Integration of HD Maps and Point Clouds: An Efficient 3D Reconstruction Framework for Autonomous Driving Applications

Gülşen Bardak[1], Matteo Sodano[2], Michael Scholz[1]

[1] Institute of Transportation Systems, German Aerospace Center (DLR), 38108 Braunschweig, Germany – (guelsen.bardak, michael.scholz)@dlr.de
[2] Institute of Geodesy and Geoinformation, University of Bonn, 53115 Bonn, Germany – matteo.sodano@uni-bonn.de

Keywords: High-definition Maps, OpenDRIVE, GDAL, Point Clouds, 3D Reconstruction, Iterative Closest Point.

Abstract

Autonomous driving approaches require simulation environments that accurately converge real-world conditions. These environments must incorporate various factors, including weather conditions, traffic patterns, and unexpected obstacles, to ensure that autonomous systems can effectively learn and adapt. But, most of the frameworks and simulators are using synthetic simulation environments to realize these conditions because of the complexity of representing real-world details and data storage capacity. In these days when autonomous vehicles are close to being included in daily life, this lack of representation could be eliminated by making use of the currently popular 3D reconstruction methodologies that simulate city and road spaces. Their level of detail enhances the training of autonomous systems and helps identifying potential weaknesses in their decision-making processes, ultimately contributing to the advancement of safer autonomous driving technologies. Currently, mapping technologies and geospatial information play a critical role in accurately constituting 3D environments. High-definition maps (HD) are often sufficiently reliable for such tasks because of lane-level representation capability. In this paper, we propose a lightweight 3D synthetic point cloud reconstruction methodology from existing real-world HD maps in the ASAM OpenDRIVE data format by using the Geospatial Data Abstraction Library (GDAL). By leveraging such road network datasets, we aim to improve the efficiency and accessibility of 3D scene reconstruction for autonomous driving applications. Additionally, we aim to provide a low-cost solution to address the annotation bottleneck in point-wise labeling for the computer vision domain with the constructed 3D models.

1. Introduction

In recent years, the demand for simulation environments has increased alongside the growing interest in autonomy across various industries. Realistic, high-fidelity simulation environments capable of replicating all possible scenarios are essential for the development and production of autonomous systems, particularly regarding security and safety. These environments not only facilitate rigorous testing and validation but also enable developers to identify and mitigate potential risks before deploying autonomous systems in real-world applications. One of the most significant focus areas within such systems is autonomous driving, due to its considerable potential benefits, including safety improvements, reduced environmental impact, and cost savings. It is also beginning to influence urban planning. Consequently, there is a high demand for secure simulation environments for autonomous driving systems, as ensuring a safe driving experience is recognized as essential in traffic settings. In response to this demand, researchers and developers are increasingly concentrating on creating advanced simulation tools that can accurately replicate real-world scenarios. As a result, the integration of sophisticated simulations is becoming a cornerstone in the advancement of autonomous driving systems. Given the scale and complexity of urban and road environments, many simulators attempt to replicate the conditions encountered in autonomous driving using synthetic simulation environments. However, this approach has proven insufficient, which has led to a growing emphasis on the production of digital twins of real urban environments. Nonetheless, challenges related to scale and data storage remain. Limitations in data storage can hinder the creation of highly detailed and accurate digital twins, which are crucial for effective testing and validation of autonomous systems. As technology advances, innovative solutions for managing and processing vast amounts of data will be vital in overcoming these challenges and enhancing the overall system reliability. In the meantime, lightweight solutions may serve as a key approach to address the ongoing lack of improvement caused by the large storage capacity required by existing solutions, such as point clouds. Vector-based geospatial information has recently gained significant importance for scenario-based automated driving because, in addition to road details, understanding the conditional and relational semantics of surroundings is crucial for managing sensor capability outages, such as GNSS outages in city canyons. In this paper, we will explore the integration of high-definition maps that provide lane-level information about road networks into simulation environments. This integration will utilize 3D shape reconstruction through the XODR driver which we contributed to the Geospatial Data Abstraction Library (GDAL Development Team, 2025, Scholz et al., 2024). The source code of the workflow presented in this paper is available on GitHub, see (Bardak, 2025).

2. Related Work

2.1 3D City and Road Models

Three-dimensional semantic city models are important for creating digital versions of cities, self-driving cars, and digital landscape models because they can provide accurate data, are easy to access and thus are useful for planning infrastructure, navigation, environmental simulations, and managing traffic (Beil and Kolbe, 2024, Schwab and Kolbe, 2019). Because of the representability of various dynamic and static agents (e.g., pedestrians), 3D city and road models are of high value for driving and sensor simulations. Automated driving depends on three main components: localization with map data, environment and self-perception using sensors, and planning and control of vehicle actions (Schwab and Kolbe, 2019). Vehicles use various sensors — such as cameras, LiDAR, RADAR, and ultrasonic — to detect and classify objects; in other words, to perform perception tasks. Those sensors have some advantages and disadvantages, which depend on environmental conditions. Additionally, semantic city and road models should align with real-world conditions to effectively predict and avoid potential issues. Testing is an accepted way to guarantee the safety of autonomous driving. Due to the high costs of real-world testing, evaluations are primarily conducted through simulation, particularly to verify performance in essential scenarios that are both risky and challenging to replicate in reality (Wang et al., 2024).

2.2 High Definition Maps

2.2.1 High-definition 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) (Tang et al., 2023b). Contemporary studies in HD map construction mostly depend on online methodologies (Liu et al., 2023, Li et al., 2022, Yang et al., 2018, Ding et al., 2023, Shin et al., 2025). These research 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 (Luo et al., 2023).

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.

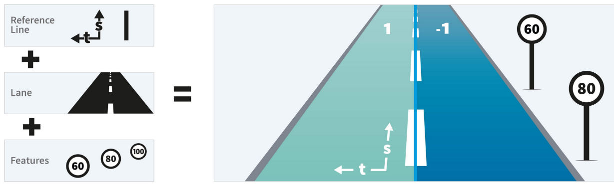

Figure 1. Modeling of road elements in OpenDRIVE; © ASAM

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 (Tang et al., 2023b). 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 (Chen et al., 2022). 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: The Divide-and-Conquer Vectorized Map Annotation System for Large-Scale Driving Scenes (Chen et al., 2023) introduces automatic annotation for online HD map construction through a scene-splitting strategy. CAMA: A vision-centric approach for consistent and accurate map annotation (Zhang et al., 2024), 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: The Tencent HD Map AI System (Tang et al., 2023a) 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.

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

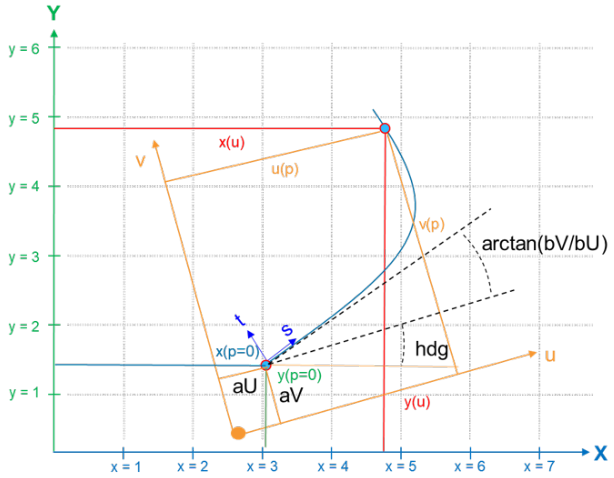

Figure 2. A parametric cubic polynomial in OpenDRIVE; © ASAM

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* as shown in Figure 1. Because of advanced geometry representation through parametric cubic polynomials (Figure 2), the modeling of complex road features can still remain lightweight:

```
<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>
```

3. Approach

Our approach depends on OpenDRIVE high-definition map data which can be an effective source for generating lightweight 3D shape reconstructions of roads and urban areas. Our goal is to create 3D objects from such data and then to refine these objects to accurately reflect real-world forms using suitable techniques. To evaluate the accuracy of our results, we assess the performance of our algorithm against a reference LiDAR point cloud dataset obtained from the same data provider's mobile mapping system which was used to derive the OpenDRIVE source data from.

3.1 GDAL with OpenDRIVE Driver (XODR)

The new XODR vector driver for the Geospatial Data Abstraction Library (GDAL) allows the conversion of highly detailed HD map data from the ASAM OpenDRIVE road description format into common spatial data formats like GeoPackage, GeoJSON, Shapefile, KML, or spatial databases. This makes OpenDRIVE easily usable in established GIS applications. The free software contribution extends the common GDAL library, making it possible to transform OpenDRIVE road elements into OGC Simple Features, which can thus be loaded and processed ad hoc by most proprietary and free and open GIS tools. This aims to stimulate interdisciplinary knowledge transfer and to create an interconnected research community between automotive engineering and GIS (Scholz et al., 2024).

The XODR driver depends on the libOpenDRIVE library, which is a lightweight, dependency-free, fast C++ library providing OpenDRIVE file parsing and simplified 3D model generation (libOpenDRIVE Development Team, 2023). 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). To facilitate this, specific hyperparameters are defined in libOpenDRIVE, which we also specify in our algorithm.

For our test case we use the openly available OpenDRIVE dataset "Schwarzer Berg" in Brunswick (Scholz et al., 2025a), which we convert to OGC Simple Feature geometries via GDAL, see Figure 3 and 4.

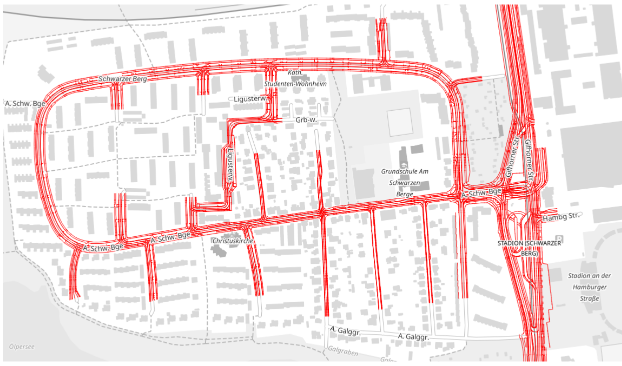

Figure 3. GDAL geometries of lane borders converted from OpenDRIVE; basemap © GeoBasis-DE/BKG 2025, CC BY 4.0 .

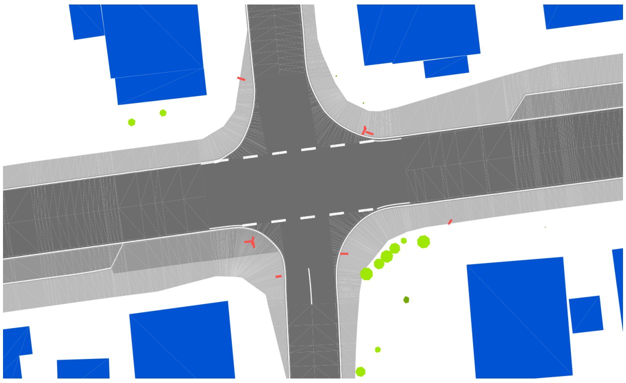

Figure 4. GDAL geometry details of lanes (gray), signals/signs (red), buildings (blue) and vegetation (green).

Parameter	Value
Epsilon ϵ	0.2
Number of points per m^3	100
Downsampling ratio	2
Max NN for KDTree	30
Radius for KDTree (m)	0.2
Initial guess for ICP transformation	I
Number of iterations for ICP	10

Table 1. Parameters to construct 3D shapes from OpenDRIVE.

4. Experiments and Evaluations

We utilized seven parameters to create a 3D synthetic point cloud from OpenDRIVE, enabling improved and optimized convergence with real-world data. Those parameters with values can be seen in Table 1. The first parameter, $epsilon$, is sourced from libOpenDRIVE and the GDAL driver; it represents the interpolation step size used to fit a cubic polynomial. To enhance the performance of our algorithm, we set this parameter to 0.2, a relatively small value that aids in generating a more densely populated vector representation. The second parameter is the number of points per m^3 that 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 (Scholz et al., 2025b) exhibits a higher point density than our generated synthetic point cloud. To achieve improved outcomes in the iterative identification of the nearest points corresponding to the reference points, the algorithm commences

Figure 5. Reference point cloud of a building.

with the downsampling of the reference point cloud. An initial estimation transformation establishes a transformation matrix to align the source and target point clouds. The modification of this transformation may necessitate the maximum number of iterations to ascertain the most suitable correspondences and attain the optimal solution. The iterations parameter is relevant to this process. By fine-tuning these parameters, we can achieve a balance between computational efficiency and the fidelity of the generated data. We will measure our algorithm quality with Iterative Closest Point (ICP) registration with reference point cloud in Section 4.1.

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

In this study, we explore the performance of three different sampling types in generating 3D synthetic point clouds. Figure 5 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 6 for uniform sampling results, Figure 7 for random sampling results, and Figure 8 for normal distribution results — illustrate the outcomes of each sampling method. At the first glance, uniform and random sampling appear to yield similar results, while the results from the normal distribution seem more distant from reality. Table 2 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 Development Team, 2024) which is an open-source 3D point cloud and mesh processing software.

4.1 Iterative Closest Point Algorithm and Point Cloud Annotation

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,

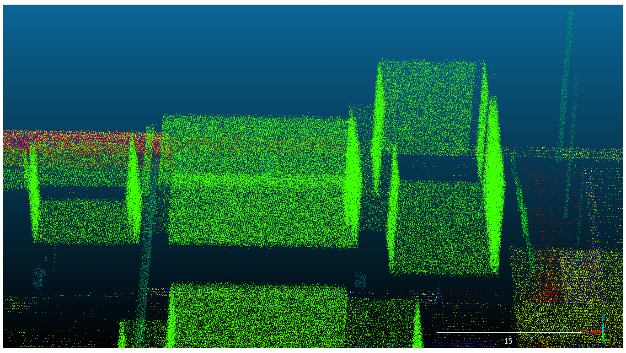

Figure 6. Synthetic point cloud generated through uniform sampling.

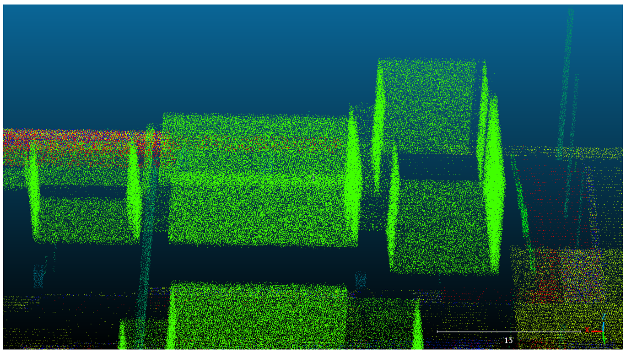

Figure 7. Synthetic point cloud generated through random sampling.

utilizing the settings specified in Table 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 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 3, those three different datasets have different numbers of point densities in 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

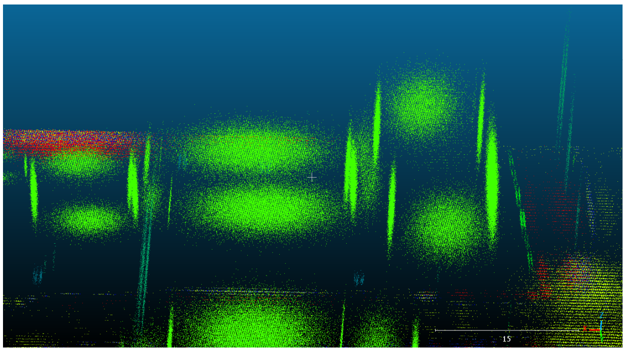

Figure 8. Synthetic point cloud generated through normal sampling.

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 3. Different sampling methods' performance for ICP registration with responsible parameters in Table 1; the reference point cloud contains 54,350,752 points.

Figure 9 illustrates a scene with buildings, parking spaces, roads, vegetation, and trees. In figure 10, the references 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 9. Orthophoto of the scene of interest; © GeoBasis-DE/LGLN 2025, CC BY 4.0

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

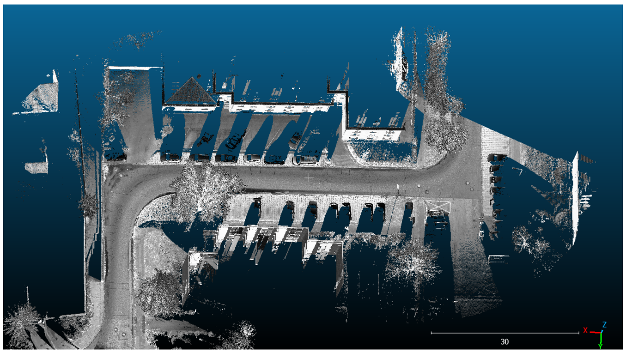

Figure 10. Reference point cloud for the same region of the orthophoto in Figure 9.

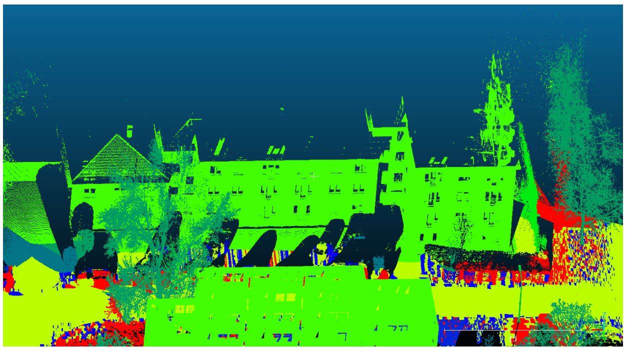

Figure 11. Transferred labels' result for reference point cloud by using ICP on uniform sampled synthetic point cloud. (Responsible region in Figure 9, Figure 10).

errors in the point cloud alignment. As shown in Table 3, 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 9, as demonstrated in Figure 11. 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 13 shows an example of how well objects are identified in areas with few points; this scene comes from the area shown in Figure 12, where both trees and buildings were correctly identified without any over- or under-segmentation.

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 14 and 15 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 16 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 17, 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

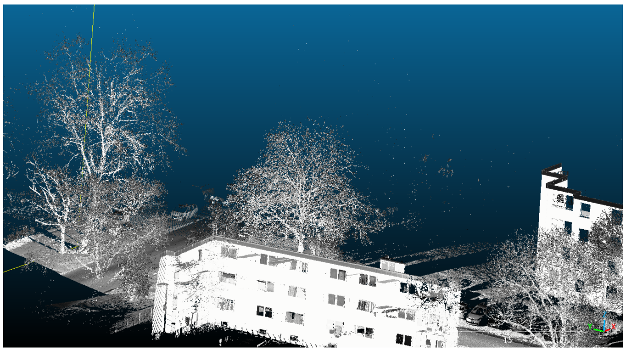

Figure 12. Sample snipped for tree and building representation.

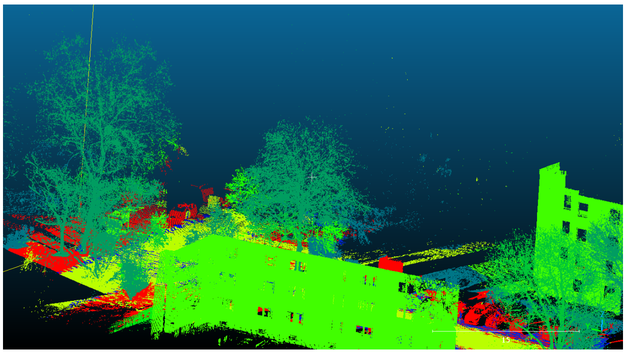

Figure 13. Labeled-sample snipped for tree and building representation.

RoadObject layer. In order 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 18. Aggregation of ground points has resulted in persistent over- and under-segmentation that cannot be rectified through modifications in data mapping during the labeling phase. A targeted approach should be employed to resolve these issues. The following section will address the future implications of this study.

5. Conclusion and Discussion

Realistic and conditionally adapted simulation environments are crucial for autonomous driving in such cases as extreme weather situations, sensor outage possibilities, etc. Recent studies are focusing on getting digital twins that can provide detail at different levels. We suggested using high-definition maps that include 3D location data, examples, and connections between places, claiming they can be a simpler way to create a digital twin of city and road environments. Our approach depends on the released GDAL XODR driver, which processes HD maps in ASAM OpenDRIVE format to get OGC Simple Feature geometries. We converted those geometries into volumetric objects to obtain a 3D representation and then interpolated them using three different methods to determine the optimal sampling methodology. We use a reference point cloud dataset collected through mobile mapping to compare our results. We created fake point clouds for certain areas, and the ICP algorithm is used to match them with the reference point clouds to find their transformations and correspondences. Experiments and results indicate that various perspectives could improve on the approach. The first aspect is data modeling; high-definition map standardization is still under development, and the community is discussing optimal representation methodologies. These data

Figure 14. Orthophoto of the road surface;
© GeoBasis-DE/LGLN 2025, CC BY 4.0

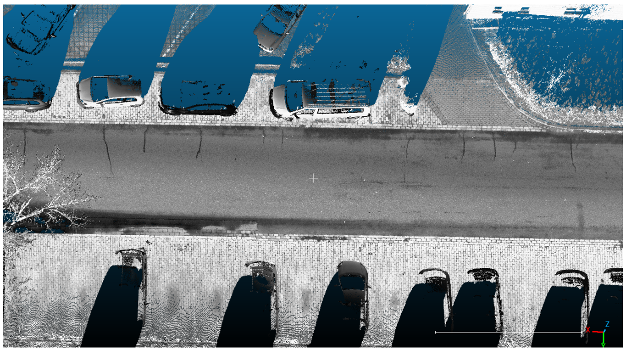

Figure 15. Reference point cloud for road surface.

models just declare how to model attributes; however, the content of attributes is still up to the data provider. This situation necessitates custom modifications for each dependent application, based on the data content. The selection of data types should be contingent upon the application domain. When concentrating on 3D synthetic point cloud reconstruction of edifices using map data, urban models in CityGML can also provide an alternative data foundation. Currently, CityGML data is widely and publicly available in Germany. In the other areas mentioned, we should look into different ways of estimating and predicting use cases, like ground-point aggregation, rather than just focusing on flat and sparse surfaces.

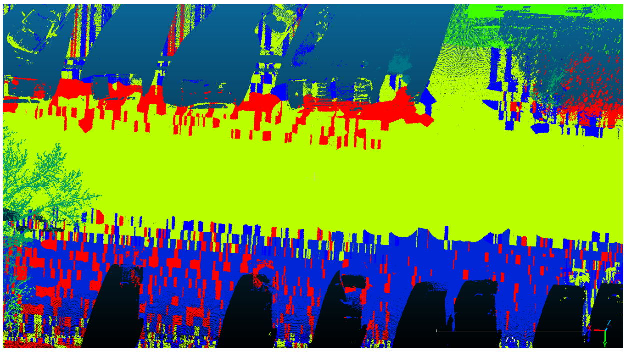

Figure 16. Labeled point cloud 3D model generated from XODR. Light green area represents driving areas, blue points are parking space and red points are road marks that cover the parking space.

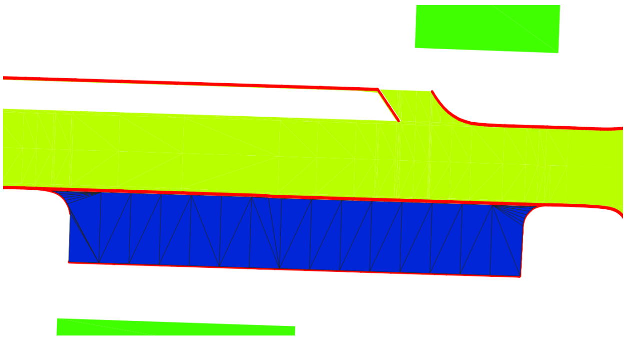

Figure 17. XODR representation where road marks (red) are bordering the parking space (blue), which is the reasons for the noisy point labeling in Figure 16.

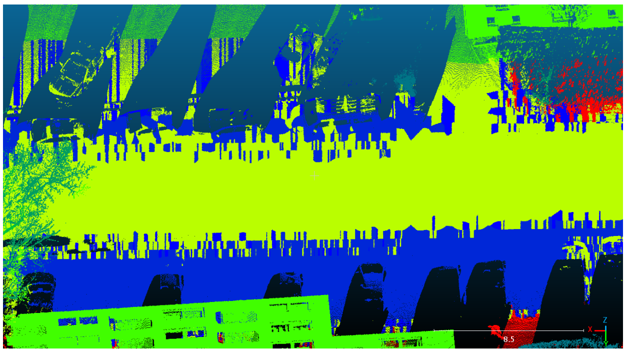

Figure 18. Labeled point cloud 3D model generated from XODR with correced mapping. Light green area represents driving areas and blue points are parking space.

References

Bardak, G., 2025. HDMap2PointCloud – Lightweight 3D point cloud generation from HD maps using GDAL. github.com/gulsenbardk/HDMap2PointCloud.

Beil, C., Kolbe, T. H., 2024. Applications for Semantic 3D Streetspace Models and Their Requirements—A Review and Look at the Road Ahead. *ISPRS International Journal of Geo-Information*, 13(10).

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

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

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

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

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

Li, Q., Wang, Y., Wang, Y., Zhao, H., 2022. HDMapNet: An Online HD Map Construction and Evaluation Framework. *2022*

International Conference on Robotics and Automation (ICRA), IEEE, 4628–4634.

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

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

Luo, Z., Gao, L., Xiang, H., Li, J., 2023. Road object detection for HD map: Full-element survey, analysis and perspectives. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197, 122–144.

Scholz, M., Böttcher, O., Bardak, G., 2024. Improving interoperability between OpenDRIVE HD map data and GIS using GDAL. *FOSS4G Europe 2024*.

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

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

Schwab, B., Kolbe, T. H., 2019. Requirement Analysis of 3D Road Space Models for Automated Driving. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 99–106.

Shin, J., Jeong, H., Rameau, F., Kum, D., 2025. InstaGraM: Instance-Level Graph Modeling for Vectorized HD Map Learning. *IEEE Transactions on Intelligent Transportation Systems*, 26(2), 1889-1899.

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

Tang, X., Jiang, K., Yang, M., Liu, Z., Jia, P., Wijaya, B., Wen, T., Cui, L., Yang, D., 2023b. High-Definition Maps Construction Based on Visual Sensor: A Comprehensive Survey. *IEEE Transactions on Intelligent Vehicles*, 1-23.

Wang, S., Li, C., Sun, T., Jia, F., Yan, R., Yan, J., 2024. Automatic Construction of HD Maps for Simulation-Based Testing of Autonomous Driving Systems. *International Symposium on Theoretical Aspects of Software Engineering*, Springer, 410–427.

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

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