UNIVERSITÄT BONN

Masterarbeit
zur Erlangung des Grades
Master of Science (M. Sc.)
der Landwirtschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation

Automatic Point Cloud Annotation using existing HD Map Data for Map Construction

von

Gülşen Bardak

aus
Isparta, Turkey

Bonn 2025

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

Second Supervisor:
Matteo Sodano, University of Bonn, Germany

Statement of Authorship

I hereby certify that this master thesis has been composed by myself. I have not made use of the work of others or presented it here unless it is otherwise acknowledged in the text. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged.

Bonn, 17.07.2025

Place, Date

Signature removed from the online version for data protection reasons. The signed declaration is available to the examination authority.

(Signature)

Acknowledgments

I would like to express my deepest gratitude to my dearest supervisors of over three years, Matteo Sodano and Michael Scholz. Their unwavering support, patience, and guidance made them the most extraordinary mentors I could have ever hoped for. I consider myself incredibly fortunate to have been their student throughout my MSc journey. My heartfelt thanks also go to my beloved family and friends, whose presence and kindness kept me grounded. To my wonderful parents, thank you especially for being a source of comfort and for helping to ease my anxiety. Your support meant everything. Last but not least, I wanna thank me.

Abstract

Perception is a critical task for autonomous driving and is challenging because of the scale and complexity of urban environments, which leads to limitations in sensor accuracy. Point-wise segmentation plays a crucial role in addressing these challenges. However, point cloud datasets have limitations related to data storage capacity, dimensionality, and the scale factor during manual annotation, making it difficult to achieve efficiency in terms of time and cost. In this study, we propose an automatic annotation pipeline sourced from another key contributor to automated driving systems: high-definition (HD) maps. HD maps provide lane-level information regarding traffic scenarios, 3-dimensional coordinates, and semantic classes of objects in a lightweight format. Our approach relies on generating a synthetic point cloud from these specific vector-type datasets and matching it with the original sensor-collected training dataset. We then utilize the point clouds and this information to train and generate reliable predictions. Ultimately, we propose that this pre-processing stage can serve as a baseline for producing new HD maps, thereby contributing to automated driving in real-world applications.

Zusammenfassung

Die Wahrnehmung ist eine entscheidende Aufgabe für autonomes Fahren und stellt aufgrund der Größe und Komplexität urbaner Umgebungen eine Herausforderung dar, die zu Einschränkungen der Sensorgenauigkeit führt. Die punktweise Segmentierung spielt bei der Bewältigung dieser Herausforderungen eine entscheidende Rolle. Punktwolken-Datensätze unterliegen jedoch Einschränkungen hinsichtlich der Datenspeicherkapazität, der Dimensionalität und des Skalierungsfaktors bei der manuellen Annotation, was eine effiziente Zeit- und Kostenersparnis erschwert. In dieser Studie schlagen wir eine automatische Annotationspipeline vor, die auf einem weiteren wichtigen Bestandteil automatisierter Fahrsysteme basiert: hochauflösenden (HD-)Karten. HD-Karten liefern fahrspurgenaue Informationen zu Verkehrsszenarien, dreidimensionalen Koordinaten und semantischen Objektklassen in einem kompakten Format. Unser Ansatz basiert auf der Generierung einer synthetischen Punktwolke aus diesen spezifischen Vektordatensätzen und deren Abgleich mit dem ursprünglichen, sensorerfassten Trainingsdatensatz. Anschließend nutzen wir die Punktwolken und diese Informationen, um zu trainieren und zuverlässige Vorhersagen zu generieren. Letztlich schlagen wir vor, dass diese Vorverarbeitungsphase als Grundlage für die Erstellung neuer HD-Karten dienen und so zum automatisierten Fahren in realen Anwendungen beitragen kann.

Task Descriptiion

Rheinische
Friedrich-Wilhelms-
Universität Bonn

Uni Bonn · Stachniss · Nussallee 15 · 53115 Bonn

Prof. Dr. Cyrill Stachniss
Professor

Institute of Geodesy
and Geoinformation

Photogrammetry
& Robotics Lab

Nussallee 15
53115 Bonn

Tel.: +49-228-73-2714
Fax: +49-228-73-2712
cyrill.stachniss@igg.uni-bonn.de

Secretary:
Birgit Klein
Tel.: +49-228-73-2713
Fax: +49-228-73-2712

Bonn, December 11, 2024

Task Description for M.Sc. thesis of Gulsen Bardak

High-definition (HD) maps play a critical role in autonomous driving technologies. Lane-level maps typically provide accurate environmental information for planning and perception tasks. However, the high-definition map construction process still has challenges due to the capability of sensors, the complexity of the real world, the labor costs for annotation, and other structural problems. Global HD map construction is usually performed offline to provide high accuracy and completeness thanks to the integration of multiple sensors and complex algorithms [5]. However, manual annotation remains the most reliable technique due to the high accuracy required; therefore, the process of producing high-definition maps requires an expensive labor cost due to this manual labeling procedure. Although point cloud datasets provide the required accuracy, point-based annotation is still an open issue as it is more complex and costly than pixel-based annotation in terms of sensitivity, recognizability, etc. Given these situations, there are strong relationships between HD map construction and point cloud annotation, both of which present challenges in terms of capability.

This thesis proposes to perform automatic point cloud annotation, where the generated high-definition maps constitute already annotated synthetic point clouds. High-definition maps cover 3D coordinates and detail at the lane level about road networks and other traffic elements such as objects and signs. We can use those details to generate a synthetic point cloud, and proper matching algorithms can function as an "automatic annotation procedure." A learning-based method can be used to produce high-definition maps on a global scale using this automatic annotation process, which uses a local sample of a region as training.

As a result, in this study, we address the problems of point-based labeling and high-definition map production. We approach these two complex problems by proposing matching of real point clouds with synthetic point clouds generated using high-definition maps in a way that they feed each other and investigate their contributions. We aim to make an efficient and high-accuracy contribution to point cloud annotation. The expected outcome of the project is generation of an efficient offline HD map construction method, similar to existing approaches such as [3], [2], and [4], using a learning-based generation method. This method utilizes automatic annotation like in [1], ensuring high accuracy and sufficient detail for self-driving instead of limited details. The quality of the annotation will be evaluated using results from a manually annotated validation dataset

UNIVERSITÄT BONN

Rheinische
Friedrich-Wilhelms-
Universität Bonn

as the ground truth, alongside standard segmentation evaluation metrics. The quality of the HD Map will be assessed through topological relations and geometrical evaluation, including gaps, overlaps, and continuity.

Point Of Departure:

- Generation of synthetic point cloud from geometries of high-definition map elements (e.g. traffic signs).
- Preparation both point clouds for matching algorithm.

Kind regards,

Signature removed from the online version for data protection reasons. The signed declaration is available to the examination authority.

Cyrill Stachniss

[1] S. Chen, Y. Zhang, B. Liao, J. Xie, T. Cheng, W. Sui, Q. Zhang, C. Huang, W. Liu, and X. Wang. Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene. *arXiv preprint arXiv:2304.09807*, 2023.

[2] X. Chen, W. Liao, B. Liu, J. Yan, and T. He. Opendenselane: A dense lidar-based dataset for hd map construction. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.

[3] Q. Li, Y. Wang, Y. Wang, and H. Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022.

[4] J. Shin, F. Rameau, H. Jeong, and D. Kum. Instagram: Instance-level graph modeling for vectorized hd map learning. *arXiv preprint arXiv:2301.04470*, 2023.

[5] X. Tang, K. Jiang, M. Yang, Z. Liu, P. Jia, B. Wijaya, T. Wen, L. Cui, and D. Yang. High-definition maps construction based on visual sensor: A comprehensive survey. *IEEE Transactions on Intelligent Vehicles*, 2023.

Contents

Chapter 1

Introduction

Scene understanding is essential for developing autonomous systems, where safety is a critical concern. Perception is a challenging task due to the scale of environments at the city and road level, as well as the complexity of these environments, sensor limitations, data scale, storage, and processing. Spatial and scene understanding of surroundings are crucial due to the necessity of safe and autonomous navigation in cars and robots, particularly in autonomous driving approaches [9]. Besides these, annotation that is required for learning-based systems is crucial to realizing perception tasks accurately. However, due to challenging conditions, it is still an open research topic that is trying to make it automatic instead of using laborious and expensive solutions. With our research, we will investigate a lightweight automatic annotation framework that can reduce labor and data storage costs by using existing high-definition maps effectively.

1.1 Motivation

Effective scene understanding enables autonomous vehicles and robots to navigate complex environments, avoiding obstacles and ensuring the predictability and reliability of participants. However, this is a challenging task due to the scale of environments at the city and road level, as well as the complexity of these environments, sensor limitations, data scale, storage, and processing. For this purpose, utilizing advanced sensors, especially cameras and LiDAR, autonomous vehicles can continuously and in real time sense their surroundings in diverse situations, including other vehicles, pedestrians, obstacles, and road signs [26]. However, even if this technology enables the understanding of surroundings, there are still some issues in terms of real-time capability, processing capacity, and sensor adaptability in extreme conditions. So, lightweight solutions for annotation in terms of LiDAR and camera-based perception, high storage capacity, and labor

are still a subject of research. Although LiDAR and cameras undertake this task and GNSS/IMU-integrated Mobile Mapping Systems (MMS) are widely used in this sense, recently high-definition (HD) maps have come to the fore in terms of playing a critical role in autonomous driving tasks and creating a reliable environment for localization, planning control, and perception tasks.

1.2 Goal and Main Contributions

As the interest and need for autonomous driving increase day by day, it also becomes inevitable to automate the elements required to achieve this. Autonomous systems must understand their environment with high accuracy to minimize human-caused traffic accidents and build safe roads and cities. However, achieving those goals has high costs on the road and city scale in terms of labor, storage space, and time management. Therefore, automating the annotation of data sets for use in learning-based systems is necessary. On the other hand, discussions have focused on high-definition maps and their production, which are crucial for completing tasks related to perception, planning, localization, and control. Therefore, the two issues that are relatively connected to each other are the main motivations for this study. We will investigate how each issue can mutually contribute to the collaboration between high-definition maps and annotation bottlenecks in point-wise segmentation.

1.3 Overview of the Thesis

This thesis explores how to improve scene understanding for autonomous systems by reducing the need for manual data annotation. It focuses on using high-definition (HD) maps to support automatic point-wise segmentation, which can lower costs related to labor, storage, and processing. The goal is to develop a lightweight, efficient framework that helps autonomous vehicles and robots better understand their surroundings while addressing the challenges of real-time perception and data management.

Chapter 2

Related Work

2.1 High-Definition Maps

2.1.1 HD Map Construction

The high-definition map construction process primarily consists of two stages: raw data collection and data processing. The data processing stage can be further divided into two categories - offline and online -, based on the approach used, a distinction that originates from Simultaneous Localization and Mapping (SLAM) methodologies. The choice between these approaches is dependent upon the processing of sensor data: either in real-time (online) or via post-processing (offline) [22]. Contemporary studies in HD map construction mostly depend on online methodologies [15, 12, 24, 7, 20]. These approaches generally utilize learning-based methodologies to generate vectorized HD maps by framing the issue as semantic segmentation from a bird's-eye view (BEV). Most online HD map generation techniques begin with BEV feature extraction from onboard sensor data, followed by the generation of vectorized map elements such as road boundaries, pedestrian crossings, and lane dividers. These extracted features are then used to construct vectorized maps, eliminating the need for localization and post-processing in many cases. Although this approach reduces the workload associated with post-processing and allows for *local* high-precision map creation, it tends to be limited in scope. Despite being referred to as "high-definition maps", such outputs offer only partial information. A truly high-definition map should include a broader set of elements. In addition to basic road features such as road markings and lanes, traffic elements like lights and signs, as well as supportive infrastructure like street lamps, trees, and static objects, are also crucial for autonomous driving systems [16]. Depending on the designated application use case, semantic information is also highly relevant to be included in HD map data. This can be realized through topological links between elements, such as linking of neighboring lanes or linking of predecessors and successors of a lane.

Additional valuable information can be the knowledge about the validity of a traffic signal for a specific set of lanes or about the association of a signal to a stop line road marking, for example. Unlike online methods for local map construction, global HD map construction is still commonly conducted through offline methodologies. These offline approaches typically offer higher accuracy and completeness by incorporating a greater variety of sensors and more complex algorithms, albeit with increased processing times [22]. Nevertheless, manual annotation remains the most reliable method due to the precision required, especially regarding semantic information, making the overall process of HD map production both labor-intensive and costly. To alleviate this, many efforts rely on image-based techniques for tasks such as road marking extraction and lane detection. Although several public datasets already exist with annotated images, the required precision for HD maps is often unattainable due to intrinsic limitations of image data. Environmental factors such as lighting conditions and shadows can degrade both the annotation quality and the accuracy of map element extraction. Moreover, projecting from 2D images to 3D space presents significant challenges, frequently leading to diminished accuracy relative to the stringent requirements for HD mapping. Conversely, LiDAR point clouds intrinsically encompass 3D spatial data at each feature point, facilitating more precise detection outcomes that may be directly applied in HD map development [5]. Annotating point-cloud data is, however, more intricate than image annotation. This complexity stems from hardware limitations such as data transport and memory usage, together with data-specific issues like sensitivity and recognizability. To address challenges on annotation and creation of high-precision maps, several approaches have been proposed. VMA [4] introduces automatic annotation for online HD map construction through a scene-splitting strategy. CAMA [25] provides automatic annotations using image-based methods enriched with elevation information. It seeks to produce dense 3D road surfaces augmented with semantic and photometric features, utilizing the nuScene dataset for evaluation. THMA [21] introduces an annotation technique grounded in self-supervised segmentation learning, with the objective of enhancing the automation of the HD map annotation process. Despite the existence of multiple methods for representing road networks as high-definition maps, data standardization remains a work in progress. This study will utilize data supplied for a specific region in the OpenDRIVE data format along with a reference point cloud to evaluate algorithm performance.

2.1.2 Characteristics of OpenDRIVE

The ASAM OpenDRIVE format is an open industry standard maintained by the Association for Standardization of Automation and Measuring Systems. It

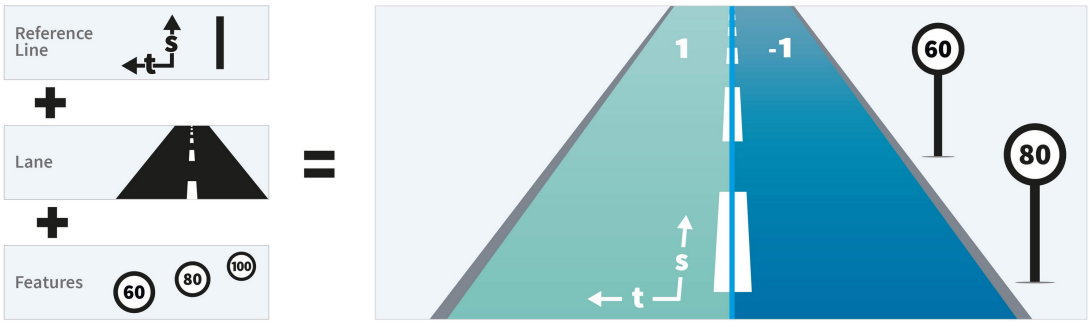


Figure 2.1: Modeling of road elements in OpenDRIVE; © ASAM

represents road networks in a file format with the extension `.xodr`, organized in a hierarchical structure commonly encoded using XML. This format captures the geometric relationships of road features and can be generated using real data or synthetically in various software environments (mostly proprietary ones). Besides the main road components (lanes, road marks, road signs, etc.) that are modeled as shown in Figure 2.2, an OpenDRIVE dataset can contain traffic-regulating infrastructure elements (traffic lights, traffic signs, etc.) and supporter elements (street lamps, trees, objects, etc.). The complexity of OpenDRIVE makes data acquisition a sophisticated task, often financed by the automotive industry and conducted by third-party mobile mapping providers. As a main characteristic, all road elements are commonly constructed in relation to and linearly referenced along a *road reference line* as shown in Figure 2.2. Because of advanced geometry representation through parametric cubic polynomials (Figure 2.4), the modeling of complex road features can still remain lightweight as shown in the Figure 2.3:

2.2 Semantic Segmentation of Point Cloud

Semantic segmentation has a critical role in scene understanding to realize automated driving. The main approach to doing semantic representation is assigning semantic classes to each basic component in the dataset that is collected during the driving: pixel-on-pixel-wise semantic segmentation and points in point-wise semantic segmentation [10]. Even though early and common approaches rely on image-based semantic segmentation because of the availability of image-based datasets, pixel-wise segmentation has some constraints in terms of scalability on different dimensions and conditions. Pixel-based segmentation has to be projected onto the 3-dimensional physical world from a 2-dimensional plane representation; however, this task is open to losing some valuable information, like depth, so post-processing is required. Additionally, cameras have limitations when capturing images in various weather conditions, such as rain, fog, or at night. 3D

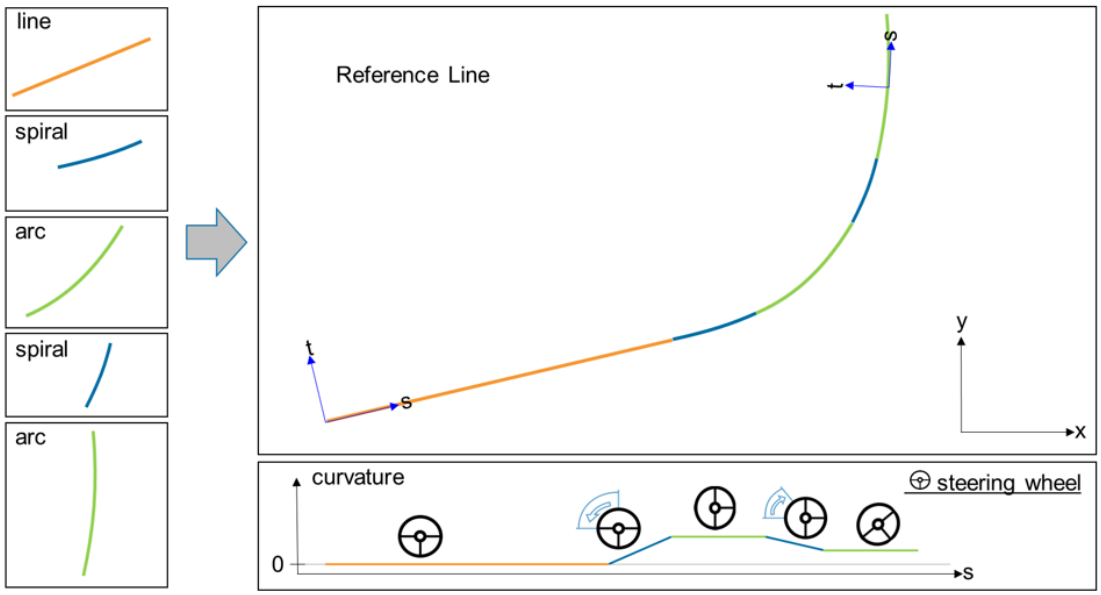


Figure 2.2: Creating a reference line from geometry elements; © ASAM

LiDAR point clouds can provide reliable information about extreme weather conditions and surface reflectivity; however, point-wise segmentation faces challenges because the complex structure of 3D data complicates storage and annotation [10]. Effective segmentation requires accurate annotation in huge amounts of sequences; however, because of inconsistency and perspective issues, labeling point clouds is challenging to maintain accurately [23]. Sourced with those issues, recent research is focusing on two aspects of alternative ways of annotation: the first one is transfer learning that uses domain adaptation from different source data, and the second one is automatic annotation with various methodologies. Transfer learning could be categorized into two subtitles: synthetic-to-original dataset that is sourced from point clouds specifically for this purpose and original-to-original transfer that is sourced from another sourced dataset. There are some publicly available datasets that consider the driving scenario, e.g., SemanticKITTI and nuScenes [2, 3].

However, this public dataset suffers from a class imbalance, known as the long-tail class problem. Besides that, those datasets rely on human annotation, so human-oriented problems and errors can occur [11]. To address this problem, some studies focus on synthetically generated point clouds. However, few studies are still considering synthetic-to-real approaches because there is a lack of extensive synthetic data with accurate semantic labels [23]. Several studies utilize the Grand Theft Auto V (GTA V) environment, a commercial video game known for its realistic driving simulation, to produce synthetic point clouds for use in 3D point cloud segmentation. Using game-engine-dependent synthetic point clouds

```
<geometry
      s="0.000000000000e+00"
      x="6.804539427645e+05"
      y="5.422483642942e+06"
      hdg="5.287405485081e+00"
      length="6.565893957370e+01">
      <paramPoly3
            aU="0.000000000000e+00"
            bU="1.000000000000e+00"
            cU="-4.666602734948e-09"
            dU="-2.629787927644e-08"
            aV="0.000000000000e+00"
            bV="1.665334536938e-16"
            cV="-1.987729787588e-04"
            dV="-1.317158625579e-09"
            pRange="arcLength">
      </paramPoly3>
</geometry>
```

Figure 2.3: Geometry XML with paramPoly3

has advantages thanks to the instance-level annotation for every object class [11].

This approach proposes to eliminate errors related to human labeling during the annotation task by using a synthetically generated 3D world, depending on the purpose. This method streamlines the annotation process and allows researchers to generate diverse datasets under controlled conditions, facilitating more robust machine learning models. Consequently, leveraging such synthetic environments can enhance the accuracy and efficiency of various computer vision applications. However, this approeaches are still facing some problems, like the requirement of preprocessing and covering all possible events and objects in real-world scenarios.

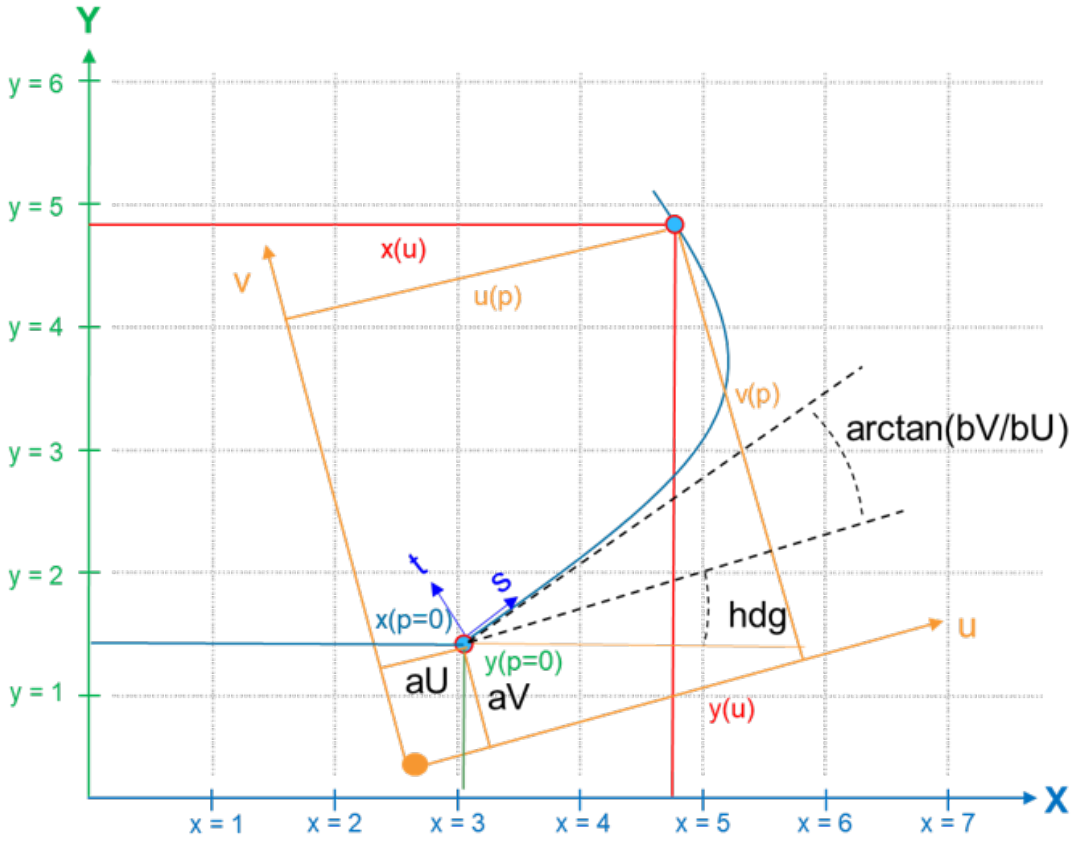


Figure 2.4: A parametric cubic polynomial in OpenDRIVE; © ASAM

Chapter 3

Basic Techniques

In this study, we proposed a lightweight automatic annotation methodology that will enable a cost-efficient approach by reducing labor and time costs associated with the manual annotation process. To complete this task, we proposed using a synthetically generated point cloud derived from the vector data, which includes all necessary information: 3D coordinates and semantic labels related to the traffic scenario participants. Because of the synthetic point cloud and the reference/original point cloud's existence, the Iterative Closest Point algorithm is the key basic technique that is used for registration and label transfer from the synthetic point cloud to the original point cloud. The second basic technique is neural networks, which are commonly used for semantic segmentation on point clouds.

3.1 Iterative Closest Point Algorithm

The Iterative Closest Point Algorithm (ICP) [1] algorithm is particularly effective for aligning two sets of data by minimizing the distance between corresponding points. This allows for improved accuracy in identifying and classifying the various elements within the traffic scenario, ultimately enhancing the system's ability to interpret real-world environments. In our case, we utilized this algorithm to get corresponding point pairs that will provide label transfer. Synthetically generated point clouds and sensor-collected original point clouds differ in density; therefore hyperparameter selection for the ICP algorithm is critical for our approach in this aspect. Besides hyperparameters, we utilized some external preprocessing like downsampling and normalization on the original point cloud to enhance algorithm performance.

3.2 Neural Network for Point-wise Segmentation

To ensure that the labels obtained through the Iterative Closest Point (ICP) algorithm could yield meaningful predictions, we employed a neural network for supervised semantic segmentation. The base model chosen was PointNet++ [17], but for the sake of simplicity, we disabled the max pooling functionality while retaining the convolutional structure. This approach enabled the model to be trained with the parameters defined by the provided data, making it ready for prediction. Network details and hyperparameter functionalities are explained in the Section 4: Approach.

3.3 High-Definition Maps

According to the definition of maps in China's cartography textbooks, "maps are graphs that follow a specific mathematical law, run a symbolic system, and often reduce various natural and socio-economic phenomena on the earth." To enable the transfer of information, these maps are designed with various intuitive, concise, easy-to-understand, and easy-to-remember map symbols to identify various spatial things and events, and service objects are mostly human [14]. However, it is not our only focus anymore. As mentioned before, it needs mapping in vehicles because, in the future, not people, but systems are expected to use maps and make the right decisions. Therefore, it is necessary to use different maps than traditional digital navigation maps for vehicles with different decision and perception mechanisms. This feature adds different analysis and display methods of various sensor integrations to mobile mapping systems that are generally dominated by Inertial Navigation Systems (INS) and supported by Global Navigation Satellite Systems (GNSS), and these are called HD maps. High-definition maps are new-generation navigation maps with high accuracy and many more components, produced to make sense of machines, unlike digital maps produced as navigation maps for human perception under normal conditions. Unquestionably, the machines' detection capacity should also incorporate a prediction function. While the term "learning" has become widespread in many fields, it is equally difficult for a system without experience to transform the information it receives into knowledge. In this context, High Definition Maps serve as the foundational unit that establishes knowledge infrastructure without relying on complex algorithms.

Chapter 4

Approach

The flowchart in Figure 4.2 shows the proposed approach from a general perspective. As shown in this flowchart, the thesis essentially covers the preliminary processes required to realize the task of OpenDRIVE Construction. The first task is to create a 3D shape from the OpenDRIVE format dataset, then compare the new synthetic point cloud with the original/reference point cloud, transfer labels this way, and finally evaluate how well supervised semantic segmentation and automatic annotation work. In this context, the approach section is organized into four main sections:

1. Geometry extraction involves acquiring OGC Simple Geometry Features from the high-definition map in OpenDRIVE format, which consists of data represented in vector form. Due to its global representation in simple geometry features, this is feasible from a computational perspective. As noted in the related works section, OpenDRIVE exhibits a complex geometric representation due to the intricacies of the real world.

2. Building a synthetic point cloud subsection that will cover how synthetic point cloud generation can be done to provide a proper baseline for the ICP that will handle the label transferring into the original/reference point cloud.

3. In the third step of our approach, we will discuss the Iterative Closest Point algorithm, which has several hyperparameters that depend on its specific purpose, and how to achieve optimal results for matching sensor-collected point clouds and synthetic point cloud methods.

4. In the last subsection in this part, the automatically labeled point cloud will be utilized in the process of selecting a model, training it, and evaluating the results of the supervised semantic segmentation procedure.

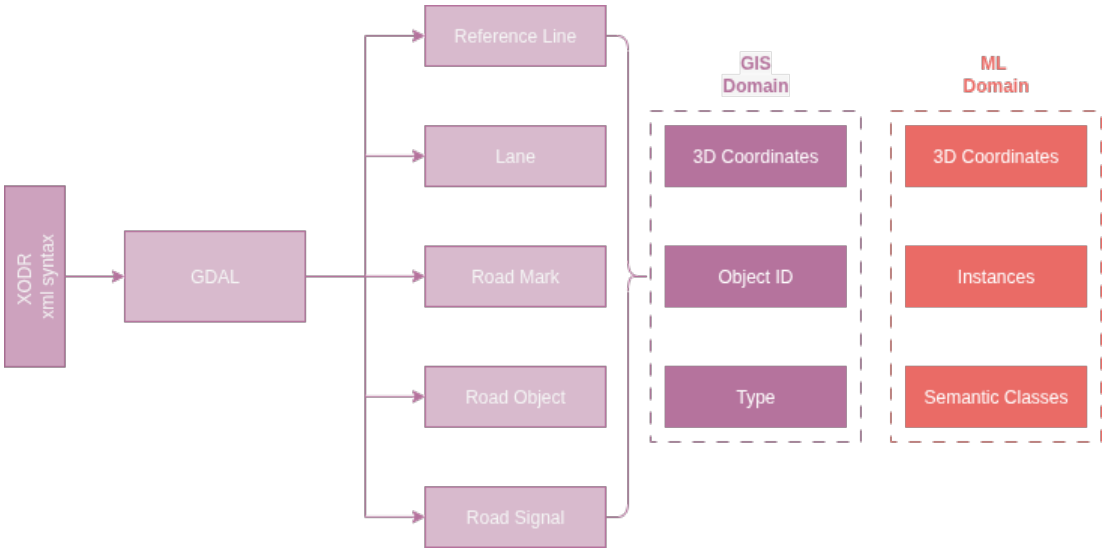


Figure 4.1: Approach design in Geographic Information Systems (GIS) and Machine Learning (ML) aspects.

4.1 Geometry Extraction from HD Map

As explained in Section 2.1.2, we used the OpenDRIVE data format because it is one of the standard types of datasets available for our region, in addition to point clouds. To process OpenDRIVE datasets, we utilized the Geospatial Data Abstraction Library (GDAL) [8], which provides a dedicated driver capable of reading and processing OpenDRIVE (`.xodr`) files. This driver converts the data into simple geometry types, as defined by the Simple Feature Standard of the Open Geospatial Consortium (OGC). This conversion allows us to extract georeferenced and simplified geometries of the road network and surrounding infrastructure. Using these geometries, a synthetic point cloud can be generated, which serves as a baseline for creating annotated datasets. Figure 4.3 illustrates this process. The left part of the figure shows the existing OpenDRIVE data. The training and validation datasets of the collected point clouds should correspond to the same geographical region. OpenDRIVE data not only includes primary road elements but also incorporates traffic infrastructure such as traffic signs, parking areas, and other support elements. These can be reflected in the annotated datasets, capturing the full context of the environment. GDAL facilitates this process by leveraging its OpenDRIVE driver, which can convert raw XODR files into a Triangular Irregular Network (TIN). This conversion enables the extraction of 3D spatial information from XML-based datasets that describe static roadway features. From this 3D data, we can synthesize a point cloud, which forms the foundation for the subsequent annotation phase.

As also discussed in Section 2.1.2, OpenDRIVE data can include complex

geometry representations, typically modeled using third-order parametric cubic polynomials. Each road element is defined relative to a reference line using these polynomials. Figure 4.4 demonstrates how OpenDRIVE employs cubic polynomial functions to accurately interpolate the shape of road elements. The GDAL XODR driver relies on `libOpenDRIVE`[13], an open-source C++ library designed for parsing and processing OpenDRIVE files. This library utilizes Bézier curves to interpolate the parametric cubic polynomials, enabling precise reconstruction of road geometry.

Through GDAL, mainly 6 different vector data layers in different geometry types are exposed, depending on the characteristics of the original OpenDRIVE geometries. Those six layers are:

- *ReferenceLine*: Road Reference line as `OGRLineString`
- *LaneBorder*: Outer road lane border as `OGRLineString`
- *Lane*: Polygonal surface (TIN) of the lane mesh as `OGRTriangulatedSurface`
- *RoadMark*: Polygonal surface (TIN) of the road mark mesh as `OGRTriangulatedSurface`
- *RoadObject*: Polygonal surface (TIN) of the road object mesh as `OGRTriangulatedSurface`
- *RoadSignal*: Polygonal surface (TIN) of the road signal mesh as `OGRTriangulatedSurface`

We mainly use the three layers RoadObject, Lane and RoadMark, which are modeled as a Triangulated Irregular Network (TIN), in order to create a 3D model of the city and roads. In our case the layer RoadObject contains basic information about building structures which in other OpenDRIVE datasets is not always included. This representation will allow volumetric representation for 3-dimensional use cases. Internally, libOpenDRIVE takes care of the linear approximation (sampling) of OpenDRIVE's continuous parametric geometries (which are illustrated in Figure 2.4). To facilitate this, specific hyperparameters are defined in libOpenDRIVE, which we also specify in our algorithm. We used OpenDRIVE data format as high-definition map representation format to complete 3D shape reconstruction task. For proof of concept we implemented our pipeline to openly avaliable dataset that covers OpenDRIVE dataset "Schwarzer Berg" in Brunswick [18], which we convert to OGC Simple Feature geometries via GDAL. Figure 4.5 is representing specific layer *LaneBorder* converted version on standard map. Figure 4.5 is shown how details represented in dataset; gray areas shown driveable area that bounded by *LaneBorder* layer, red objects

are signal that is responsible for objects like traffic light and signs ares stored as *RoadSignals* with their local types (e.g. Sign 101-11 Pedesterian Crossing in Germany) , green objects are trees or vegetations; and blue objects are buildings. Those geometries will be constituted as a baseline to generate a synthetic point cloud that is explained in the following section in detail.

4.2 Building Synthetic Point Cloud

As explained in Section 4.1, the resulting geometries were generated in a TIN structure to be used in constructing the synthetic point cloud. This way, the objects or participants in the scene are expected to be synthetically generated with the same high detail (e.g., convex geometries) as the original point cloud. However, it should be noted that the performance of this approach is directly related to the level of detail provided in the OpenDRIVE data generation. The GDAL XODR driver generates the triangles and the corner points of the triangles that form the object for triangulated irregular networks; therefore, a proper interpolation method is required. For this purpose, three different sampling methods were used, and the number of points per m^3 expresses the point density to be generated per volume. The model's geometry becomes more detailed in volumetric instances as this value increases. The third parameter is the downsampling ratio, as the reference point cloud [19] exhibits a higher point density than our generated synthetic point cloud. We will measure our algorithm quality with Iterative Closest Point (ICP) registration with a reference point cloud in Section 5.

4.3 ICP Integration

To get label transfer from a synthetic point cloud, the method for creating this point cloud is explained in Section 4.2: Building a synthetic point cloud, and the Iterative Closest Point (ICP) algorithm is used to align it with the original point cloud gathered by sensors. Table 4.1 lists the hyperparameters used to realize a proper matching algorithm in our approach.In the beginning, the sparsity of the synthetic point cloud is dependent on set parameters that are explained in Section 4.2. This situation leads us to conclude that a higher density is an efficient approach initially. However, because of computational cost, densifying synthetic point clouds is not the best approach. To deal with these situations, we reduce the number of points in the original point cloud to lower costs and avoid mismatches due to differences in density with the synthetic point cloud. Therefore, the downsampling ratio is the most important setting for using the Iterative Closest Point algorithm, which helps match points between two point

clouds so that labels can be transferred.The next two hyperparameters are important for the *KDTree* search algorithm, which helps estimate normals and is used in the registration process for both point clouds. The maximum number of nearest neighbors determines the number of points available in the search space. Additionally, we used this parameter to identify corresponding nearest neighbor points for estimating the intensity values of synthetic point clouds; however, we have not implemented this value in the network.

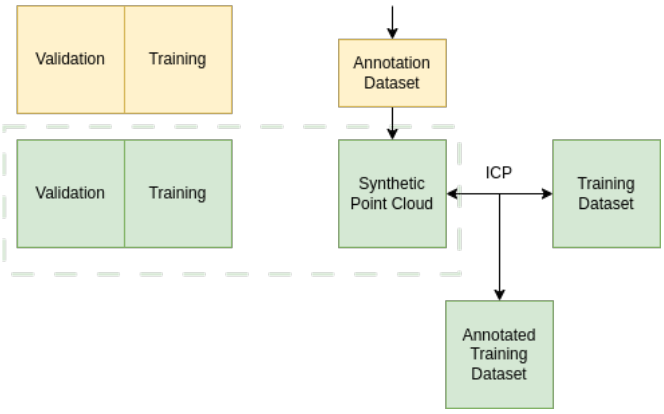


Figure 4.7: Process sub-diagram for automatic annotation

Parameter	**Functionality**
Downsampling ratio	Regulates sparsity mismatch between the original point cloud and the synthetic point cloud
Max NN for KDTree	Maximum number of nearest neighbors used in KDTree search
Radius for KDTree (m)	Radius for local neighborhood search in KDTree; controls the scale of local geometry
Initial guess for ICP transformation	Initial transformation matrix for ICP
Transformation type	ICP method: either point-to-point or point-to-plane depending on input quality
Max correspondence distance	Threshold for accepting point correspondences during ICP alignment

Table 4.1: Hyperparameters for Iterative Closest Point Algortihm

4.4 Model Structure for Semantic Segmentation

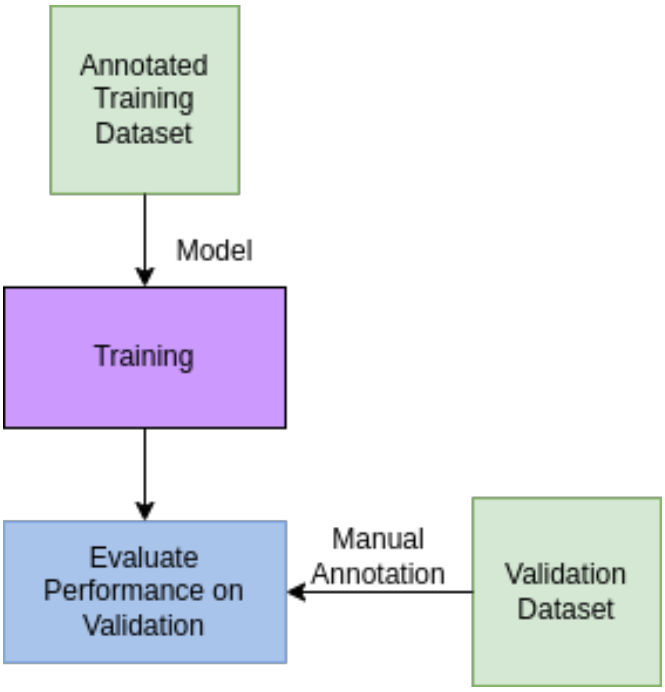


Figure 4.8: Process sub-diagram for training

As explained in previous sections, this study proposed an automatic annotation pipeline for use in point-wise semantic segmentation. Our problem definition will centers on how to use the supervised learning methodology to effectively predict semantic classes, adhering to the previously described approaches. Debugging and investigating ready-to-use model structures was difficult and time-consuming due to the custom-generated dataset. So to realize semantic segmentation, we explored two different custom architectures that mainly rely on the PointNet [17] network structure. The first architecture we explored is PointNet, which processes 3D coordinates using convolutional layers. It could not provide sufficient accuracy for predictions even if different loss functions and hyperparameters were used. The second network is called GeoPointNet and uses point normals in addition to 3D coordinates in a similar structure and pseudostructure, as can be seen in Algorithm 2. Besides the model selection, data preprocessing is one of the most important steps in refining our model performance. To handle computational issues, we downsampled the input dataset that provides a certain point number. On the other hand, we normalized point coordinates to ensure stability and reduce scale variance during the training stage. Because the datasets are unbalanced, meaning that some semantic categories occur more often others, we figured out the weights for each class and chose a loss function that takes these weights into account to handle the differences between classes and prevent problems caused by

the imbalance. As shown in Algorithm 3, we employed a combination of cross-entropy loss and focal loss to address class imbalance in the dataset, utilizing class weights. As regular hyperparameters for model structure, different batch sizes and optimal epoch numbers are also experienced to get better predictions.

Algorithm 1 Preprocessing Pipeline for Point Cloud Batching

Require: Point cloud dataset $\mathcal{D} = \{(X_i, Y_i)\}$ where $X_i \in \mathbb{R}^{N_i \times d}$
Ensure: Batches of shape $[B, N_{\max}, d]$ with normalized coordinates
for all (X_i, Y_i) in dataset **do**
    Downsample X_i to N' points
    Normalize X_i: subtract mean, divide by std
    Pad X_i and Y_i to $N_{\max}$ points
Custom `collate_fn` to batch padded samples

Algorithm 2 Model Structure for Semantic Segmentation with GeoPointNet

Require: Input tensor $x \in \mathbb{R}^{B \times N \times 6}$, where B is batch size, N is number of points
Ensure: Output tensor $y \in \mathbb{R}^{B \times N \times C}$, where C is number of classes
$x \leftarrow \text{permute}(x)$ ▷ Change shape to $[B, 6, N]$ for Conv1D
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, \text{in} = 6, \text{out} = 64, \text{kernel} = 1))$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 64 \to 128))$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 128 \to 256))$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 256 \to 512))$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 512 \to 256))$
$x \leftarrow \text{Dropout}(x, p = 0.3)$
$x \leftarrow \text{Conv1D}(x, 256 \to C)$
$y \leftarrow \text{permute}(x)$ ▷ Return to $[B, N, C]$
return y

Algorithm 3 Focal Loss for Multi-class Segmentation

Require: Predictions $P \in \mathbb{R}^{B \times C \times N}$, Targets $T \in \mathbb{N}^{B \times N}$, γ, α
Compute Cross-Entropy Loss: $L_{\text{CE}} = \text{CrossEntropy}(P, T)$
Compute $p_t = \exp(-L_{\text{CE}})$ ▷ probability of correct class
Compute Focal Loss: $L = (1 - p_t)^{\gamma} \cdot L_{\text{CE}}$
if reduction == "mean" **then**
    Return $\frac{1}{B} \sum L$
else
    Return L

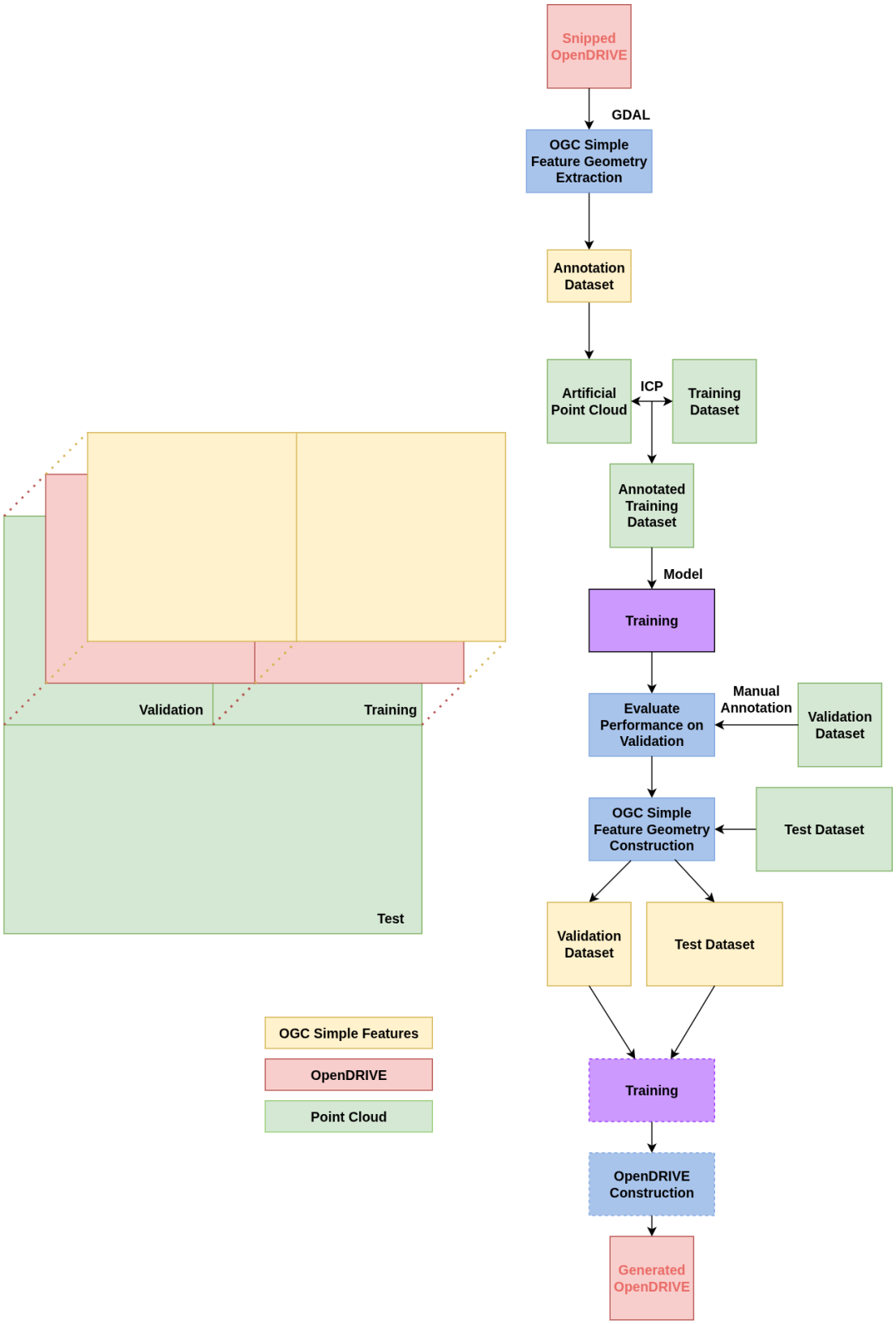


Figure 4.2: Overview for approach

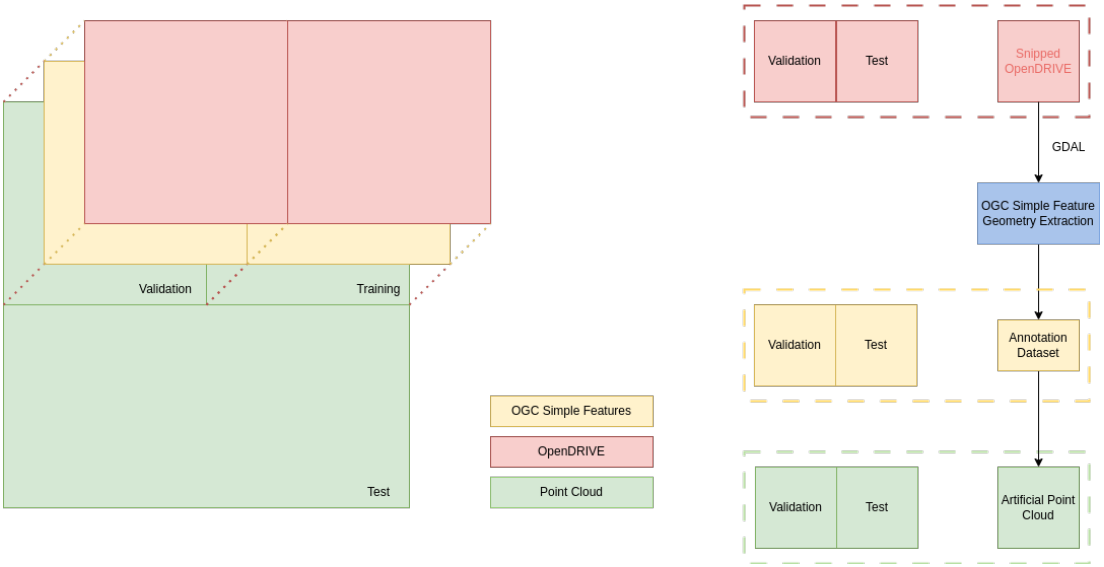
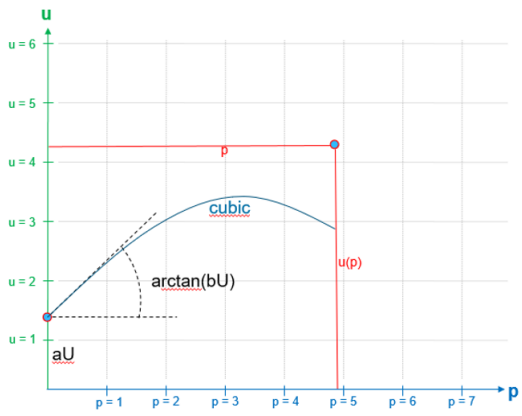
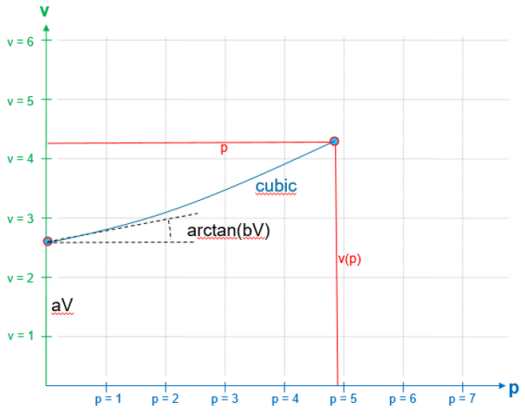


Figure 4.3: Process sub-diagram and simple area representation for automatic annotation



(a) Parametric cubic polynomial for interpolation of the u coordinate



(b) Parametric cubic polynomial for interpolation of the u coordinate

Figure 4.4: Interpolation using parametric cubic polynomials

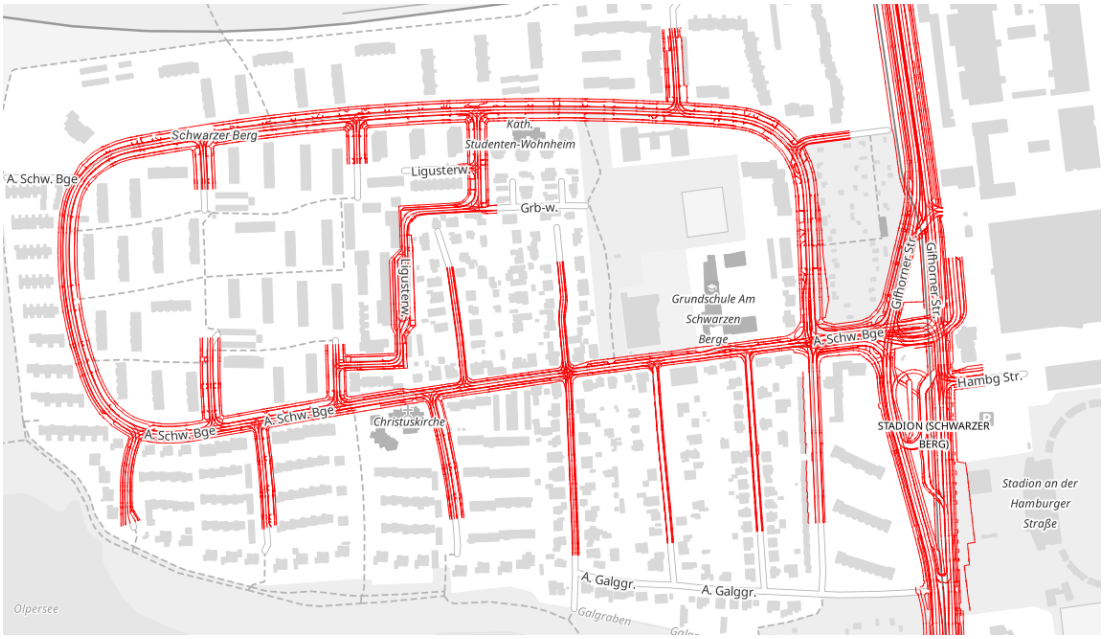


Figure 4.5: GDAL geometries of lane borders converted from OpenDRIVE; basemap © GeoBasis-DE/BKG 2025, CC BY 4.0

.

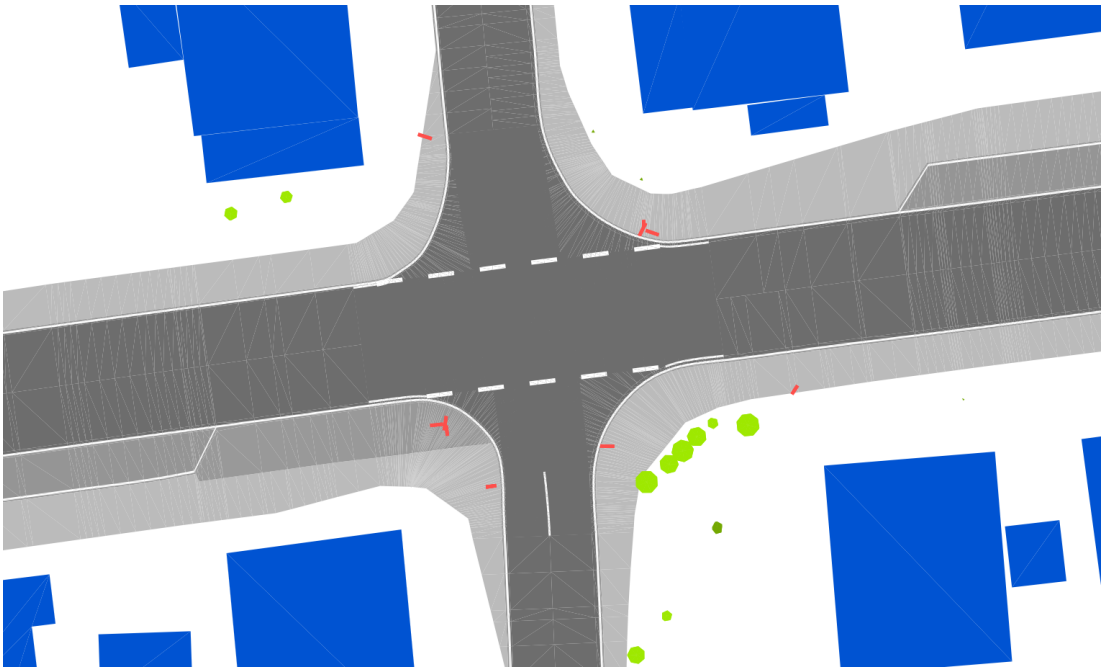


Figure 4.6: GDAL geometry details of lanes (gray), signals/signs (red), buildings (blue) and vegetation (green).

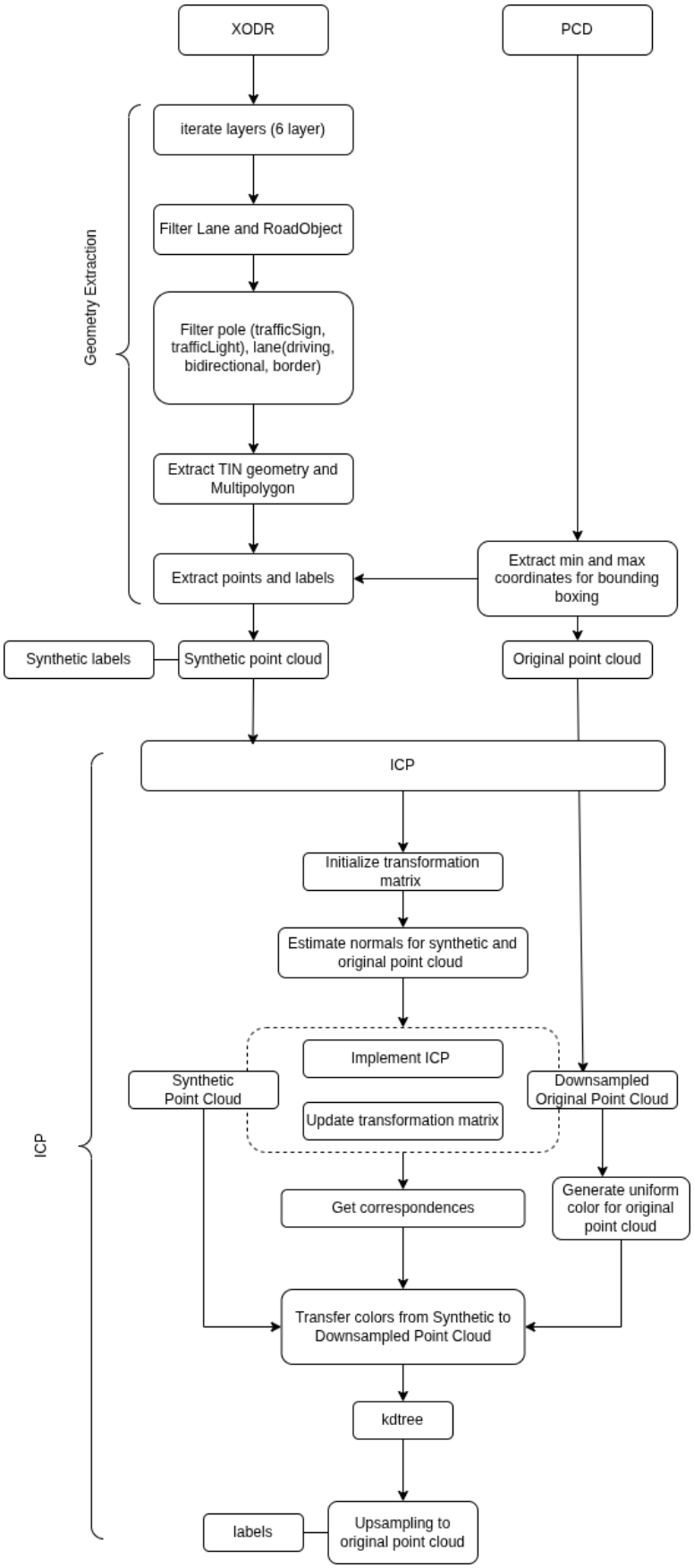


Figure 4.9: Algorithm flowchart for Iterative Closest Point in between synthetic and original point clouds.

Chapter 5

Experiments

As described in previous sections, we proposed a system that automatically labels sensor-collected original datasets using 3D synthetic point clouds generated from vector datasets using high-resolution maps in OpenDRIVE format. The motivation and overview are described in the Introduction section; the main techniques are described in the Basic Techniques section; and the methods and algorithms used are described in the Approach section. The results and evaluations obtained from this pipeline will be described in this section.

5.1 Synthetic Point Cloud Construction, ICP Integration and Automatic Annotation Results

As mentioned in Section 4, in this study we proposed a lightweight automatic annotation pipeline for point-wise semantic segmentation. To achieve this goal, we assumed the use of high-definition maps that already include detailed information about the surroundings, such as 3D coordinates and semantic classes. Therefore, synthetically generated point clouds serve as reference frames for getting original point cloud labels in this context. However, building a synthetic point cloud from a vector-based dataset has many challenges in terms of sampling/interpolation methodology. On this baseline, we explore the performance of three different sampling types in generating 3D synthetic point clouds. Figure 5.1 displays a snippet that serves as the point cloud reference or ground truth, providing an initial overview of the results used to construct a 3D representation. The subsequent figures Figure 5.2 for uniform sampling results, Figure 5.3 for random sampling results, and Figure 5.4 for normal distribution results illustrate the outcomes of each sampling method. At first glance, uniform and random sampling appear to

yield similar results, while the results from the normal distribution seem more distant from reality. Table 5.1 presents the calculated cloud-to-cloud distances among the three synthetic point clouds, with the standard deviation indicating that the Gaussian-distributed synthetic point cloud differs slightly from both the uniform and randomly sampled point clouds. This calculation and visualizations made on CloudCompare[6] which is an open-source 3D point cloud and mesh processing software.

Sampling	Gaussian	Random	Uniform
Gaussian	0.0	0.0518 / 0.066	0.0518 / 0.066
Random	0.0518 / 0.066	0.0	0.0524 / 0.042
Uniform	0.0518 / 0.066	0.0524 / 0.042	0.0

Table 5.1: Cloud-to-cloud mean distances and standard deviations between raw point clouds in meters.

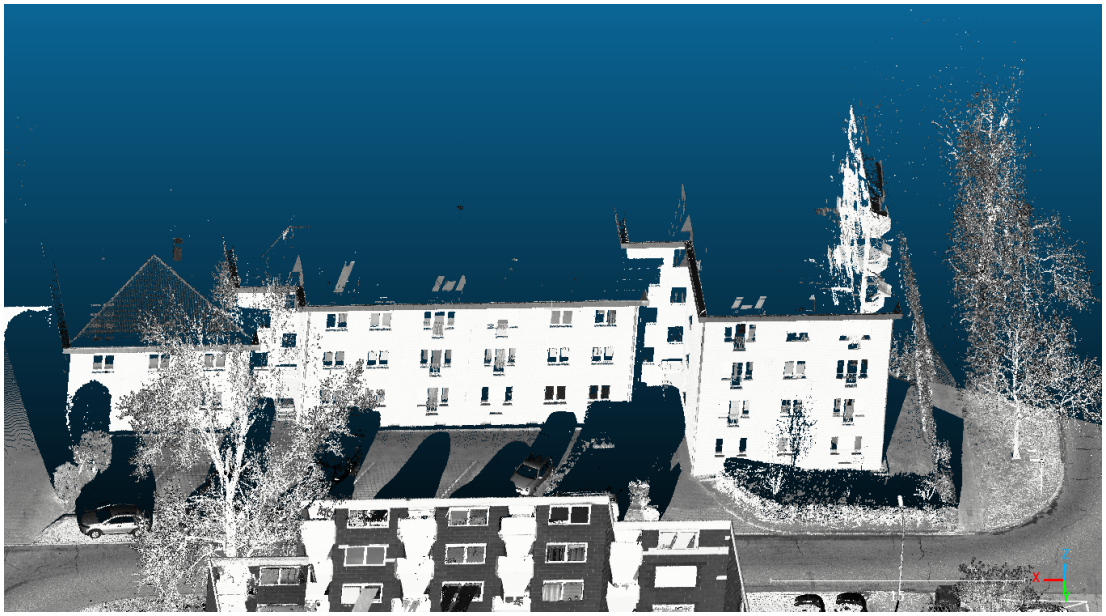


Figure 5.1: Reference point cloud of a building.

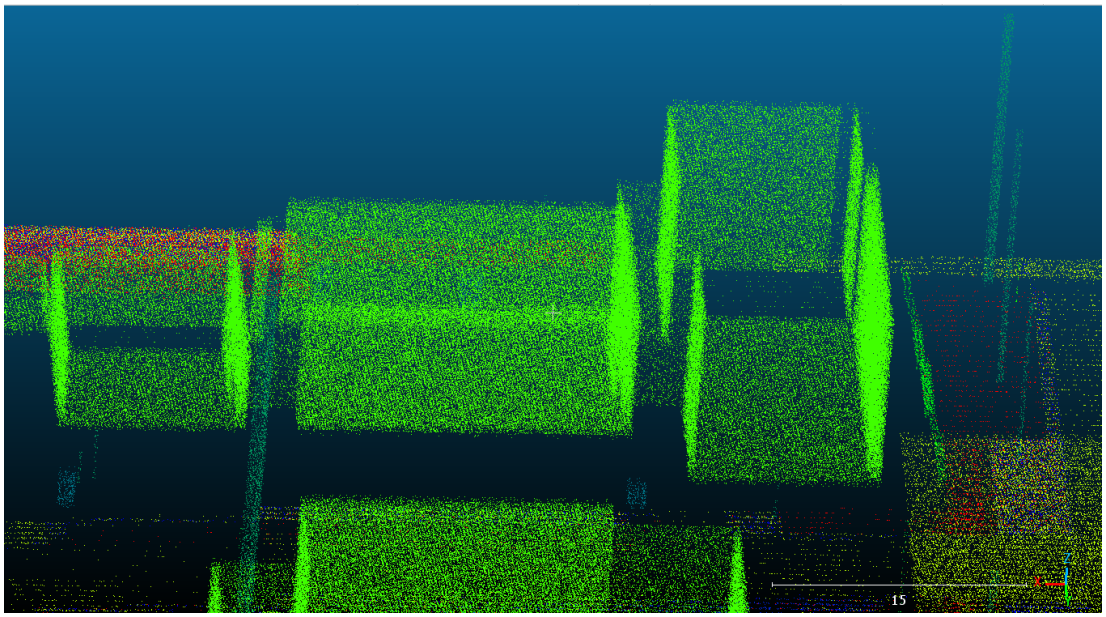


Figure 5.2: Synthetic point cloud generated through uniform sampling.

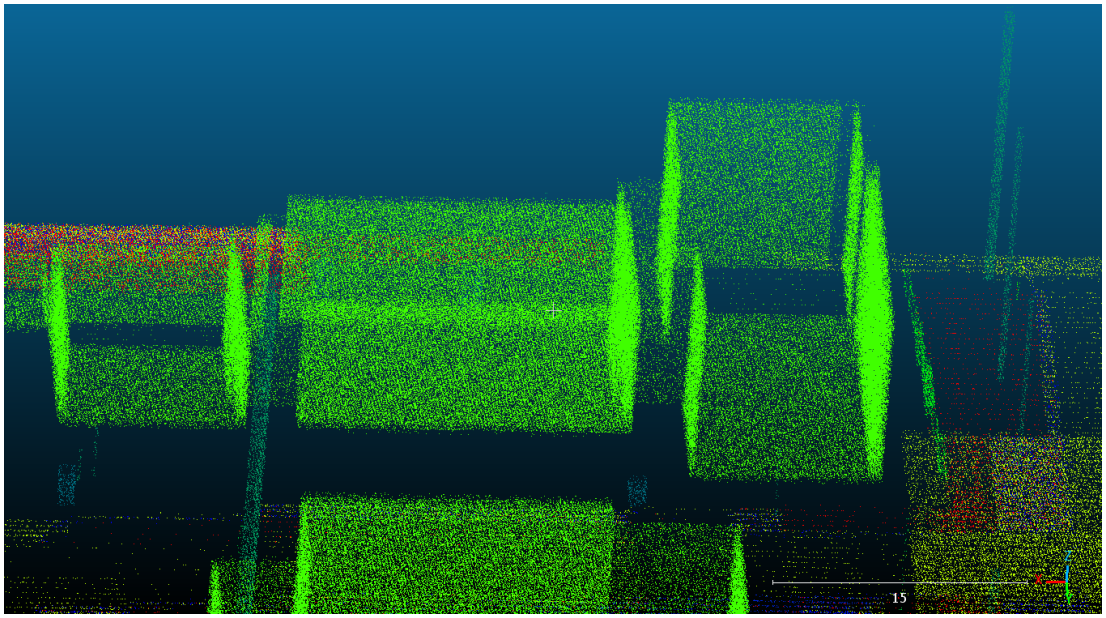


Figure 5.3: Synthetic point cloud generated through random sampling.

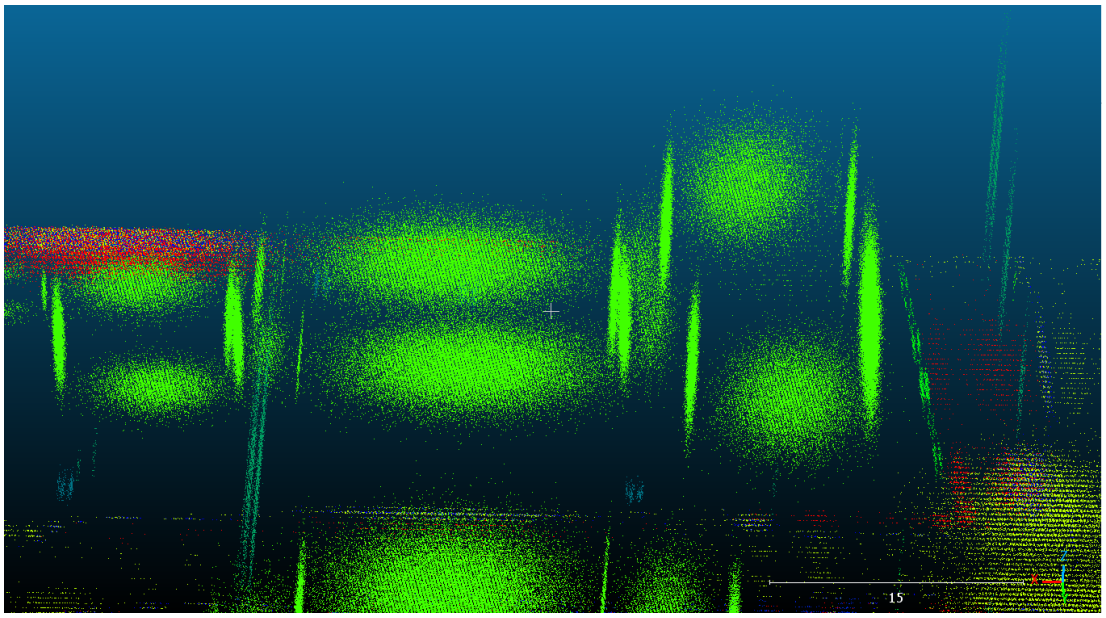


Figure 5.4: Synthetic point cloud generated through normal sampling.

The Iterative Closest Point Algorithm (ICP) algorithm was utilized to quantitatively measure the performance of three different sampling methods. Synthetic point clouds were produced utilizing uniform, random, and normal sampling methods as source datasets, alongside the reference LiDAR-acquired and georeferenced point cloud as the target dataset. The ICP registration algorithm subsequently processed these point clouds, utilizing the settings specified in Table 4.1. Considering that synthetic point clouds generally exhibit a lower point density than the reference point cloud, downsampling is applied to the reference point cloud to handle this issue. The maximum number of nearest neighbor parameters indicates how many points will be examined for each point to search the corresponding points located within the radius specified by the KDTree parameter. In our case, the radius is 0.2 meters and the nearest neighbor number is 30, as shown in Table 4.1. This configuration is selected to enable a feasible downsampling approach, ensuring a balance between preserving geometric detail and reducing computational complexity. The initial guess transformation involves initializing a 4x4 transformation matrix that aligns the source and target point clouds. In our approach, we used an identity matrix for this purpose. The term ”iteration number” refers to the process of adjusting the transformation, which will undergo a maximum number of iterations to identify the best correspondences and achieve the optimal result. We set it as 10 to reduce computational consumption; however, a larger number of iterations could yield a better result when hardware provides the required performance in a discrete time window. As can be observed in Table 5.2, those three different datasets have differen-

t numbers of point densities in the explained parameter set; normal sampling has slightly denser points than the others because this type of sampling has the attitude to interpolate more precisely on edges, corners, and curvy areas. The initial approach focuses on the prevalence of flat surfaces, leading us to compare the root mean square error (RMSE) and fitness values of uniform versus random sampling. Fitness is defined as the number of points that align with the reference point cloud, while RMSE measures the extent of overlap between the predicted results and the ground truth. As seen in the table, they have almost similar values, but uniform sampling has slightly better results than random sampling. So we will proceed with uniform sampling to make further analyses.

Sampling Type	**Number of Points**	**RMSE**	**Fitness**
Normal	5,143,308	1.09767	0.99923
Random	5,123,088	1.05360	0.99921
Uniform	5,123,424	1.05346	0.99922

Table 5.2: Different sampling methods' performance for ICP registration with responsible parameters in Table 4.1; the reference point cloud contains 54,350,752 points.

Figure 5.5 illustrates a scene with buildings, parking spaces, roads, vegetation, and trees. In Figure 5.6, the reference point cloud sample that was collected using mobile mapping shows the same region as the orthophoto area. The data map, which is used for transferring labels to the actual point cloud, contains 13 known classes: road mark, parking, obstacle, vegetation, tree, streetlamp, barrier, traffic sign, traffic light, pole, building, driving area, and restricted-to-driving areas; and 1 unknown class that contains points that cannot be defined exactly or are not relevant to our approach. Besides 3D coordinates, those data maps are derived from OpenDRIVE files.

Figure 5.5: Orthophoto of the scene of interest; © GeoBasis-DE/LGLN 2025, CC BY 4.0

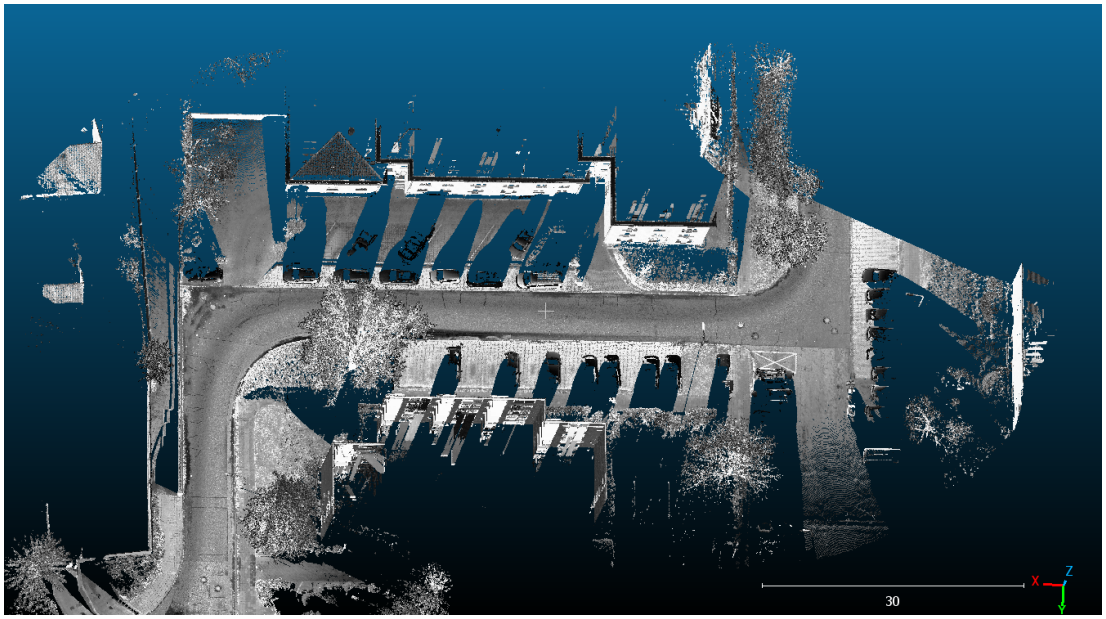


Figure 5.6: Reference point cloud for the same region of the orthophoto in Figure 5.5.

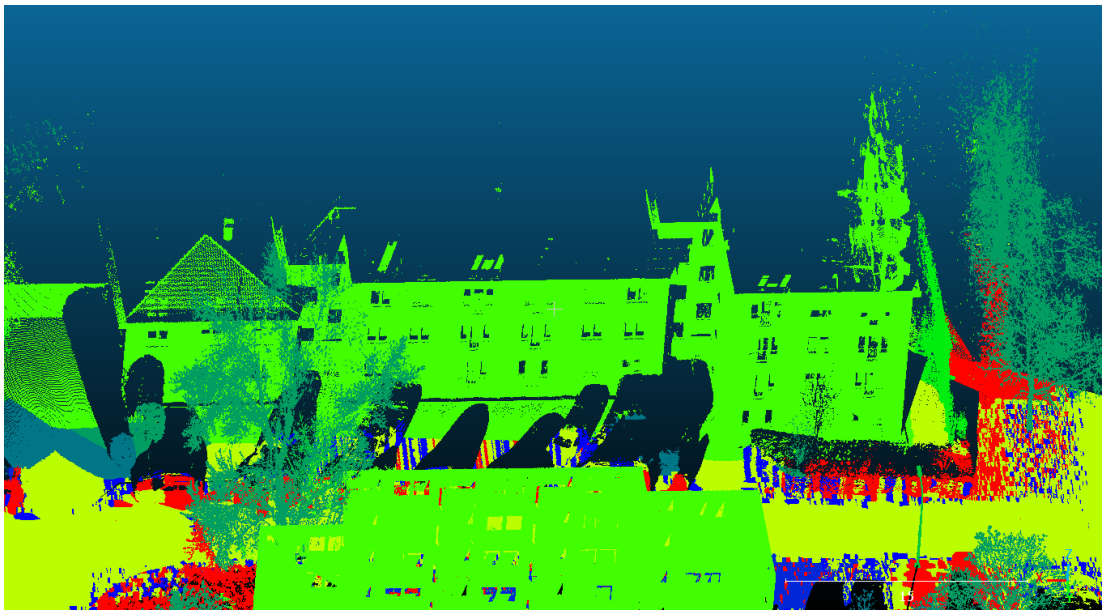


Figure 5.7: Transferred labels' result for reference point cloud by using ICP on uniform sampled synthetic point cloud. (Responsible region in Figure 5.5, Figure 5.6).

The subsequent phase involves completing the upsampling process through a nearest neighbor search utilizing KDTree, followed by the identification of corresponding points between the point clouds aligned with the transformation matrix derived from ICP. This step is critical for guaranteeing accurate alignment and maintaining the integrity of the data. Once the corresponding points are identified, further refinement can be applied to enhance the registration process and minimize any residual errors in the point cloud alignment. As shown in Table 5.2, alignment has reliable results with approximately 0.99 fitness value and 1.053 m RMSE. As the downsampling and upsampling ratios are identical, we acquire an accurately scaled, annotated point cloud. The annotated point cloud is depicted as a segmented representation of the scene illustrated in Figure 5.5, as demonstrated in Figure 5.7. Structures, vegetation, driving and non-driving areas are categorized under distinct classifications. In details of specific objects, the KDTree-based label transfer algorithm operates with high precision and provides accurate annotations if a point on an object, such as a building, in sparse regions lacks neighboring points at that elevation. Figure 5.9 shows an example of how well objects are identified in areas with few points; this scene comes from the area shown in Figure 5.8, where both trees and buildings were correctly identified without any over- or under-segmentation.

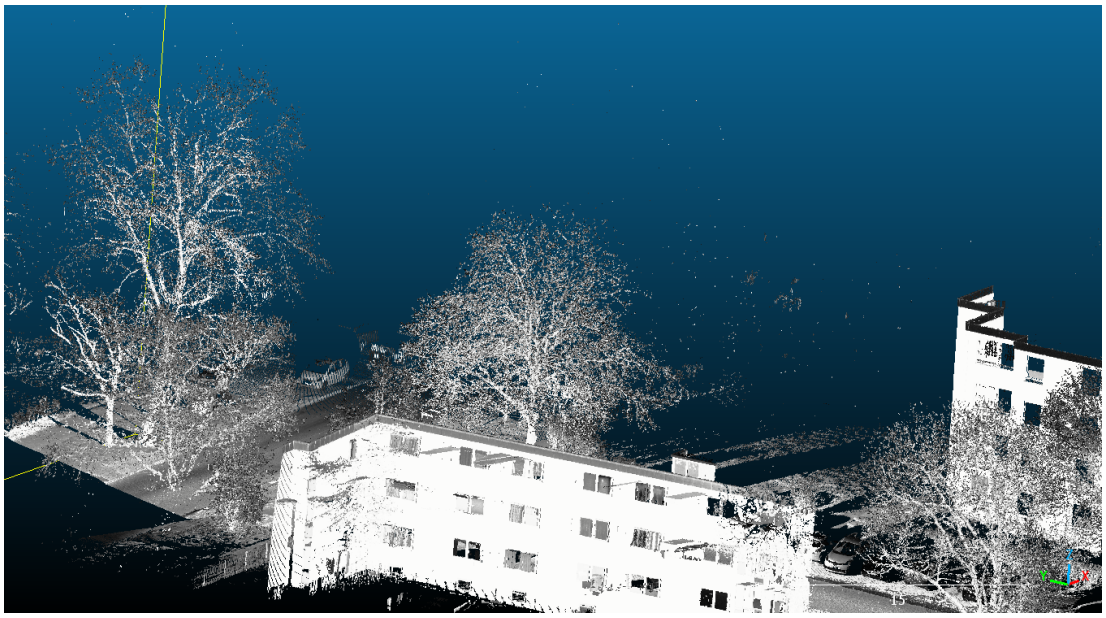


Figure 5.8: Sample snipped for tree and building representation.

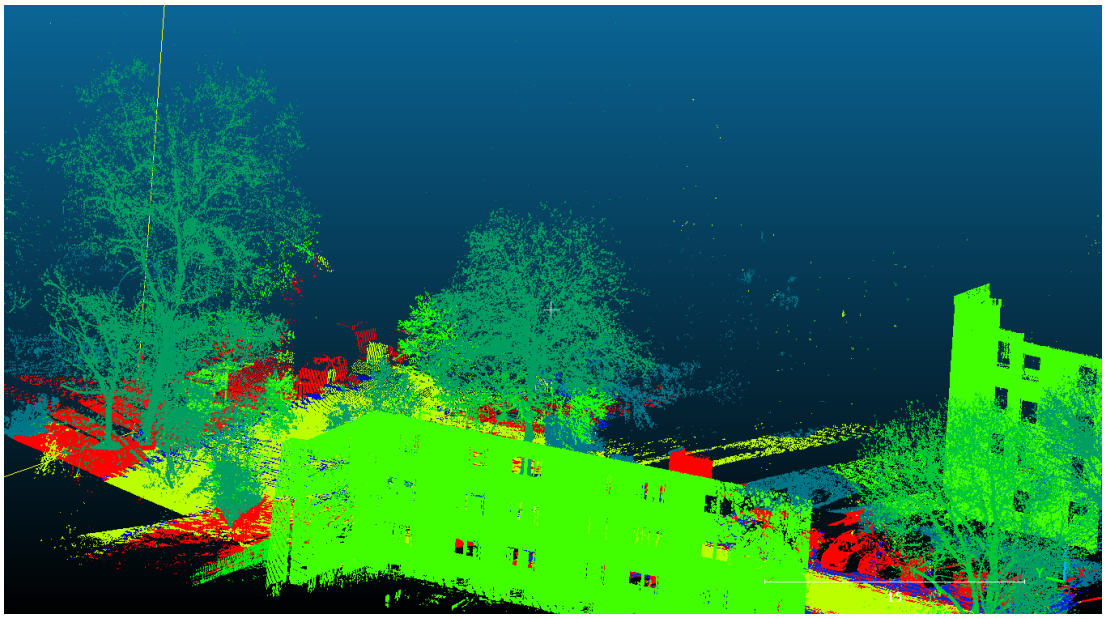


Figure 5.9: Labeled sample snipped for tree and building representation.

Upon analyzing ground points that encompass the road surfaces, the algorithm's performance deteriorates qualitatively due to the clustering of proximate objects within the same area. Figures 5.10 and Figure 5.11 depict the same area, which includes parking and driving spaces, with the former represented as an orthophoto and the latter as a point cloud. Figure 5.12 shows the upsampled point cloud is over-segmented for the same region. Because attributes are closely

located, aggregated, and overlapping at the ground level, the algorithm should be improved with different strategies. As shown in Figure 5.13, the investigated parking area is framed by a road mark; for that reason, the borders have confusion on the transferring label stage. Due to the modeling habit of our OpenDRIVE test dataset, the feature ”parking space” has two separate indexes, one in the Lane layer and one in the RoadObject layer. To solve this confusing data modeling, mapping can be modified with the same label for both, and a corrected version of labeling can be seen in Figure 5.14. The way ground points are grouped has led to ongoing problems with too many or too few sections, which can't be fixed just by changing the data mapping during labeling. A targeted approach should be employed to resolve these issues. In the following section, this issue will be pointed out in terms of remapping of data mapping to enhance model solution.

Figure 5.10: Orthophoto of the road surface; © GeoBasis-DE/LGLN 2025, CC BY 4.0

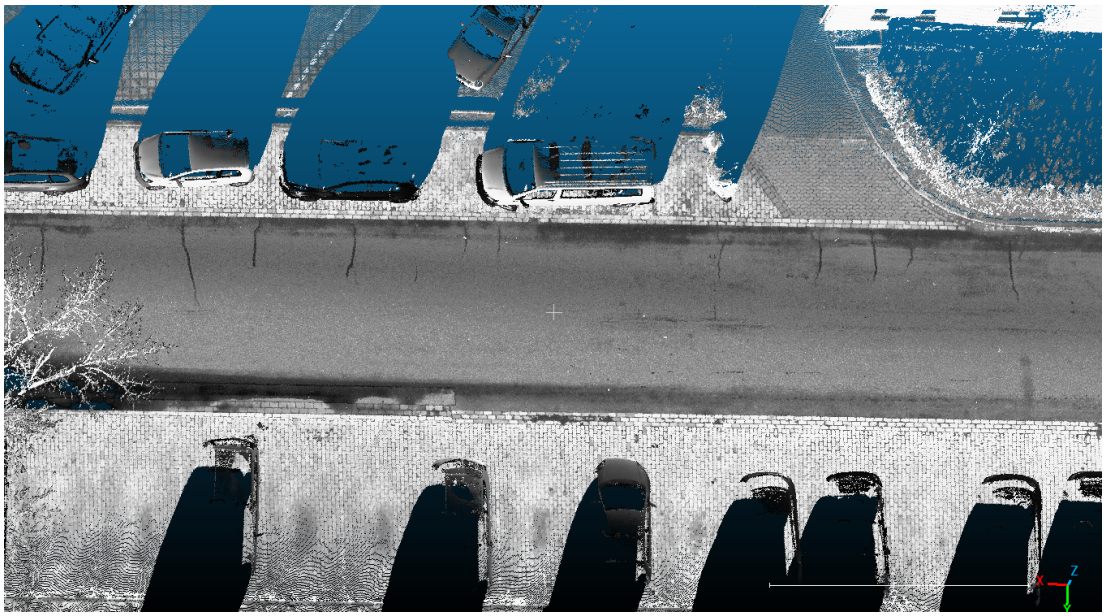


Figure 5.11: Reference point cloud for road surface.

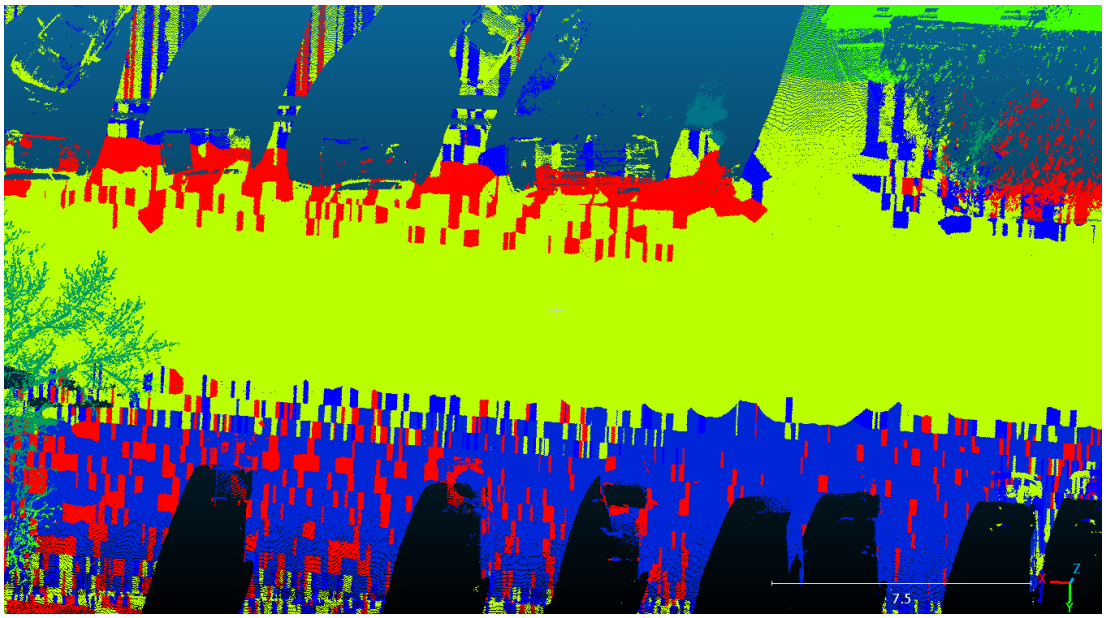


Figure 5.12: Labeled point cloud 3D model generated from XODR. The light green area represents driving areas, blue points are parking spaces, and red points are road marks that cover the parking space.

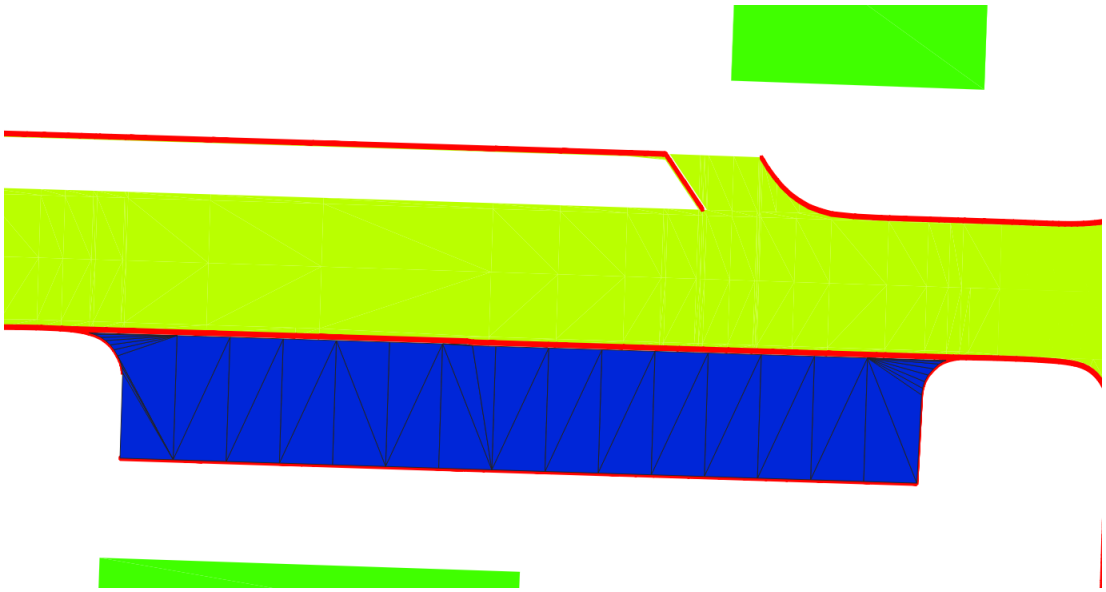


Figure 5.13: XODR representation where road marks (red) are bordering the parking space (blue), which is the reason for the noisy point labeling in Figure 5.12.

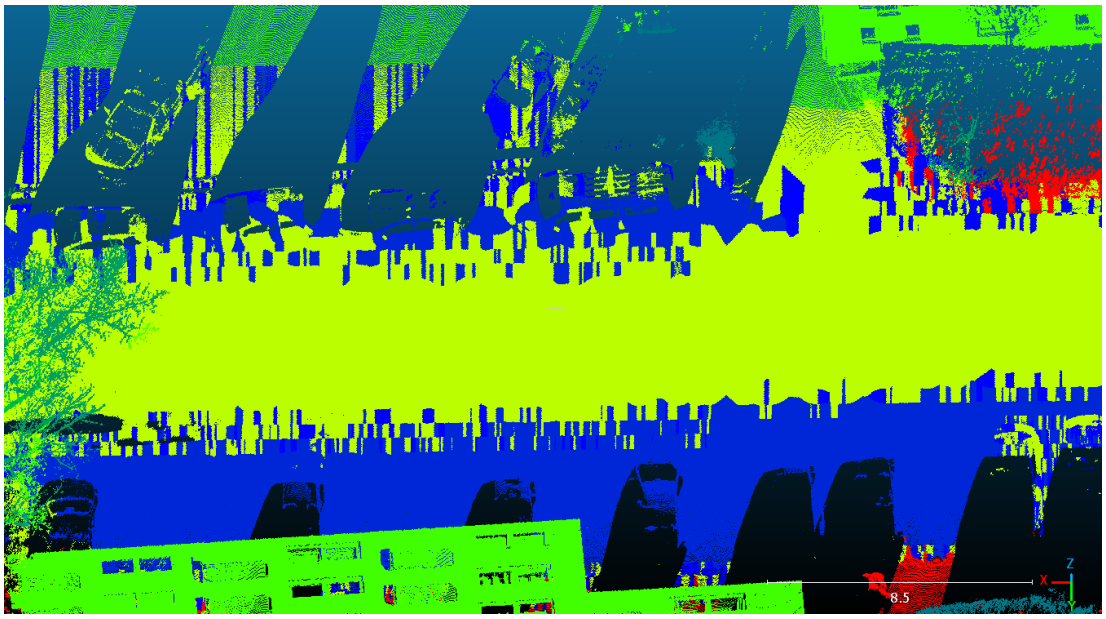


Figure 5.14: Labeled point cloud 3D model generated from XODR with corrected mapping. The light green area represents driving areas, and the blue points are parking spaces.

As described in the previous section, synthetic point cloud construction was done for the responsible area, and label transfer was completed by using the iterative closest point algorithm. All possible labels that extracted from OpenDRIVE file are represented in Table 5.5 Figure 5.15 and Figure 5.17 show how a synthetically generated point cloud represents an object in overview and detail.

Figure 5.16 and Figure 5.18 illustrate the success of the ICP transfer algorithm in label transfer, providing both an overview and detailed views of a specific area. In these scenes, it is evident that buildings, trees, street lamps, and other types of objects can be labeled properly. Therefore, the next step would be to build a network and train a model to predict a non-manually annotated dataset in this way. The following part will investigate how a model can be structured and how it affects qualitative and quantitative results.

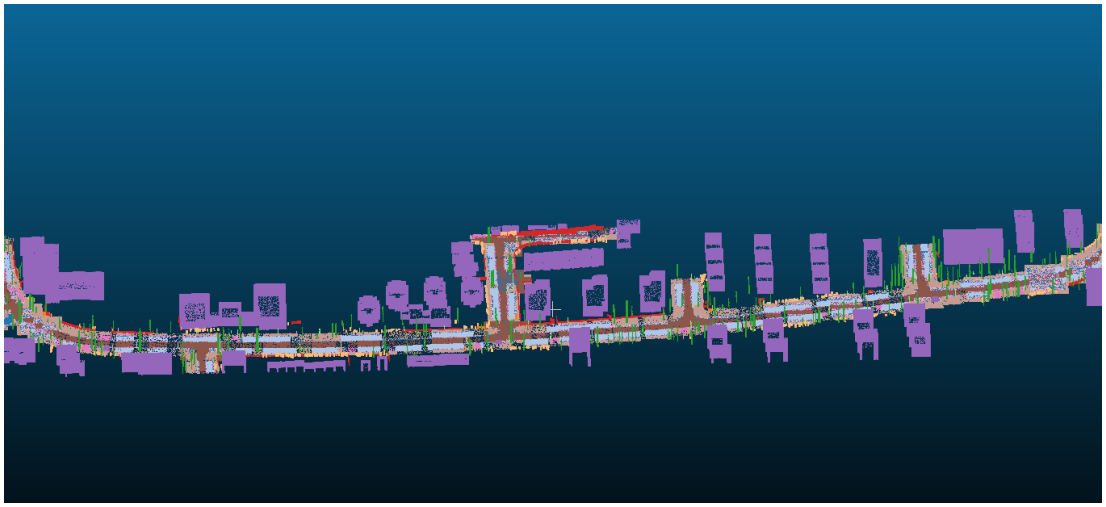


Figure 5.15: Overview for labelled synthetically generated point cloud through the OpenDRIVE file.

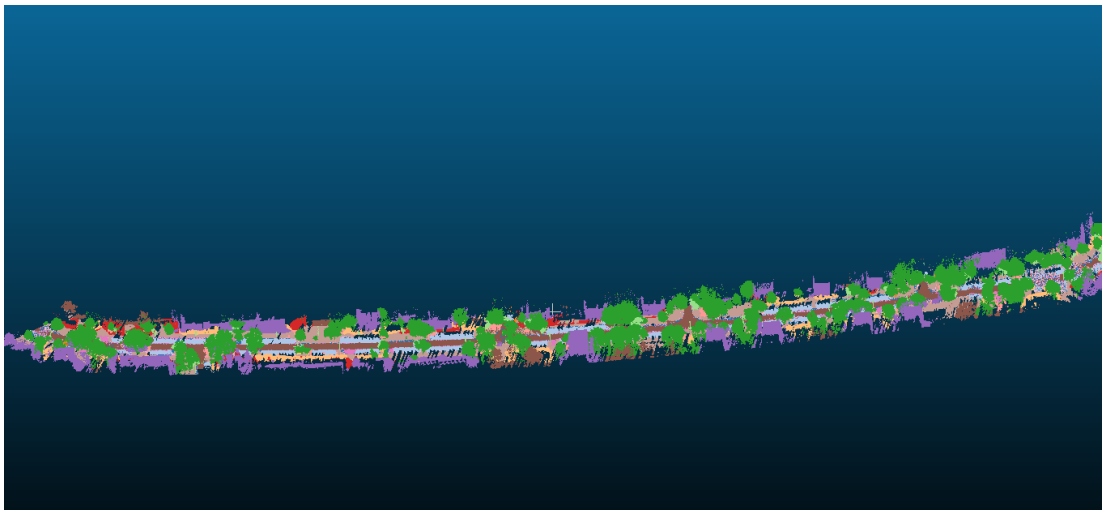


Figure 5.16: Overview for transfered labels on original point cloud responsible area with Figure 5.15

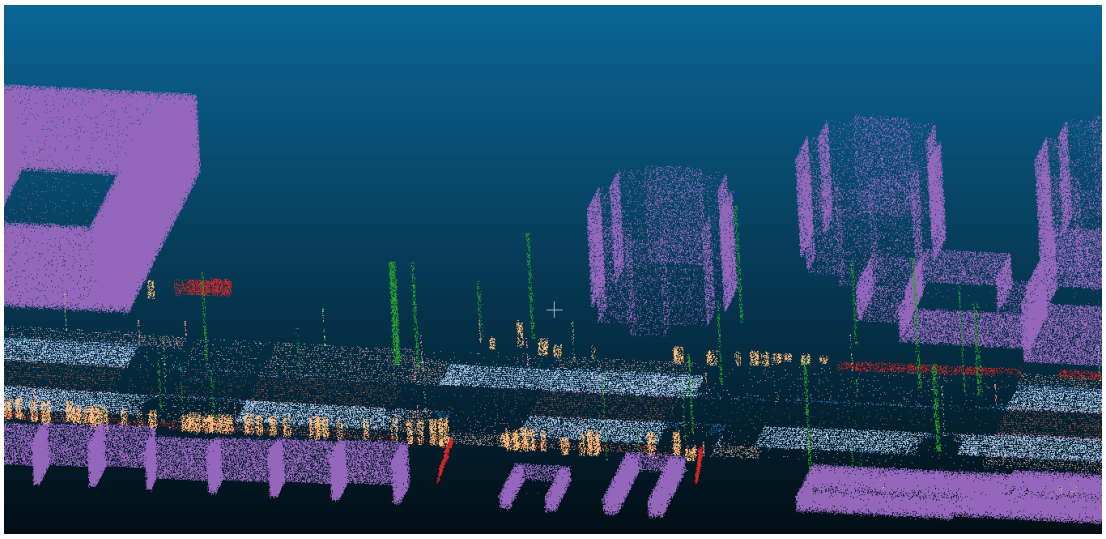


Figure 5.17: Details for labelled synthetically generated point cloud through the OpenDRIVE file.

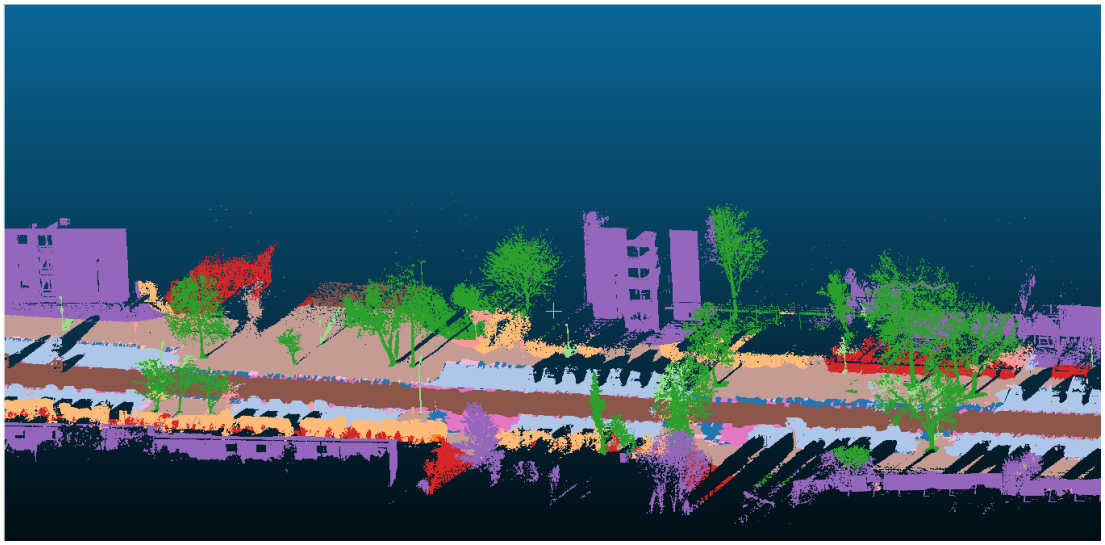


Figure 5.18: Details for transfered labels on original point cloud responsible area with Figure 5.17

Table 5.3: RMSE and Fitness values for synthetic-to-original point cloud alignment using ICP algorithm across multiple scenes.

Scene ID	**RMSE**	**Fitness**
Scene 1	0.8463	0.9991
Scene 2	1.0869	0.9980
Scene 3	0.9976	0.9993
Scene 4	0.9538	0.9998
Scene 5	0.9447	0.9993

Table 5.4: Per-class RMSE across different scenes (lower is better).

Class ID	**Scene 1**	**Scene 2**	**Scene 3**	**Scene 4**	**Scene 5**
0 (roadmark)	0.3690	0.4377	0.3976	0.4034	0.4239
1 (parking)	0.4807	0.5554	0.6902	0.5066	0.4747
2 (obstacle)	2.2511	1.2092	1.7860	1.8797	0.8961
3 (vegetation)	2.2061	1.7354	0.6172	1.0551	1.4512
4 (tree)	2.5747	2.9946	2.6489	3.1606	2.9055
5 (streetLamp)	3.0969	3.2220	2.1034	3.2438	1.9056
6 (barrier)	0.7497	2.4770	0.6950	0.9181	1.1032
7 (trafficSign)	2.3129	3.3498	2.4035	3.5884	2.1113
8 (building)	1.6301	1.3385	1.1984	1.4366	1.0760
9 (pole)	1.2995	1.9176	3.9575	2.9959	3.2248
10 (trafficLight)	1.7831	—	1.9500	—	1.6970
11 (driving)	0.4015	0.4261	0.3437	0.3374	0.4390
19 (restricted)	1.2363	1.0447	1.1028	0.8139	1.3886
21 (border)	0.2655	0.3369	0.2386	0.2938	0.3773
22 (curb)	0.6421	0.4985	0.3574	0.4427	0.7582

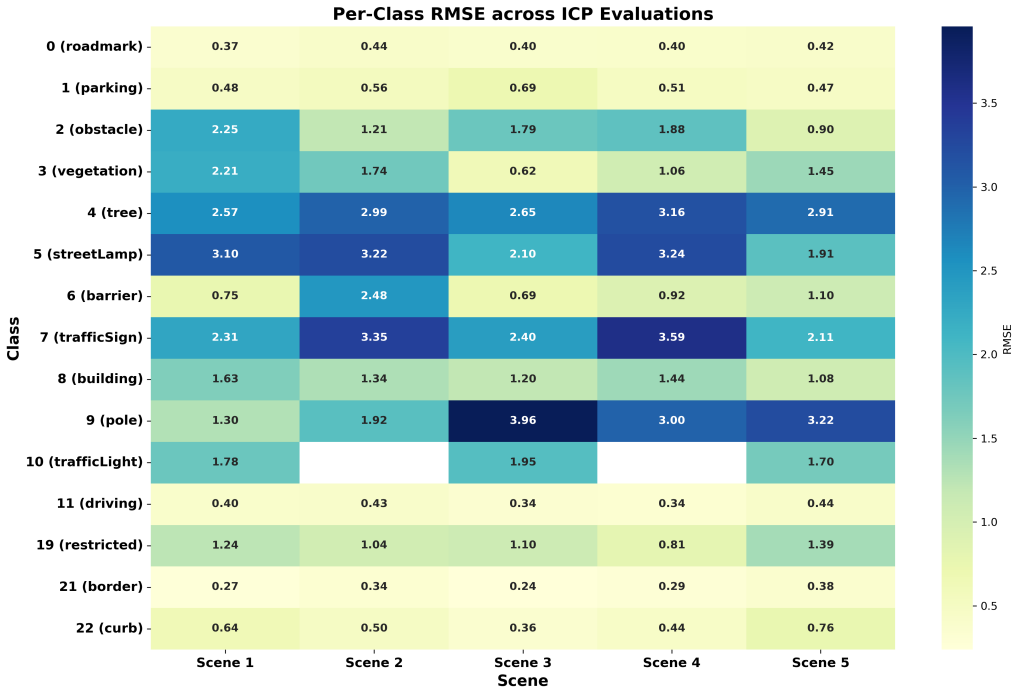


Figure 5.19: Heatmap showing per-class RMSE for synthetic-to-original alignment using ICP across 5 scenes.

Table 5.5: Label mapping from the original 22 synthetic point cloud classes to the 10 simplified model classes. Labels not used in the simplified set are marked as 'Not used'.

Original ID	Class Name	Usage on Model
-1	unknown	Not used
0	roadmark	Not used
1	parking	Not used
2	obstacle	1
3	vegetation	2
4	tree	3
5	streetLamp	4
6	barrier	5
7	trafficSign	6
8	building	7
9	pole	8
10	trafficLight	9
11	driving	Not used
19	restricted	Not used
20	bidirectional	Not used
21	border	Not used
22	curb	Not used

5.2 Supervised Semantic Segmentation with Automatic Annotation

As shown in Table 5.5, a subset of 10 classes was selected from the full set of 22 for model training to facilitate the detection of certain easily recognizable objects, such as trees, buildings, traffic lights, and signs. This approach allows us to evaluate our model from both quantitative and qualitative perspectives. Future refinement will cover the multiple classes learning to construct proper and accurate map construction. Details regarding to future refinements will discussed in Section 6: Discussion and Future Work. After enhancing the label transferring system using synthetic point cloud creation and the iterative closest point method, the pointwise semantic segmentation model was used on the labeled original point cloud, as explained in the approach section. The GeoPointNet architecture Algorithm 2 was trained with 4 different batch sizes for 400 epochs using the focal loss algorithm Algorithm 3, coordinate normalization, and a method to reduce the amount of data. As shown in Table 5.6 and Figure 5.35, class imbalance and geometric differences are the reasons for the diverse metric distribution. Tables show a huge number of points affects model performance; however, it is not the only reason for lower accuracy. Even if the Street Lamp class has a relatively higher number of points than several classes, precision is comparatively low. This behavior can be interpreted as surface distribution because normals are also an input for the model, which is another key factor for precision when we use only coordinate space-considered models. Flat and continuous objects like barriers and buildings have relatively higher precision.

Table 5.6: Classification report for the 10-class model on the synthetic point cloud dataset (excluding ignored index).

ID	Class Name	Precision	Recall	F1-score	Support
1	obstacle	0.2041	0.7574	0.3216	2477
2	vegetation	0.4782	0.7436	0.5821	10927
3	tree	0.8525	0.3346	0.4806	45581
4	streetLamp	0.2785	0.4436	0.3422	10831
5	barrier	0.7689	0.6890	0.7267	25809
6	trafficSign	0.1983	0.5003	0.2841	4301
7	building	0.7992	0.6390	0.7102	29679
8	pole	0.2191	0.6526	0.3280	2504
9	trafficLight	0.2448	0.7218	0.3656	2883
Accuracy		0.5383			
Macro Avg		0.4493	0.6091	0.4601	-
Weighted Avg		0.6910	0.5383	0.5608	134992

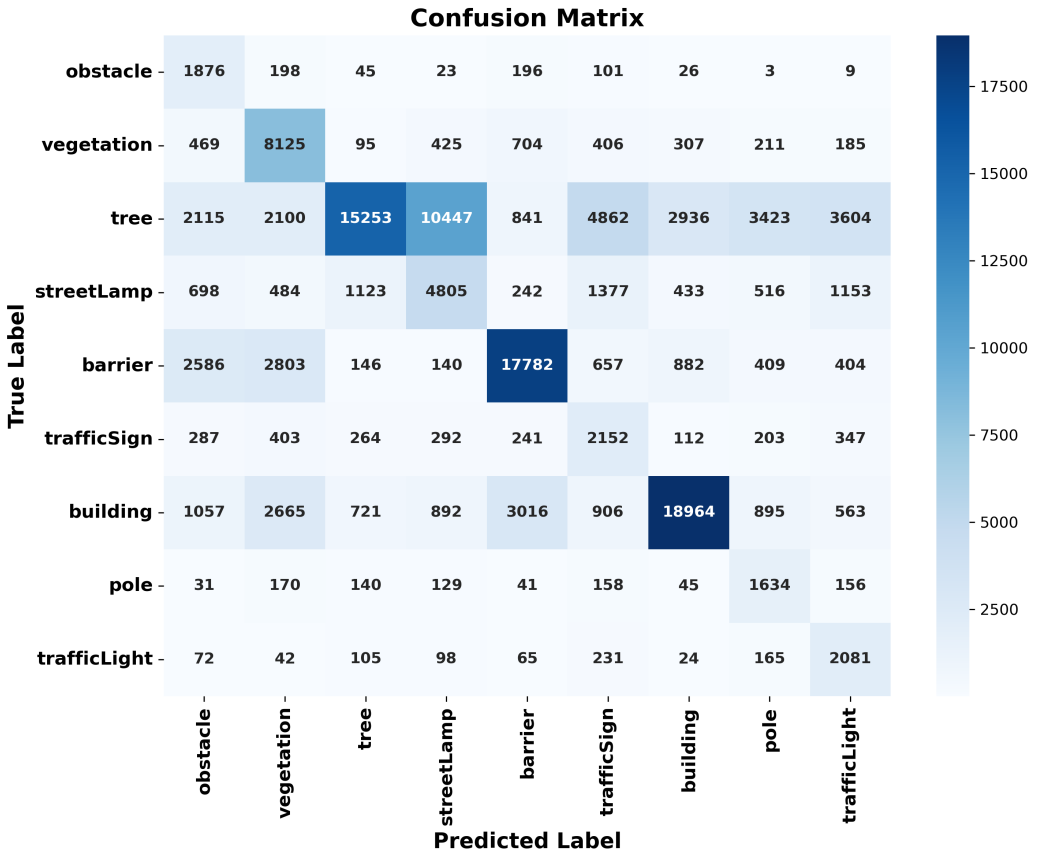


Figure 5.20: Heatmap showing per-class RMSE for synthetic-to-original alignment using ICP across 5 scenes.

Figure5.21, Figure5.22, and Figure 5.23 show snapshots of model performance in predicting the classes of points belonging to the building. Due to downsampling for the training and validation steps, the point cloud appears in a simplified version. Figure5.21 displays sensor-collected point clouds without intensity values. Figure 5.22 illustrates the performance of label transfers on this building following downsampling. Figure 5.23 presents the predicted classes for this building. As previously mentioned, the building contains a vast number of points, which indicates that This type of object is at the beginning of the long-tail problem and has certain advantages; in addition, it features flat and continuous geometries. GeoPointNet is an effective network for predicting point-wise semantic classes for this type of object.

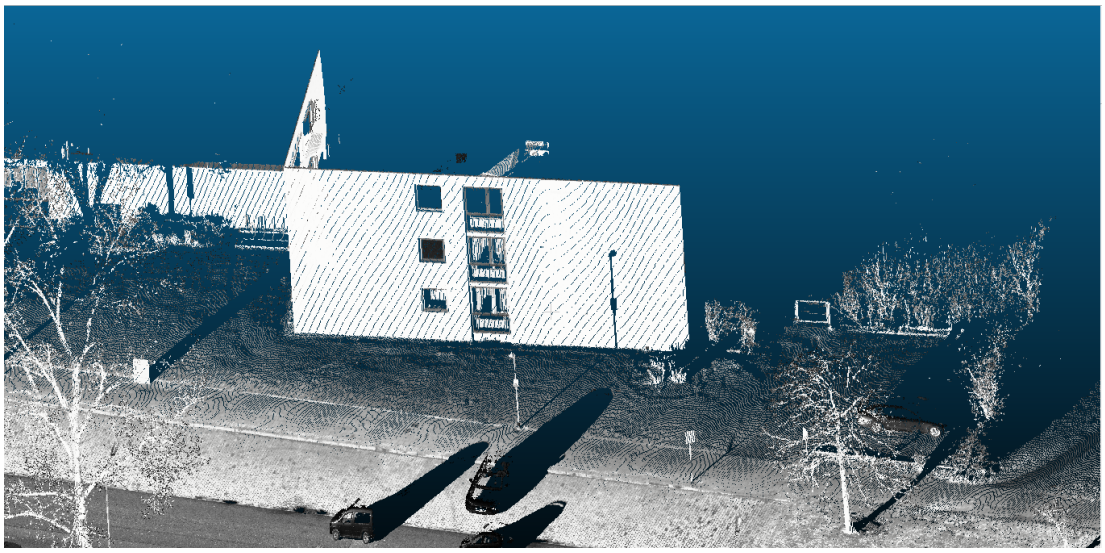


Figure 5.21: Original, unlabeled point cloud sample for building.

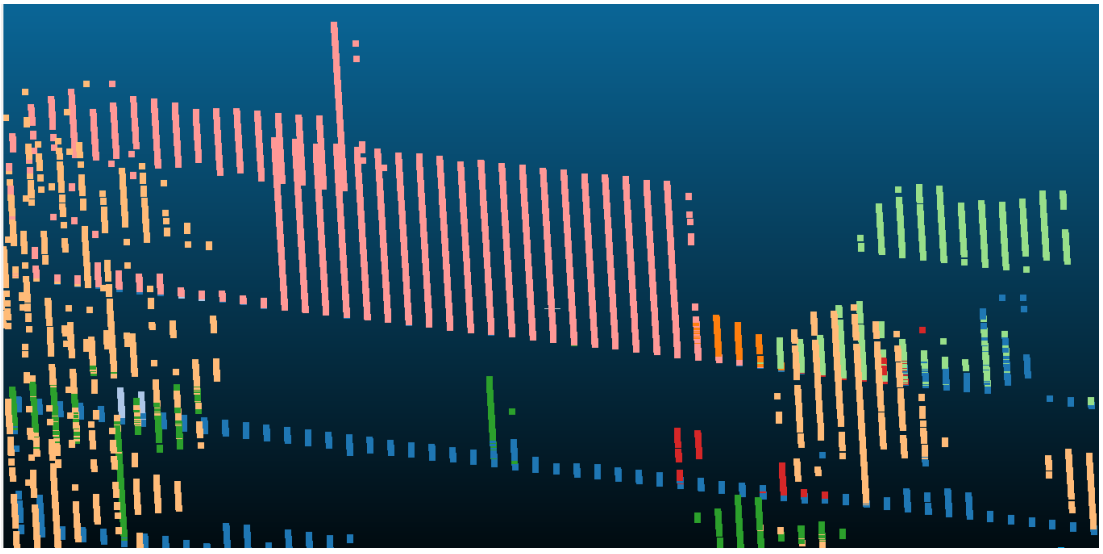


Figure 5.22: Ground truth - transfered labels from ICP-stage for responsible building in Figure 5.21

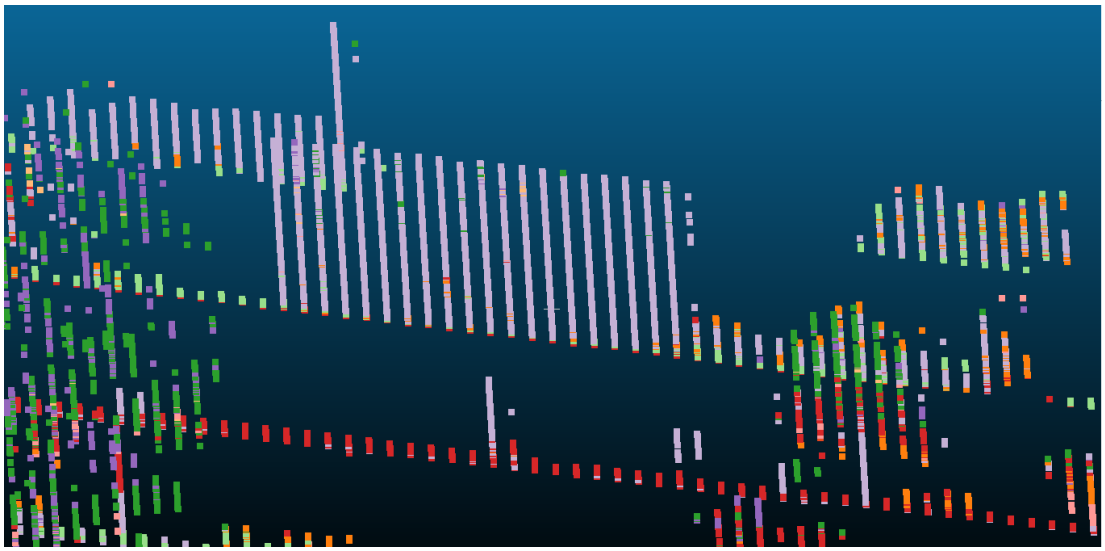


Figure 5.23: Predicted labels for responsible building in Figure 5.21 and 5.22

The following figures illustrate the validation area used for evaluation. Figure 5.24 presents the original point cloud dataset, offering a baseline visualization of the raw data. In contrast, Figure 5.25 depicts the output of the label transferring stage, which includes downsampling to align the labels with the dataset resolution. Figure 5.26 illustrates how the model performs on semantic predictions. Figure 5.27 represents how the model predicts correctly for semantic classes; green points are correctly predicted points, and red points are misalignments. As shown in Figure 5.27 and Table 5.6, the model performs with 53% accuracy, and a significant amount of correct precision is found in flat objects like buildings, barriers, or dominated classes like trees. The following figures illustrate this situation.

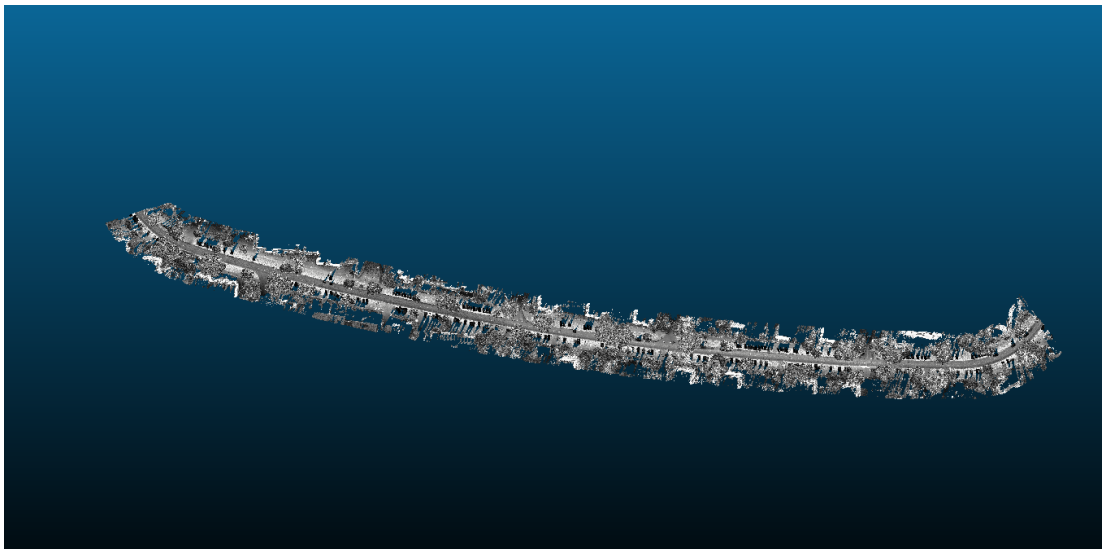


Figure 5.24: Original point cloud representation of the validation area. This visualization shows the raw 3D structure without any semantic annotations.

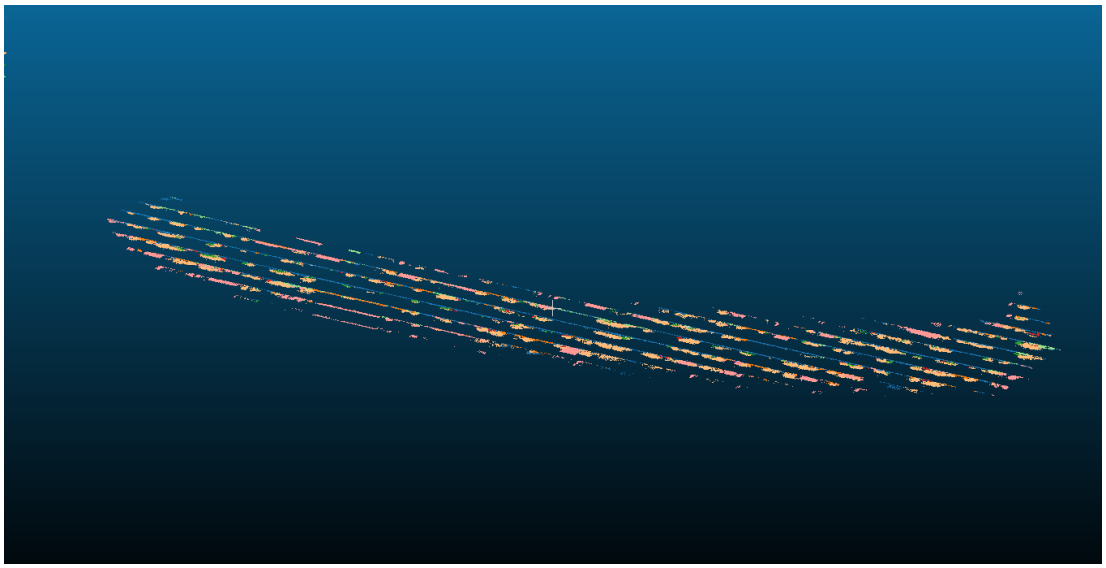


Figure 5.25: Ground truth semantic labels projected onto the point cloud after the label transfer and downsampling process.

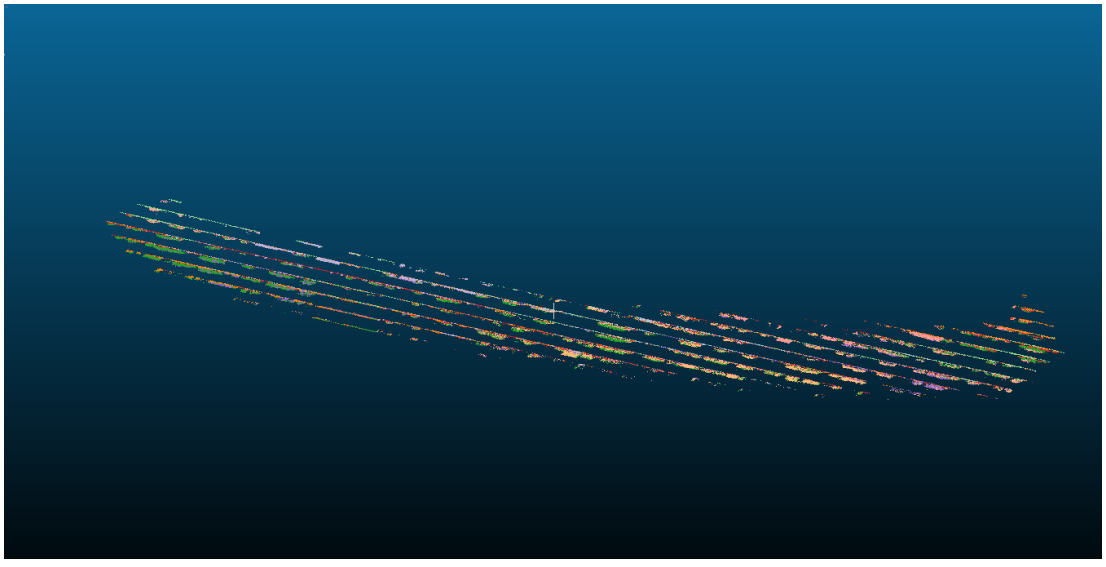


Figure 5.26: Semantic predictions produced by the model for the corresponding validation area. Each point is colored according to its predicted class.

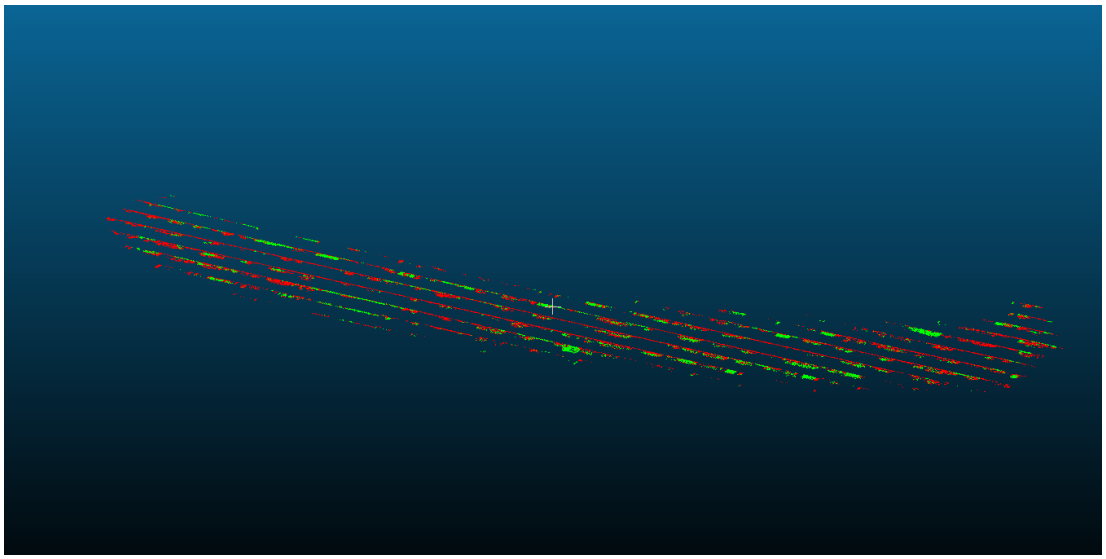


Figure 5.27: Comparison between predicted and ground truth semantic labels. Green points represent correctly predicted classes, while red points indicate misclassified ones.

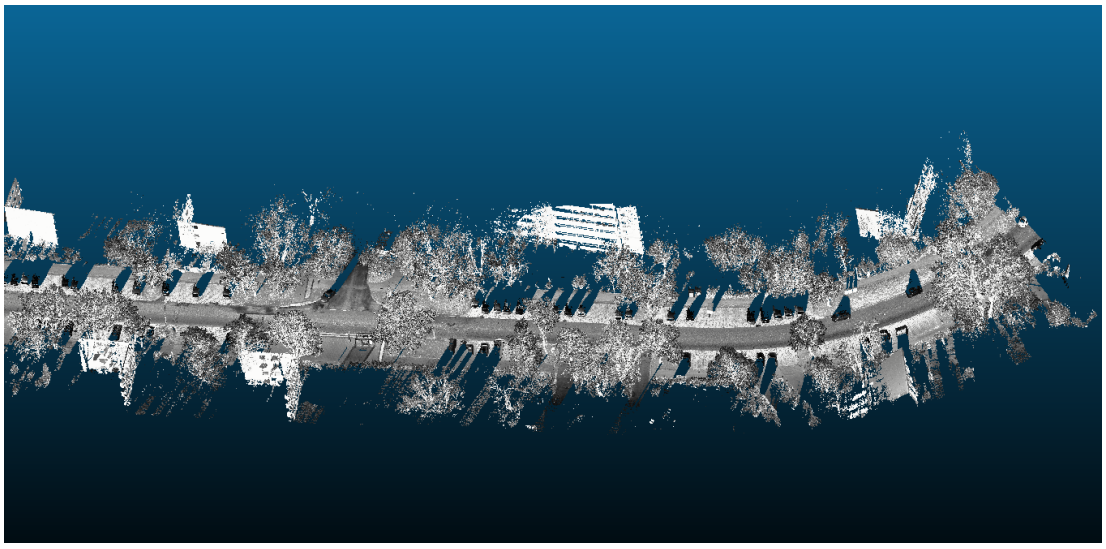


Figure 5.28: Original point cloud of the building area without semantic annotations.

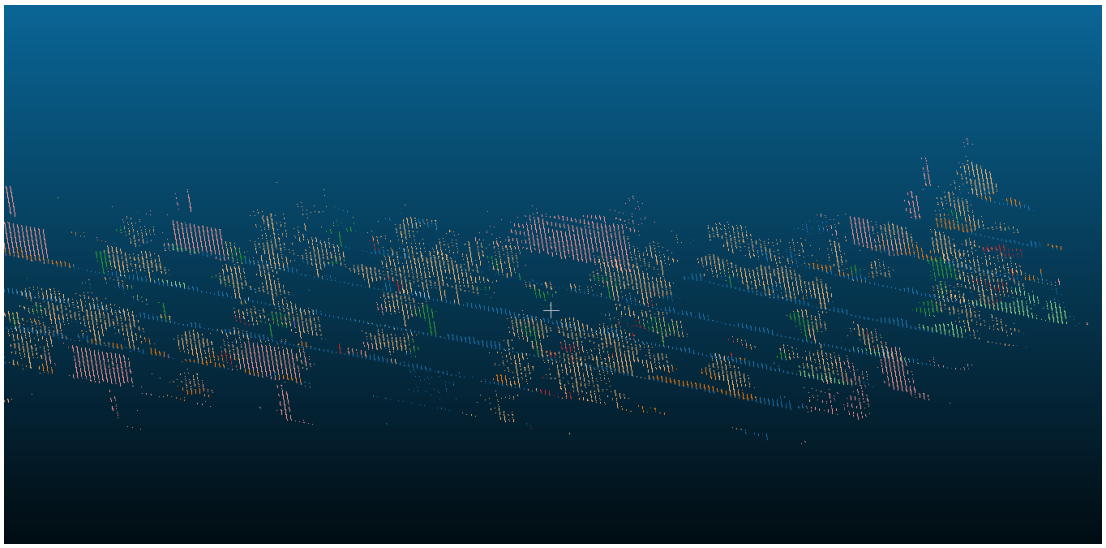


Figure 5.29: Ground truth semantic labels for the same building area after label transfer and downsampling.

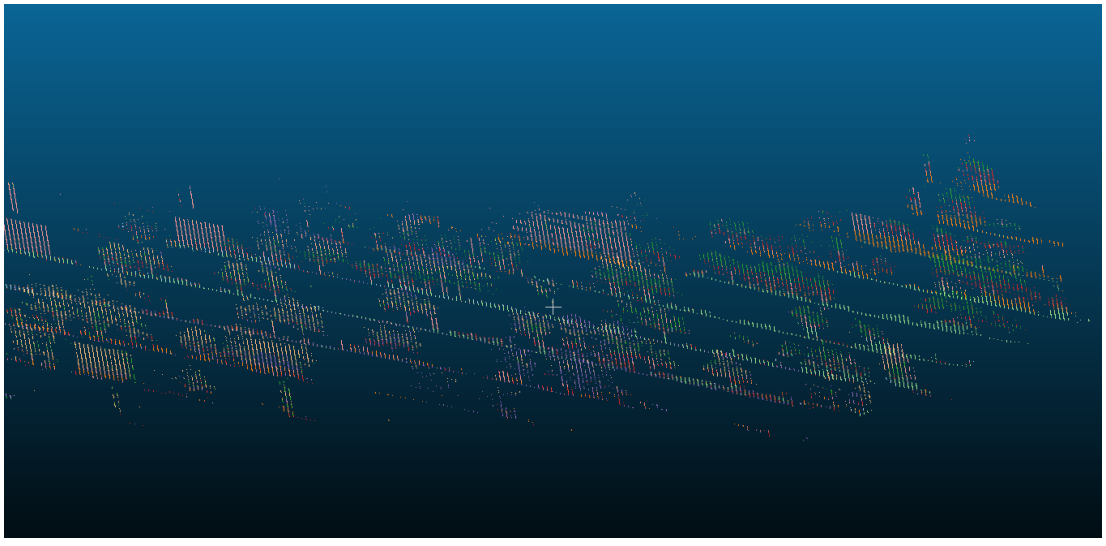


Figure 5.30: Model predictions for the same building area. Points are colored by the predicted semantic classes.

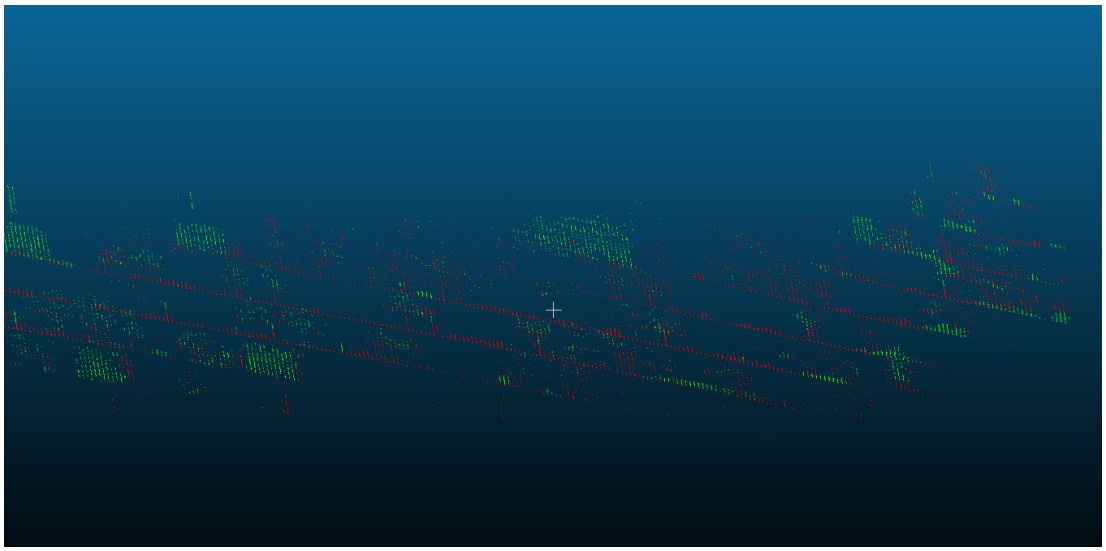


Figure 5.31: Comparison of predictions with ground truth for the building area. Green points are correctly predicted; red points indicate misclassifications.

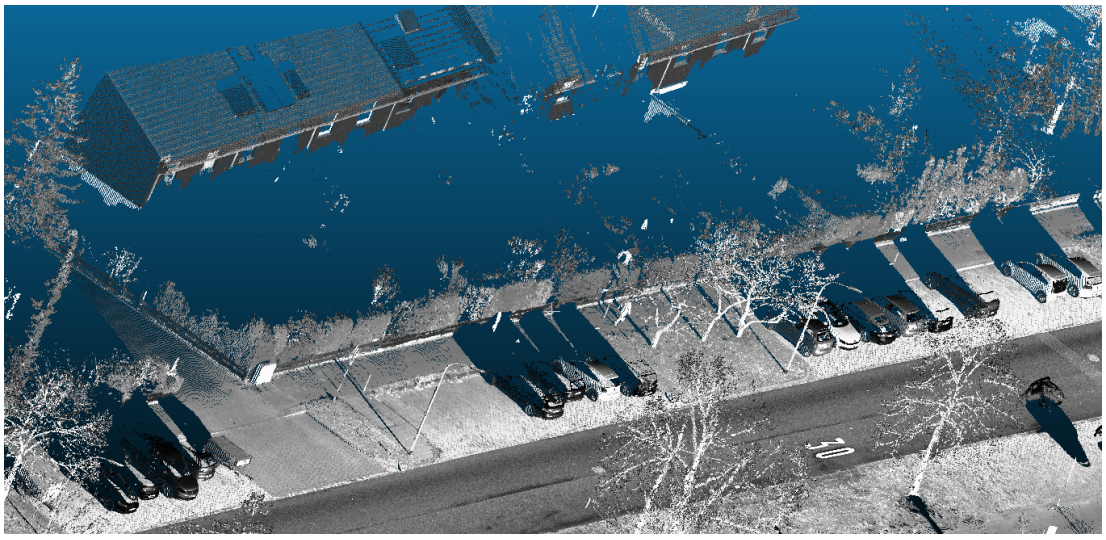


Figure 5.32: Original point cloud of the barrier area without semantic annotations.

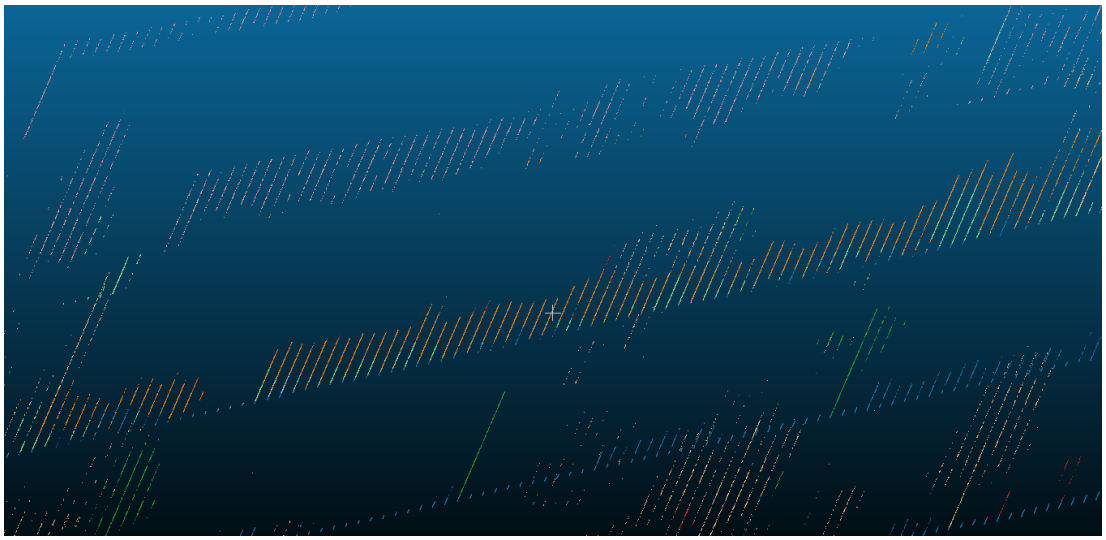


Figure 5.33: Ground truth semantic labels for the same barrier area after label transfer and downsampling.

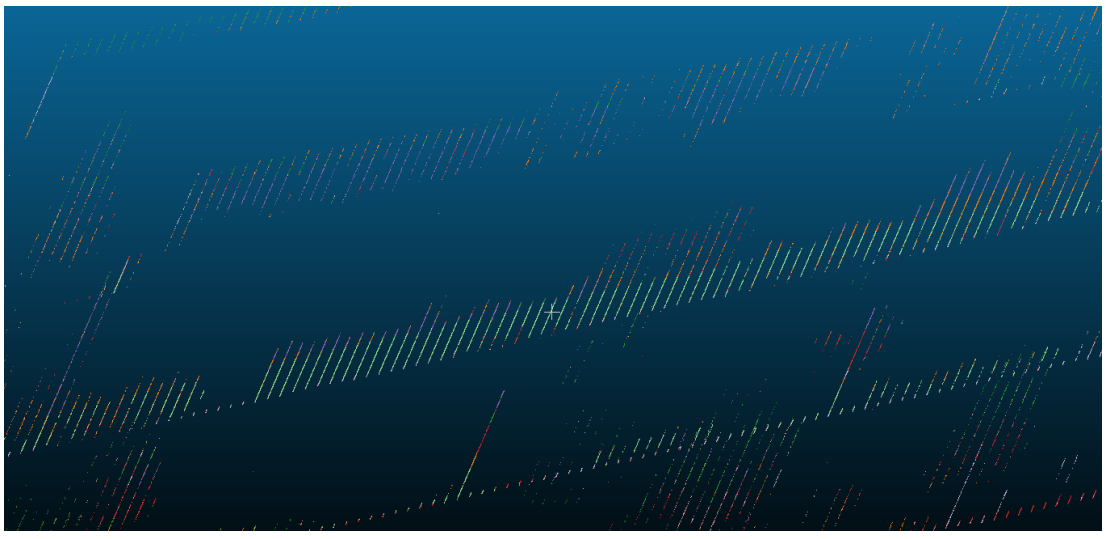


Figure 5.34: Model predictions for the same barrier area. Points are colored by the predicted semantic classes.

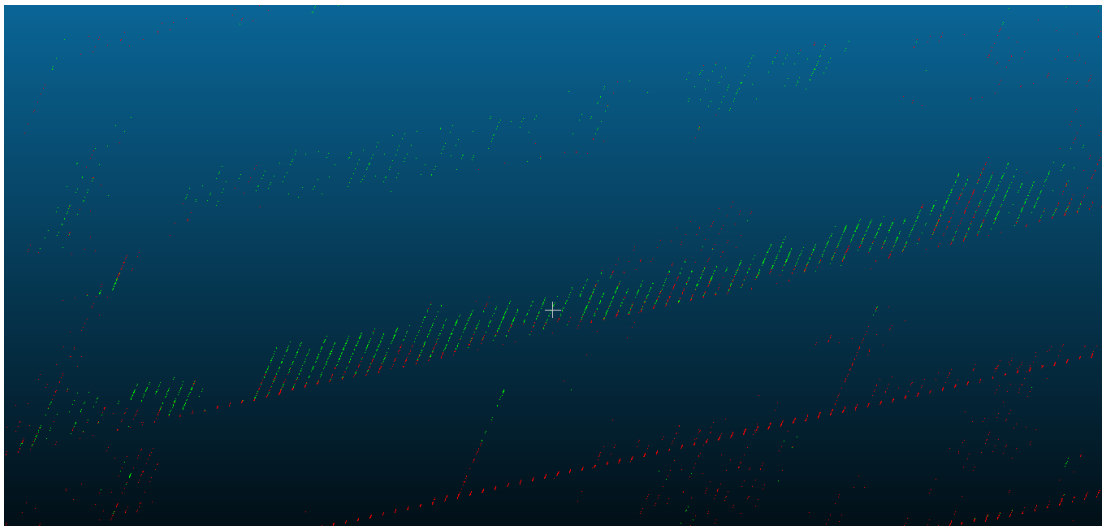


Figure 5.35: Comparison of predictions with ground truth for the barrier area. Green points indicate correct predictions; red points indicate misclassifications.

Chapter 6

Discussion and Future Work

6.1 Limitations

As shown in the Section 5: Experiments, even if predictions look reasonable, accuracy is quite low. Therefore, the network structure can be improved or modified from scratch. During the training stage, we avoid the maximum pooling layer in the PointNet++ network to simplify our approach, so the first option could be adding up this refinement. Another refinement option could be to change and test hyperparameters on the training structure. Because of point cloud density and hardware capabilities, we just trained under certain hyperparameter settings, so playing around extreme cases could give better accuracy.

6.2 OpenDRIVE Construction

As described in Section 1: Introduction, the main motivation behind this study is producing a lightweight automatic annotation pipeline for point-wise segmentation. OpenDRIVE data format that covers 3D coordinate information and semantic labels regarding traffic surroundings are used to realize this purpose. And we suggested this way could be a better baseline to produce further OpenDRIVE files without manually annotated datasets as described in Section 2.

Figure 6.1, most online HD map production methods depend on bird's-eye view extraction from on-board sensor data. After that, extracted features are used for the production of vectorized maps. Those vectorized maps include road boundaries, pedestrian crossings, and lane dividers that are provided as features from a bird's-eye view. In this way, they offer the necessity of localization, and post-processing can be eliminated. Even if it can reduce the workload on post-processing and provide local high-precision map elements, this method only focuses on the extraction of road boundaries, lane dividers, and pedestrian crossings. In Figure 6.2, the left images show ground truth and the right images show the

results of VectorMapNet [15]. Even if they called those results "high-definition maps" those vectorized maps only include limited information about roads. But a high-definition map should cover more than these elements. Besides road elements (road marks, lanes, road signs, etc.), traffic elements (traffic lights, traffic signs, etc.), and supporter elements (street lamps, trees, objects, etc.) still have great importance for autonomous driving.[16] Therefore, this task would be considered the further step of this study.

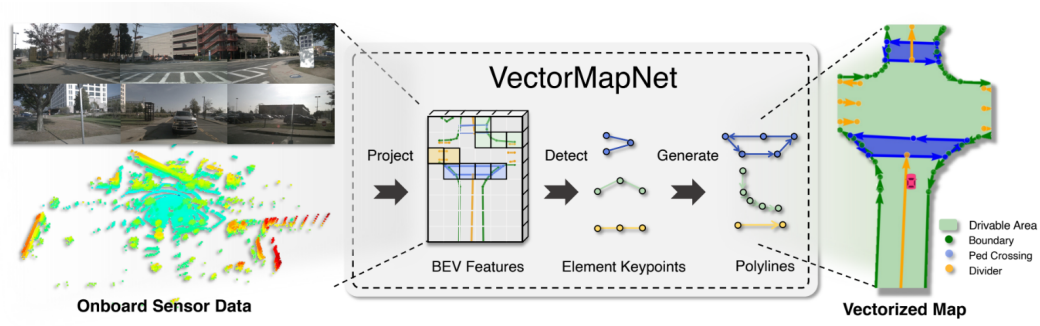


Figure 6.1: Overview for VectorMapNet [15]

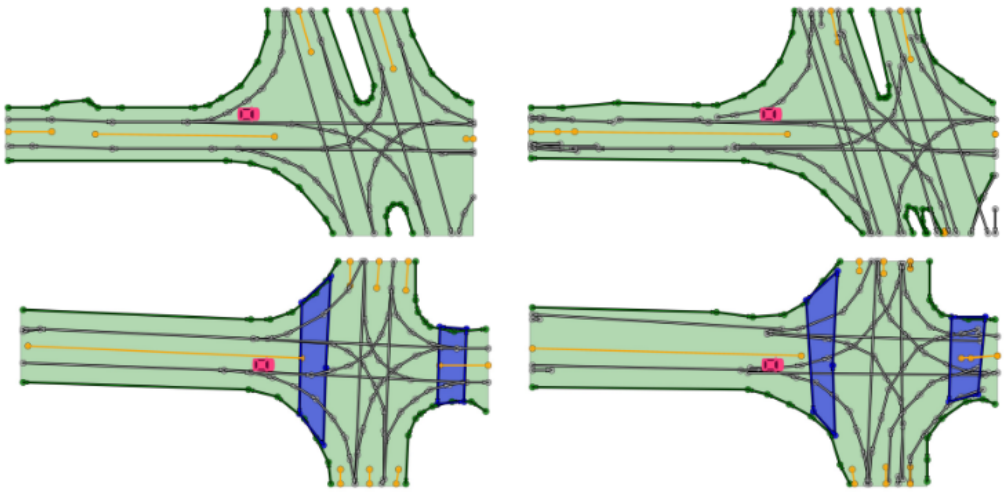


Figure 6.2: Example of output for VectorMapNet (left: ground truth, right: VectorMapNet) [15]

Chapter 7

Conclusion

In conclusion, perception plays a crucial role in autonomous driving, and point-wise segmentation significantly contributes to the scientific community in this regard. Even though early and common approaches rely on image-based semantic segmentation because of the availability of image-based datasets, pixel-wise segmentation has some constraints in terms of scalability on different dimensions and conditions. LiDAR-collected 3D point clouds can provide reliable information about extreme weather conditions and surface reflectivity; however, point-wise segmentation faces challenges because the complex structure of 3D data complicates storage and annotation. Annotating point cloud datasets is still challenging in terms of time, cost, attention, and data aspects. Effective segmentation requires accurate annotation in huge amounts of sequences; however, because of inconsistency and perspective issues, labeling point clouds is challenging to maintain accurately. Therefore, automatic annotation is an emerging method for obtaining semantic classes from sensor-collected point clouds. To address these challenges, several approaches have been proposed, including the use of synthetically generated point clouds and game-engine-dependent synthetic environments with instance-level annotation for every object class. These approaches aim to eliminate errors related to human labeling and streamline the annotation process. However, these methods still face some problems, such as the requirement of preprocessing and covering all possible events and objects in real-world scenarios. In this study, we discussed whether an existing high-definition map could be used as a base map to automatically annotate a point cloud. The OpenDRIVE dataset is used to create a synthetic point cloud in this direction. Then we proceed to the Iterative Closest Point algorithm to find corresponding points for both the synthetically generated point cloud and the originally collected point cloud. This step enables the transfer of labels to original point clouds. After this implementation, the problem definition turns into supervised semantic segmentation on a point cloud. We implemented a neural network,

which generated predictions. Our findings show our approach should improve in terms of metric predictions, even if synthetic point cloud construction and label transfer work well. Despite the existence of multiple methods for representing road networks as high-definition maps, data standardization remains a work in progress.

Bibliography

[1] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987.

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences, 2019.

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020.

[4] Shaoyu Chen, Yunchi Zhang, Bencheng Liao, Jiafeng Xie, Tianheng Cheng, Wei Sui, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene, 2023.

[5] Xiaolei Chen, Wenlong Liao, Bin Liu, Junchi Yan, and Tao He. Opendenselane: A Dense Lidar-Based Dataset for HD Map Construction. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.

[6] CloudCompare Development Team. CloudCompare (version 2.13.2). cloudcompare.org, 2024.

[7] Wenjie Ding, Limeng Qiao, Xi Qiu, and Chi Zhang. PivotNet: Vectorized Pivot Learning for End-to-end HD Map Construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3672–3682, 2023.

[8] GDAL Development Team. GDAL Geospatial Data Abstraction software Library, version 3.10.2. Open Source Geospatial Foundation. 10.5281/zenodo.14871456, 2025.

[9] Elias Greve, Martin Büchner, Niclas Vödisch, Wolfram Burgard, and Abhinav Valada. Collaborative dynamic 3d scene graphs for automated driving. In

2024 IEEE International Conference on Robotics and Automation (ICRA), pages 11118-11124. IEEE, May 2024.

[10] Dan Halperin and Niklas Eisl. Point cloud based scene segmentation: A survey, 2025.

[11] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2522–2529. IEEE, 2019.

[12] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. HDMapNet: An Online HD Map Construction and Evaluation Framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022.

[13] libOpenDRIVE Development Team. libOpenDRIVE – small, lightweight C++ library for handling OpenDRIVE files. doi.org/10.5281/zenodo.7771707, 2023.

[14] Jun Liu, Jihua Xiao, HongJie Cao, and Jiakai Deng. The status and challenges of high precision map for automated driving. In *China Satellite Navigation Conference*, pages 266–276. Springer, 2019.

[15] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *International Conference on Machine Learning*, pages 22352–22369. PMLR, 2023.

[16] Zhipeng Luo, Lipeng Gao, Haodong Xiang, and Jonathan Li. Road object detection for hd map: Full-element survey, analysis and perspectives. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197:122–144, 2023.

[17] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017.

[18] Michael Scholz, Oliver Böttcher, Jörg Peter Schäfer, Franz Andert, and iMAR Navigation GmbH. OpenDRIVE road network dataset of Schwarzer Berg in Brunswick. German Aerospace Center (DLR). doi.org/10.5281/zenodo.15395840, May 2025.

[19] Michael Scholz, Oliver Böttcher, Jörg Peter Schäfer, Franz Andert, and iMAR Navigation GmbH. Point cloud road space dataset of Schwarzer Berg in Brunswick. German Aerospace Center (DLR). doi.org/10.5281/zenodo.15527622, May 2025.

[20] Juyeb Shin, Hyeonjun Jeong, Francois Rameau, and Dongsuk Kum. Instagram: Instance-level graph modeling for vectorized hd map learning. *IEEE Transactions on Intelligent Transportation Systems*, 26(2):1889–1899, 2025.

[21] Kun Tang, Xu Cao, Zhipeng Cao, Tong Zhou, Erlong Li, Ao Liu, Shengtao Zou, Chang Liu, Shuqi Mei, Elena Sizikova, et al. THMA: Tencent HD Map AI System for Creating HD Map Annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15585–15593, 2023.

[22] Xuewei Tang, Kun Jiang, Mengmeng Yang, Zhaoyang Liu, Peijin Jia, Benny Wijaya, Tuopu Wen, Le Cui, and Diange Yang. High-definition maps construction based on visual sensor: A comprehensive survey. *IEEE Transactions on Intelligent Vehicles*, pages 1–23, 2023.

[23] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation, 2021.

[24] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155. PMLR, 2018.

[25] Jiaxin Zhang, Shiyuan Chen, Haoran Yin, Ruohong Mei, Xuan Liu, Cong Yang, Qian Zhang, and Wei Sui. A vision-centric approach for static map element annotation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15861–15867, 2024.

[26] Chaoran Zhao, Bo Peng, and Takuya Azumi. Point cloud automatic annotation framework for autonomous driving. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 3063–3070. IEEE, 2024.

List of Figures

List of Tables

List of Algorithms

Appendix A

Paper

Automatic Point Cloud Annotation using existing HD Map Data for Map Construction

Gülşen Bardak

***Abstract*— Perception is a critical task for autonomous driving and is challenging because of the scale and complexity of urban environments, which leads to limitations in sensor accuracy. Point-wise segmentation plays a crucial role in addressing these challenges. However, point cloud datasets have limitations related to data storage capacity, dimensionality, and the scale factor during manual annotation, making it difficult to achieve efficiency in terms of time and cost. In this study, we propose an automatic annotation pipeline sourced from another key contributor to automated driving systems: high-definition (HD) maps. HD maps provide lane-level information regarding traffic scenarios, 3-dimensional coordinates, and semantic classes of objects in a lightweight format. Our approach relies on generating a synthetic point cloud from these specific vector-type datasets and matching it with the original sensor-collected training dataset. We then utilize the point clouds and this information to train and generate reliable predictions. Ultimately, we propose that this preprocessing stage can serve as a baseline for producing new HD maps, thereby contributing to automated driving in real-world applications.**

I. Introduction

Scene understanding is essential for developing autonomous systems, where safety is a critical concern. Perception is a challenging task due to the scale of environments at the city and road level, as well as the complexity of these environments, sensor limitations, data scale, storage, and processing. Spatial and scene understanding of surroundings are crucial due to the necessity of safe and autonomous navigation in cars and robots, particularly in autonomous driving approaches [8]. Annotation that is required for learning-based systems is crucial to realizing perception tasks accurately. However, due to challenging conditions, it is still an open research topic that is trying to make it automatic instead of using laborious and expensive solutions. With our research, we will investigate a lightweight automatic annotation framework that can reduce labor and data storage costs by using existing high-definition maps effectively. Effective scene understanding enables autonomous vehicles and robots to navigate complex environments, avoiding obstacles and ensuring the predictability and reliability of participants. Autonomous vehicles employ sophisticated sensors, particularly cameras and LiDAR, to continually and in real time perceive their environment in various scenarios, including other vehicles, pedestrians, barriers, and road signs [21]. However, even if this technology enables the understanding of surroundings, there are still some issues in terms of real-time capability, processing capacity, and sensor adaptability in extreme conditions. Lightweight solutions for annotation in terms of LiDAR and camera-based perception, high storage capacity, and labor are still a subject of research. Although LiDAR and cameras undertake this task and GNSS/IMU-integrated Mobile Mapping Systems (MMS) are widely used in this sense, recently high-definition (HD) maps have come to the fore in terms of playing a critical role in autonomous driving tasks and creating a reliable environment for localization, planning control, and perception tasks. As the interest and need for autonomous driving increase day by day, it also becomes inevitable to automate the elements required to achieve this. Autonomous systems must understand their environment with high accuracy to minimize human-caused traffic accidents and build safe roads and cities. However, achieving those goals has high costs on the road and city scale in terms of labor, storage space, and time management. Therefore, automating the annotation of data sets for use in learning-based systems is necessary. Discussions have focused on high-definition maps and their production, which are crucial for completing tasks related to perception, planning, localization, and control. The two issues that are relatively connected to each other are the main motivations for this study. We will investigate how each issue can mutually contribute to the collaboration between high-definition maps and annotation bottlenecks in point-wise segmentation. This study explores how to improve scene understanding for autonomous systems by reducing the need for manual data annotation. It focuses on using high-definition (HD) maps to support automatic point-wise segmentation, which can lower costs related to labor, storage, and processing. The goal is to develop a lightweight, efficient framework that helps autonomous vehicles and robots better understand their surroundings while addressing the challenges of real-time perception and data management.

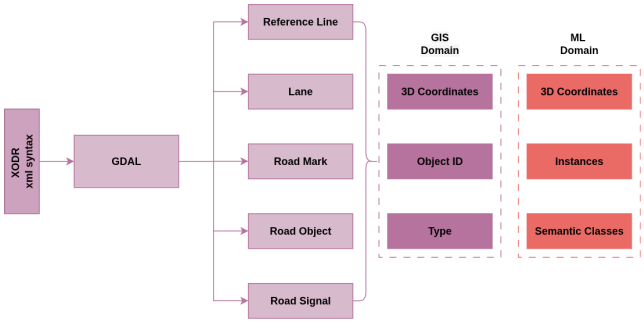


Fig. 1: Overview for approach

II. Related Work

A. High-Definition Maps

HD Map Construction The high-definition map construction process primarily consists of two stages: raw data collection and data processing. The data processing stage can be further divided into two categories —offline and online–, based on the approach used, a distinction that originates from Simultaneous Localization and Mapping (SLAM) methodologies. Contemporary studies in HD map construction mostly depend on online methodologies [6], [11], [12], [16], [19]. Most online HD map generation techniques begin with BEV feature extraction from onboard sensor data, followed by the generation of vectorized map elements such as road boundaries, pedestrian crossings, and lane dividers. Although this approach reduces the workload associated with post-processing and allows for *local* high-precision map creation, it tends to be limited in scope. A truly high-definition map should include a broader set of elements. Unlike online methods for local map construction, global HD map construction is still commonly conducted through offline methodologies. Manual annotation remains the most reliable method due to the precision required—especially regarding semantic information—making the overall process of HD map production both labor-intensive and costly. LiDAR point clouds intrinsically encompass 3D spatial data at each feature point, facilitating more precise detection outcomes that may be directly applied in HD map development [4]. To address challenges on annotation and creation of high-definiton maps, several approaches have been proposed. VMA [3] introduces automatic annotation for online HD map construction through a scene-splitting strategy. CAMA [20], provides automatic annotations using image-based methods enriched with elevation information. It seeks to produce dense 3D road surfaces augmented with semantic and photometric features, utilizing the nuScene dataset for evaluation. THMA [17] introduces an annotation technique grounded in self-supervised segmentation learning, with the objective of enhancing the automation of the HD map annotation process. Despite the existence of multiple methods for representing road networks as high-definition maps, data standardization remains a work in progress.

Characteristics of OpenDRIVE The ASAM OpenDRIVE format is an open industry standard maintained by the Association for Standardization of Automation and Measuring Systems. It represents road networks in a file format with the extension `.xodr`, organized in a hierarchical structure commonly encoded using XML. This format captures the geometric relationships of road features and can be generated using real data or synthetically in various software environments (mostly proprietary ones). Besides the main road components (lanes, road marks, road signs, etc.), an OpenDRIVE dataset can contain traffic-regulating infrastructure elements (traffic lights, traffic signs, etc.) and supporter elements (street lamps, trees, objects, etc.). The complexity of OpenDRIVE makes data acquisition a sophisticated task, often financed by the automotive industry and conducted by third-party mobile mapping providers. As a main characteristic, all road elements are commonly constructed in relation to and linearly referenced along a *road reference line*, that is the main reason why this type of file is still lightweight, this approach is represented in Figure 2

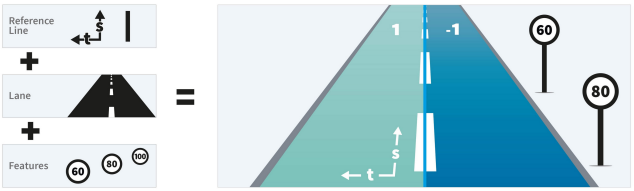


Fig. 2: Modeling of road elements in OpenDRIVE; © ASAM

Semantic Segmentation of Point Cloud Semantic segmentation has a critical role in scene understanding to realize automated driving. The main approach to doing semantic representation is assigning semantic classes to each basic component in the dataset that is collected during the driving: pixel-on-pixel-wise semantic segmentation and points in point-wise semantic segmentation [9].

Even though early and common approaches rely on image-based semantic segmentation because of the availability of image-based datasets, pixel-wise segmentation has some constraints in terms of scalability on different dimensions and conditions. Pixel-based segmentation has to be projected onto the 3-dimensional physical world from a 2-dimensional plane representation; however, this task is open to losing some valuable information, like depth, so post-processing is required. Additionally, cameras have limitations when capturing images in various weather conditions, such as rain, fog, or at night. LiDAR-collected 3D point clouds can provide reliable information about extreme weather conditions and surface reflectivity; however, point-wise segmentation faces challenges because the complex structure of 3D data complicates storage and annotation [9].

Effective segmentation requires accurate annotation in huge amounts of sequences; however, because of inconsistency and perspective issues, labeling point clouds is challenging to maintain accurately [18]. Sourced with those issues, recent research is focusing on two aspects of alternative ways of annotation: the first one is transfer learning that uses domain adaptation from different source data, and the second one is automatic annotation with various methodologies. Transfer learning could be categorized into two subtitles: synthetic-to-original dataset that is sourced from point clouds specifically for this purpose and original-to-original transfer that is sourced from another sourced dataset.

There are some publicly available datasets that consider the driving scenario, e.g., SemanticKITTI and nuScenes [1], [2]. However, this public dataset suffers from a class imbalance, known as the long-tail class problem. Besides that, those datasets rely on human annotation, so human-oriented problems and errors can occur [10].

To address this problem, some studies focus on synthetically generated point clouds. However, few studies are still

considering synthetic-to-real approaches because there is a lack of extensive synthetic data with accurate semantic labels [18]. Several studies utilize the Grand Theft Auto V (GTA V) environment, a commercial video game known for its realistic driving simulation, to produce synthetic point clouds for use in 3D point cloud segmentation. Using game-engine-dependent synthetic point clouds has advantages thanks to the instance-level annotation for every object class [10]. This approach proposes to eliminate errors related to human labeling during the annotation task by using a synthetically generated 3D world, depending on the purpose. This method streamlines the annotation process and allows researchers to generate diverse datasets under controlled conditions, facilitating more robust machine learning models. Consequently, leveraging such synthetic environments can enhance the accuracy and efficiency of various computer vision applications. However, these approaches are still facing some problems, like the requirement of preprocessing and covering all possible events and objects in real-world scenarios.

III. Approach

A. Geometry Extraction from HD Map

As explained in Section II, we used the OpenDRIVE data format because it is one of the standard types of datasets available for our region, in addition to point clouds. To process OpenDRIVE datasets, we utilized the Geospatial Data Abstraction Library (GDAL) [7], which provides a dedicated driver capable of reading and processing OpenDRIVE (`.xodr`) files. Using these geometries, a synthetic point cloud can be generated, which serves as a baseline for creating annotated datasets. OpenDRIVE data includes primary road elements as well as traffic infrastructure such as traffic signs, parking areas, and other support elements. This enables the extraction of 3D spatial information from XML-based datasets that describe static roadway features. From this 3D data, we can synthesize a point cloud that forms the foundation for the subsequent annotation phase. OpenDRIVE data can include complex geometry representations, typically modeled using third-order parametric cubic polynomials. Each road element is defined relative to a reference line using these polynomials. We mainly use the layers *RoadObject*, *Lane*, and *RoadMark*, which are modeled as a TIN, to create a 3D model of the city and roads. This representation enables volumetric modeling for 3D use cases. We used the OpenDRIVE format as a high-definition map representation for the 3D shape reconstruction task. For proof of concept, we implemented our pipeline using the openly available OpenDRIVE dataset "Schwarzer Berg" in Brunswick [14], which we converted to OGC Simple Feature geometries via GDAL. These geometries serve as a baseline to generate a synthetic point cloud, as explained in the following section.

B. Building Synthetic Point Cloud

As explained in Section III-A, the resulting geometries were generated in a TIN structure to be used in constructing the synthetic point cloud. This way, the objects or participants in the scene are expected to be synthetically generated with the same high detail (e.g., convex geometries) as the original point cloud. However, it should be noted that the performance of this approach is directly related to the level of detail provided in the OpenDRIVE data generation. The GDAL XODR driver generates the triangles and the corner points of the triangles that form the object for triangulated irregular networks; therefore, a proper interpolation method is required. For this purpose, three different sampling methods were used, and the number of points per m^3 expresses the point density to be generated per volume. The model's geometry becomes more detailed in volumetric instances as this value increases. The third parameter is the downsampling ratio, as the reference point cloud [15] exhibits a higher point density than our generated synthetic point cloud. We will measure our algorithm quality with Iterative Closest Point (ICP) registration with a reference point cloud in Section IV.

C. ICP Integration

To get label transfer from a synthetic point cloud, the method for creating this point cloud is explained in Section III-B: Building a synthetic point cloud, and the Iterative Closest Point (ICP) algorithm is used to align it with the original point cloud gathered by sensors. Table I lists the hyperparameters used to realize a proper matching algorithm in our approach. In the beginning, the sparsity of the synthetic point cloud is dependent on set parameters that are explained in Section III-B. This situation leads us to conclude that a higher density is an efficient approach initially. However, because of computational cost, densifying synthetic point clouds is not the best approach. To deal with these situations, we reduce the number of points in the original point cloud to lower costs and avoid mismatches due to differences in density with the synthetic point cloud. Therefore, the downsampling ratio is the most important setting for using the Iterative Closest Point algorithm, which helps match points between two point clouds so that labels can be transferred. The next two hyperparameters are important for the *KDTree* search algorithm, which helps estimate normals and is used in the registration process for both point clouds. The maximum number of nearest neighbors determines the number of points available in the search space. Additionally, we used this parameter to identify corresponding nearest neighbor points for estimating the intensity values of synthetic point clouds; however, we have not implemented this value in the network.

Parameter
Downsampling ratio
Max NN for KDTree
Radius for KDTree (m)
Initial guess for ICP transformation
Transformation type
Max correspondence distance

TABLE I: Hyperparameters for Iterative Closest Point Algorithm

D. Model Structure for Semantic Segmentation

As explained in previous sections, this study proposed an automatic annotation pipeline for use in point-wise semantic segmentation. Our problem definition will centers on how to use the supervised learning methodology to effectively predict semantic classes, adhering to the previously described approaches.Debugging and investigating ready-to-use model structures was difficult and time-consuming due to the custom-generated dataset. So to realize semantic segmentation, we explored two different custom architectures that mainly rely on the PointNet [13] network structure. The first architecture we explored is PointNet, which processes 3D coordinates using convolutional layers. It could not provide sufficient accuracy for predictions even if different loss functions and hyperparameters were used. The second network is called GeoPointNet Algortihm 1 and uses point normals in addition to 3D coordinates Besides the model selection, data preprocessing is one of the most important steps in refining our model performance. To handle computational issues, we downsampled the input dataset that provides a certain point number. On the other hand, we normalized point coordinates to ensure stability and reduce scale variance during the training stage. Because the datasets are unbalanced, meaning that some semantic categories occur more often others, we figured out the weights for each class and chose a loss function that takes these weights into account to handle the differences between classes and prevent problems caused by the imbalance. We employed a combination of cross-entropy loss and focal loss to address class imbalance in the dataset, utilizing class weights. As regular hyperparameters for model structure, different batch sizes and optimal epoch numbers are also experienced to get better predictions.

Algorithm 1 GeoPointNet Model for Semantic Segmentation

Require: Input $x \in \mathbb{R}^{B\times N\times 6}$
Ensure: Output $y \in \mathbb{R}^{B\times N\times C}$
$x \leftarrow \text{permute}(x) \in \mathbb{R}^{B\times 6\times N}$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 6 \rightarrow 64))$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 64 \rightarrow 128))$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 128 \rightarrow 256))$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 256 \rightarrow 512))$
$x \leftarrow \text{ReLU}(\text{Conv1D}(x, 512 \rightarrow 256))$
$x \leftarrow \text{Dropout}(x, p = 0.3)$
$x \leftarrow \text{Conv1D}(x, 256 \rightarrow C)$
$y \leftarrow \text{permute}(x) \in \mathbb{R}^{B\times N\times C}$
return y

IV. Experimental Evaluation

In this study we proposed a lightweight automatic annotation pipeline for point-wise semantic segmentation; in this section we will discuss how our synthetic point cloud generation approach works for transferring labels and how our network performs on predictions of point-wise semantic classes.

A. Synthetic Point Cloud Construction, ICP Integration, and Automatic Annotation Results

Building a synthetic point cloud from a vector-based dataset has many challenges in terms of sampling/interpolation methodology. On this baseline, we explore the performance of three different sampling types in generating 3D synthetic point clouds. At first glance, uniform and random sampling appear to yield similar results, while the results from the normal distribution seem more distant from reality. Table II presents the calculated cloud-to-cloud distances among the three synthetic point clouds, with the standard deviation indicating that the Gaussian-distributed synthetic point cloud differs slightly from both the uniform and randomly sampled point clouds. This calculation and visualization was made on CloudCompare [5] which is an open-source 3D point cloud and mesh processing software.

Sampling	Gaussian	Random	Uniform
Gaussian	0.0	0.0518 / 0.066	0.0518 / 0.066
Random	0.0518 / 0.066	0.0	0.0524 / 0.042
Uniform	0.0518 / 0.066	0.0524 / 0.042	0.0

TABLE II: Cloud-to-cloud mean distances and standard deviations between raw point clouds in meters.

The Iterative Closest Point Algorithm (ICP) algorithm was utilized to quantitatively measure the performance of three different sampling methods. Synthetic point clouds were produced utilizing uniform, random, and normal sampling methods as source datasets, alongside the reference LiDAR-acquired and georeferenced point cloud as the target dataset. The ICP registration algorithm subsequently processed these point clouds, utilizing the settings specified in Table I. As shown in Table III, alignment has reliable results with approximately 0.99 fitness value and 1.053 m RMSE. As the downsampling and upsampling ratios are identical, we acquire an accurately scaled, annotated point cloud.

Sampling Type	**Number of Points**	**RMSE**	**Fitness**
Normal	5,143,308	1.09767	0.99923
Random	5,123,088	1.05360	0.99921
Uniform	5,123,424	1.05346	0.99922

TABLE III: Different sampling methods' performance for ICP registration with responsible parameters in Table I; the reference point cloud contains 54,350,752 points.

In details of specific objects, the KDTree-based label transfer algorithm operates with high precision and provides accurate annotations if a point on an object, such as a building, in sparse regions lacks neighboring points at that elevation. Figure 4 shows an example of how well objects are identified in areas with few points; this scene comes from the area shown in Figure 3, where both trees and buildings were correctly identified without any over- or under-segmentation.

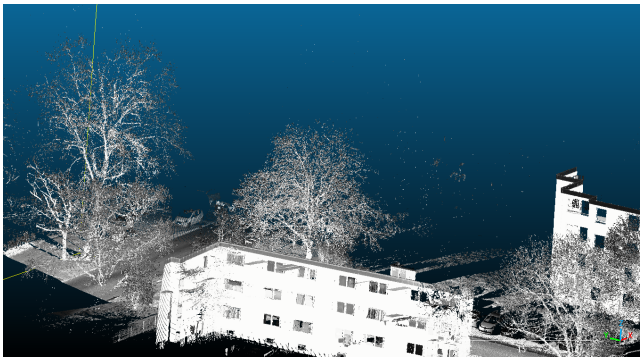


Fig. 3: Sample snipped for tree and building representation.

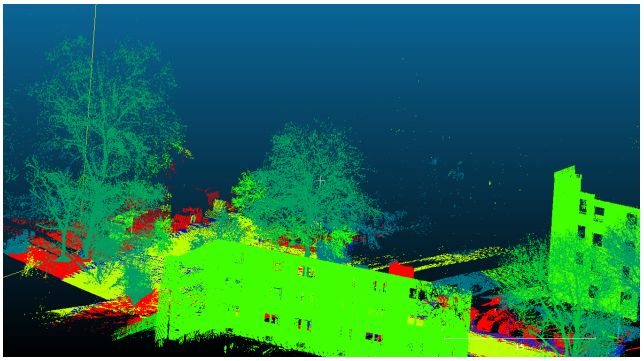


Fig. 4: Labeled sample snipped for tree and building representation.

B. Supervised Semantic Segmentation with Automatic Annotation

A subset of 10 classes was selected from the full set of 22 for model training to facilitate the detection of certain easily recognizable objects, such as trees, buildings, traffic lights, and signs. This approach allows us to evaluate our model from both quantitative and qualitative perspectives. After enhancing the label transferring system using synthetic point cloud creation and the iterative closest point method, the pointwise semantic segmentation model was used on the labeled original point cloud, as explained in the approach section. The GeoPointNet architecture Algorithm 1 was trained with 4 different batch sizes for 400 epochs using the focal loss and cross entropy loss combination, coordinate normalization, and downsampling. As shown in Table IV, class imbalance and geometric differences are the reasons for the diverse metric distribution. The table shows a huge number of points affecting model performance; however, it is not the only reason for lower accuracy. Even if the Street Lamp class has a relatively higher number of points than several classes, precision is comparatively low. This behavior can be interpreted as surface distribution because normals are also an input for the model, which is another key factor for precision when we use only coordinate space-considered models. Flat and continuous objects like barriers and buildings have relatively higher precision.

Figure 5, Figure 6, and Figure 7 show snapshots of model

TABLE IV: Segmentation report for the 10-class model on the synthetic point cloud dataset (excluding ignored index).

ID	Class Name	Precision	Recall	F1-score	Support
1	obstacle	0.2041	0.7574	0.3216	2477
2	vegetation	0.4782	0.7436	0.5821	10927
3	tree	0.8525	0.3346	0.4806	45581
4	streetLamp	0.2785	0.4436	0.3422	10831
5	barrier	0.7689	0.6890	0.7267	25809
6	trafficSign	0.1983	0.5003	0.2841	4301
7	building	0.7992	0.6390	0.7102	29679
8	pole	0.2191	0.6526	0.3280	2504
9	trafficLight	0.2448	0.7218	0.3656	2883
Accuracy		0.5383			
Macro Avg		0.4493	0.6091	0.4601	-
Weighted Avg		0.6910	0.5383	0.5608	134992

Fig. 5: Original, unlabeled point cloud sample for building.

performance in predicting the classes of points belonging to the building. Due to downsampling for the training and validation steps, the point cloud appears in a simplified version. Figure 5 displays sensor-collected point clouds without intensity values. Figure 6 illustrates the performance of label transfers on this building following downsampling. Figure 7 presents the predicted classes for this building. As previously mentioned, the building contains a vast number of points, which indicates that this type of object is at the beginning of the long-tail problem and has certain advantages; in addition, it features flat and continuous geometries. GeoPointNet is an effective network for predicting point-wise semantic classes for this type of object.

V. Conclusion

In conclusion, perception plays a crucial role in autonomous driving, and point-wise segmentation significantly contributes to the scientific community in this regard. Even though early and common approaches rely on image-based semantic segmentation because of the availability of image-based datasets, pixel-wise segmentation has some constraints in terms of scalability on different dimensions and conditions. LiDAR-collected 3D point clouds can provide reliable information about extreme weather conditions and surface reflectivity; however, point-wise segmentation faces challenges because the complex structure of 3D data complicates storage and annotation. Annotating point cloud datasets is still challenging in terms of time, cost, attention, and data aspects. Effective segmentation requires accurate annotation in huge amounts of sequences; however, because of inconsistency and perspective issues, labeling point clouds is challenging

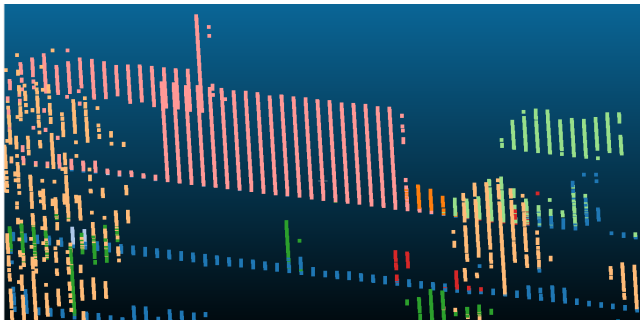


Fig. 6: Ground truth - transfered labels from ICP-stage for responsible building in Figure 5

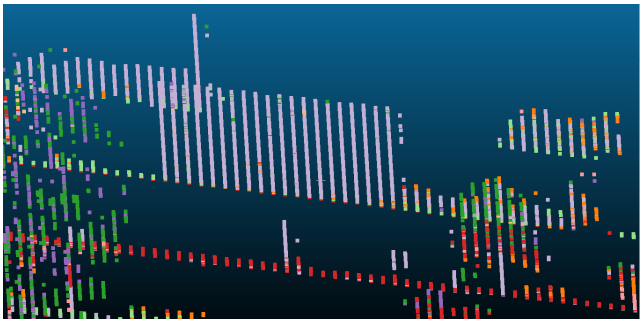


Fig. 7: Predicted labels for responsible building in Figure 5 and 6

to maintain accurately. Therefore, automatic annotation is an emerging method for obtaining semantic classes from sensor-collected point clouds. To address these challenges, several approaches have been proposed, including the use of synthetically generated point clouds and game-engine-dependent synthetic environments with instance-level annotation for every object class. These approaches aim to eliminate errors related to human labeling and streamline the annotation process. However, these methods still face some problems, such as the requirement of preprocessing and covering all possible events and objects in real-world scenarios. In this study, we discussed whether an existing high-definition map could be used as a base map to automatically annotate a point cloud. The OpenDRIVE dataset is used to create a synthetic point cloud in this direction. Then we proceed to the Iterative Closest Point algorithm to find corresponding points for both the synthetically generated point cloud and the originally collected point cloud. This step enables the transfer of labels to original point clouds. After this implementation, the problem definition turns into supervised semantic segmentation on a point cloud. We implemented a neural network, which generated predictions. Our findings show our approach should improve in terms of metric predictions, even if synthetic point cloud construction and label transfer work well. Despite the existence of multiple methods for representing road networks as high-definition maps, data standardization remains a work in progress.

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences, 2019.

[2] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020.

[3] S. Chen, Y. Zhang, B. Liao, J. Xie, T. Cheng, W. Sui, Q. Zhang, C. Huang, W. Liu, and X. Wang. Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene, 2023.

[4] X. Chen, W. Liao, B. Liu, J. Yan, and T. He. Opendenselane: A Dense Lidar-Based Dataset for HD Map Construction. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.

[5] CloudCompare Development Team. CloudCompare (version 2.13.2). cloudcompare.org, 2024.

[6] W. Ding, L. Qiao, X. Qiu, and C. Zhang. PivotNet: Vectorized Pivot Learning for End-to-end HD Map Construction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3672–3682, 2023.

[7] GDAL Development Team. GDAL Geospatial Data Abstraction software Library, version 3.10.2. Open Source Geospatial Foundation. 10.5281/zenodo.14871456, 2025.

[8] E. Greve, M. Büchner, N. Vödisch, W. Burgard, and A. Valada. Collaborative dynamic 3d scene graphs for automated driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11118-11124. IEEE, May 2024.

[9] D. Halperin and N. Eisl. Point cloud based scene segmentation: A survey, 2025.

[10] B. Hurl, K. Czarnecki, and S. Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2522–2529. IEEE, 2019.

[11] Q. Li, Y. Wang, Y. Wang, and H. Zhao. HDMapNet: An Online HD Map Construction and Evaluation Framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022.

[12] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *International Conference on Machine Learning*, pages 22352–22369. PMLR, 2023.

[13] C.R. Qi, H. Su, K. Mo, and L.J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017.

[14] M. Scholz, O. Böttcher, J.P. Schäfer, F. Andert, and iMAR Navigation GmbH. OpenDRIVE road network dataset of Schwarzer Berg in Brunswick. German Aerospace Center (DLR). doi.org/10.5281/zenodo.15395840, May 2025.

[15] M. Scholz, O. Böttcher, J.P. Schäfer, F. Andert, and iMAR Navigation GmbH. Point cloud road space dataset of Schwarzer Berg in Brunswick. German Aerospace Center (DLR). doi.org/10.5281/zenodo.15527622, May 2025.

[16] J. Shin, H. Jeong, F. Rameau, and D. Kum. Instagram: Instance-level graph modeling for vectorized hd map learning. *IEEE Transactions on Intelligent Transportation Systems*, 26(2):1889–1899, 2025.

[17] K. Tang, X. Cao, Z. Cao, T. Zhou, E. Li, A. Liu, S. Zou, C. Liu, S. Mei, E. Sizikova, et al. THMA: Tencent HD Map AI System for Creating HD Map Annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15585–15593, 2023.

[18] A. Xiao, J. Huang, D. Guan, F. Zhan, and S. Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation, 2021.

[19] B. Yang, M. Liang, and R. Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155. PMLR, 2018.

[20] J. Zhang, S. Chen, H. Yin, R. Mei, X. Liu, C. Yang, Q. Zhang, and W. Sui. A vision-centric approach for static map element annotation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15861–15867, 2024.

[21] C. Zhao, B. Peng, and T. Azumi. Point cloud automatic annotation framework for autonomous driving. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 3063–3070. IEEE, 2024.

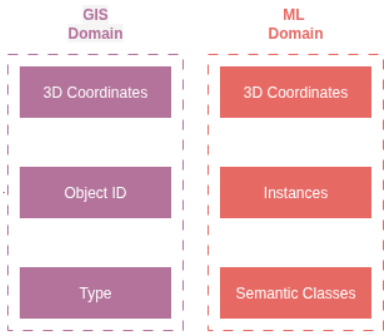
Appendix B

Poster

Automatic Point Cloud Annotation using existing HD Map Data for Map Construction

Gülşen Bardak

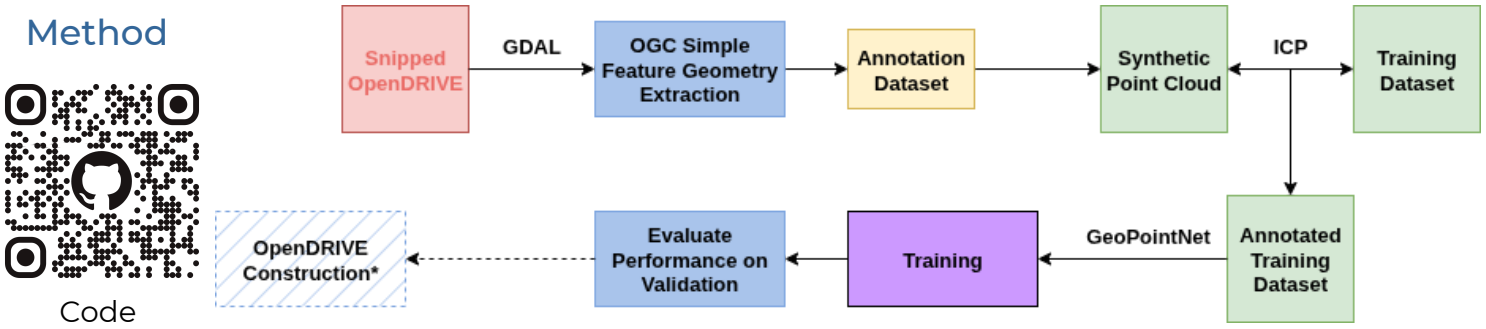
UNIVERSITÄT BONN



Abstract

- Annotation is a bottleneck for point-wise segmentation task
- High-definition maps can provide 3D coordinate information and semantic classes
- Building synthetic point cloud and transferring labels from synthetic point cloud to original point cloud
- Learning with this labels; supervised semantic segmentation

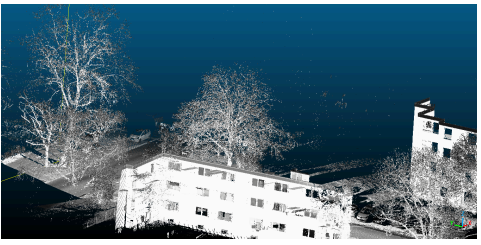
Method

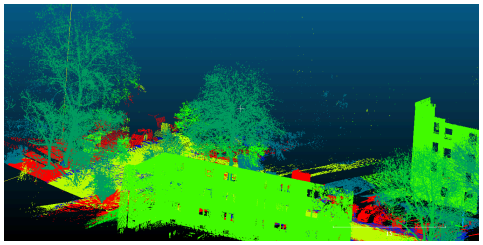


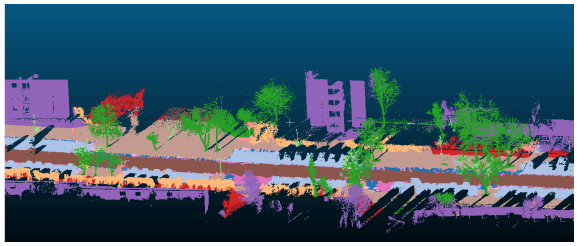
Code

- Getting 3D coordinate and semantic classes from high-definition map.
- Construct synthetic point clouds.
- Transfer labels to the original point cloud.
- Train dataset with these labels.

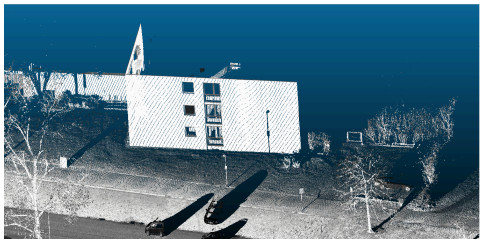
Experiments and Results

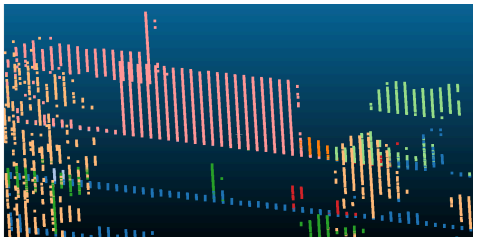


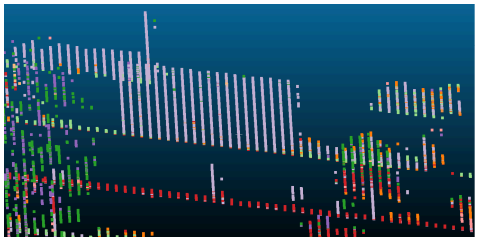




- Synthetic point cloud construction
- Label transferring to original point cloud







- Network design
- Predicting semantic classes

%53