1}$	47.0	31.5	19.5	17.8	28.9	base
	sv_4	av_{c4}	f_{M}	47.3	31.7	22.1	15.7	29.2	+0.3
	sv_4	av_{c4}	f_{Mges}	45.5	33.5	25.7	16.0	30.2	+1.3

Results on the overlap-split test set While the main evaluation is conducted on the geo-split test set, additional evaluation results on the overlap-split test set are provided in the appendix. Table A.5 shows the performance on the overlap-split across different input resolutions and training sample frequencies. It demonstrates the performance boost obtained if the model is trained and evaluated on data from the same geographic region. The performance of the base model improves from 28.9% to 60.4% mAP. This performance increase is also observed in the qualitative results of the base model on examples from the overlap-split test set in Figure A.1 and Figure A.2.

6.6 Further Work in Bird's Eye View Perception

This section briefly highlights seven publications directly related to BEV perception, which the author contributed to, but which are not applied in this thesis. The publications range from proposing a BEV environment model, to defining and exploring the task of dense semantic BEV segmentation from sparse LiDAR data, and incorporating priors for online HD map completion.

A dual evidential top-view representation as environmental model Richter et al. [RBW24]† propose a dual evidential top-view representation as an environment model for automated vehicles. By using evidential multi-modal top-view grid maps, it is possible to model free space, traffic participants, and information on their semantics in a common representation. In this representation, measurements from heterogeneous sensors can be combined, conflicts resolved, and the corresponding evidence can be propagated over time. The method introduces a novel dual-layer representation that separates the environment into two complementary layers: One for evidential tracking of dynamic objects and another for evidential mapping of the ground surface.

Dense semantic BEV segmentation from sparse LiDAR data Bieder et al. [BWJ20]* define the task of dense semantic BEV segmentation from sparse LiDAR data and tackle it by exploiting a multi-layer grid map representation

combined with convolutional neural networks originally developed for image processing. The approach is enhanced in [BLR21]* by fusing deeply learned features from complementary representations, i.e., range view and bird's eye view within a unified network architecture. Two subsequent works by Fei et al. [FPH21]† and Peng et al. [FPH21]† further explore this task by employing PointNet [QSM17] and a multi-attention mechanism.

Priors for online HD map construction and completion When relying solely on scene information observed from the on-board sensor suite, online map construction is inherently constrained by the sensor's range, noise, and limited field of view. To address these constraints, several approaches have proposed to incorporate priors to either extend the range or improve performance of the prediction. Immel et al. propose two methods for incorporating priors in HD map construction: First, [IFB25a]† present M3TR, a single Generalist HD map completion model, capable of handling diverse map priors originated from different HD map degradation scenarios by leveraging a unique training regime and query embedding. Second, SDTagNet [IPF25]† introduce an encoding module for SD map priors, which is based on natural language processing and is capable of utilizing a wide range of available map annotations without manual feature engineering.

6.7 Conclusion

This chapter conducted a comprehensive study on how to effectively train an online HD map construction model on a custom surround-view camera dataset. By integrating recent advancements in label definition and generation for online HD map construction, this work achieves state-of-the-art label quality in terms of high-fidelity, geometric representation and inclusion of crucial map element categories.

The presented ablation studies analyzed the influence of various factors on model performance, including input resolution, camera setup, model initialization, and training sample frequency. Each study was evaluated holistically, considering both overall performance and real-world deployability.

The results demonstrate that, even with a challenging sensor setup and limited dataset size, effective training strategies lead to competitive performance in online HD map construction. The deployed models were able to accurately predict not only simple road layouts, but complex intersection topologies in previously unseen geographic regions. However, performance in regions which are already observed during training does demonstrate even higher accuracy, highlighting the importance of considering the domain gap between training and deployment scenarios.

7 Perspective View Map Perception

The previous chapter pointed out the advantages of a BEV representation as environment model for automated driving, as its spatial consistency brings benefits for multiple downstream tasks like sensor data fusion, tracking or motion planning. While it makes sense for many applications to exploit these advantages in the inference stage, there are several scenarios in which interpreting the environment in the perspective view is preferred or even required. Being able to understand the environment in the sensor's native representation, instead of a reprojection, enables to interpret the field of view on a fine-grained level with pixel or point accuracy in a continuous 3D space. A BEV representation inherently loses spatial information by discretizing into grid cells or voxels, which are typically much coarser than the sensor resolution. Especially in case of 2D BEV representation, height information or vertical arrangements of objects are lost if not explicitly encoded. While this may be negligible for features like lane topology, it becomes disadvantageous when assigning a close arrangement of vertically and horizontally stacked traffic lights to their respective lanes. Even with the shift towards more and more end-to-end learning approaches in automated driving, ground truth annotations in the perspective view of different sensors remain valuable as auxiliary tasks in multitask learning setups, improving feature embeddings and overall performance. Notably, examples for which the loss of spatial information caused by BEV projection negatively affects performance include 3D reconstruction, semantic mapping, and many localization pipelines.

In this chapter, learning from maps is applied to generate annotations in the perspective view of different sensor modalities. It can be divided into two main parts: First, learning from maps is exploited as a cross-modal domain adaptation strategy to transfer knowledge from a richly annotated source domain,

i.e., image domain, to an underrepresented target domain, i.e., LiDAR, via automatically created HD maps. Unlike previous domain transfer approaches, this fully self-supervised method does not require similar sensor modalities or direct overlap of field of view. Instead, it enables extending the angular perception range from a front-view camera to a full 360° view. In the second part, learning from maps is applied in a pure perception context for the front-view camera. The high-resolution and context-rich image domain allows examining learning from maps in perspective view with a more fine-grained level of detail, including a broad variety of map elements, ranging from geometric models to large surface features and very fine-grained road markings. Examples for challenging cases are dashed lines with a width of 12 cm, perceived on a flat angle and at a distance of up to 50 m.

Requirements, limitations and error propagation When learning a perspective map perception with maps as a supervisory signal, a point-or-pixel-accurate reprojection of map elements into the sensor space is essential for multiple learning tasks, such as semantic segmentation. This poses several strict requirements including a highly accurate intrinsic and extrinsic calibration of all sensors, as well as a precise 6D pose graph. Calibration errors harm the precision of the annotation pipeline in multiple ways. First, they introduce errors in the pose graph estimation and the manual or automatic generation of the HD map as both rely on the precise localization of perceived landmarks and objects. Second, calibration errors lead to misalignments in the reprojection of map elements in the sensor space. Furthermore, the resolution of the sensor must be appropriate for the size, detail level, and maximum distance of reprojected map elements. Similarly, pose errors, regardless of their origin, also compromise both HD map creation and reprojection process, causing spatial mismatches between the reprojected map elements and their actual position in the sensor data. Another significant source of errors arises from occlusions. In the mapping stage, occlusions can result in incomplete or erroneous map elements. During the reprojection stage, occluded map elements may appear incorrectly over foreground objects, creating implausible annotations, further addressed in Section 7.2.1.

7.1 Learning from Maps as a Cross-Modality Domain Adaptation Strategy

The fact that large-scale annotated datasets are often restricted to specific sensing modalities or even particular sensor models poses a significant challenge for deploying models across changing sensor setups. For this reason, extensive research has focused on domain adaptation techniques, i.e., methods that transfer knowledge from a source domain with abundant labeled data to a target domain with limited or no annotations.

Most existing domain adaptation approaches in the field of autonomous driving fall into one of two categories: First, techniques that align the source and target domains by adapting sensor-specific characteristics, such as the ray distribution in LiDAR-to-LiDAR adaptation [LMH20, REG19]. The second category includes methods that exploit overlapping sensor fields of view to transfer knowledge from simultaneously perceived objects or structures [CCR23, MGW23]. Both methods impose strong requirements, either on the existence and extent of their shared fields of view or that the sensor modalities are similar enough to allow for effective knowledge transfer. These specific constraints limit the range of applications of such methods on real-world sensor setups.

This section applies the concept of learning from maps to overcome these limitations: It proposes XD-MAP, a fully self-supervised cross-modality domain adaptation framework that leverages HD maps as an intermediate representation to transfer knowledge between heterogeneous sensor modalities without requiring overlapping fields of view. In order to examine this concept, maps with no manually annotated labels are used in order to keep the process fully self-supervised. Instead, the automatically generated parametric map M_{EP} is employed. Its semantically tailored geometric representation of map elements are capable of conserving semantic knowledge from an open source image dataset, i.e., the Mapillary Vistas dataset [NOB17], and serving as a bridge to enable pixel-and-point-accurate pseudo labels for the target sensor, i.e., LiDAR.

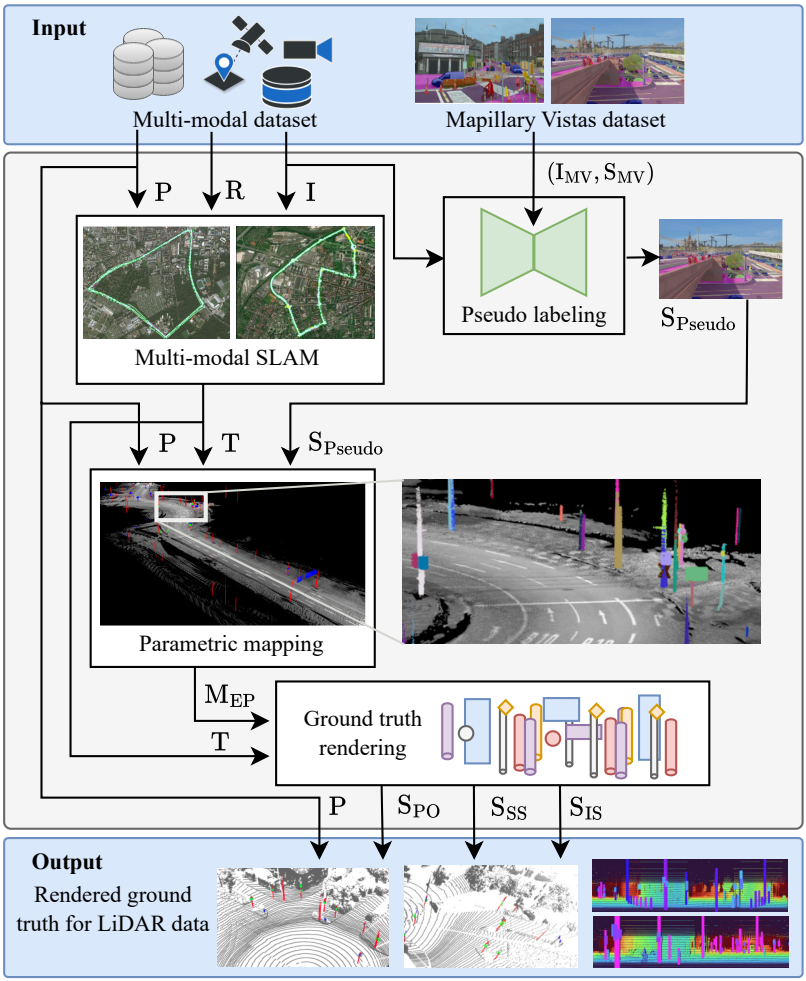


Figure 7.1: Overview of the cross-modal domain adaptation approach XD-MAP. The input consists of two parts: First raw sensor data from the multi-modal dataset, i.e., images, LiDAR, and GPS/IMU data, denoted as I, P, and R, respectively. Second, the Mapillary Vistas dataset for image segmentation, denoted as (I_{MV},S_{MV}). Intermediate results are pseudo labels S_{Pseudo}, a pose graph T and a semantic parametric HD map M_{EP}. The output consists of rendered ground truth for LiDAR data for panoptic segmentation, semantic segmentation and instance segmentation, denoted as S_{PO}, S_{SS}, S_{IS}, respectively.

An overview of the proposed XD-MAP framework is shown in Figure 7.1. It illustrates the individual steps of the pipeline, which starts with a multi-modal dataset and detections of a well-generalizing pretrained neural network in the source domain and ends with rendered ground truth for the target sensor modality. Most of the processing steps have been described in detail in previous chapters. The process includes the pseudo labeling from Section 4.4.2, the multi-modal SLAM from Section 4.4.5 and the parametric mapping from Section 5.6 to create the semantic HD map M_{EP}. The following section describes the pseudo label generation process, the implementation of two baseline approaches and the experimental evaluation of the method. This work has also been accepted for publication in Bieder et al. [BKH26]*.

7.1.1 Label Generation

The map M_{EP} contains three distinct types of elements in semantically tailored geometric representations, i.e., poles and traffic lights, modeled as cylinders, and road signs, modeled as upright planes of various shapes. This section details the process of rendering pseudo labels for 2D and 3D LiDAR perception models as well as two baseline approaches that use single shot pseudo labels instead of a semantic parametric HD map.

Perspective data annotation by projecting map elements In a first step, a set of relevant landmarks $\mathscr{L}_i = \{\ell \in \mathrm{M}_{\mathrm{EP}} \mid \left\|\ell - \mathrm{T}_i\right\|_2 < \tau\}$ from the semantic map M_{EP} is obtained for each ego pose T_i, where the distance between a landmark ℓ and the ego pose is below a predefined threshold τ. τ has to be chosen slightly larger than the maximum range objects are labeled for, as its extent might not be accounted for in the landmark's position. The set of relevant landmarks $\mathscr{L}_i$ is then transformed from the global world frame to the sensor frame with $\mathrm{T}_i^{\mathrm{l}} = \mathrm{T}_l^{\mathrm{E}}\mathrm{T}_i$, using both the vehicle pose and the sensor's extrinsic calibration. Subsequently, key points of each landmark are constructed in the sensor frame and projected using the spherical sensor model of the LiDAR, introduced in 4.4.3. This reconstructs the shape of the landmark in the sensor space producing a polygon set S representing object boundaries. If landmarks

are modeled as 3D geometric primitives, such as cylinders, it might be necessary to first transform it into the sensor frame before constructing key points by a primitive-specific rendering function as the key points depend on the direction of observation. The map elements are processed in the order of decreasing distance from the sensor, while each newly projected element overwrites previously projected ones. This procedure is necessary to account for occlusions and ensures that objects are correctly represented from the sensor's perspective. This representation is the basis from which labels for various 2D or 3D perception tasks can be derived, i.e., semantic segmentation, panoptic segmentation or bounding box detection. While rendering of 2D labels is straightforward from this representation, the 3D point cloud annotation pipeline utilizes the same intermediate representation. It combines each 2D shape with an object-specific depth range, creating a frustum that encapsulates the corresponding map element. All LiDAR measurements which fall within this frustum are then assigned the label of the respective map element. This method ensures that the 3D labels accurately reflect the semantic information from the map while maintaining spatial consistency within the point cloud. Additionally, it effectively handles occlusions by leveraging the depth information inherent in the frustum representation.

Mitigation of annotation uncertainty As addressed at the beginning of the chapter, even small errors in calibration, localization or mapping can lead to misalignments between the rendered labels and the actual sensor data, i.e., cause points near object contours to fall outside the primitive. Since the objects of interest, i.e., traffic lights, traffic signs, and poles, are sparsely distributed in the scene, the reduction of false negative instance annotations is prioritized even at the cost of introducing false positives. To mitigate the impact of these misalignments in 3D, an uncertainty margin is introduced by increasing the radius of each cylindrical object by 5 cm and by 7 cm for traffic lights and poles, respectively. Additionally, traffic signs which are modeled as upright 2D planes are extended to 3D boxes with a depth of 10 cm. For 2D annotations, a minimal dilation operation is applied to the rendered shape, resulting in a 1-pixel expansion of each segment. This adjustment also prevents artifacts such as poles from

being rendered with a width of zero. All points or pixels not assigned to any object are labeled as background to clearly distinguish them.

Baseline approaches Two baseline approaches are implemented that generate pseudo labels directly from single shot 2D instance segmentations without utilizing the semantic parametric HD map. One specifically aims to preserve the object's shape, while the other focuses on retaining accurate depth information for each point within an object. The major challenge for these approaches is to precisely infer an object's 3D position and the spatial extent in LiDAR space based on a 2D polygon in the camera image. In addition to that, there are inherent limitations: First, the accuracy suffers from the parallax effect due to the misalignment of the optical centers of the sensors, especially due to the close-to-production sensor mounting. The parallax effect is illustrated in Figure 7.2. Second, annotations are only possible within the shared field of view between camera and LiDAR sensor, which amounts to roughly 100°. While object instances detected in the camera image within that range are projected into LiDAR space and annotated accordingly, the remaining 260° of the LiDAR are labeled to be ignored during training to avoid negatively affecting the optimization process.

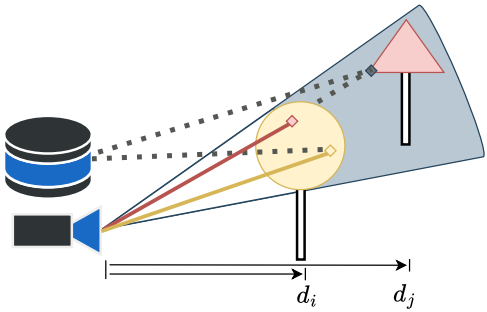


Figure 7.2: The parallax effect is caused by the different locations of the optical centers of the sensors. The gray area represents the occlusion by the yellow traffic sign in the field of view of a camera. The red traffic sign is fully occluded in the field of view of the camera, yet detections of a slightly higher mounted LiDAR sensor still capture it. When LiDAR points are projected into the camera image, the parallax effect causes detection of objects occluded by the yellow traffic sign in the camera image to be projected onto it.

The first baseline, *the Shape-Preserving Lifting* XD-B1, lifts 2D front-view instance masks into 3D. It assigns a single representative LiDAR depth to each instance, estimating it by using the 30^{th} percentile of the depths of the LiDAR points falling within the respective instance. Selecting the 30^{th} percentile instead of the mean aims to mitigate parallax effects, as background LiDAR points may be incorrectly projected onto foreground objects in camera space. In order to preserve geometric contours of the objects, the lifted instance polygons are then reprojected into the spherical LiDAR representation.

The second baseline, *the Depth-Preserving Lifting* XD-B2, uses the 2D semantic instance masks as a lookup table associating instance labels with LiDAR points. Instead of estimating a single depth, all LiDAR points corresponding to an instance are used to construct a convex hull. This pseudo labeling mechanism prioritizes the accurate retention of depth information for each LiDAR point within an instance without enforcing a specific shape.

Both baselines apply the same uncertainty mitigation strategies as described in Section 7.1.1 to reduce false negative annotations. A direct comparison of the three labeling approaches is presented in Figure A.4. It demonstrates the advantages of the map-based labeling approach, both in terms of annotation accuracy and coverage.

7.1.2 Experimental Evaluation

Experiment layout The XD-Map framework is evaluated on three different tasks: 2D semantic segmentation, 2D panoptic segmentation, and 3D semantic segmentation. It is benchmarked against two single-shot baseline approaches, XD-B1 and XD-B2, using the corresponding metrics defined in Section 3.3.3. In ablation studies, the influence of motion compensation, training sample frequency and map element range on performance is investigated, as these factors are important for practical considerations in data collection, preprocessing and downstream applications. Since our labels are not pixels in an individual image, but generated from mapped geometric primitives in 3D space, motion compensation impacts the congruence of the labels with the range image. By comparing all approaches with and without motion compensation, its impact is

systematically evaluated. For training sample frequency the same notations as in Section 7.2.2 are used, i.e., $f_{S/1}$ for 10 Hz, $f_{S/5}$ for 2 Hz and $f_{S/20}$ for 0.5 Hz. The maximum possible range of detected map elements is important for different downstream applications. While a higher detection range is beneficial, it must be balanced with system parameters such as map element size, sensor resolution, and backprojection quality. Figure 7.3 illustrates the height and width of different map elements as a function of their distance to the ego vehicle. It demonstrates that the spatial extent of objects in the sensor data already becomes very small at medium ranges of 50 m, especially for the width of traffic lights and poles. Therefore, XD-MAP performance is evaluated for three maximum detection ranges: 30 m, 50 m, and 70 m. Including farther ranges is also important, as motion compensation might affect farther points more severely. Thus, the interaction of motion compensation and range is analyzed.

For the 2D input representation, a spherical range image with a resolution of 1812 px × 200 px is used, covering a vertical field of view of approximately 30° and a horizontal field of view of 360°. While the logarithmic intensity is encoded in one 8-bit channel, two 8-bit channels are used to encode the depth information. The experiments are conducted using all five sequences and a geographic split for training and evaluation is applied as described in Section 6.3, i.e., separation of training and test set by at least 100 m.

Models and hyperparameter setting Two architectures are employed, Mask2Former [CMS22] and Cylinder3D [ZZW21], to evaluate the quality of the generated labels for 2D and 3D segmentation, respectively.

To evaluate the quality of the generated 2D labels, Mask2Former [CMS22] is employed as the unified model architecture for both semantic and panoptic segmentation. The model is optimized using AdamW [LH19] with a base learning rate of 0.0001, a weight decay of 0.05, and gradient clipping set to 0.01 for the entire model. Group Normalization [WH18] with 32 groups is applied as well as a backbone learning-rate multiplier of 0.1 to partially preserve the pretrained feature representations of the ResNet-50 [HZR16] backbone, initialized with ImageNet [DDS09].

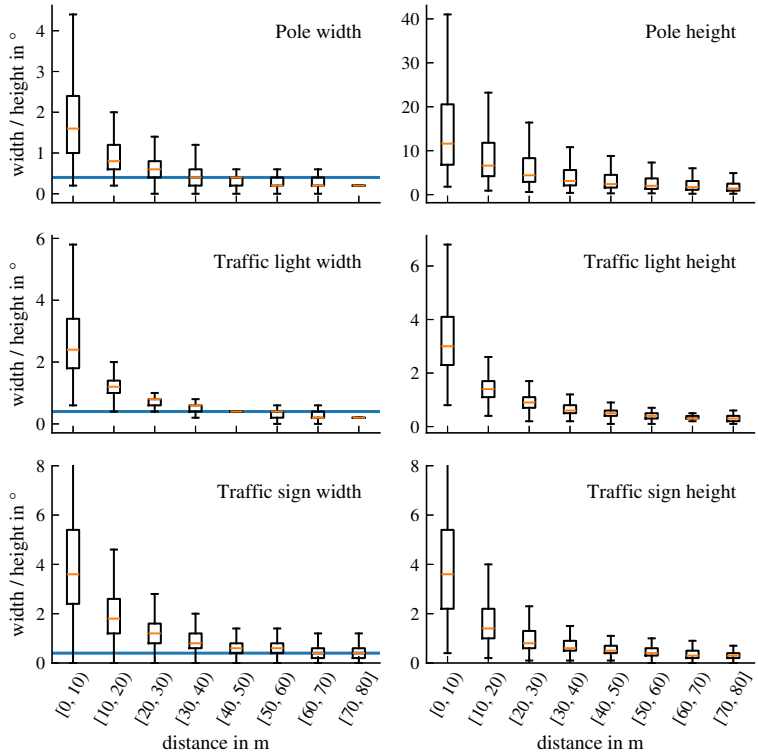


Figure 7.3: Boxplot of instance height and width in the field of view of the LiDAR as a function of its distance to the ego vehicle. The blue line symbolizes the constant horizontal resolution of 0.2°. Even with the high-resolution LiDAR, objects like traffic lights and poles are in a distance above 50 m often not even a pixel in width due to their limited spatial extent.

To account for the modality differences between range images and RGB images, RGB-specific augmentations are disabled, and the learning rate multiplier of the pretrained backbone is increased to 0.4 from 0.1. Random cropping is also disabled during training due to the already limited input height. The spherical range images are cropped in height resulting in a 1812 px × 200 px input size, as regions outside this vertical span contain almost no LiDAR detections.

For Cylinder3D, the hyperparameters are adopted from the ones used for training on SemanticKITTI [TDC23], except for the number of epochs. The model is trained 12 epochs with a batch size of 16. AdamW [LH19] optimizer is used in combination with a base learning rate of 0.001 and weight decay of 0.01.

7.1.3 Results

2D semantic and panoptic segmentation Table 7.1 summarizes the quantitative results for the 2D semantic segmentation and panoptic segmentation tasks. For configuration parameters not specified, the default configuration is a maximum element range of 50 m and a training sample frequency of 10 Hz. All three map element types are considered as *thing* classes, while only one *stuff* class, i.e., background, exists, which is excluded from the evaluation. Compared to the single shot baselines XD-B1 and XD-B2, XD-MAP yields strong performance increases. It improves the mIoU in semantic segmentation by +27.2 and +19.5 points and the PQ_{th} by +21.7 and +19.5 points for XD-B1 and XD-B2, respectively. This is present across all element classes for semantic segmentation. For panoptic segmentation, performance increases can almost entirely be attributed to higher recognition quality PQ_{th}, when compared with other metrics. Hence, models trained with XD-MAP have much higher precision and recall of detected elements. This showcases the benefits of a systematic approach combining a highly accurate SLAM with the parametric mapping of geometric primitives for cross-modal domain adaptation against simpler approaches limited by camera field of view or pseudo-label inaccuracies. Qualitative results for 2D panoptic segmentation are presented in Figure 7.4 demonstrating the self-supervised cross-modal domain adaptation capabilities of XD-MAP.

The evaluation with different element ranges presents a lower performance for higher label element ranges. Between 30 m range and 70 m, the mIoU for the semantic segmentation task results in a difference of -3.4, while for the panoptic segmentation task it is slightly larger at -8.7 PQ_{th}. The main factor here is the lower recognition quality RQ_{th}. The segmentation quality SQ_{th} only declines by a small amount, implying the model's trouble with detecting smaller

elements appearing due to an increased element range. Another factor contributing to lower performance is decreased training sample frequency. However, a frequency of 2 Hz decreases mIoU by -0.6 for semantic segmentation and by -0.4 PQ_{th} for panoptic segmentation. This implies that for use cases with stricter requirements on dataset size, similar performance can be reached with only a quarter of the storage. In the case of 2D semantic segmentation, inactive motion compensation leads to steadily decreasing performance across tasks, with mIoU drops ranging from -0.5 to -2.9. These results further underline the importance of careful dataset curation, including the reduction of sensor noise when possible.

A further observation is that training Mask2Former for panoptic segmentation tasks yields a model that also performs slightly better in semantic segmentation compared to training it directly in a semantic segmentation setup. Table A.7 also shows the semantic segmentation evaluation of the base XD-MAP panoptic

Table 7.1: Quantitative results for the 2D semantic segmentation and panoptic segmentation tasks. All results are reported in %.

	Experiments	Mot. Comp.	Semantic				Panoptic		
			IoU_{Po}	IoU_{TL}	IoU_{TS}	**mIoU**	SQ_{th}	RQ_{th}	$\mathbf{PQ_{th}}$
Baseline	XD-B1	✗	9.6	9.1	9.2	9.3	65.2	7.4	4.8
	XD-B2	✗	15.0	18.6	17.0	16.9	63.4	11.7	7.4
	XD-MAP	✗	34.7	39.4	29.2	34.4	67.8	35.9	24.4
	XD-B1	✓	9.9	11.3	8.1	9.8	64.3	8.9	5.8
	XD-B2	✓	17.9	18.6	16.1	17.5	63.6	12.6	8.0
	XD-MAP	✓	**37.1**	**42.3**	**31.8**	**37.0**	**69.4**	**39.6**	**27.5**
Range	30 m	✗	35.4	42.9	30.2	36.2	68.1	41.2	28.1
	50 m	✗	34.7	39.4	29.2	34.4	67.8	35.9	24.4
	70 m	✗	32.5	37.1	29.1	32.9	67.1	33.0	22.1
	30 m	✓	40.0	45.4	31.9	39.1	70.3	47.4	33.4
	50 m	✓	37.1	42.3	31.8	37.0	69.4	39.6	27.5
	70 m	✓	36.4	40.1	30.8	35.7	69.0	35.7	24.7
Freq.	$f_{S/20}$	✓	33.9	38.5	28.4	33.6	68.7	36.1	24.8
	$f_{S/5}$	✓	**37.4**	41.2	30.8	36.4	69.2	39.1	27.1
	$f_{S/1}$	✓	37.1	**42.3**	**31.8**	**37.0**	**69.4**	**39.6**	**27.5**

model and its relative performance increase. This might be due to the additional supervision signal provided by instance annotations.

3D semantic segmentation Table 7.2 summarizes the quantitative results for the 3D semantic segmentation task. It shows similar results to the other tasks with a few notable differences. XD-MAP shows large improvements in mIoU compared to the single-shot baselines. For both baselines, yet XD-B1 in particular, many objects remain unlabeled, leading models to predict mostly background. While increasing the element range again reduces performance, the decrease is smaller than in 2D. This implies that long-range segmentation benefits from the richer 3D representation of point clouds. In contrast, the absence of motion compensation leads to a much stronger performance decline, which could indicate that our 3D labeling is more dependent on a precise alignment of LiDAR points and HD map to ensure points are located in an object's frustum. This highlights that one relative advantage of 2D labeling is its lower sensitivity to localization, mapping or calibration errors. Qualitative results are presented in Figure 7.5. It presents that, to the human eye, the prediction appears to surpass the ground truth annotation in a lot of the instances.

Table 7.2: Quantitative results for 3D semantic segmentation. For configuration parameters not specified, default configuration is [XD-MAP, 10 Hz, 50 m]. Results are reported in %.

	Experiments	Mot. Comp.	Semantic			
			IoU_{Po}	IoU_{TL}	IoU_{TS}	**mIoU**
Baseline	XD-B1	✓	0.5	5.1	17.1	7.6
	XD-B2	✓	0.5	7.6	26.6	11.6
	XD-MAP	✗	26.8	42.8	30.4	33.3
	XD-MAP	✓	**41.4**	**52.3**	**38.0**	**43.9**
Range	30 m	✓	40.5	53.4	38.0	44.0
	50 m	✓	41.4	52.3	38.0	43.9
	70 m	✓	40.2	50.9	37.8	43.0
Freq.	$f_{S/20}$	✓	39.9	50.3	34.1	41.5
	$f_{S/5}$	✓	41.6	51.9	37.1	43.5
	$f_{S/1}$	✓	**41.4**	**52.3**	**38.0**	**43.9**

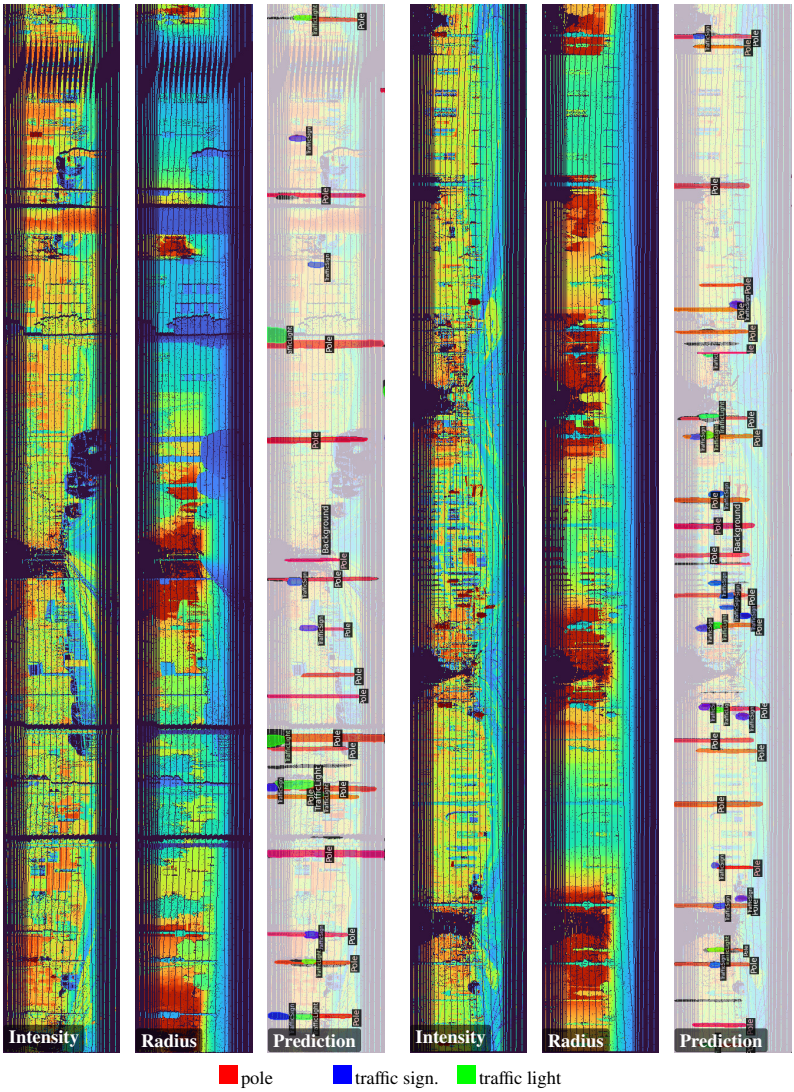
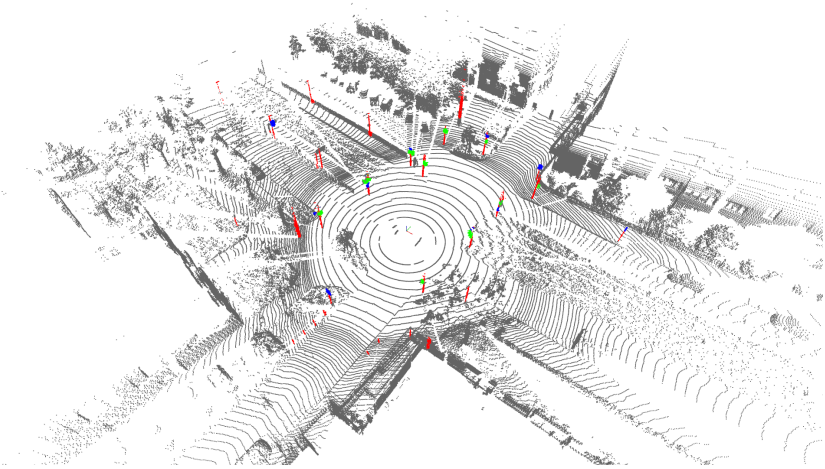
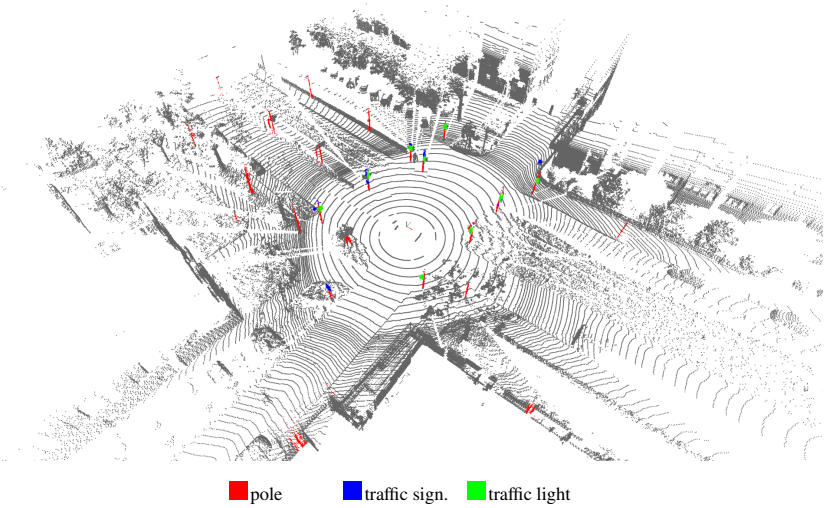


Figure 7.4: Qualitative 2D prediction results of XD-MAP model, showing panoptic segmentation predictions for two example scenes from the geo. split evaluation set. For each scene, the top rows show the 2D input, i.e., intensity and range encoded as a spherical projection, while the bottom row displays the corresponding panoptic segmentation results.



(a) Prediction results for 3D semantic segmentation on eval set with geographic split.



(b) Ground Truth example from the eval set, generated using the XD-Map approach.

Figure 7.5: Qualitative 3D prediction results of XD-MAP model, showing an example of 3D segmentation prediction and corresponding ground truth annotations. Notably, for some objects, the model's prediction even appears to surpass the ground truth in visual quality.

7.2 Front-View Map Perception

Cameras, with their unique capabilities, are an indispensable sensor modality in the context of autonomous driving. They are used across a wide range of tasks, many of which benefit from high-quality annotations in the perspective view. Therefore, this section applies learning from maps to generate pixel-accurate annotations for the front-facing camera C^{c}_{FV} as sensor modality. Towards this goal, map elements from the HD maps M_{EP} and M_{DS} are rendered into the image space. Besides the map elements modeled as geometric primitives, the resulting label set also includes very large surface features and very fine-grained road markings, introducing specific challenges for the annotation process and testing the limits of learning from maps further.

7.2.1 Label Generation

The label rendering process for front-view map perception is nearly identical to the one presented in Section 7.1.1 for the 2D case. However, some specific considerations arise from camera models with limited projection fields and more heterogeneous map elements, due to the addition of M_{DS}. In this context, this section discusses how to mitigate edge cases, annotation artifacts and ensure annotation fidelity.

Mitigate annotation and projection artifacts Map elements can be annotated in different ways which, if not accounted for, may introduce inconsistencies in the training data and negatively affect learning. For example, polylines with uneven point spacing can cause annotation-specific artifacts in the training samples. In addition, wide point spacing that appears linear in BEV may produce distortions in the perspective view due to projection effects and varying point heights. To address this, polylines are densely resampled to ensure a consistent spacing with a maximum of 10 cm between points. Smaller annotation spacings are preserved, yet very rare and typically occur only for very small map elements. Further artifacts can result from inconsistent labeling of non-confined map elements in large annotation spaces. Considering long solid

lines, represented as polygons, annotators often divide them into multiple adjacent segments, sharing a line segment. While this is irrelevant when deriving semantic segmentation annotations, it may lead to artifacts in instance-level annotations. This issue can be mitigated through a post-processing step after the rendering of map elements in the target sensor space which aims to identify and merge instances which belong together.

Additional label artifacts can occur during the projection of map elements using camera models, for example when large surfaces such as roads have corners located behind the camera that cannot be projected into the image plane in a pinhole model. As ignoring these element corners can lead to annotation artifacts, a reprojecting or resampling strategy is applied to ensure a complete and meaningful annotation.

Annotation fidelity, map verification and occlusion handling When maps are used as a supervision signal, the quality of the generated training data depends critically on the accuracy and plausibility of the annotations. Erroneous labels may result from misalignments between reprojected map elements and their true position in the sensor data, or from outdated map content. While the former can be mitigated through precise calibration, localization, and mapping, the latter can be reduced by frequent map updates or resolved through consistent map verification before training data generation, as discussed by Pauls [Pau24]. Other implausible annotations can arise from occlusions. In some cases, partially occluded map elements remain meaningful annotations and desired to be predicted from sensor data. Typical examples are training instances from the HD map construction task, which were discussed in Chapter 6, i.e., continuous road borders, lane divider or road topology annotations in a BEV representation. Even if partially occluded, these elements can be reasonably inferred from their visible parts or purely from contextual cues, such as a road border hidden behind parked vehicles. However, there are also more confined map elements whose position or even existence cannot be reasonably inferred if fully occluded such as traffic signs blocked by a truck or dashed lines obscured by a construction fence, making their annotation unfavorable.

A general approach to identify occluded map elements is to leverage depth information from the sensor data. By comparing the distance of a reprojected map element with the measured depth along its line of sight, occlusions can be identified if the deviation is above a certain threshold. In the master's thesis by Kirik [Kir20], a stereo-based approach for occlusion handling in the context of learning from maps is developed, which searches for homogeneous surfaces in the depth maps that are in the line of sight of a map element. Similarly, the 3D LiDAR annotation pipeline presented in Section 7.1.1 constructs a frustum for each map element, ensuring that only LiDAR points within the frustum are assigned the corresponding label.

Within this work, a simple yet effective specialized alternative for occlusion handling in the context of learning from maps is developed. It is based on two key observations. First, most occlusions of relevant traffic-scene elements are caused by objects belonging to dynamic classes such as cars, trucks, or pedestrians, regardless of whether they are moving or not. Second, state-of-the-art perception models for the autonomous driving domain can detect these dynamic object classes with high accuracy across different sensor modalities and tasks, given their high relevance in autonomous driving. Since these dynamic objects are not part of a static HD map and map elements are typically not transparent, an arbitrary map element projected into a region where a dynamic object is detected can be considered as occluded. This idea, also presented in [BHS23]*, is illustrated in Figure 7.6 for map elements of M_{DS}, using a binary semantic

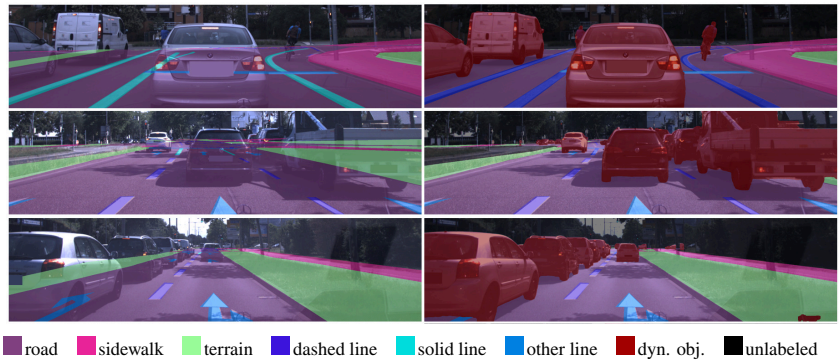


Figure 7.6: Examining dynamic occlusion handling by binary semantic mask.

mask for dynamic objects obtained by Section 4.4.2. It illustrates scenarios in which road features are fully occluded by dynamic objects, rendering the annotations implausible. In other cases, only parts of a map element are occluded, yet the visible parts can still be reasonably inferred. This occlusion handling approach is applied to most perspective-view front-facing camera experiments.

7.2.2 Experimental Evaluation

Experiment layout Similar to Section 7.1.2, the front-view map perception is evaluated on 2D semantic segmentation and 2D panoptic segmentation. However, the label set is extended compared to the previous section as it also includes elements from the dense surface map M_{DS}. An overview of the complete set of map elements for M_{DS} and M_{EP} is provided in Table A.3 and Table A.4, respectively. For the panoptic segmentation task, thing classes are represented by all three parametric map element types, i.e., poles, traffic lights and road signs, and the three lane marking types, i.e., solid lines, dashed lines and other lines. The three remaining dense surface elements, i.e., road, sidewalk and terrain, are considered as stuff classes. Additional stuff labels are the background class and the dynamic occlusion class, both later removed for the mIoU calculation of the semantic segmentation task. This amounts to a total of 11 semantic classes, for which six are considered thing and five stuff classes.

In the experiments, the influence of different factors on the segmentation performance is investigated in ablation studies. The base configuration employs both the dynamic occlusion handling M_{dy} and the fusion-based ground surface reconstruction method G_f. This configuration is evaluated on different training sample frequencies including two multi-drive setups, i.e., f_M and f_{Mges}. In addition to that, experiments without dynamic occlusion handling and with labels generated using the stereo-based ground surface reconstruction method are conducted to examine their influence on performance. Here, a direct comparison between the experiments has to be conducted carefully, as the generated ground truth labels differ for the evaluation.

Models and hyperparameter setting For the front-view perception experiments, the same Mask2Former [CMS22] architecture as in Section 7.1.2 is used to evaluate the quality of the generated labels for 2D semantic and panoptic segmentation. The hyperparameters are chosen analogous with only minor modifications: The adjustments to account for the LiDAR-specific input representation are reverted, namely the backbone learning-rate multiplier is set to 0.1, random cropping is enabled and RGB-specific augmentations are applied. For all experiments, images of the front-view camera C^{c}_{FV} are used as input to the model. As region of interest, a bottom-center crop of 2048 px × 1024 px from the original image is defined for model input and ground truth annotation. During the training process, images are further randomly cropped to 1024 px × 512 px for data augmentation. Evaluation is performed on the full 2048 px × 1024 px crop.

Furthermore, the usual geographic data split is applied for training and evaluation, i.e., separation of training and test set by at least 100 m. The only exception is that SQ_C is excluded from the training due to quality issues in the M_{DS} map.

7.2.3 Results

Figure 7.7 and Figure 7.8 show qualitative examples of the panoptic segmentation results on the geographically distinct test set using the base model trained with $f_{S/1}$ and both G_f and M_{dy} enabled. It can be observed that even small and distant elements like traffic lights, traffic signs and various lane marking types are consistently detected. Quantitative results are summarized in Table 7.3 and Table 7.4 for the panoptic segmentation and semantic segmentation tasks, respectively. Regarding training sample frequency for the base experiment setting, the results match the trends observed in the previous chapters, with performance improving as the frequency increases. Again the geographically-equal-sampled multi-drive setup f_{Mges} achieves the highest performance in both, tasks outperforming the standard multi-drive setting and improving the mIoU by 2.1 points and the PQ by 1.1 points compared to the single-drive setup $f_{S/1}$. In case of panoptic segmentation, the main improvement of it can be attributed to better recognition quality.

The gray lines in both tables separate different ground truth types, which cannot be directly compared or at least only with limitations regarding the generality of conclusions. Both the different ground surface reconstruction and the dynamic occlusion handling influence the annotation strategy and label distribution. All experiments are evaluated on the respective ground truth type they were trained on.

Table 7.3: Quantitative results for panoptic segmentation of front-view images on the geo. split evaluation set. Models are trained with different combinations of dynamic occlusion handling, ground surface reconstruction methods, and training sample frequencies. The use of dynamic occlusion handling and fusion-based ground reconstruction is denoted by M_{dy} and G_f, respectively. Gray lines separate distinct ground-truth types, which are not directly comparable across groups. Results are reported in %.

Parameters			**Recognition Qual.**			**Segmentation Qual.**			**Panoptic Qual.**		
M_{dy}	G_f	freq.	RQ_{st}	RQ_{th}	RQ	SQ_{st}	SQ_{th}	SQ	PQ_{st}	PQ_{th}	PQ
✗	✗	$f_{S/1}$	68.3	28.0	44.1	77.8	70.1	73.2	56.6	19.8	34.5
✓	✗	$f_{S/1}$	74.0	27.8	48.8	77.8	70.2	73.6	60.4	19.6	38.2
✗	✓	$f_{S/1}$	70.2	26.4	43.9	76.5	70.0	72.6	57.6	18.7	34.2
✓	✓	$f_{S/40}$	71.9	18.5	42.8	75.9	67.7	71.4	57.6	12.6	33.1
✓	✓	$f_{S/20}$	73.5	23.0	45.9	75.1	68.1	71.3	58.5	15.8	35.2
✓	✓	$f_{S/5}$	75.9	27.3	49.4	76.8	69.3	72.7	61.0	19.1	38.1
✓	✓	$f_{S/1}$	75.7	28.1	49.8	**77.3**	69.2	72.9	61.3	19.7	38.6
✓	✓	f_{M}	76.2	29.0	50.5	**77.3**	**69.6**	**73.1**	61.5	20.4	39.1
✓	✓	f_{Mges}	**76.7**	**30.0**	**51.2**	**77.3**	69.5	73.0	**62.0**	**21.1**	**39.7**

When comparing the semantic segmentation results, it can be observed that the base configuration with both G_f and M_{dy} enabled yields the higher performances on the respective ground truth. The improvement is even more pronounced when comparing to the models without dynamic occlusion handling. Here, it cannot be stated conclusively if the improvement is mainly due to the fact that more difficult predictions, e.g., road markings behind trucks, are also included in the ground truth or if more meaningful labels have improved the supervision. An argument for the latter is that when comparing the base setting with its counterpart without M_{dy} all three elevated map element classes also

show performance increases, while being occluded less frequently. A comparison of two qualitative prediction examples for these two models is shown in Figure A.5.

Table 7.4: Quantitative results for semantic segmentation of front-view images on the geo. split evaluation set. Models are trained with different combinations of dynamic occlusion handling, ground surface reconstruction methods, and training sample frequencies. The use of dynamic occlusion handling and fusion-based ground reconstruction is denoted by M_{dy} and G_f, respectively. Gray lines separate distinct ground-truth types, which are not directly comparable across groups. Results are reported in %.

Parameters			**Semantic Segmentation.**									
M_{dy}	G_f	freq.	IoU_{RO}	IoU_{SW}	IoU_{TE}	IoU_{DL}	IoU_{SL}	IoU_{OL}	IoU_{TS}	IoU_{TL}	IoU_{Po}	**mIoU**
✗	✗	$f_{S/1}$	81.7	28.0	58.0	47.8	32.5	34.4	54.6	52.4	43.7	48.1
✓	✗	$f_{S/1}$	81.7	27.9	59.1	48.9	35.4	38.5	55.6	54.3	43.9	49.5
✗	✓	$f_{S/1}$	81.7	32.3	57.6	42.8	38.9	36.1	53.0	52.6	44.2	48.8
✓	✓	$f_{S/20}$	82.1	34.4	59.0	41.8	37.9	38.3	54.3	51.7	43.3	49.2
✓	✓	$f_{S/5}$	83.0	37.0	60.3	43.7	**41.5**	**41.3**	54.1	53.2	44.8	51.0
✓	✓	$f_{S/1}$	82.0	32.0	59.7	45.3	40.7	40.6	55.3	53.2	44.5	50.4
✓	✓	f_M	84.6	40.8	59.8	45.0	39.3	39.7	57.8	54.5	**45.0**	51.8
✓	✓	f_{Mges}	**85.0**	**42.2**	**62.3**	**45.4**	41.0	39.7	57.4	**54.9**	44.7	**52.5**

RO: road; SW: sidewalk; TE: terrain; DL: dashed lines; SL: solid lines; OL: other line markings; TS: traffic signs; TL: traffic lights; Po: poles.

Figure 7.7: Qualitative evaluation of perspective panoptic segmentation. Top to bottom: RGB image, annotated ground truth, prediction results of panoptic segmentation shown as semantic masks and full panoptic output. Examples are from the geographically distinct test set.

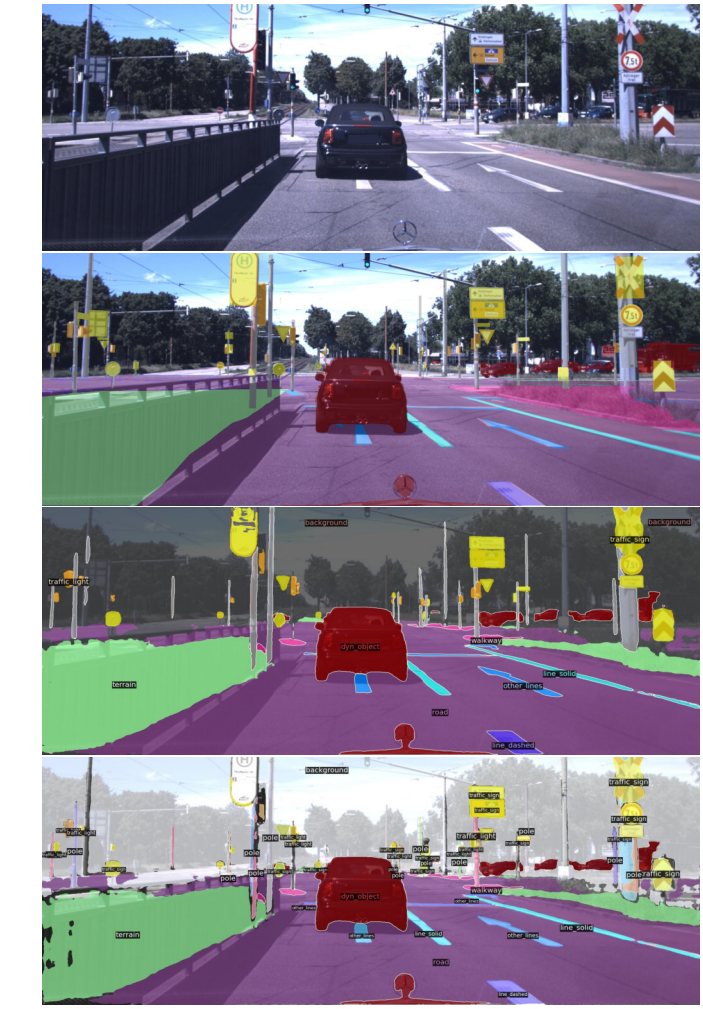


Figure 7.8: Qualitative evaluation of perspective panoptic segmentation. Top to bottom: RGB image, annotated ground truth, prediction results of panoptic segmentation shown as semantic masks and full panoptic output. Examples are from the geographically distinct test set.

7.3 Conclusion

This chapter conducted a comprehensive exploration of training perspective view map perception models using maps as a supervisory signal. Two main application scenarios were investigated: First, a framework for fully self-supervised cross-modal domain adaptation from images to LiDAR via automatically generated HD maps was presented. In contrast to previous domain adaptation methods, this approach does not require similar sensor modalities or overlapping fields of view, enabling knowledge transfer from a front-view camera, to a 360° LiDAR as target. This method makes knowledge, encapsulated in richly annotated image datasets, accessible to underrepresented or specialized sensor modalities, such as LiDAR. Although demonstrated for three exemplary static object classes, the approach can be extended to arbitrary semantic categories, including dynamic objects, if 4D reconstruction is performed, as discussed in Section 8.2.

Second, the generation of high-fidelity annotations for front-view cameras was explored, comprising a wide range of map elements, from geometric models to large surface features and fine-grained road markings. Experiments have shown that the presented approach can effectively generate pixel-accurate annotations even for challenging object classes, such as dashed lines or traffic lights. This also indicates an even broader applicability beyond the representative set of map elements examined in this work. Overall, the conducted experiments highlight that learning from maps is a versatile strategy for training perspective view map perception models across different application scenarios.

8 Conclusion and Outlook

8.1 Conclusion

Regardless of whether an autonomous vehicle aims to localize itself within an HD map, verify that the map is still up-to-date, or perceive the map information on-the-fly from onboard sensors to enable map-less driving: In all cases, a reliable and powerful map perception is essential.

To scale the training of map perception systems, this work proposed the concept of *learning from maps* as a strategy to leverage HD maps as supervision source to train diverse map perception models across tasks and sensor modalities. Although individual aspects of this concept have been applied in prior works, this dissertation is the first to provide a comprehensive exploration of *learning from maps* as a general framework for scalable ground truth generation in autonomous driving.

Chapter 4 defined the key requirements for datasets that enable the effective use of HD maps as a supervision source for diverse map perception tasks. These requirements comprise a high-resolution sensor suite with production-like mounting positions, accurate calibration and a precise multi-drive SLAM framework. The presented work successfully applied the process of creating a multi-drive dataset that completely fulfills these criteria, which then served as the foundation for generating complementary HD map layers in Chapter 5. These map layers differ fundamentally in their generation process and contained map elements, varying in geometric representation, abstraction level, and functional role. The resulting HD maps achieve a centimeter-level precision when back-projected into the sensor space, allowing to consistently match the back-projection of tiny map elements with the corresponding sensor measurements. The included map elements range from physical road borders to higher-level

features such as lane centerlines and from large surface features like solid line markings to fine-grained cylindrical traffic lights.

This enables the exploration of *learning from maps* across different feature representations and functional roles, e.g., learning a pixel-wise segmentation of traffic lights in images or the lane topology in BEV. Chapter 6 conducted a comprehensive study on how to optimize the training of BEV map perception models for online HD map construction. By integrating recent advancements in label generation and definition, it achieved state-of-the-art label quality in terms of geometric fidelity and inclusion of often overlooked map element categories, e.g., divider types, crucial for downstream driving tasks. The chapter concluded that, even with a challenging sensor setup and limited dataset size, effective training strategies lead to competitive performance in online HD map construction.

Complementary to the BEV experiments, Chapter 7 extended the concept to perspective view map perception. It demonstrated that map elements of diverse geometric representations and functional roles can be effectively exploited as supervision source for different tasks and sensor modalities in perspective view. This was first shown in a fully self-supervised cross-modal domain adaptation framework, transferring knowledge from a richly annotated image dataset to the LiDAR domain via automatically generated HD maps. A second line of experiments generated high-fidelity, pixel-accurate annotations for front-view cameras, where trained models consistently predict diverse map element classes, ranging from geometric models to large surface features and fine-grained road markings.

This thesis demonstrated that *learning from maps* is an effective approach to train map perception models across different tasks, representations, and sensor modalities. The perception models and map elements of the conducted experiments are chosen to span a representative set that fulfills the needs of various map perception tasks. Hence, *learning from maps* can be extended naturally to a much broader range of elements and perception tasks.

8.2 Outlook

Learning from maps represents a promising step toward scalable autonomous driving. It facilitates the transition from map-based to less-map or even map-less driving, extending both coverage and reliability. A natural extension is the use of foundation models, i.e., neural networks trained on huge amounts of data with unprecedented generalization capabilities, to refine and improve the quality of HD map generation or the addition of novel tasks using zero-shot labeling capabilities. However, there are two even more foundational ideas that can be built upon this thesis.

Closing the Cycle of Semantic Mapping, Map Verification and Learning from Maps

While learning perception models from maps imposes the requirement of an accurate and up-to-date map, the automatic generation of such maps and their verification requires the capability of detecting map elements from on-board sensors, i.e., a reliable map perception capability. It is a classical chicken-and-egg problem, which can initially be resolved through manual annotation of HD maps or sensor data. Once this foundation is established, a cyclic dependency emerges between the accuracy of mapping, map verification and learning from maps. More accurate perception models enable more precise maps, which in turn can be used to generate more accurate training data. While errors in any component can propagate and potentially amplify through the cycle, a well-designed framework can aim to turn this dependency into a *self-improving feedback cycle*. For example, shortcomings of the perception model, i.e., misclassifications or missed detections, can be mitigated by leveraging unlimited computational resources and extensive sensor data from both multiple sensors and viewpoints in the mapping stage. While the dissertation of Pauls [Pau24] addresses map verification and Hu [Hu26] semantic mapping, this work focused on *learning from maps*. As a next step, the joint interplay of all three components aiming to create the self-improving cycle should be considered.

Integration in a Wider Framework to Learn from 4D Scene Representations

By leveraging 3D maps for large-scale supervision, generated ground truth is restricted to static objects and environment. In order to overcome these limitations, the 3D map representation can be extended into the temporal domain. The resulting 4D scenario representation is built upon the 3D map, incorporating next to the 3D shape of objects also their time-varying 6D poses. This allows for the reprojection of both static and dynamic objects into different sensor modalities, and thus the generation of comprehensive ground truth for a wide range of tasks. Here, the presented multi-drive SLAM approach is particularly well suited, as it allows to accumulate multiple recordings of different times in one 4D scenario representation. Recent works by Qi et al. [QZN21] and Ma et al. [MYZ23] particularly focus on offboard 3D object detection from point cloud sequences. Others like Shi et al. [SWZ25] aim for full offboard semantic scene completion and 3D scene reconstruction. Both lines of works would be potential candidates to be integrated into such a 4D scenario representation framework.

Bibliography

[AMT23] Agarwal, S.; Mierle, K. and Team, T. C. S.: Ceres Solver. Version 2.2. Oct. 2023. url: https://github.com/ceres-solver/ceres-solver (cit. on p. 63).

[ARB15] Aeberhard, M.; Rauch, S.; Bahram, M.; Tanzmeister, G.; Thomas, J.; Pilat, Y.; Homm, F.; Huber, W. and Kaempchen, N.: "Experience, Results and Lessons Learned from Automated Driving on Germany's Highways". In: *IEEE Intelligent Transportation Systems Magazine* 7.1 (2015), pp. 42–57. doi: 10.1109/MITS.2014.2360306 (cit. on p. 158).

[Atl20] Atlatec GmbH: How Accurate Are HD Maps for Autonomous Driving and ADAS Simulation? 2020. url: medium.com/atlatec-gmbh/how-accurate-are-hd-maps-for-autonomous-driving-and-adas-simulation-9f68fa89f840 (last retrieved 2025-01-21) (cit. on pp. 7, 9).

[Aur91] Aurenhammer, F.: "Voronoi diagrams—a survey of a fundamental geometric data structure". In: *ACM Computing Surveys* 23.3 (1991), pp. 345–405 (cit. on p. 160).

[Aut23] Autoware Foundation: Autoware Documentation. 2023. url: autowarefoundation.github.io/autoware-documentation/main/ (last retrieved 2025-01-21) (cit. on p. 9).

[BE12] Biagioni, J. and Eriksson, J.: "Inferring Road Maps from Global Positioning System Traces". In: *Transportation Research Record: Journal of the Transportation Research Board* 2291 (Dec. 2012). doi: 10.3141/2291-08 (cit. on p. 9).

[Bec21] Beck, J.: “Camera Calibration with Non-Central Local Camera Models”. PhD Thesis. Karlsruhe, Germany: Karlsruher Institut für Technologie (KIT), 2021. doi: 10.5445/IR/1000131090 (cit. on pp. 21, 32, 51).

[BHL23] Bao, Z.; Hossain, S.; Lang, H. and Lin, X.: “A review of high-definition map creation methods for autonomous driving”. In: *Engineering Applications of Artificial Intelligence* 122 (June 2023), p. 106125. doi: 10.1016/j.engappai.2023.106125 (cit. on p. 4).

[BHS23] Bieder, F.; Hu, H.; Schantz, J.; Kirik, O.; Ries, F.; Haueis, M. and Stiller, C.: “Ein Ansatz zur Automatisierten Erstellung von Trainingsdaten unter Verwendung von HD Karten und Mehrfachbefahrungen”. In: *15. Workshop Fahrerassistenz Und Automatisiertes Fahren (FAS)* (Berkheim, Germany). (Best Paper Award). 2023 (cit. on pp. 34, 57, 106).

[Bis06] Bishop, C.: Pattern Recognition and Machine Learning. Springer International Publishing, Jan. 2006 (cit. on p. 19).

[BKH26] Bieder, F.; Königshof, H.; Hu, H.; Immel, F.; Shen, Y.; Pauls, J.-H. and Stiller, C.: “XD-MAP: Cross-Modal Domain Adaptation via Semantic Parametric Maps for Scalable Training Data Generation”. In: *2026 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Denver (CO), United States). 2026 (cit. on p. 93).

[BKM20] Bock, J.; Krajewski, R.; Moers, T.; Runde, S.; Vater, L. and Eckstein, L.: “The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections”. In: *2020 IEEE Intelligent Vehicles Symposium (IV)* (Las Vegas (NV), United States). 2020, pp. 1929–1934. doi: 10.1109/IV47402.2020.9304839 (cit. on pp. 9, 12).

[BLR21] Bieder, F.; Link, M.; Romanski, S.; Hu, H. and Stiller, C.: “Improving Lidar-Based Semantic Segmentation of Top-View

Grid Maps by Learning Features in Complementary Representations". In: *2021 IEEE International Conference on Information Fusion (FUSION)* (Sun City, South Africa). 2021. doi: 10.23919/FUSION49465.2021.9627069 (cit. on pp. 15, 87).

[BMF25] Blumberg, A.; Merkert, J.; Fehler, R.; Immel, F.; Bieder, F.; Pauls, J.-H. and Stiller, C.: "Impact of Localization Errors on Label Quality for Online HD Map Construction". In: *2025 IEEE Intelligent Vehicles Symposium (IV)* (Cluj, Romania). June 2025, pp. 1833–1840. doi: 10.1109/IV64158.2025.11097513 (cit. on pp. 78, 85, 164).

[BPF16] Beyerer, J.; Puente León, F. and Frese, C.: Automatische Sichtprüfung-Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung. Springer-Verlag, 2016. doi: 10.1007/978-3-662-47786-1 (cit. on p. 19).

[Bra00] Bradski, G.: "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000) (cit. on p. 35).

[BS18] Beck, J. and Stiller, C.: "Generalized B-spline Camera Model". In: *2018 IEEE Intelligent Vehicles Symposium (IV)* (Changshu, China). June 2018, pp. 2137–2142. doi: 10.1109/IVS.2018.8500466 (cit. on p. 32).

[BTB77] Barrow, H. G.; Tenenbaum, J. M.; Bolles, R. C. and Wolf, H. C.: "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching". In: *International Joint Conference on Artificial Intelligence (IJCAI)* (Cambridge (MA), USA). 1977 (cit. on p. 23).

[BTV06] Bay, H.; Tuytelaars, T. and Van Gool, L.: "SURF: Speeded Up Robust Features". In: *2006 European Conference on Computer Vision (ECCV)* (Graz, Austria). Ed. by Leonardis, A.; Bischof, H. and Pinz, A. Springer International Publishing, 2006, pp. 404–417. doi: 10.1007/11744023_32 (cit. on p. 17).

[Bun13] Bundesrepublik Deutschland: Straßenverkehrs-Ordnung (StVO). In der Fassung der Bekanntmachung vom 6. März 2013 (BGBl. I S. 367), zuletzt geändert durch Artikel 1 der Verordnung vom 20. März 2024 (BGBl. I Nr. 93). 2013. url: https://www.gesetze-im-internet.de/stvo_2013/ (cit. on p. 58).

[BWJ20] Bieder, F.; Wirges, S.; Janosovits, J.; Richter, S.; Wang, Z. and Stiller, C.: "Exploiting Multi-Layer Grid Maps for Surround-View Semantic Segmentation of Sparse LiDAR Data". In: *2020 IEEE Intelligent Vehicles Symposium (IV)* (Virtual). Oct. 2020, pp. 1892–1898. doi: 10.1109/IV47402.2020.9304848 (cit. on pp. 15, 86).

[BZS14] Bender, P.; Ziegler, J. and Stiller, C.: "Lanelets: Efficient map representation for autonomous driving". In: *2014 IEEE Intelligent Vehicles Symposium (IV)* (Dearborn (MI), United States). 2014, pp. 420–425. doi: 10.1109/IVS.2014.6856487 (cit. on p. 9).

[CBL20] Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G. and Beijbom, O.: " nuScenes: A Multimodal Dataset for Autonomous Driving ". In: (June 2020), pp. 11618–11628. doi: 10.1109/CVPR42600.2020.01164 (cit. on pp. 10, 11, 13, 37, 69).

[CCR23] Cardace, A.; Conti, A.; Ramirez, P. Z.; Spezialetti, R.; Salti, S. and Di Stefano, L.: "Boosting Multi-Modal Unsupervised Domain Adaptation for LiDAR Semantic Segmentation by Self-Supervised Depth Completion". In: *IEEE Access* 11 (Aug. 2023) (cit. on p. 91).

[CCW22] Chen, S.; Cheng, T.; Wang, X.; Meng, W.; Zhang, Q. and Liu, W.: "Efficient and Robust 2D-to-BEV Representation Learning via Geometry-guided Kernel Transformer". In: *arXiv preprint arXiv:2206.04584* (June 2022). doi: 10.48550/arXiv.2206.04584 (cit. on pp. 17, 32).

[CDF23] Chen, J.; Deng, R. and Furukawa, Y.: "Polydiffuse: Polygonal shape reconstruction via guided set diffusion models". In: *2023 Advances in Neural Information Processing Systems (NeurIPS)*

(New Orleans (LA), United States). Vol. 36. 2023, pp. 1863–1888 (cit. on p. 16).

[CDP06] Caron, F.; Duflos, E.; Pomorski, D. and Vanheeghe, P.: "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects". In: *Information Fusion* 7.2 (June 2006), pp. 221–230. doi: 10.1016/j.inffus.2004.07.002 (cit. on p. 31).

[Ced04] Cederberg, J.: A Course in Modern Geometries. Undergraduate Texts in Mathematics. Springer International Publishing, 2004 (cit. on p. 19).

[CER21] Campos, C.; Elvira, R.; Rodríguez, J. J. G.; M. Montiel, J. M. and D. Tardós, J.: "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM". In: *IEEE Transactions on Robotics (T-RO)* 37.6 (2021), pp. 1874–1890. doi: 10.1109/TRO.2021.3075644 (cit. on p. 17).

[CLS19] Chang, M.-F.; Lambert, J. W.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D. et al.: "Argoverse: 3D Tracking and Forecasting with Rich Maps". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach (CA), USA). 2019 (cit. on p. 13).

[CMS20] Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A. and Zagoruyko, S.: "End-to-End Object Detection with Transformers". In: *2020 European Conference for Computer Vision (ECCV)* (Edinburgh, United Kingdom). Springer International Publishing, 2020 (cit. on pp. 16, 72).

[CMS22] Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A. and Girdhar, R.: "Masked-attention Mask Transformer for Universal Image Segmentation". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (New Orleans, United States). 2022, pp. 1280–1289. doi: 10.1109/CVPR52688.2022.00135 (cit. on pp. 15, 36, 37, 97, 108).

[COR16] Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S. and Schiele, B.: “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas (NV), United States). 2016, pp. 3213–3223 (cit. on pp. 13, 37, 160).

[CPA23] Chalvatzaras, A.; Pratikakis, I. and Amanatiadis, A. A.: “A Survey on Map-Based Localization Techniques for Autonomous Vehicles”. In: *IEEE Transactions on Intelligent Vehicles* 8.2 (2023) (cit. on p. 17).

[Cru24] Cruise: Technology - Cruise. 2024. url: `getcruise . com / technology` (last retrieved 2025-01-27) (cit. on p. 1).

[CSK21] Cheng, B.; Schwing, A. and Kirillov, A.: “Per-pixel classification is not all you need for semantic segmentation”. In: *2021 Advances in Neural Information Processing Systems (NeurIPS)* (Virtual). Vol. 34. 2021, pp. 17864–17875 (cit. on p. 15).

[CV18] Cai, Z. and Vasconcelos, N.: “Cascade r-cnn: Delving into high quality object detection”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, United States). 2018, pp. 6154–6162 (cit. on p. 15).

[CWT24] Chen, J.; Wu, Y.; Tan, J.; Ma, H. and Furukawa, Y.: “MapTracker: Tracking with Strided Memory Fusion for Consistent Vector HD Mapping”. In: *arXiv preprint arXiv:2403.15951* (Oct. 2024). doi: `10.48550/arXiv.2403.15951` (cit. on pp. 10, 16, 77).

[DDG17] Dabeer, O.; Ding, W.; Gowaiker, R.; Grzechnik, S. K.; Lakshman, M. J.; Lee, S.; Reitmayr, G.; Sharma, A.; Somasundaram, K.; Sukhavasi, R. T. et al.: “An end-to-end system for crowdsourced 3D maps for autonomous vehicles: The mapping component”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, Canada). Sept. 2017, pp. 634–641. doi: `10.1109/IROS.2017.8202218` (cit. on p. 9).

[DDS09] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K. and Fei-Fei, L.: "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Miami (FL), USA). 2009, pp. 248–255 (cit. on pp. 73, 97).

[DEH23] Deichman, J.; Ebel, E.; Heineke Kersten und Heuss, R.; Kellner, M. and Steiner, F.: McKinsey & Company: Autonomous driving's future: Convenient and connected. 2023. url: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected (last retrieved 2025-01-16) (cit. on p. 1).

[DRC17] Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A. and Koltun, V.: "CARLA: An Open Urban Driving Simulator". In: *arXiv preprint arXiv:1711.03938* (Nov. 2017). doi: 10.48550/arXiv.1711.03938 (cit. on p. 14).

[EE14] Einecke, N. and Eggert, J.: "Block-matching stereo with relaxed fronto-parallel assumption". In: *2014 IEEE Intelligent Vehicles Symposium (IV)* (Dearborn (MI), USA). June 2014, pp. 700–705. doi: 10.1109/IVS.2014.6856414 (cit. on p. 36).

[EE15] Einecke, N. and Eggert, J.: "A multi-block-matching approach for stereo". In: *2015 IEEE Intelligent Vehicles Symposium (IV)* (Seoul, South Korea). June 2015, pp. 585–592. doi: 10.1109/IVS.2015.7225748 (cit. on p. 36).

[EEG14] Everingham, M.; Eslami, S. M. A.; Gool, L. V.; Williams, C. K. I.; Winn, J. M. and Zisserman, A.: "The Pascal Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision* 111 (June 2014), pp. 98–136 (cit. on p. 24).

[EFH23] Elghazaly, G.; Frank, R.; Harvey, S. and Safko, S.: "High-Definition Maps: Comprehensive Survey, Challenges, and Future Perspectives". In: *IEEE Open Journal of Intelligent Transportation Systems* 4 (July 2023), pp. 527–550. doi: 10.1109/OJITS.2023.3295502 (cit. on p. 158).

[EKC18] Engel, J.; Koltun, V. and Cremers, D.: “Direct Sparse Odometry”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40.3 (2018). doi: 10.1109/TPAMI.2017.2658577 (cit. on p. 18).

[ESC14] Engel, J.; Schöps, T. and Cremers, D.: “LSD-SLAM: Large-Scale Direct Monocular SLAM”. In: *2014 European Conference on Computer Vision (ECCV)* (Zurich, Switzerland). Springer International Publishing, 2014. doi: 10.1007/978-3-319-10605-2_54 (cit. on p. 18).

[Esr23] Esri: World Imagery. © Esri, DigitalGlobe, GeoEye, i-cubed, USDA FSA, USGS, AEX, Getmapping, Aerogrid, IGN, IGP, swisstopo, und the GIS User Community. Accessed 15. August 2023 via JOSM with switch:services,server. 2023. url: arcgisonline.com/arcgis/rest/services/World_Imagery/MapServer/tile (cit. on pp. 34, 49).

[Fab21] Fabulos: Robot bus fleets have been successfully tested in 5 European cities - Fabulos. 2021. url: fabulos.eu (last retrieved 2025-01-27) (cit. on p. 1).

[Fen22] Fenske, H.: “Recursive fusion of sequential LiDAR measurements considering dynamic occlusions for the creation of grid maps in the context of autonomous driving”. Master thesis. Karlsruhe, Germany: Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, 2022 (cit. on p. 50).

[FLI17] FLIR Integrated Imaging Solutions Inc: Flir Flea3 GigE Vision. 2017. url: eureca.de/files/pdf/optoelectronics/flir/Datasheet-Flea3-gige.pdf (last retrieved 2025-01-21) (cit. on p. 29).

[FLI20] FLIR Integrated Imaging Solutions Inc: Blackfly S BFS-U3-88S6M Datasheet. 2020. url: softwareservices.flir.com/BFS-U3-88S6/latest/Model/spec.html (last retrieved 2025-01-21) (cit. on p. 29).

[FPH21] Fei, J.; Peng, K.; Heidenreich, P.; Bieder, F. and Stiller, C.: "PillarSegNet: Pillar-based Semantic Grid Map Estimation using Sparse LiDAR Data". In: *2021 IEEE Intelligent Vehicles Symposium (IV)* (Nagoya, Japan). 2021, pp. 838–844. doi: 10.1109/IV48863.2021.9575694 (cit. on p. 87).

[FPZ22] Fan, L.; Pang, Z.; Zhang, T.; Wang, Y.-X.; Zhao, H.; Wang, F.; Wang, N. and Zhang, Z.: "Embracing single stride 3d object detector with sparse transformer". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (New Orleans, United States). 2022, pp. 8458–8468 (cit. on p. 16).

[FSG17] Fan, H.; Su, H. and Guibas, L.: "A Point Set Generation Network for 3D Object Reconstruction from a Single Image". In: *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu (HI), USA). Honolulu, HI, July 2017, pp. 2463–2471. doi: 10.1109/CVPR.2017.264 (cit. on p. 23).

[FTV00] Fusiello, A.; Trucco, E. and Verri, A.: "A compact algorithm for rectification of stereo pairs". In: *Machine Vision and Applications* 12.1 (July 2000), pp. 16–22. doi: 10.1007/s001380050120 (cit. on p. 35).

[FU11] Fairfield, N. and Urmson, C.: "Traffic light mapping and detection". In: *2011 IEEE International Conference on Robotics and Automation (ICRA)* (Shanghai, China). 2011, pp. 5421–5426. doi: 10.1109/ICRA.2011.5980164 (cit. on p. 10).

[GBC16] Goodfellow, I.; Bengio, Y. and Courville, A.: Deep Learning. MIT Press, 2016. url: http://www.deeplearningbook.org (cit. on p. 19).

[GLU12] Geiger, A.; Lenz, P. and Urtasun, R.: "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *2012 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Providence (RI), USA). June 2012, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074 (cit. on pp. 20, 35).

[HB22] He, S. and Balakrishnan, H.: “Lane-Level Street Map Extraction from Aerial Imagery”. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Waikoloa, HI, USA). Jan. 2022, pp. 1496–1505. doi: 10.1109/WACV51458.2022.00156 (cit. on p. 10).

[HBL20] Hu, H.; Beck, J.; Lauer, M. and Stiller, C.: “Continuous Fusion of IMU and Pose Data using Uniform B-Spline”. In: *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (Virtual). 2020, pp. 173–178. doi: 10.1109/MFI49285.2020.9235248 (cit. on p. 42).

[HER] HERE Technologies: HERE HD Live Map Technical Paper: A self-healing map for reliable autonomous driving. url: https://go.engage.here.com/self-healing.html (last retrieved 2025-01-21) (cit. on pp. 8, 158).

[HGD17] He, K.; Gkioxari, G.; Dollár, P. and Girshick, R.: “Mask r-cnn”. In: *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, United States). 2017, pp. 2961–2969 (cit. on p. 15).

[HHB22] Hu, H.; Han, F.; Bieder, F.; Pauls, J.-H. and Stiller, C.: “TEScalib: Targetless Extrinsic Self-Calibration of LiDAR and Stereo Camera for Automated Driving Vehicles with Uncertainty Analysis”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Kyoto, Japan). 2022, pp. 6256–6263. doi: 10.1109/IROS47612.2022.9981651 (cit. on p. 33).

[HSH14] Haubrich, T.; Seele, S.; Herpers, R.; Müller, M. E. and Becker, P.: “A Semantic Road Network Model for traffic simulations in virtual environments: Generation and integration”. In: *2014 IEEE Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)* (Minneapolis (MN), United States). 2014, pp. 43–50. doi: 10.1109/SEARIS.2014.7152800 (cit. on p. 9).

[Hu26] Hu, H.: “Multi-Modal Continuous Time SLAM and Semantic Parametric Mapping for Autonomous Driving”. PhD Thesis. Karlsruhe, Germany: Karlsruher Institut für Technologie (KIT), 2026 (cit. on pp. 42, 49, 51, 62, 63, 117, 160).

[HWS22] Hou, J. J.; Wu, X.; Shan, J.; Li, D. and Wang, H.: “Robust optimization-based fusion of GNSS and Visual-Inertial-Wheel Odometry”. In: *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Xishuangbanna, China). Dec. 2022, pp. 1–6. doi: 10.1109/ROBIO55434.2022.10011839 (cit. on p. 31).

[HYW23] Hu, H.; Yang, H.; Wu, J.; Lei, X.; Bieder, F.; Pauls, J.-H. and Stiller, C.: “Large-Scale 3D Semantic Reconstruction for Automated Driving Vehicles with Adaptive Truncated Signed Distance Function”. In: *2023 IEEE Intelligent Vehicles Symposium (IV)* (Anchorage (AK), United States). June 2023. doi: 10.1109/IV55152.2023.10186691 (cit. on p. 55).

[HZG20] Hou, Y.; Zheng, L. and Gould, S.: “Multiview Detection with Feature Perspective Transformation”. In: *2020 European Conference on Computer Vision (ECCV)* (Glasgow, United Kingdom). Ed. by Vedaldi, A.; Bischof, H.; Brox, T. and Frahm, J.-M. Springer International Publishing, 2020, pp. 1–18 (cit. on p. 72).

[HZR16] He, K.; Zhang, X.; Ren, S. and Sun, J.: “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, United States). Las Vegas, NV, USA, June 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90 (cit. on pp. 72, 97).

[Ian18] Ian Endres, H.: Making and Maintaining Maps Accurate to the Centimeter: Automation Demands It. 2018. url: here.com/learn/blog/making-and-maintaining-maps-accurate-to-the-centimeter-automation-demands-it (last retrieved 2025-01-21) (cit. on p. 9).

[IBM22] IBM: What is a digital twin? 2022. url: ibm.com/topics/what-is-a-digital-twin (last retrieved 2025-01-21) (cit. on p. 7).

[IFB25a] Immel, F.; Fehler, R.; Bieder, F.; Pauls, J.-H. and Stiller, C.: “M3TR: Generalist HD Map Construction with Variable Map Priors”. In: *IEEE Robotics and Automation Letters (RAL)* 10.12 (2025) (cit. on pp. 16, 73, 77, 84, 87).

[IFB25b] Immel, F.; Fehler, R.; Bieder, F. and Stiller, C.: “Generation of Training Data from HD Maps in the Lanelet2 Framework”. In: *16. Workshop Fahrerassistenz Und Automatisiertes Fahren (FAS)* (Kaufbeuren, Germany). 2025 (cit. on pp. 8, 48, 73, 75).

[IFG23] Immel, F.; Fehler, R.; Ghanaat, M. M.; Ries, F.; Haueis, M. and Stiller, C.: “HD Map Generation from Noisy Multi-Route Vehicle Fleet Data on Highways with Expectation Maximization”. In: *2023 IEEE Intelligent Vehicles Symposium (IV)* (Anchorage (AK), United States). June 2023, pp. 1–7. doi: 10.1109/IV55152.2023.10186773 (cit. on p. 9).

[Int11] International Organization for Standardization: ISO 8855:2011 Road Vehicles - Vehicle Dynamics and Road-Holding Ability. Geneva, Switzerland, 2011 (cit. on p. 20).

[IPF25] Immel, F.; Pauls, J.-H.; Fehler, R.; Bieder, F.; Merkert, J. and Stiller, C.: “SDTagNet: Leveraging Text-Annotated Navigation Maps for Online HD Map Construction”. In: *2025 Advances in Neural Information Processing Systems (NeurIPS)* (San Diego (CA), United States). Vol. 39. 2025 (cit. on pp. 16, 87).

[JGB20] Janai, J.; Güney, F.; Behl, A. and Geiger, A.: “Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art”. In: *Found. Trends. Comput. Graph. Vis.* 12.1-3 (July 2020), pp. 1–308. doi: 10.1561/0600000079 (cit. on p. 2).

[JLC23] Jain, J.; Li, J.; Chiu, M. T.; Hassani, A.; Orlov, N. and Shi, H.: “OneFormer: One Transformer To Rule Universal Image Segmentation”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Vancover, Canada). 2023, pp. 2989–2998 (cit. on p. 15).

[JOSM] The Openstreetmap Project: JOSM. url: https://josm.openstreetmap.de (last retrieved 2023-03-24) (cit. on p. 49).

[JZL24] Jiang, Z.; Zhu, Z.; Li, P.; Gao, H.-a.; Yuan, T.; Shi, Y.; Zhao, H. and Zhao, H.: "P-MapNet: Far-Seeing Map Generator Enhanced by Both SDMap and HDMap Priors". In: *IEEE Robotics and Automation Letters (RAL)* 9.10 (2024), pp. 8539–8546. doi: 10.1109/LRA.2024.3447450 (cit. on pp. 16, 79).

[Kar20] Karpathy, A.: Tesla Autopilot, Neural networks in production and Neural Networks for FSD. Workshop on Scalability in Autonomous Driving, CVPR 2020. 2020. url: https://www.youtube.com/watch?v=g2R2T631x7k (cit. on p. 3).

[KGH19] Kirillov, A.; Girshick, R.; He, K. and Dollár, P.: "Panoptic feature pyramid networks". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, United States). 2019, pp. 6399–6408 (cit. on p. 15).

[KHG19] Kirillov, A.; He, K.; Girshick, R.; Rother, C. and Dollar, P.: "Panoptic Segmentation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA). June 2019, pp. 9396–9405. doi: 10.1109/CVPR.2019.00963 (cit. on pp. 15, 25).

[Kir20] Kirik, O.: "Occlusion Handling for Automatic Data Generation using HD Maps and a highly accurate SLAM". Master thesis. Karlsruhe, Germany: Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, 2020 (cit. on p. 106).

[KK20] Kümmerle, J. and Kühner, T.: "Unified Intrinsic and Extrinsic Camera and LiDAR Calibration under Uncertainties". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Virtual). May 2020, pp. 6028–6034. doi: 10.1109/ICRA40945.2020.9197496 (cit. on p. 33).

[KKL18] Kümmerle, J.; Kühner, T. and Lauer, M.: "Automatic Calibration of Multiple Cameras and Depth Sensors with a Spherical Target". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau, China). Oct. 2018, pp. 1–8. doi: 10.1109/IROS.2018.8593955 (cit. on p. 33).

[KL21] Krontschieder, P. and Lorenzo, P.: Upgrading to Vistas 2.0. 2021. url: blog.mapillary.com/update/2021/01/18/vistas-2-dataset.html (last retrieved 2025-01-21) (cit. on pp. 36, 37, 159).

[KLC23] Kong, L.; Liu, Y.; Chen, R.; Ma, Y.; Zhu, X.; Li, Y.; Hou, Y.; Qiao, Y. and Liu, Z.: "Rethinking Range View Representation for LiDAR Segmentation". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (Paris, France). Oct. 2023, pp. 228–240. doi: 10.1109/iccv51070.2023.00028 (cit. on p. 15).

[KMB20] Krajewski, R.; Moers, T.; Bock, J.; Vater, L. and Eckstein, L.: "The rounD Dataset: A Drone Dataset of Road User Trajectories at Roundabouts in Germany". In: *2020 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (Rhodes, Greece). 2020, pp. 1–6. doi: 10.1109/ITSC45102.2020.9294728 (cit. on pp. 9, 12).

[KNH22] Khan, S.; Naseer, M.; Hayat, M.; Zamir, S. W.; Khan, F. S. and Shah, M.: "Transformers in vision: A survey". In: *ACM Computing Surveys (CSUR)* 54.10s (2022), pp. 1–41 (cit. on p. 15).

[KSP19] Kümmerle, J.; Sons, M.; Poggenhans, F.; Kühner, T.; Lauer, M. and Stiller, C.: "Accurate and Efficient Self-Localization on Roads using Basic Geometric Primitives". In: *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, Canada). May 2019, pp. 5965–5971. doi: 10.1109/ICRA.2019.8793497 (cit. on pp. 4, 10).

[Küm20] Kümmerle, J. V.: "Multimodal Sensor Calibration with a Spherical Calibration Target". PhD Thesis. Karlsruhe, Germany: Karlsruher Institut für Technologie (KIT), 2020. 185 pp. doi: 10.5445/IR/1000124721 (cit. on p. 33).

[KWL19] Kühner, T.; Wirges, S. and Lauer, M.: "Automatic Generation of Training Data for Image Classification of Road Scenes". In: *2019*

IEEE International Conference on Intelligent Transportation Systems(ITSC) (Auckland, New Zealand). Oct. 2019, pp. 1097–1103. doi: 10.1109/ITSC.2019.8917089 (cit. on p. 33).

[LAB11] Levinson, J.; Askeland, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Kolter, J. Z.; Langer, D.; Pink, O.; Pratt, V. et al.: "Towards fully autonomous driving: Systems and algorithms". In: *2011 IEEE Intelligent Vehicles Symposium (IV)* (Baden-Baden, Germany). June 2011, pp. 163–168. doi: 10.1109/IVS.2011.5940562 (cit. on pp. 8, 10, 158).

[LBH15] LeCun, Y.; Bengio, Y. and Hinton, G.: "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. doi: 10.1038/nature14539 (cit. on p. 19).

[LBK13] Lategahn, H.; Beck, J.; Kitt, B. and Stiller, C.: "How to learn an illumination robust image feature for place recognition". In: *2013 IEEE Intelligent Vehicles Symposium (IV)* (Gold Coast, Australia). June 2013, pp. 285–291. doi: 10.1109/IVS.2013.6629483 (cit. on p. 42).

[LBS14] Lategahn, H.; Beck, J. and Stiller, C.: "DIRD is an illumination robust descriptor". In: *2014 IEEE Intelligent Vehicles Symposium (IV)* (Dearborn (MI), United States). June 2014, pp. 756–761. doi: 10.1109/IVS.2014.6856421 (cit. on pp. 41, 42).

[LCJ24] Liao, B.; Chen, S.; Jiang, B.; Cheng, T.; Zhang, Q.; Liu, W.; Huang, C. and Wang, X.: "Lane Graph as Path: Continuity-Preserving Path-Wise Modeling for Online Lane Graph Construction". In: *2024 European Conference on Computer Vision (ECCV)* (Milan, Italy). Ed. by Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T. and Varol, G. Vol. 15102. Cham: Springer International Publishing, 2024, pp. 334–351. doi: 10.1007/978-3-031-72784-9_19 (cit. on p. 69).

[LCW22] Liao, B.; Chen, S.; Wang, X.; Cheng, T.; Zhang, Q.; Liu, W. and Huang, C.: "MapTR: Structured Modeling and Learning for Online Vectorized HD Map Construction". In: *2022 International*

Conference on Learning Representations (ICLR) (Virual). Sept. 2022 (cit. on pp. 10, 16, 32, 57, 72, 77, 79).

[LCW23] Li, T.; Chen, L.; Wang, H.; Li, Y.; Yang, J.; Geng, X.; Jiang, S.; Wang, Y.; Xu, H.; Xu, C. et al.: "Graph-based Topology Reasoning for Driving Scenes". In: *arXiv preprint arXiv:2304.05277* (Aug. 2023). doi: 10.48550/arXiv.2304.05277 (cit. on p. 4).

[LCZ24] Liao, B.; Chen, S.; Zhang, Y.; Jiang, B.; Zhang, Q.; Liu, W.; Huang, C. and Wang, X.: "MapTRv2: An End-to-End Framework for Online Vectorized HD Map Construction". In: *International Journal of Computer Vision* 133 (Oct. 2024). doi: 10.1007/s11263-024-02235-z (cit. on pp. 10, 16, 72, 77, 79).

[Len20] Lensation GmbH: Lensation BM4018S118 Datasheet. 2020. url: lensation.de/pdf/BM4018S118.pdf (last retrieved 2025-01-21) (cit. on p. 29).

[LFS24] Lilja, A.; Fu, J.; Stenborg, E. and Hammarstrand, L.: "Localization is All You Evaluate: Data Leakage in Online Mapping Datasets and How to Fix it". In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, USA). June 2024, pp. 22150–22159. doi: 10.1109/CVPR52733.2024.02091 (cit. on pp. 12, 76).

[LGK19] Ling, H.; Gao, J.; Kar, A.; Chen, W. and Fidler, S.: "Fast Interactive Object Annotation With Curve-GCN". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach (CA), USA). IEEE, June 2019, pp. 5252–5261. doi: 10.1109/CVPR.2019.00540 (cit. on p. 14).

[LH19] Loshchilov, I. and Hutter, F.: "Decoupled Weight Decay Regularization". In: *arXiv preprint arXiv:1711.05101v3* (Jan. 2019). doi: 10.48550/arXiv.1711.05101v3 (cit. on pp. 79, 97, 99).

[LH21] Lambert, J. W. and Hays, J.: "Trust, but Verify: Cross-Modality Fusion for HD Map Change Detection". In: *2021 Advances in Neural Information Processing Systems (NeurIPS)* (Virtual). 2021 (cit. on p. 4).

[LLC21] Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S. and Guo, B.: “Swin transformer: Hierarchical vision transformer using shifted windows”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (Virtual). 2021, pp. 10012–10022 (cit. on p. 72).

[LMH20] Langer, F.; Milioto, A.; Haag, A.; Behley, J. and Stachniss, C.: “Domain transfer for semantic segmentation of LiDAR data using deep neural networks”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Las Vegas (NV), USA). 2020, pp. 8263–8270 (cit. on p. 91).

[Low04] Lowe, D. G.: “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision (IJCV)* 60.2 (Nov. 2004), pp. 91–110. doi: 10.1023/B:VISI.0000029664.99615.94 (cit. on p. 17).

[LS14] Lategahn, H. and Stiller, C.: “Vision-Only Localization”. In: *IEEE Transactions on Intelligent Transportation Systems (T-ITS)* 15.3 (June 2014), pp. 1246–1257. doi: 10.1109/TITS.2014.2298492 (cit. on pp. 17, 19, 41, 42).

[LSD15] Long, J.; Shelhamer, E. and Darrell, T.: “Fully convolutional networks for semantic segmentation”. In: *2015 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, United States). 2015, pp. 3431–3440 (cit. on p. 15).

[LWL22] Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y. and Dai, J.: “BEVFormer: Learning Bird’s-Eye-View Representation from Multi-camera Images via Spatiotemporal Transformers”. In: *2022 European Conference on Computer Vision (ECCV)* (Tel Aviv, Israel). Ed. by Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M. and Hassner, T. Vol. 13669. Cham: Springer International Publishing, 2022, pp. 1–18. doi: 10.1007/978-3-031-20077-9_1 (cit. on p. 72).

[LWW22] Li, Q.; Wang, Y.; Wang, Y. and Zhao, H.: “HDMapNet: An Online HD Map Construction and Evaluation Framework”. In: *2022 IEEE International Conference on Robotics and Automation*

(ICRA) (Philadelphia (PA), United States). May 2022, pp. 4628–4634. doi: 10.1109/ICRA46639.2022.9812383 (cit. on pp. 16, 57, 72, 73).

[LXG23] Liao, Y.; Xie, J. and Geiger, A.: "KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.3 (2023), pp. 3292–3310. doi: 10.1109/TPAMI.2022.3179507 (cit. on pp. 13, 37).

[Lyf18] Lyft Level 5: Rethinking Maps for Self-Driving. 2018. url: medium.com/wovenplanetlevel5/https-medium-com-lyftlevel5-rethinking-maps-for-self-driving-a147c24758d6 (last retrieved 2025-01-21) (cit. on pp. 7, 8, 13, 158).

[LYW23] Liu, Y.; Yuan, T.; Wang, Y.; Wang, Y. and Zhao, H.: "VectorMapNet: End-to-end Vectorized HD Map Learning". In: *2023 International Conference on Machine Learning (ICML)* (Honolulu, United States). Vol. 202. PMLR, July 2023, pp. 22352–22369 (cit. on pp. 16, 57, 72, 77, 79).

[Mei17] Meike: Meike Fisheye 6.5 mm F2. 2017. url: meike-shop.de/Fisheye-Objektiv-MK-65mm-F-20-fuer-Sony-E-Mount (last retrieved 2025-01-21) (cit. on p. 29).

[MFG22] Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C. and Woodall, W.: "Robot Operating System 2: Design, architecture, and uses in the wild". In: *Science Robotics* 7.66 (2022). doi: 10.1126/scirobotics.abm6074 (cit. on p. 9).

[MGW23] Man, Y.; Gui, L.-Y. and Wang, Y.-X.: "DualCross: Cross-Modality Cross-Domain Adaptation for Monocular BEV Perception". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Detroit (MI), USA). 2023 (cit. on p. 91).

[MM15] Matthaei, R. and Maurer, M.: "Autonomous driving - a top-down-approach". In: *at - Automatisierungstechnik* 63.3 (Mar. 2015), pp. 155–167. doi: 10.1515/auto-2014-1136 (cit. on p. 3).

[Mov22] Movella: Xsense MTi-300 Datasheet. 2022. url: mouser.de/datasheet/2/1484/MTi_300-3241580.pdf (last retrieved 2025-01-21) (cit. on p. 31).

[MSP21] Meyer, A.; Skudlik, P.; Pauls, J.-H. and Stiller, C.: "YOLinO: Generic Single Shot Polyline Detection in Real Time". In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (Virtual). Montreal, BC, Canada, Oct. 2021, pp. 2916–2925. doi: 10.1109/ICCVW54120.2021.00326 (cit. on p. 57).

[MT17] Mur-Artal, R. and Tardós, J. D.: "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Transactions on Robotics (T-RO)* 33.5 (Oct. 2017), pp. 1255–1262. doi: 10.1109/TRO.2017.2705103 (cit. on p. 17).

[MVB19] Milioto, A.; Vizzo, I.; Behley, J. and Stachniss, C.: "RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau, China). 2019, pp. 4213–4220. doi: 10.1109/IROS40897.2019.8967762 (cit. on pp. 15, 39).

[MWB24] Ma, Y.; Wang, T.; Bai, X.; Yang, H.; Hou, Y.; Wang, Y.; Qiao, Y.; Yang, R. and Zhu, X.: "Vision-Centric BEV Perception: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (Dec. 2024), pp. 10978–10997. doi: 10.1109/TPAMI.2024.3449912 (cit. on p. 72).

[MWF16] Mattyus, G.; Wang, S.; Fidler, S. and Urtasun, R.: "HD Maps: Fine-Grained Road Segmentation by Parsing Ground and Aerial Images". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas (NV), United States). Las Vegas, NV, USA, June 2016, pp. 3611–3619. doi: 10.1109/CVPR.2016.393 (cit. on p. 10).

[MXN21] Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H. and Xu, C.: "Voxel transformer for 3d object detection". In: *2021*

IEEE/CVF International Conference on Computer Vision (ICCV) (Virtual). 2021, pp. 3164–3173 (cit. on p. 16).

[MYZ23] Ma, T.; Yang, X.; Zhou, H.; Li, X.; Shi, B.; Liu, J.; Yang, Y.; Liu, Z.; He, L.; Qiao, Y. et al.: "DetZero: Rethinking Offboard 3D Object Detection with Long-term Sequential Point Clouds". In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (Paris, France). Oct. 2023 (cit. on p. 118).

[NHG23] Naumann, A.; Hertlein, F.; Grimm, D.; Zipfl, M.; Thoma, S.; Rettinger, A.; Halilaj, L.; Luettin, J.; Schmid, S. and Caesar, H.: "Lanelet2 for nuScenes: Enabling Spatial Semantic Relationships and Diverse Map-based Anchor Paths". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Vancouver, Canada). Vancouver, BC, Canada, June 2023, pp. 3248–3257. doi: 10.1109/CVPRW59228.2023.00327 (cit. on p. 11).

[NHT24] NHTSA's National Center for Statistics and Analysis: "Overview of Motor Vehicle Traffic Crashes in 2022". In: *Traffic Safety Facts - National Highway Traffic Safety Administration* (June 2024) (cit. on p. 1).

[NOB17] Neuhold, G.; Ollmann, T.; Bulo, S. R. and Kontschieder, P.: "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes". In: *2017 IEEE/CVF International Conference on Computer Vision (ICCV)* (Venice, Italy). Oct. 2017, pp. 5000–5009. doi: 10.1109/ICCV.2017.534 (cit. on pp. 37, 91).

[OFO23] Ochs, S.; Fleck, T.; Orf, S.; Schotschneider, A.; Gontscharow, M.; Polley, R.; Zofka, M. R.; Viehl, A.; Zöllner, J. M.; Simon, K. et al.: "TAF-BW - Real Laboratory as Enabler for Autonomous Driving". In: *2023 Mobility 4.0* (Dubai, United Arab Emirates). SAE International, 2023, pp. 2023-01–1909. doi: 10.4271/2023-01-1909 (cit. on p. 1).

[Ope] Open Source Geospatial Foundation: Tile Map Service Specification. url: wiki.osgeo.org/wiki/Tile_Map_Service_Specification (last retrieved 2025-01-21) (cit. on p. 49).

[Pau24] Pauls, J.-H.: “Continuous Verification and Safe Localization in Semantic High Definition Maps for Automated Driving”. PhD Thesis. Karlsruhe, Germany: Karlsruher Institut für Technologie (KIT), 2024. doi: 10.5445/IR/1000179521 (cit. on pp. 105, 117).

[PBC19] Porzi, L.; Bulo, S. R.; Colovic, A. and Kontschieder, P.: “Seamless Scene Segmentation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach (CA), USA). June 2019, pp. 8269–8278. doi: 10.1109/CVPR.2019.00847 (cit. on pp. 37, 63, 159).

[PCF23] Peng, L.; Chen, Z.; Fu, Z.; Liang, P. and Cheng, E.: “BEVSegFormer: Bird’s Eye View Semantic Segmentation From Arbitrary Camera Rigs”. In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Waikoloa (HI), United States). Jan. 2023, pp. 5924–5932 (cit. on p. 17).

[Pet18] Pete Goldin: 10 Advantages of Autonomous Vehicles | ITSdigest. Feb. 20, 2018. url: https://www.itsdigest.com/10-advantages-autonomous-vehicles (last retrieved 2025-01-16) (cit. on p. 1).

[PF10] Pfeiffer, D. and Franke, U.: “Efficient representation of traffic scenes by means of dynamic stixels”. In: *2010 IEEE Intelligent Vehicles Symposium (IV)* (San Diego (CA), United States). La Jolla, CA, USA, June 2010, pp. 217–224. doi: 10.1109/IVS.2010.5548114 (cit. on p. 50).

[PF20] Philion, J. and Fidler, S.: “Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D”. In: *2020 European Conference on Computer Vision (ECCV)* (Glasgow, United Kingdom). Ed. by Vedaldi, A.; Bischof, H.; Brox, T. and Frahm, J.-M. Springer International Publishing, 2020, pp. 194–210 (cit. on pp. 72, 73).

[PFY22] Peng, K.; Fei, J.; Yang, K.; Roitberg, A.; Zhang, J.; Bieder, F.; Heidenreich, P.; Stiller, C. and Stiefelhagen, R.: “MASS: Multi-Attentional Semantic Segmentation of LiDAR Data for Dense

Top-View Understanding". In: *IEEE Transactions on Intelligent Transportation Systems (T-ITS)* 23.9 (2022), pp. 15824–15840. doi: 10.1109/TITS.2022.3145588 (cit. on p. 87).

[PJ20] Poggenhans, F. and Janosovits, J.: "Pathfinding and Routing for Automated Driving in the Lanelet2 Map Framework". In: *2020 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (Rhodes, Greece). 2020. doi: 10.1109/ITSC45102.2020.9294376 (cit. on p. 9).

[PLH20] Pannen, D.; Liebner, M.; Hempel, W. and Burgard, W.: "How to Keep HD Maps for Automated Driving Up To Date". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Virtual). 2020, pp. 2288–2294. doi: 10.1109/ICRA40945.2020.9197419 (cit. on p. 4).

[Pog19] Poggenhans, F.: "Generierung hochdetaillierter Karten für das automatisierte Fahren". PhD Thesis. Karlsruhe, Germany: Karlsruher Institut für Technologie (KIT), 2019. doi: 10.5445/IR/1000100719 (cit. on pp. 57–60, 75).

[PPJ18] Poggenhans, F.; Pauls, J.-H.; Janosovits, J.; Orf, S.; Naumann, M.; Kuhnt, F. and Mayr, M.: "Lanelet2: A high-definition map framework for the future of automated driving". In: *2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (Maui (HI), USA). Nov. 2018, pp. 1672–1679. doi: 10.1109/ITSC.2018.8569929 (cit. on pp. 8, 9, 47, 73, 158).

[PS10] Pink, O. and Stiller, C.: "Automated map generation from aerial images for precise vehicle localization". In: *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (Funchal, Portugal). Sept. 2010, pp. 1517–1522. doi: 10.1109/ITSC.2010.5625276 (cit. on p. 10).

[PSH18] Pauls, J.-H.; Strauss, T.; Hasberg, C.; Lauer, M. and Stiller, C.: "Can We Trust Our Maps? An Evaluation of Road Changes and a Dataset for Map Validation". In: *2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (Maui (HI),

USA). Nov. 2018, pp. 2639–2644. doi: 10.1109/ITSC.2018.8569249 (cit. on pp. 3, 4).

[PSS15] Poggenhans, F.; Schreiber, M. and Stiller, C.: "A Universal Approach to Detect and Classify Road Surface Markings". In: *2015 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (Las Palmas, Spain). Sept. 2015, pp. 1915–1921. doi: 10.1109/ITSC.2015.310 (cit. on p. 49).

[PSS21] Pauls, J.-H.; Schmidt, B. and Stiller, C.: "Automatic Mapping of Tailored Landmark Representations for Automated Driving and Map Learning". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)* (Xi'an, China). 2021, pp. 6725–6731. doi: 10.1109/ICRA48506.2021.9561432 (cit. on pp. 10, 62, 63).

[QDQ23] Qiao, L.; Ding, W.; Qiu, X. and Zhang, C.: "End-to-end vectorized hd-map construction with piecewise bezier curve". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Vancouver, Canada). 2023, pp. 13218–13228 (cit. on p. 16).

[QSM17] Qi, C. R.; Su, H.; Mo, K. and Guibas, L. J.: "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, United States). 2017, pp. 652–660 (cit. on p. 87).

[QYS17] Qi, C. R.; Yi, L.; Su, H. and Guibas, L. J.: "PointNet++: deep hierarchical feature learning on point sets in a metric space". In: *2017 Advances in Neural Information Processing Systems (NeurIPS)* (Long Beach (CA),USA). 2017, pp. 5105–5114 (cit. on p. 16).

[QZN21] Qi, C. R.; Zhou, Y.; Najibi, M.; Sun, P.; Vo, K.; Deng, B. and Anguelov, D.: "Offboard 3D Object Detection from Point Cloud Sequences". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville (TN), USA). June 2021 (cit. on p. 118).

[RBW24] Richter, S.; Bieder, F.; Wirges, S. and Stiller, C.: “A Dual Evidential Top-View Representation to Model the Semantic Environment of Automated Vehicles”. In: *IEEE Transactions on Intelligent Vehicles (T-IV)* 9.1 (2024), pp. 2688–2700. doi: 10.1109/TIV.2023.3284400 (cit. on p. 86).

[RD12] Ranft, B. and Denninger, O.: “Run-time Adaptation to Heterogeneous Processing Units for Real-time Stereo Vision”. In: *2012 IEEE International Conference on High Performance Computing and Communication & International Conference on Embedded Software and Systems* (Liverpool, United Kingdom). Liverpool, United Kingdom, June 2012, pp. 1592–1599. doi: 10.1109/HPCC.2012.232 (cit. on p. 36).

[REG19] Rist, C. B.; Enzweiler, M. and Gavrila, D. M.: “Cross-Sensor Deep Domain Adaptation for LiDAR Detection and Segmentation”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)* (Paris, France). 2019, pp. 1535–1542. doi: 10.1109/IVS.2019.8814047 (cit. on p. 91).

[RFB15] Ronneberger, O.; Fischer, P. and Brox, T.: “U-net: Convolutional networks for biomedical image segmentation”. In: *2015 International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (Munich, Germany). Springer International Publishing, Oct. 2015, pp. 234–241 (cit. on p. 15).

[RKC18] Roddick, T.; Kendall, A. and Cipolla, R.: “Orthographic Feature Transform for Monocular 3D Object Detection”. In: *arXiv preprint arXiv:1811.08188* (Nov. 2018). doi: 10.48550/arXiv.1811.08188 (cit. on p. 72).

[RRK11] Rublee, E.; Rabaud, V.; Konolige, K. and Bradski, G.: “ORB: An efficient alternative to SIFT or SURF”. In: *2011 IEEE/CVF International Conference on Computer Vision (ICCV)* (Barcelona, Spain). Nov. 2011, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544 (cit. on p. 17).

[RS14] Ranft, B. and Strauß, T.: “Modeling arbitrarily oriented slanted planes for efficient stereo vision based on block matching”. In: *2014 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (Qingdao, China). Oct. 2014, pp. 1941–1947. doi: 10.1109/ITSC.2014.6957990 (cit. on pp. 35, 36).

[RSM16] Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D. and Lopez, A. M.: “The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes”. In: *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas (NV), United States). June 2016, pp. 3234–3243. doi: 10.1109/CVPR.2016.352. (Last retrieved 2025-01-14) (cit. on p. 14).

[RVR16] Richter, S. R.; Vineet, V.; Roth, S. and Koltun, V.: “Playing for Data: Ground Truth from Computer Games”. In: *2016 European Conference on Computer Vision (ECCV)* (Amsterdam, Netherlands). Ed. by Leibe, B.; Matas, J.; Sebe, N. and Welling, M. Cham: Springer International Publishing, 2016, pp. 102–118. doi: 10.1007/978-3-319-46475-6_7 (cit. on p. 14).

[SAE21] SAE On-Road Automated Driving Committee: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International. Apr. 1, 2021. url: https://www.sae.org/standards/content/j3016_202104/ (last retrieved 2025-01-16) (cit. on p. 1).

[SCC24] Shi, A.; Cai, Y.; Chen, X.; Pu, J.; Fu, Z. and Lu, H.: “GlobalMapNet: An Online Framework for Vectorized Global HD Map Construction”. In: *arXiv preprint arXiv:2409.10063* (Sept. 2024). doi: 10.48550/arXiv.2409.10063 (cit. on p. 16).

[Sch21] Schantz, J.: “Automated Data Generation with HD-Maps for Machine Learning in the Context of Automated Driving”. Master thesis. Karlsruhe, Germany: Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, 2021 (cit. on pp. 56, 57, 159).

[SDS17] Sefati, M.; Daum, M.; Sondermann, B.; Kreisköther, K. D. and Kampker, A.: “Improving vehicle localization using semantic and pole-like landmarks”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)* (Los Angeles, CA, United States). 2017, pp. 13–19. doi: 10.1109/IVS.2017.7995692 (cit. on p. 10).

[SGL21] Strudel, R.; Garcia, R.; Laptev, I. and Schmid, C.: “Segmenter: Transformer for semantic segmentation”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Virtual). 2021, pp. 7262–7272 (cit. on p. 15).

[SKH16] Schreiber, M.; Königshof, H.; Hellmund, A.-M. and Stiller, C.: “Vehicle localization with tightly coupled GNSS and visual odometry”. In: *2016 IEEE Intelligent Vehicles Symposium (IV)* (Gothenburg, Sweden). June 2016, pp. 858–863. doi: 10.1109/IVS.2016.7535488 (cit. on p. 31).

[SLK15] Sons, M.; Lategahn, H.; Keller, C. G. and Stiller, C.: “Multi trajectory pose adjustment for life-long mapping”. In: *2015 IEEE Intelligent Vehicles Symposium (IV)* (Seoul, South Korea). June 2015, pp. 901–906. doi: 10.1109/IVS.2015.7225799 (cit. on pp. 41, 42).

[SLK17] Sons, M.; Lauer, M.; Keller, C. G. and Stiller, C.: “Mapping and localization using surround view”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)* (Los Angeles (CA), United States). June 2017, pp. 1158–1163. doi: 10.1109/IVS.2017.7995869 (cit. on pp. 41, 42).

[SPV13] Soheilian, B.; Paparoditis, N. and Vallet, B.: “Detection and 3D reconstruction of traffic signs from multiple view color images”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 77 (Mar. 2013), pp. 1–20. doi: 10.1016/j.isprsjprs.2012.11.009 (cit. on p. 10).

[SRG11] Sturm, P.; Ramalingam, S.; Gasparini, S. and Barreto, J. IEEE Now Foundations and Trends, 2011. doi: 10.1561/0600000023 (cit. on p. 21).

[SS18] Sons, M. and Stiller, C.: “Efficient Multi-Drive Map Optimization towards Life-long Localization using Surround View”. In: *2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (Maui (HI), United States). Nov. 2018, pp. 2671–2677. doi: 10.1109/ITSC.2018.8570011 (cit. on pp. 41, 42).

[SW22] Schwarz, C. and Wang, Z.: “The Role of Digital Twins in Connected and Automated Vehicles”. In: *IEEE Intelligent Transportation Systems Magazine* 14.6 (2022), pp. 41–51. doi: 10.1109/MITS.2021.3129524 (cit. on p. 7).

[SWZ25] Shi, H.; Wang, S.; Zhang, J.; Yin, X.; Wang, G.; Zhu, J.; Yang, K. and Wang, K.: “Offboard Occupancy Refinement with Hybrid Propagation for Autonomous Driving”. In: *arXiv preprint arXiv:2403.08504* (Mar. 2025). doi: 10.48550/arXiv.2403.08504 (cit. on p. 118).

[SYL25] Sun, R.; Yang, L.; Lingrand, D. and Precioso, F.: “Mind the Map! Accounting for Existing Maps When Estimating Online HDMaps from Sensors”. In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Tucson (AZ), USA). 2025, pp. 1671–1681 (cit. on p. 16).

[SZB14] Strauß, T.; Ziegler, J. and Beck, J.: “Calibrating multiple cameras with non-overlapping views using coded checkerboard targets”. In: *2014 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (Qingdao, China). Oct. 2014, pp. 2623–2628. doi: 10.1109/ITSC.2014.6958110 (cit. on p. 32).

[TBF05] Thrun, S.; Burgard, W. and Fox, D.: Probabilistic Robotics. Cambridge, MA, USA: MIT Press, 2005 (cit. on pp. 17, 19).

[TDC23] Thisanke, H.; Deshan, C.; Chamith, K.; Seneviratne, S.; Vidanaarachchi, R. and Herath, D.: “Semantic segmentation using Vision Transformers: A survey”. In: *Engineering Applications of Artificial Intelligence* 126 (Nov. 2023), p. 106669 (cit. on p. 99).

[Tel17] Teledyne FLIR LLC: Flir Blackfly Gige Vision. 2017. url: flir.app.boxcn.net/s/iicqenjhtth41dt13951qh5toidx29ih/file/418608635277 (last retrieved 2025-01-21) (cit. on p. 29).

[Tom18] TomTom: HD Map with RoadDNA: High definition map with sensor-agnostic localization. 2018. url: download.tomtom.com/open/banners/HD_Map_with_RoadDNA_Product_Info_Sheet.pdf (last retrieved 2025-01-21) (cit. on pp. 8, 158).

[TSP18] Taş, Ö. Ş.; Salscheider, N. O.; Poggenhans, F.; Wirges, S.; Bandera, C.; Zofka, M. R.; Strauss, T.; Zöllner, J. M. and Stiller, C.: "Making Bertha Cooperate-Team AnnieWAY's Entry to the 2016 Grand Cooperative Driving Challenge". In: *IEEE Transactions on Intelligent Transportation Systems (T-ITS)* 19.4 (Apr. 2018), pp. 1262–1276. doi: 10.1109/TITS.2017.2749974 (cit. on pp. 28, 30).

[TuS23] TuSimple: TuSimple Benchmark. 2023. url: github.com/TuSimple/tusimple-benchmark (last retrieved 2023-08-31) (cit. on p. 13).

[U-B22] U-Blox: NEO-M8P Datasheet: u-blox M8 high precision GNSS modules. 2022. url: content.u-blox.com/sites/default/files/NEO-M8P_DataSheet_UBX-15016656.pdf (last retrieved 2025-01-21) (cit. on p. 31).

[Vel19] Velodyne Lidar, Inc.: Velodyne VLP-16 Datasheet. 2019. url: hexagondownloads.blob.core.windows.net/public/AutonomouStuff/wp-content/uploads/2019/05/Puck__Datasheet_whitelabel.pdf (last retrieved 2025-01-21) (cit. on p. 29).

[Vel20] Velodyne Lidar, Inc.: Velodyne Alpha Prime Datasheet. 2020. url: autonomoustuff.com/-/media/Images/Hexagon/Hexagon%20Core/autonomousstuff/pdf/velodyne-alpha-prime-datasheet.ashx (last retrieved 2025-01-21) (cit. on p. 29).

[VGM23] Vizzo, I.; Guadagnino, T.; Mersch, B.; Wiesmann, L.; Behley, J. and Stachniss, C.: "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way". In: *IEEE Robotics and Automation Letters (RA-L)* 8.2

(2023), pp. 1029–1036. doi: 10.1109/LRA.2023.3236571 (cit. on p. 38).

[VYF13] Vu, A.; Yang, Q.; Farrell, J. A. and Barth, M.: "Traffic sign detection, state estimation, and identification using onboard sensors". In: *2013 IEEE International Conference on Intelligent Transportation Systems (ITSC)* (The Hague, Netherlands). 2013, pp. 875–880. doi: 10.1109/ITSC.2013.6728342 (cit. on p. 10).

[Way24a] Waymo LLC: Next stop: Miami. 2024. url: waymo.com/blog/2024/12/next-stop-miami (last retrieved 2025-01-27) (cit. on p. 1).

[Way24b] Waymo LLC: Redefine how you move around San Francisco. 2024. url: waymo.com/waymo-one-san-francisco (last retrieved 2025-01-27) (cit. on p. 1).

[Way24c] Waymo LLC: Waymo Open Dataset – Motion Prediction Challenge. 2024. url: waymo.com/open/challenges/2024/motion-prediction/ (last retrieved 2025-01-20) (cit. on p. 10).

[WH18] Wu, Y. and He, K.: "Group Normalization". In: *2018 European Conference on Computer Vision (ECCV)* (Munich, Germany). Ed. by Ferrari, V.; Hebert, M.; Sminchisescu, C. and Weiss, Y. Springer International Publishing, Sept. 2018, pp. 3–19 (cit. on p. 97).

[WJY24] Wijaya, B.; Jiang, K.; Yang, M.; Wen, T.; Wang, Y.; Tang, X.; Fu, Z.; Zhou, T. and Yang, D.: "High Definition Map Mapping and Update: A General Overview and Future Directions". In: *arXiv preprint arXiv:2409.09726* (Sept. 2024). doi: 10.48550/arXiv.2409.09726 (cit. on p. 4).

[WLL23] Wang, H.; Li, T.; Li, Y.; Chen, L.; Sima, C.; Liu, Z.; Wang, B.; Jia, P.; Wang, Y.; Jiang, S. et al.: "OpenLane-V2: A Topology Reasoning Benchmark for Unified 3D HD Mapping". In: *2023 Advances in Neural Information Processing Systems (NeurIPS)* (New Orleans (LA), United States). Vol. 36. Curran Associates, Inc., 2023, pp. 18873–18884 (cit. on pp. 4, 12).

[WMB22] Wei, Y.; Mahnaz, F.; Bulan, O.; Mengistu, Y.; Mahesh, S. and Losh, M. A.: "Creating Semantic HD Maps From Aerial Imagery and Aggregated Vehicle Telemetry for Autonomous Vehicles". In: *IEEE Transactions on Intelligent Transportation Systems (T-TITS)* 23.9 (2022), pp. 15382–15395. doi: 10.1109/TITS.2022.3140423 (cit. on p. 10).

[WNS24] Wu, K.; Nian, S.; Shen, C.; Yang, C. and Li, Z.: "LGmap: Local-to-Global Mapping Network for Online Long-Range Vectorized HD Map Construction". In: *arXiv preprint arXiv:2406.13988* (June 2024). doi: 10.48550/arXiv.2406.13988 (cit. on p. 4).

[WQA21] Wilson, B.; Qi, W.; Agarwal, T.; Lambert, J.; Singh, J.; Khandelwal, S.; Pan, B.; Kumar, R.; Hartnett, A.; Pontes, J. K. et al.: "Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting". In: *2021 Advances in Neural Information Processing Systems (NeurIPS)* (Virtual). Dec. 2021 (cit. on pp. 10, 12, 13, 69).

[WRB21] Wirges, S.; Roesch, K.; Bieder, F. and Stiller, C.: "Fast and Robust Ground Surface Estimation from LiDAR Measurements using Uniform B-Splines". In: *2021 IEEE International Conference on Information Fusion (FUSION)* (Sun City, South Africa). 2021. doi: 10.23919/FUSION49465.2021.9626921 (cit. on p. 51).

[WZA21] Wang, H.; Zhu, Y.; Adam, H.; Yuille, A. and Chen, L.-C.: "Max-deeplab: End-to-end panoptic segmentation with mask transformers". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Virtual). 2021, pp. 5463–5474 (cit. on p. 15).

[XKS16] Xie, J.; Kiefel, M.; Sun, M.-T. and Geiger, A.: "Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer". In: *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas (NV), USA). June 2016, pp. 3688–3697. doi: 10.1109/CVPR.2016.401 (cit. on p. 13).

[XLY23] Xiong, X.; Liu, Y.; Yuan, T.; Wang, Y.; Wang, Y. and Hang, Z.: "Neural Map Prior for Autonomous Driving". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Vancouver, Canada). 2023 (cit. on p. 16).

[XWW20] Xu, C.; Wu, B.; Wang, Z.; Zhan, W.; Vajda, P.; Keutzer, K. and Tomizuka, M.: "SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation". In: *2020 European Conference on Computer Vision (2020)* (Edinburgh, United Kingdom). Ed. by Vedaldi, A.; Bischof, H.; Brox, T. and Frahm, J.-M. Springer International Publishing, 2020, pp. 1–19. doi: 10.1007/978-3-030-58604-1_1 (cit. on p. 15).

[YLW24] Yuan, T.; Liu, Y.; Wang, Y.; Wang, Y. and Zhao, H.: "StreamMapNet: Streaming Mapping Network for Vectorized Online HD Map Construction". In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Waikoloa, HI, USA). Jan. 2024, pp. 7341–7350. doi: 10.1109/WACV57701.2024.00719 (cit. on pp. 16, 77, 79).

[YŞU19] Yurtkulu, S. C.; Şahin, Y. H. and Unal, G.: "Semantic segmentation with extended DeepLabv3 architecture". In: *2019 IEEE Conference on Signal Processing and Communications Applications (SIU)* (Sivas, Turkey). Apr. 2019, pp. 1–4 (cit. on p. 15).

[ZBS14] Ziegler, J.; Bender, P.; Schreiber, M.; Lategahn, H.; Strauss, T.; Stiller, C.; Dang, T.; Franke, U.; Appenrodt, N.; Keller, C. G. et al.: "Making Bertha Drive—An Autonomous Journey on a Historic Route". In: *IEEE Intelligent Transportation Systems Magazine* 6.2 (2014), pp. 8–20. doi: 10.1109/MITS.2014.2306552 (cit. on pp. 8, 28, 158).

[ZK22] Zhou, B. and Krahenbuhl, P.: "Cross-view Transformers for real-time Map-view Semantic Segmentation". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (New Orleans (LA), United States). 2022. doi: 10.1109/CVPR52688.2022.01339 (cit. on pp. 17, 32, 72).

[ZLW23] Zhang, G.; Lin, J.; Wu, S.; Luo, Z.; Xue, Y.; Lu, S.; Wang, Z. et al.: "Online map vectorization for autonomous driving: A rasterization perspective". In: *2023 Advances in Neural Information Processing Systems (NeurIPS)* (New Orleans (LA), United States). Vol. 36. 2023, pp. 31865–31877 (cit. on p. 16).

[ZLZ21] Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H. et al.: "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Virtual). 2021, pp. 6881–6890 (cit. on p. 15).

[ZS14] Zhang, J. and Singh, S.: "LOAM: Lidar Odometry and Mapping in Real-time". In: *2014 Robotics: Science and Systems (RSS)* (Berkeley, United States, July 12–16, 2014). Ed. by Fox, D.; Kavraki, L. E. and Kurniawati, H. 2014 (cit. on p. 10).

[ZSW19] Zhan, W.; Sun, L.; Wang, D.; Shi, H.; Clausse, A.; Naumann, M.; Kummerle, J.; Konigshof, H.; Stiller, C.; Fortelle, A. d. L. et al.: "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps". In: *arXiv preprint arXiv:1910.03088* (Sept. 2019). doi: 10.48550/arXiv.1910.03088 (cit. on pp. 9, 12).

[ZZD24] Zhang, Z.; Zhang, Y.; Ding, X.; Jin, F. and Yue, X.: "Online vectorized hd map construction using geometry". In: *2024 European Conference on Computer Vision (ECCV)* (Milan, Italy). Ed. by Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T. and Varol, G. Springer International Publishing, 2024, pp. 73–90 (cit. on p. 16).

[ZZW21] Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Ma, Y.; Li, W.; Li, H. and Lin, D.: "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Virtual). 2021, pp. 9939–9948 (cit. on pp. 16, 97).

List of Figures

List of Tables

A Appendix

A.1 HD Map Layer Composition

Table A.2 supports Figure 2.1 by providing an overview of different HD map layer compositions. Depending on its application, the map definitions vary in the number of layers, their semantic meaning and the contained metadata.

A.2 Dataset Statistics

Table A.1 provides additional statistics about the recorded sequences, discussed in Section 4.3, before and after filtering based on minimum traveled distance and loop closure. The procedure ensures that the dataset contains scenes with a similar density of training samples. In total, the filtered dataset contains 62 km of driven trajectory, and approximately 65 000 sensor frames. Typically, only the first drive of each sequence covers the full circle, while the subsequent drives are shorter and contain approximately 60 %-80 % of the unique road coverage.

Table A.1: Summary of available data frames before and after filtering.

SQ#	# frames per drive, initial				# frames per drive, selected			
	$SQ_{\square 1}$	$SQ_{\square 2}$	$SQ_{\square 3}$	$SQ_{\square 4}$	$SQ_{\square 1}$	$SQ_{\square 2}$	$SQ_{\square 3}$	$SQ_{\square 4}$
SQ_A	10463	5890	7062	6663	5801	4349	4207	4113
SQ_B	6462	4888	3687	3627	3918	3100	2701	2731
SQ_C	6662	3493	2182	2308	4090	2283	945	1875
SQ_D	5532	3017	4074	2780	4783	2691	3047	2667
SQ_E	7001	3658	3769	4254	5084	3499	3752	3622

Table A.2: Definition HD maps regarding different layer composition. Layer architectures are used in autonomous driving projects, suggested in conjunction with a corresponding map framework or were independently presented in research papers.

HD map layer and metadata
Levison *et. al.*[LAB11], 2011, 2 layers *Probabilistic map*: Localization layer consisting of a 2D Gaussian-modeled infrared reflectivity *Road map*: Contains traffic signs and other elements
Ziegler *et. al.* [ZBS14], 2014, 3 layers *Digital road map*: Layout of drivable lanelets including lane-relations like merge, yield or stoplines *Point-feature map*: Localization layer consisting of dense point-features *Lane-marking map*: Localization layer with road features e.g. road markings, curbs, stop lines.s
Aeberhard *et. al.* [ARB15], 2015, 2 layers *Semantic, geometric layer*: Lane geometry and high-level semantic inform. e.g. lane connectivity. *Localization layer*: Loc. layer consisting of lane markings and road boundaries.
HERE Technologies [HER], 2017, 3 layers *HD localization model*: Loc. layer using roadside objects like guard rails, walls, signs or poles *HD lane model*: Precise lane-level details e.g. lane direction, type or boundary, also lane markings *Road model*: Understand local insights e.g. high-occupany lanes or country-specific roads
TomTom [Tom18], 2018, 2-7 layers *TomTom HD map*: Road rep. featuring lane models and geometry, traffic signs and road furniture. *RoadDNA*: Localization layer consisting of six sub-layers: (1) Roadside patterns for LiDAR-based localization, (2) Traffic signs for camera-based localization, (3) Lane markings along the roadway for camera-based localization, (4) Roadway objects perceived by radar, (5) Vertical poles for LiDAR, camera or radar localization, (6) Road surface reflectivity for LiDAR-based localization.
Poggenhans *et. al* [PPJ18], 2018, 3 layers *Physical layer*: Containing the real observable semantic elements *Relational layer*: Describes how the elements of the physical layer are connected. *Topological layer* Combines elements of the rational layer to a topology graph of passable regions.
Lyft Level 5 [Lyf18], 2018, 5 layers *Base map layer*: SD map which includes basic road network data as offered by web map servvices, *Geometric map layer*: Rep. of the 3D world by a voxel or ground map, derived from a point cloud. *Semantic map layer*: Semantic, physical obj., such as lane boundaries, crosswalks, stop signs, etc. *Map priors layer*: Pre-computed partial or interm. results. Bayesian prior prob. about dyn. world. *Real-time Layer*: Real-time updated map information.
Elghazaly *et. al.* [EFH23], 2023, 6 layers *Base map*: 3D environment representation made of raw sensor data. *Geometric map*: High-precision lane-level geometric primitives. *Semantic map*: Semantic information about road features incl. traffic lights, road signs, crossings. *Road connectivity*: Defines connection between geometric primitives. *Priors map*: Experience information from past rides including temporal changes etc. *Real time map*: Real time updated map information.

A.3 Definition of Label Set with Class Priority

Table A.3 and Table A.4 provide an overview of the label sets used for annotation of M_{DS} and M_{EP}. Each class is associated with a priority value $\mathbb{p}_c$ defining its annotation hierarchy. While M_{DS} is annotated manually and requires a stringent label definition, M_{EP} is generated fully automatically and its semantic definition is based on the employed detection model. In this work, the respective model Seamseg [PBC19] was trained on [KL21], as described in Section 4.4.2.

Table A.3: Label set of ground surface features map M_{DS} and corresponding class priority $\mathbb{p}_c$ indicating their annotation hierarchy. More details about the definition and annotation process can be found in [Sch21], a master thesis supervised by the author.

	Name	$\mathbb{p}_c$	Description
	road	1	Areas intended as drivable space for vehicles, including all lanes and road types. Areas separated from lanes by markings, e.g., bike lanes or parking areas, are also included.
	sidewalk	1	Areas intended for pedestrians and cyclists, separated from the road by physical structures such as curbs, usually elevated and located alongside the road. The class also includes traffic islands and pedestrian zones.
	terrain	1	All other area types not used for regular traffic participants, such as grass, forests, or isolated parking lots. This class also includes curbs.
	dashed line	3	Dashed line markings, e.g., dashed lane dividers which may run parallel or perpendicular to the driving direction.
	solid line	4	Solid line markings, e.g., solid lane dividers aligned with the direction of travel, excluding stop lines.
	other markings	2	All other road markings such as arrows, symbols, or stop lines. If the marking has a clear shape, e.g., a line or arrow, the exact contour is annotated; otherwise, the convex hull is used.
	unlabeled	0	All remaining areas, not defined by any of the other classes.

Table A.4: Label set of semantically tailored traffic element map M_{EP} and corresponding class priority $\mathbb{p}_c$. There are two color themes: c_1, used when also including ground surface features, is similar to Cityscapes [COR16], and c_2, designed for maximum distinctiveness with only three classes present. Detailed information about the modelling, optimization and mapping process can be found in [Hu26].

c_1	c_2	**Name**	$\mathbb{p}_c$	**Geometric representation**
		traffic sign	2	Modeled as planar surface, i.e., rectangle, triangle, circle or arbitrary polygon.
		traffic light	1	Modeled as cylindrical object.
		pole	3	Modeled as cylindrical object.

A.4 Definition of Voronoi Diagrams

Voronoi diagrams [Aur91] partition a plane into segments based on the distance to a set of seed points. Each segment, called Voronoi cell, is associated to one seed point and each point in the segment is closer to its seed point than to any other seed point. Consequently, the segment borders are equidistant to the closest seed points, making them a natural fit for elements like centerlines if seed points are sampled on road borders.

A.5 Results of Online HD Map Construction on Overlap Split

The overlap evaluation split purely consists of scenes that are also represented within drives in the training set. However, the same sensor recordings, i.e., same drive of the same scene, are never used for both training and evaluation.

Qualitative prediction examples of the base model are shown in Figure A.1 and Figure A.2. Table A.5 shows the performance of online HD map construction models on the overlap split across different input resolutions and training sample frequencies. Both qualitative and quantitative performance on the overlap split is significantly higher than on the geographically distinct split, as expected.

Table A.5: Quantitative results of online HD map construction on overlap split, along two experimental dimensions evaluated, sensor setup and frequency of training samples. The scaling factor of the input images is indicated in the resolution column as indices, e.g., r_5 corresponds to a scale factor of 0.5 and an image input resolution of 1500 px × 750 px.

	Parameter		Average Precision on overlap split				
	Reso.	Freq.	AP_{bou}	AP_{cen}	AP_{dsh}	AP_{sol}	mAP
Setup	r_{15}	$f_{S/1}$	0.606	0.521	0.529	0.446	0.525
	r_2	$f_{S/1}$	0.658	0.548	0.569	0.515	0.573
	r_3	$f_{S/1}$	0.711	0.579	0.621	0.555	0.617
	r_4	$f_{S/1}$	0.715	0.583	0.616	0.564	0.619
	r_5	$f_{S/1}$	0.707	0.575	0.593	0.542	0.604
	r_6	$f_{S/1}$	0.727	0.598	0.610	0.561	0.624
	r_7	$f_{S/1}$	0.724	0.593	0.607	0.561	0.621
	r_8	$f_{S/1}$	0.733	0.587	0.591	0.532	0.611
Frequency	r_5	$f_{S/40}$	0.517	0.417	0.372	0.253	0.390
	r_5	$f_{S/20}$	0.573	0.470	0.437	0.355	0.459
	r_5	$f_{S/5}$	0.677	0.558	0.568	0.507	0.577
	r_5	$f_{S/1}$	0.707	0.575	0.593	0.542	0.604
	r_5	f_{Mges}	0.772	0.629	0.671	0.621	0.673

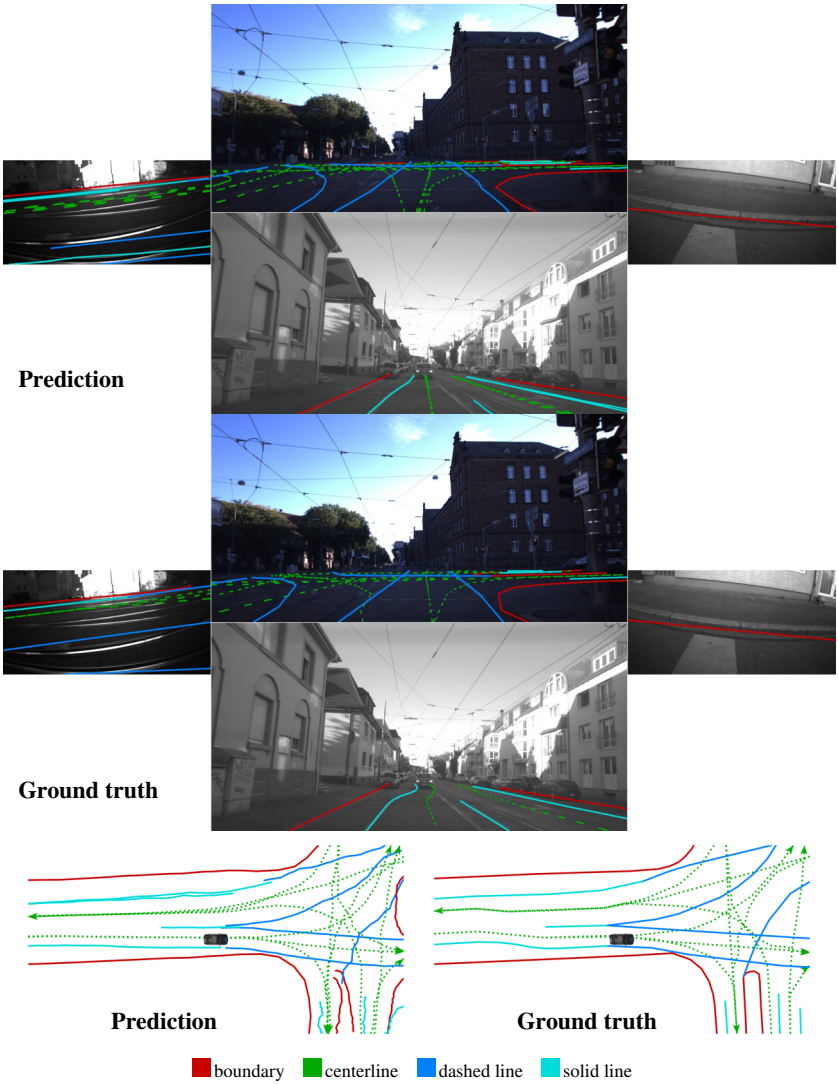


Figure A.1: Example of HD map construction performance on the overlap split. Top: Predicted map instances overlaid on the surround-view camera images. Middle: Ground truth map instances overlaid on the surround-view camera images. Bottom left: Predicted map instances in BEV. Bottom right: Ground truth map instances in BEV.

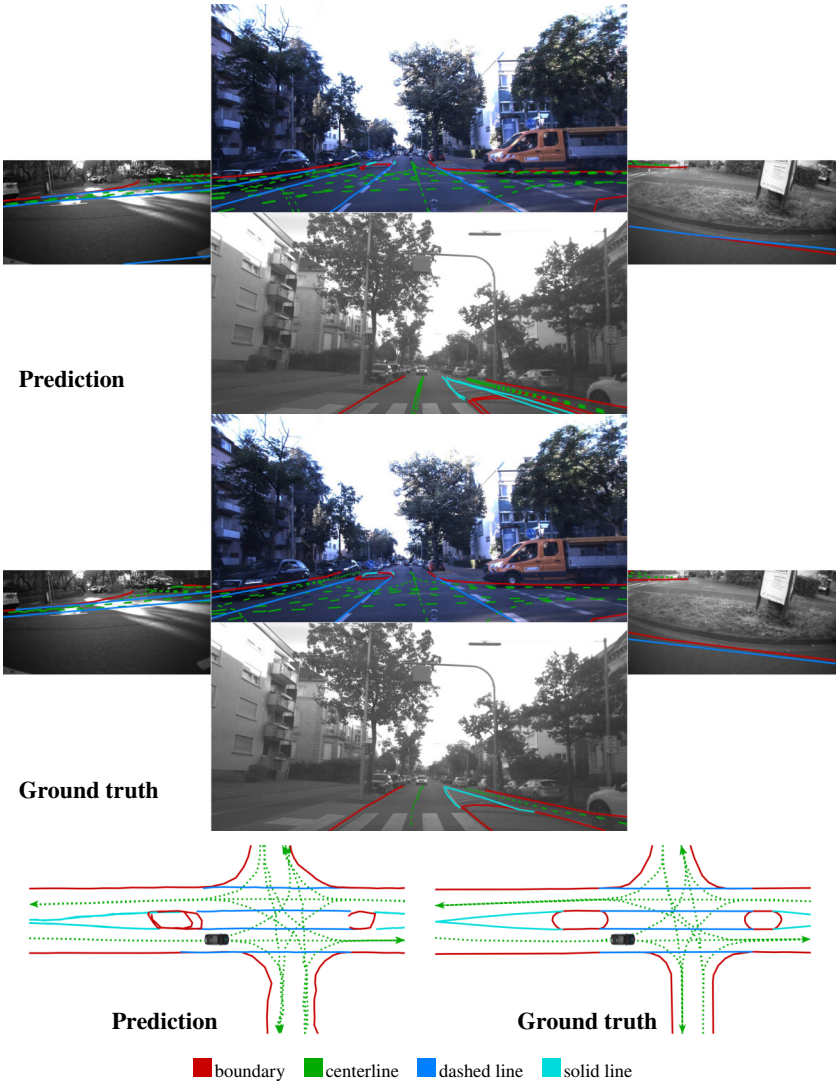


Figure A.2: Example of HD map construction performance on the overlap split. Top: Predicted map instances overlaid on the surround-view camera images. Middle: Ground truth map instances overlaid on the surround-view camera images. Bottom left: Predicted map instances in BEV. Bottom right: Ground truth map instances in BEV.

A.6 Results of Online HD Map Construction for Different Noise Patterns

Table A.6 shows the results of the robustness study regarding the localization noise patterns Ramp, Gaussian, and Perlin, as modelled and described in [BMF25]†. The results are reported on the geo.-split test set using the base model with input resolution r_5 and training sample frequency $f_{S/1}$. While ϵ_L and ϵ_R are the maximum translation and angular error, σ_L and σ_R are the standard deviations of the Gaussian noise in translation and rotation, respectively. h_c is a boolean indicating whether a heading correction is applied for the Ramp noise type and d_f is the division factor of the Perlin noise. All further parameters are chosen as in [BMF25]†. Overall the results mimic those observed by Blumberg et al. The system is quite robust towards small localization noise levels and starts to degrade especially if angular noise is introduced.

Table A.6: Online HD map construction results for the noise patterns Ramp, Gaussian, and Perlin. The three error types are modeled as described in [BMF25]†.

Noise Parameter				Average Precision					
				AP_{bou}	AP_{cen}	AP_{dsh}	AP_{sol}	mAP	vs.ba.
	ϵ_L	ϵ_R	h_c						
Ramp	2.0	0.0	✓	46.3	30.1	19.9	14.5	27.7	-1.2
	2.0	1.0	✓	43.7	31.0	20.0	12.5	26.8	-2.1
	4.0	1.0	✓	40.3	28.0	20.1	11.6	25.0	-3.9
	σ_L	ϵ_R	σ_R						
Gaussian	0.5	0.0	0.0	46.1	30.0	19.0	13.9	27.3	-1.6
	0.5	6	2.0	47.3	25.4	18.9	12.6	26.0	-2.9
	0.5	10	3.0	41.8	16.2	12.2	8.3	19.6	-9.3
	ϵ_L	ϵ_R	d_f						
Perlin	1.0	0.0	1000	46.3	32.7	21.0	15.1	28.8	-0.1
	1.0	0.5	1000	45.2	32.5	21.1	15.1	28.5	-0.4
	4.0	2.0	1000	37.2	23.0	14.6	13.3	22.0	-6.9
		base		47.0	31.5	19.5	17.8	28.9	-

A.7 Example of Localization Issues

In most cases, the initial SLAM-framework delivers a highly accurate and consistent pose graph. However, in rare occasions, such as in the event of the turning maneuver shown in Figure A.3, the SLAM can fail to provide accurate poses. In the shown example the estimated turning rate is too low, resulting in a temporary drift which is corrected after the turn is completed. Although, the drift is only temporary, this behavior is critical for the automatic semantic mapping process described in Section 5.6, which lead to the decision to switch to the multi-modal, continuous-time trajectory SLAM approach described in Section 4.4.5.

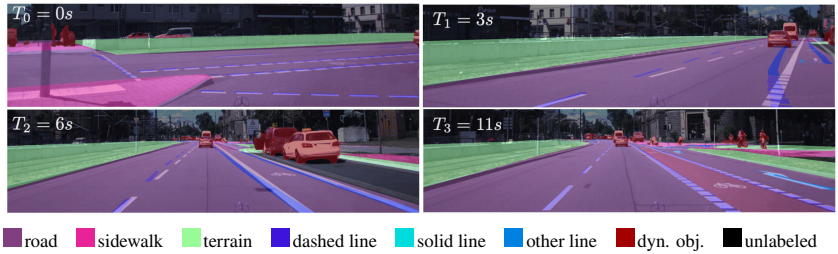


Figure A.3: A pose drift occurs during a turning maneuver, leading to misalignment between the back-projected map elements and the corresponding sensor data.

A.8 Ground Truth Comparison for Different Cross-modal Domain Adaptation

Figure A.4 illustrates the differences in annotations between XD-MAP and two baseline approaches, XD-B1 and XD-B2. In contrast to the baselines, XD-Map provides a more complete 360° annotation which preserves geometric details of the map elements, thanks to the tailored modeling and mapping process.

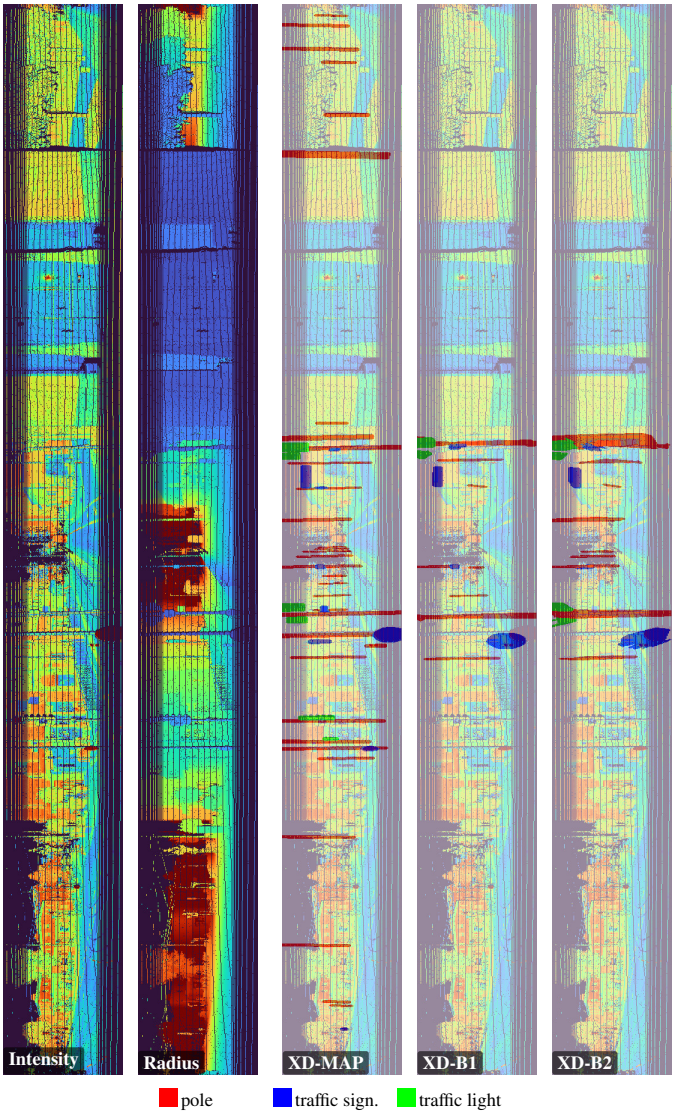


Figure A.4: Comparison of pseudo ground-truth annotations for 2D segmentation. Shown are annotations generated using the XD-MAP approach and two baseline methods, XD-B1 and XD-B2. The top row shows the input intensity and range encoded as spherical projections, while the bottom row shows the corresponding pseudo ground-truth annotations.

A.9 Semantic Segmentation Performance of Panoptic Models

Table A.7 compares the semantic segmentation performance of two models with identical architectures: one trained using panoptic labels and the other trained directly with semantic segmentation labels. Across most configurations and training runs, the model trained for the panoptic task achieves slightly higher performance when evaluated on semantic segmentation labels. Improvements in these cases are highlighted in green.

Table A.7: Comparison of semantic segmentation performance of models trained on using panoptic labels and semantic segmentation labels. Left are the panoptic models of table 7.1 evaluated on semantic segmentation labels. Right is their performance in comparison of corresponding models which are directly trained on semantic segmentation labels.

	Experiments	Mot. Comp.	Panoptic Experiments				Diff. to Semantic Experiments			
			IoU_{Po}	IoU_{TL}	IoU_{TS}	**mIoU**	IoU_{Po}	IoU_{TL}	IoU_{TS}	**mIoU**
Baseline	XD-B1	✗	7.0	11.0	10.9	9.6	-2.6	+1.9	+1.7	+0.3
	XD-B2	✗	11.0	14.8	15.0	13.6	-4.0	-3.8	-2.0	-3.3
	XD-MAP	✗	35.6	40.0	30.7	35.4	+8.2	+2.5	+1.3	+4.0
	XD-B1	✓	3.9	13.2	8.2	8.4	-6.0	+1.9	+0.1	-1.4
	XD-B2	✓	4.1	16.1	13.2	11.1	-13.8	-2.5	-2.9	-6.4
	XD-MAP	✓	39.6	39.9	32.6	37.4	+2.5	-2.4	+0.8	+0.4
Range	30 m	✗	36.2	42.7	32.3	37.1	+0.8	-0.2	+2.1	+0.9
	50 m	✗	35.6	40.0	30.7	35.4	+0.9	+0.6	+1.5	+1.0
	70 m	✗	34.8	38.6	29.7	34.4	+2.3	+1.5	+0.6	+1.5
	30 m	✓	41.7	44.4	33.7	39.9	+1.7	-1.0	+1.8	+0.8
	50 m	✓	39.6	39.9	32.6	37.4	+2.5	-2.4	+0.8	+0.4
	70 m	✓	37.8	39.7	30.3	35.9	+1.4	-0.4	-0.5	+0.2
Freq.	0.5 Hz	✓	35.7	37.1	29.2	34.0	+1.8	-1.4	+0.8	+0.4
	2 Hz	✓	38.7	40.4	32.1	37.0	+1.3	-0.8	+1.3	+0.6
	10 Hz	✓	39.6	39.9	32.6	37.4	+2.5	-2.4	+0.8	+0.4

A.10 Perspective View Segmentation Results with and without Dynamic Occlusion Handling

Figure A.5 shows two prediction examples for models trained with and without dynamic occlusion handling. The images illustrate two scenes from the geo.-split test set.

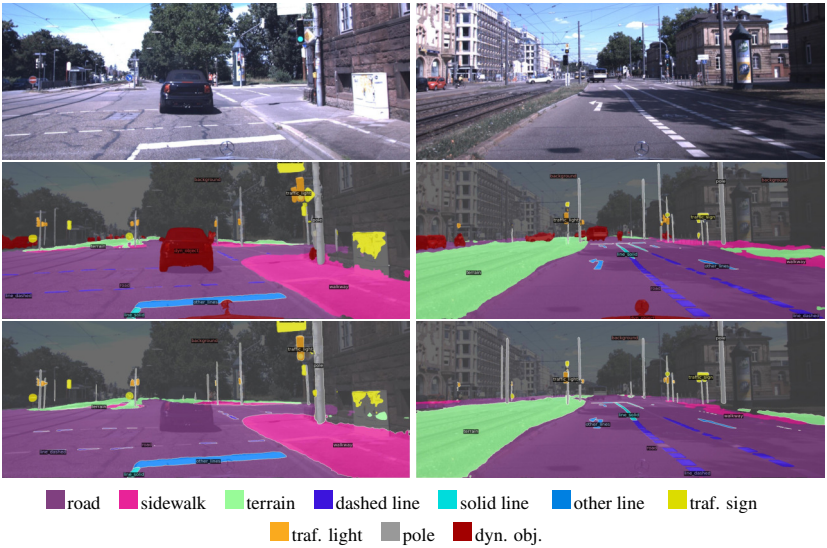


Figure A.5: Qualitative evaluation of dynamic occlusion handling. Two prediction examples are shown for a model trained with dynamic occlusion handling and without. Top to bottom: RGB image, prediction results of model trained with dynamic occlusion handling, prediction results of model trained without dynamic occlusion handling. Both examples are from the geo.-split test set.