Geo-referencing Autonomous Vehicles Using LoD2 and HD Maps: Performance Assessment in Simulated Urban Environments

Mohamad Wahbah[1], Lukas Ramme[2], Dominik Ernst[1], Sören Vogel[1], Ingo Neumann[1], Hamza Alkhatib[1]

[1]Geodetic Institute, Leibniz Universität Hannover, Germany - (wahbah, ernst, vogel, neumann, alkhatib)@gih.uni-hannover.de
[2]Leibniz Universität Hannover, Germany - lukas.ramme@gmx.de

Keywords: Digital Maps, Geo-referencing, LIDAR, Simulation

Abstract

Autonomous vehicles (AVs) require accurate global pose estimation to operate effectively. A common approach involves utilizing perception sensors to extract environmental features which are used to geo-reference the vehicle with pre-defined maps. High Definition (HD) maps are frequently used for this purpose due to their detailed feature sets. However, the use of HD maps presents challenges as they are not frequently unavailable and their custom generation involves considerable complexity and cost. Conversely, Level of Detail 2 (LoD2) maps are freely available for numerous cities and are regularly updated, hence they can offer a potential solution. However, due to their geometric simplifications, the applicability of LoD2 maps for AV pose estimation remains uncertain. In this study, we investigate the impact of these simplifications and assess the suitability of LoD2 maps for AV pose estimation. We perform a comparative analysis between HD and LoD2 maps in a simulated CARLA environment, employing an Error State Kalman Filter (ESKF) to estimate the position, velocity, and orientation of an AV. We showcase our results using ideal sensors to isolate the effects of LoD2 maps, as well as realistic sensors to evaluate their performance in real-world scenarios.

1. Introduction

In Autonomous Vehicles (AVs), lower levels of driving automation require the vehicle to perform basic driving tasks such as emergency braking, lane centering, and adaptive cruise control. These tasks are performed based on real-time assessments of the vehicle's immediate environment and do not require precise knowledge of the vehicle's position or long-term planning. Conversely, higher levels of automation require AVs to take full control and perform the navigation tasks autonomously. To plan the journey and execute the required driving maneuvers successfully, the AV's pose (typically consisting of the position, speed, and orientation) must be known within a global reference frame. Additionally, safe operation necessitates high positional accuracy, of $0.1\ m$ at 95% confidence level (Reid et al., 2019). To meet these requirements, geo-referencing techniques are frequently utilized. Geo-referencing involves using sensors to collect geo-spatial data and aligning it with a predefined reference frame within a map to determine the vehicle's pose. A prominent example of these techniques is Global Navigation Satellite Systems (GNSS), which can provide up to $3\ cm$ accurate global position measurements, especially when augmented with techniques like Differential GNSS (D-GNSS) (SAPOS, 2015). However, their performance suffers in urban environments due to factors such as signal blockage and multipath reflections, which reduce accuracy and introduce service interruptions. To mitigate these challenges, self-contained sensors are employed. Unlike GNSS, these sensors don't rely on externally transmitted signals, instead, they collect measurements of their surroundings and identify landmarks, which are then matched and aligned with the pre-existing map. For this geo-referencing approach to be viable in AVs, the maps should meet several criteria; they must describe accurate and distinct landmarks for the sensors to identify, be frequently updated to match changes in the real world, be computationally efficient to search and store, and ubiquitously available.

A popular choices of maps in the literature is High-Definition (HD) maps. While they lack a unified definition, HD maps are described as multi-layered representations of the environment containing heterogeneous data regarding road connectivity, road features such as road signs, intersections, and traffic lights, accurate lane information such as lane lines and road markings, and sensor representations of the environment including 3D point clouds from laser scanners, as well as mono or stereo images (Liu et al., 2020, Elghazaly et al., 2023). To utilize HD maps, some studies detect lane lines, either by utilizing LiDARs (Ghallabi et al., 2018, Rohde et al., 2016), or though cameras (Suhr et al., 2017, Han et al., 2018), and in some cases, the detected features are extended to include road signs as well (Guo et al., 2021). While previous works focused on road-specific features, various studies employed alternative methods. For instance, in (Wan et al., 2018, Chen et al., 2021, Tao et al., 2022), the authors rasterized LiDAR scans into 2D images, while in (Li et al., 2018), non-descriptive features were extracted from camera images. Nevertheless, the processed measurements are then matched to a map in order to estimate the pose of the vehicle. Due to the variation in utilized features, the previous works had to mostly construct their own maps. The maps were created prior to the experiments by scanning the environment for the required features, and then estimating the trajectory using a mixture of selecting a path with optimal D-GNSS coverage, conducting multiple iterations, manually measuring and geo-tagging the desired map features, post-processing techniques, or relying on third-party companies. To further refine the estimated pose, data from Inertial Measurement Units (IMU), wheel odometry, and in some cases GNSS, are integrated using a variety of filters, such as Kalman Filters (KF), Extended Kalman Filters (EKF), or Particle Filters (PF).

The literature review reveals that despite their popularity, HD maps have several drawbacks (Elghazaly et al., 2023):

- Lack of standardization: the literature varies significantly in terms of the data included within the map and the manner in which the data is represented.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-915-2025 |

- High costs and labor-intensive: Due to the non standardized nature, nearly every HD map-based approach required the researchers to create custom reference maps, often necessitating specialized equipment, sophisticated post-processing techniques, and multiple scans. Replicating these procedures on a wider scale with frequent updates presents substantial challenges.
- Storage limitations: In scenarios where access to images and point clouds is necessary, HD maps are computationally expensive to store, and in some cases, they require constant communication with a server.

Given the challenges associated with HD maps, researchers have explored alternative mapping solutions that balance accuracy, updatability, and computational efficiency. For instance, (Kassas et al., 2020) relied on cellular Signals of Opportunity (SOP) to estimate the vehicle’s pose, while (Yoneda et al., 2018) utilized Millimeter-Wave Radar (MWR) to detect objects such as poles and trees and align them with a map. Another promising approach was proposed by (Javanmardi et al., 2019, Bureick et al., 2019), where planar maps containing simplified representations of the environment were employed. This approach simplifies the mapping procedure, with the added benefit of reducing the map storage size and feature alignment time. In the case of (Javanmardi et al., 2019), the map was acquired by scanning the test environment with a LiDAR and subsequently post-processing the scans using a Random Sample Consensus (RANSAC) algorithm to estimate building planes. Conversely, authors in (Bureick et al., 2019) employed Level of Detail 2 (LoD2) maps and Digital Terrain Models (DTM).

The use of LoD2 maps is particularly noteworthy, as these maps offer several advantages over HD maps, such as being standardized and adhering to the CityGML standard (Kolbe et al., 2021), requiring less storage space, being easier to generate and thus maintain, and having freely available LoD2 maps for numerous cities worldwide (Wysocki, 2024, Peters et al., 2022), some of which are updated annually (3D-Gebäudemodelle LoD2 Deutschland, n.d.), thereby eliminating the added complexity of generating a custom map. Nevertheless, the impact of LoD2 maps’ geometrical simplifications on geo-referencing accuracy remains unexplored. This study aims to quantify this impact by comparing the performance of LoD2 and HD maps in a simulated urban environment. The comparison is conducted utilizing the Error-State Kalman Filter (ESKF) presented in (Ernst et al., 2023). This filter employs the implicit measurement approach from (Bureick et al., 2019), but replaces the IEKF with an ESKF to reduce linearization errors, and substitutes the constant velocity model with a kinematic model based on IMU measurements. By using one estimation method for both maps, the effects of geometrical simplifications are isolated from other factors specific to the geo-referencing and filtering approach..

This article is organized as follows: In section 2, we discuss the simulation environment and map generation, and we introduce the pose estimation filter and the evaluation criteria used for assessing the performance of the filter. In section 3, we present the results from two different scenarios. Finally, in section 4 we conclude the article with a summary and an outlook.

2. Methodology

2.1 ESKF & Implicit measurements

Given that we are interested in global pose estimation, it is beneficial to distinguish between the different reference frames and how the quantities are represented in them. In our case, the global frame of reference is denoted by the letter o, while the IMU and LiDAR frames are denoted by i and l, respectively. Since a LiDAR measurement $\mathbf{z}$ is measured with respect to the l frame, we have to calculate the representation of that measurement in the o frame to carry out our estimation procedure. To do so, we use the 4×4 transformation matrix ${}^o_lT_k$ as follows:

$$ {}^o\mathbf{z}_k = {}^o_l T_k \ {}^o\mathbf{z}_k = \begin{bmatrix} {}^o_lR_k & {}^o_l\mathbf{t}_k \\ 0 & 1 \end{bmatrix} {}^o\mathbf{z}_k \tag{1} $$

where, ${}^o\mathbf{z}_k$ is measurement in the global frame, ${}^o\mathbf{z}_k$ is the measurement in the LiDAR frame, ${}^o_lR_k$ and ${}^o_l\mathbf{t}_k$ are the 3×3 rotation matrix and the 3×1 translation vector of frame l with respect to frame o, respectively, and the subscript k denotes the time step at witch the measurement occurred.

Naturally, the transformation matrix is not known a priori, and relies on the states estimated by our filter. In our context, the state vector $\mathbf{x}$ from (Ernst et al., 2023) is defined as:

$$ {\mathbf{x}_k}^T = \begin{bmatrix} {}^o_l{\mathbf{t}_k}^T & {}^o_l{\mathbf{v}_k}^T & {}^o_l{\mathbf{q}_k}^T & {}^i{\mathbf{a}_{b,k}}^T & {}^i{\boldsymbol{\omega}_{b,k}}^T \end{bmatrix} \tag{2} $$

where, ${}^o_l{\mathbf{t}_k}^T$ is the 3D translation vector, ${}^o_l\mathbf{v}_k$ is the 3D velocity vector, ${}^o_l\mathbf{q}_k$ is the orientation represented in a 4D quaternion vector, and ${}^i\mathbf{a}_{b,k}$ and ${}^i\boldsymbol{\omega}_{b,k}$ are the 3D accelerometer and 3D gyroscope biases in the IMU frame i, respectively. Unlike other KFs, an ESKF does not directly estimate the states, instead it estimates the prediction errors, which are subsequently used to correct the states. Since linearization around the errors tends to be more stable, this method leads to improved accuracy (Solà, 2017). We therefore introduce the error state vector $\delta\mathbf{x}$, which is defined as:

$$ \delta{\mathbf{x}_k}^T = \begin{bmatrix} \delta^o_l{\mathbf{t}_k}^T & \delta^o_l{\mathbf{v}_k}^T & \delta^o_l{\boldsymbol{\theta}_k}^T & \delta^i{\mathbf{a}_{b,k}}^T & \delta^i{\boldsymbol{\omega}_{b,k}}^T \end{bmatrix} \tag{3} $$

where the operator $\delta(\cdot)$ indicates the error of a component, and $\delta^o_l\boldsymbol{\theta}_k$ is the orientation error represented as axis-angle.

The filter iterates between a prediction step, where the system physical model is used to predict the pose, and a correction step, where our measurements are geo-referenced to estimate the errors and correct the prediction, Figure 1 depicts the flowchart of our filter, which is explained in this section. The physical model used in the prediction step is a 3D point kinematic motion model $f()$:

$$ \mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1},\ \mathbf{u}_k) \tag{4} $$

where $\mathbf{u}_k$ is the input vector consisting of the IMU measured acceleration $\boldsymbol{a}_k$ and rotational velocity $\boldsymbol{\omega}_k$. These measurements are processed by removing the estimated biases and gravitational vector, and are then integrated to estimate the linear velocity, and the linear and angular displacements. Since the prediction step falls outside of our scope, we refer to (Ernst et al., 2023) for the full mathematical description. Next, the covariance matrix P is computed as:

$$ P_{k|k-1} = F_x P_{k-1} F_x^T + F_n Q_n F_n^T \tag{5} $$

where F_x is the Jacobian of $f()$ with respect to the error estimates at time k, F_n is the Jacobin of $f()$ with respect to noise model at time k, and Q_n is the model noise covariance matrix.

In the subsequent phase, we initiate the correction step by geo-

https://doi.org/10.5194/isprs-annals-X-G-2025-915-2025

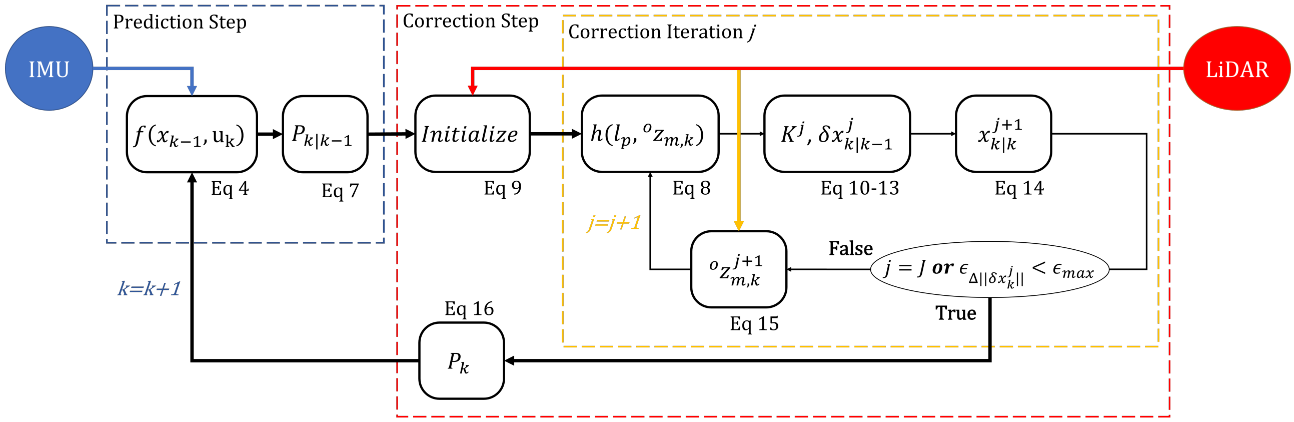


Figure 1. Flowchart of the filter algorithm

referencing the LiDAR measurements. To enhance computational efficiency, we downsample the measured point cloud to M points utilizing a voxel grid filter. Unlike conventional methods that use the centroid of each voxel, we select the nearest measured point within each voxel to preserve the original data and prevent the loss of details. To align the downsampled point cloud with our map frame, it is transformed into the global coordinate system, as described in equation 1. Then, for each measurement point, we identify the closest map feature within a predefined spherical threshold radius r, assigning the measurement to the corresponding feature. For LoD2 maps, these features comprise the planes of buildings or the DTM cells, whereas in HD maps, the features are represented by PCL points. To incorporate these assignments, we employ the implicit plane equation as our measurement function h:

$$h = \begin{bmatrix} n_x, n_y, n_z, d \end{bmatrix} \begin{bmatrix} {}^{o}\mathbf{z}_{m,k} \\ 1 \end{bmatrix} \tag{6}$$

where, n_x, n_y, n_z are the plane normal vector components, d is the plane distance from the origin, and ${}^{o}\mathbf{z}_{m,t}$ is the m^{th} measured point assigned to that plane. The update step is based on the IEKF from (Bureick et al., 2019), hence it iterates until the change in the error state magnitude $\epsilon_{\Delta||\delta x_k||}$ is below a certain threshold ϵ_{max}, or it reaches the maximum number of iterations J. The first iteration is initialized as follows:

$$\mathbf{x}_{k|k}{}^{j} = \mathbf{x}_{k|k-1}, \quad \delta\mathbf{x}_k^{j-1} = \mathbf{0}, \quad {}^{o}\mathbf{z}^{j}_{m,k} = {}^{o}\,\mathbf{z}_{m,k} \tag{7}$$

where j is the number of the current iteration. For each subsequent iteration, we calculate $\delta\mathbf{x}_{k|k}$ using:

$$P_z^j = A^j P_{k|k-1}{}^{j} A^{j^T} + B^j Q_z B^{j^T} \tag{8}$$

$$K^j = P^j_{k|k-1} A^{j^T} (P_z^j)^{-1} \tag{9}$$

$$\mathbf{W}^j = h^j + B^j({}^{o}\mathbf{z}^j_{m,k} - {}^{o}\,\mathbf{z}_{m,k}) + A^j \delta\mathbf{x}_k{}^{j-1} \tag{10}$$

$$\delta\mathbf{x}_k{}^{j} = -K^j \mathbf{W}^j \tag{11}$$

where, A^j and B^j are the Jacobians of $h()$ with respect to $\delta\mathbf{x}_{k|k}{}^{j}$ and ${}^{o}\mathbf{z}^j_{m,k}$ respectively, and Q_z is the measurement noise covariance matrix. We then adjust our estimates vector using the estimated errors:

$$\mathbf{x}_{k|k}{}^{j+1} = \mathbf{x}_{k|k-1} \bigoplus \delta\mathbf{x}_k{}^{j} \tag{12}$$

the $\bigoplus$ symbol indicates that each adjustment is carried out separately, as required by the estimated quantity. If the stopping conditions aren't met, we update the observations for the next iteration:

$${}^{o}\mathbf{z}^{j+1}_{m,k} = {}^{o}\,\mathbf{z}^j_{m,k} - Q_z (B^j)^T P_z^{j^{-1}} W^j \tag{13}$$

However, if the iterations are completed, we compute the new P_k using:

$$P_k = (I - KA) P_{k|k-1} (I - KA)^T + K(BQ_zB^T)K^T \tag{14}$$

The filter then loops back to the prediction step based on a chosen frequency, while the update step gets carried out whenever a new measurement is available for correction.

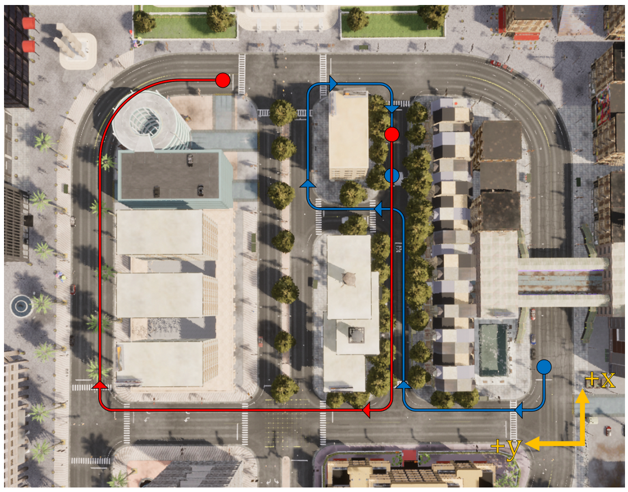


Figure 2. The two trajectories. Trajectory $P1$ shown in blue and Trajectory $P2$ shown in red

2.2 Simulation environment

There are multiple motivations for choosing a simulated environment. First, we require a dense and highly accurate point cloud to create and compare the maps. Acquiring such point clouds in the real world introduces new sources of errors, thus affecting our comparison. A simulated environment on the other hand, provides the absolute ground truth. Secondly, since the algorithm requires assigning LiDAR points to planes, points belonging to dynamic objects such as pedestrians and vehicles might be wrongly assigned. The problem of point assignments affects both maps and is an attribute of the chosen filter; hence it falls outside the scope of this article, which is specifically concerned with the effect of geometrical simplifications. Another motivation is the ability to isolate and focus solely on the effects of LoD2 maps, by simulating ideal sensor measurements that exclude sensor noise influences. Additionally, a simulation allows us to repeat the experiment with added sensor noise and show how the two maps compare in more realistic scenarios. Finally, a simulation enables a more generalized comparison by allowing the selection of an environment with diverse building structures and road infrastructure.

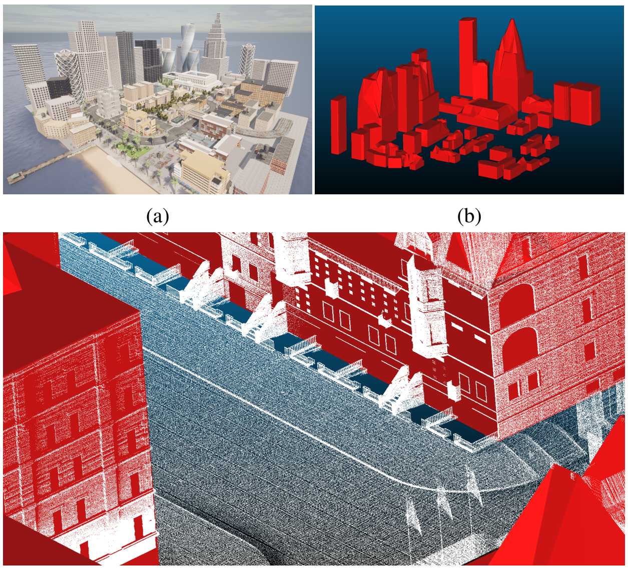
CARLA (Dosovitskiy et al., 2017) is a simulation environment for autonomous driving, developed using Unreal Engine. We selected this software due to its extensive validation in the literature and its flexible programming interface, which enables the selection of specific sensors and map configurations tailored to our scope. We decided to use the pre-loaded map "Town10" as it represents a diverse urban city, including skyscrapers, residential, industrial, and public buildings, as well as varying road and junction sizes. In addition, we used the built-in autonomous driving function to operate the test vehicle along two different trajectories, shown in Figure 2. The first trajectory $P1$ (blue) follows a curvy path between residential buildings, while the second trajectory $P2$ (red) consists of longer straight paths, wider turns, a larger intersection, and a mixture of residential buildings and high-rises. $P1$ was $419\ m$ in length and took $56.7\ s$ to complete, while $P2$ was $417\ m$ and took $70.75\ s$. The scenarios were limited in length by the size of the map but were carefully designed to represent various driving conditions, including low-speed and high-speed segments, multiple start-and-stop instances, as well as the specific trajectory features mentioned previously.

2.3 Maps Generation

Since we are working in a simulated environment, we had to create both maps. For the HD map, we acquired measurements using a high-density scanning LiDAR. Since CARLA does not simulate the rolling shutter effect, all LiDAR points within an epoch shared the same pose. Given that we have access to the true LiDAR pose and noise-free distance measurements, the HD map was constructed by transforming the LiDAR frame l measurements into the global frame o using equation 1, then collating the results into a comprehensive map.

As for the LoD2 map, we used the buildings' 3D models and manually traced each building footprint. Subsequently, we simulated an airborne LiDAR to capture measurements of building heights and roof geometries. Utilizing ArcGIS, we then converted the acquired aerial map into polygon meshes, extending the building footprints to the roofs in accordance with the CityGML standard (Kolbe et al., 2021). among the original map, the HD point cloud, and the LoD2 map. Upon examining Figure 3a and Figure 3b, it becomes apparent that certain buildings are absent from the LoD2 map. Nonetheless, the absence of these buildings does not impact the performance of our filter. When comparing the HD and LoD2 maps, as depicted in Figure 3c, it is evident that the HD map captures more detailed features than the geometrically simplified LoD2 map. The influence of these additional features on geo-referencing and pose estimation is our main scope of interest.

Lastly, given that our geo-referencing technique relies on plane equations, it was necessary to provide the filter with plane normals. For LoD2 maps, this process is straightforward, as the



(a) (b) (c)

Figure 3. The generated maps. Figure 3a shows the original "Town10" map. Figure 3b shows the generated LoD2 map. Figure 3c highlights the differences between the HD map point cloud (white) and the simplified LoD2 map (red)

environment is represented by planes. In contrast, for HD maps we estimated the plane normals offline and stored the results to enhance the filter's runtime efficiency.

2.4 Parameters Tuning

By analyzing Equation 9, it is evident that the Kalman gain K_j, which determines the extent to which a measurement corrects a predicted state, is significantly influenced by the noise matrices Q_n and Q_z. It is therefore important to set these matrices to achieve good performance. While the conventional approach of populating these matrices with sensor variances generally leads to convergence, it is advised to further tune these values to gain better performance (Abbeel et al., 2005). The adjusted values are intended to account for additional perturbations arising from model mismatch, filter linearization, and discretization errors. In our implementation, the Q_n matrix is diagonal and consists of the accelerometer and gyroscope bias noise densities σ_{a_n}, σ_{ω_n} and bias instabilities σ_{a_w}, σ_{ω_w}:

$$Q_n = diag(\sigma_{a_n}^2 \Delta t^2,\ \sigma_{\omega_n}^2 \Delta t^2, \sigma_{a_w} \Delta t, \sigma_{\omega_w} \Delta t) \tag{15}$$

where, the scalar Δt is the time step between each updates, which we set to match the accelerometer and gyroscope sampling rate. The Q_z matrix on the other hand is obtained by converting the LiDAR distance σ_d, horizontal angle σ_ϕ, and and azimuth angle σ_θ uncertainties into their Cartesian equivalents, following the approach provided in (Ernst et al., 2024).

To fine-tune our filter, a comprehensive parametric sweep was conducted across the aforementioned parameters and the point-plane assignment threshold radius r. The range of parameters explored during this sweep is presented in Table 1. This sweep was performed independently on $P2$ for four distinct scenarios: an HD map with noiseless measurements, an LoD2 map with noiseless measurements, an HD map with noisy measurements, and an LoD2 map with noisy measurements. The rationale for

Parameter	HD Map		LoD2 Map	
	Min	Max	Min	Max
σ_{a_n} $(m/(s^2\sqrt{Hz}))$	4.8	91.2	4.8	4.8
σ_{ω_n} $(deg/(s\sqrt{Hz}))$	8	80	16	16
σ_{a_w} (m/s^2)	26	416	26	26
σ_{ω_w} (deg/sec)	34	323	68	68
σ_d (mm)	8.5	8.5	550	2500
σ_ϕ $(mdeg)$	48.7	48.7	50	5050
σ_θ $(mdeg)$	29.8	29.8	50	5050
r (m)	0.1	0.1	0.05	0.7

Table 1. List of the tuned parameters and the start and end range for the sweep function

selecting different parameter ranges for each scenario is detailed in Section 3.1. In each case, the parameters that minimized the average translation errors, as defined in Equation 17, were selected.

2.5 Evaluation criteria

To evaluate the filter's performance, we compute the errors using the Euclidean distance, denoted as $L2_k$ between the estimated and true values of a vector at time k:

$$L2_{\boldsymbol{u},k} = \sqrt{(x_k - x_{r,k})^2 + (y_k - y_{r,k})^2 + (z_k - z_{r,k})^2} \quad (16)$$

Here, $x_k,\ y_k,\ z_k$ represent the three-dimensional components of the estimated vector $\boldsymbol{u}$, while $x_{r,k},\ y_{r,k},\ z_{r,k}$ denote the corresponding components of the true vector. The vector $\boldsymbol{u}$ represents any of the translation, velocity, or orientation (expressed in Euler angles) vectors. Here, we assume that the small-angle approximation applies, and difference between the estimated and true orientation values is minimal, typically within a few degrees. Additionally, we assume that the vehicle does not encounter gimbal lock conditions, as it is driving on relatively flat ground. Hence, the Euclidean distance provides a reliable and consistent metric for representing the filter's performance across the estimated variables. To represent the error over the complete trajectory, we calculate the Mean Euclidean Distance (MED) of each vector:

$$MED_{\boldsymbol{u}} = \frac{1}{K}\sum_{k=1}^{K} L2_{\boldsymbol{u},k} \quad (17)$$

To investigate the loss of accuracy associated with LoD2 maps, we introduce the generalization factor γ:

$$\gamma_{\boldsymbol{u}} = \frac{MED_{\boldsymbol{u},LoD2}}{MED_{\boldsymbol{u},HD}} \quad (18)$$

Ideally, we seek γ to be as low as possible, which would suggest a reduced negative impact from the geometrical simplifications inherent in LoD2 maps. It is important to note that while calculating γ using MED offers greater robustness to outliers, the value of γ can become skewed, particularly in the context of ideal measurements where the HD map may significantly outperform LoD2 maps. Therefore, it is crucial to employ both metrics when evaluating the suitability of LoD2 maps.

3. Results

For our tests, we conducted two experiments for each trajectory: one with ideal measurements, and one with added sensor noise. The first experiment demonstrates the impact of LoD2 simplifications on geo-referencing, while the second experiment compares LoD2 maps to HD maps in realistic scenarios. The simulation consisted of a sedan vehicle equipped with a ”Velodyne VLP16” LiDAR, and a ”Microstrain 3DM-GQ4-45” IMU. The parameters for these sensors were based on their prospective data-sheets, and are provided in Table 2. For tests with ideal measurements, the noise, instability, and uncertainty parameters were set to zero. The CARLA simulation physics engine time-step was set to 1 ms in synchronous mode, and the ESKF filter was implemented in an unoptimized MATLAB code. The tuned filter parameters from Section 2.1 were employed as shown in Table 3. It is important to distinguish between the sensor simulation parameters in Table 2 and the tuned filter parameters in Table 3. The former were used exclusively for simulating sensor behavior, while the latter were adjusted to account for additional perturbations and enhance filter performance. It is important to note that the real-time capabilities of the geo-referencing techniques are not evaluated in this section, as they are influenced by external factors such as hardware choice and code optimization. These aspects fall outside the scope of this research, which aims to benchmark the performance of LoD2 maps regarding their geo-referencing accuracy.

IMU Settings	
Frequency	100 Hz
Accelerometer noise density	$(6.16,\ 4.61,\ 3.62) \cdot 10^{-3}\ m/(s^2\sqrt{Hz})$
Accelerometer bias instability	$(22.26,\ 20.35,\ 36.36) \cdot 10^{-3}\ m/s^2$
Gyroscope noise density	$(14.89,\ 19.48,\ 13.75) \cdot 10^{-3}\ deg/(s\sqrt{Hz})$
Gyroscope bias instability	$(70.47,\ 65.31,\ 68.18) \cdot 10^{-3}\ deg/s$
LiDAR Settings	
Frequency	10 Hz
Range	100 m
Num. Vertical scan lines	16
Upper FOV	$15°$
Lower FOV	$-15°$
Range uncertainty	8.5 mm
Horizontal angle uncertainty	48.7 $mdeg$
Azimuth angle uncertainty	29.8 $mdeg$

Table 2. Sensor parameters used for the experiments

3.1 Parameters Tuning

Before analyzing the test results, it is crucial to examine Table 1, which showcases the range of parameters within the sweep. Notably, for the HD map, parameters pertaining to the prediction step σ_{a_n}, σ_{a_w}, σ_{ω_n}, and σ_{ω_w} were varied while maintaining the measurement parameters, σ_d, σ_ϕ, and σ_θ, at their default values. This is contrasted with the LoD2 maps, where the opposite was applied. This methodology was developed after observing that when all parameters were varied, certain parameters consistently remained at their default value, hence the parameters were fixed to reduce the time required for the tuning process.

These findings suggest that when the HD map is utilized, the filter places greater emphasis on the correction step due to the map's accurate representation of the environment, which reduces reliance on the prediction step to minimize errors arising from filter linearization and physical model mismatch. Conversely, the use of the LoD2 map necessitated an increase in measurement noise, compensating for geometric simplifications and resultant discrepancies between the map and the real environment as measured by the LiDAR. Consequently, the prediction parameters were kept at their minimum to maintain the Kalman gain at an optimized value. It should be noted that the parameter tuning was bounded below by the sensors' default

https://doi.org/10.5194/isprs-annals-X-G-2025-915-2025 | © Author(s) 2025. CC BY 4.0 License.

Filter Tuned Parameter	Noiseless Measurementes		Noisy Measurements	
	HD Map	LoD2 Map	HD Map	LoD2 Map
Point-plane assignment Threshold r	$0.1m$			0.4 m
Accelerometer noise density σ_{a_n}	$76.8\ \frac{mm}{s^2\sqrt{Hz}}$	Values from Table 2	$86.4\ \frac{mm}{s^2\sqrt{Hz}}$	Values from Table 2
Accelerometer bias instability σ_{a_w}	$312\ \frac{mm}{s^2}$	Values from Table 2	$390\ \frac{mm}{s^2}$	Values from Table 2
Gyroscope noise density σ_{ω_n}	$16\ \frac{mdeg}{s\sqrt{Hz}}$	Values from Table 2	$64\ \frac{mdeg}{s\sqrt{Hz}}$	Values from Table 2
Gyroscope bias instability σ_{ω_w}	$68\ \frac{mdeg}{s}$	Values from Table 2	$272\ \frac{mdeg}{s}$	Values from Table 2
Range uncertrainty σ_d	$8.5\ mm$	$0.7\ m$	$8.5\ mm$	$2.3\ m$
Horizontal angle uncertainty σ_ϕ	$48.7\ mdeg$	$0.55\ deg$	$48.7\ mdeg$	$4.05\ deg$
Azimuth angle uncertainty σ_θ	$29.8\ mdeg$	$0.55\ deg$	$29.8\ mdeg$	$4.05\ deg$

Table 3. Tuned filter parameters used for the different test scenarios

Eval. Criteria	Noiseless Measurementes				Noisy Measurements			
	Traj. P1		Traj. P2		Traj. P1		Traj. P2	
	HD Map	LoD2 Map	HD Map	LoD2 Map	HD Map	LoD2 Map	HD Map	LoD2 Map
$MED_t\ (mm)$	0.4	137	0.2	92	1.9	153	1.8	97
$MED_v\ (mm/s)$	14.3	76	7.7	30	25.9	64	22.6	40
$MED_\theta\ (mdeg)$	4	148	1.7	57	8.1	178	7.2	123
γ_t	342.5		460		80.5		53.9	
γ_v	5.31		3.90		2.47		1.77	
γ_θ	37.0		33.5		22.0		17.1	

Table 4. Evaluation results for the different test scenarios

values to prevent overconfident predictions; thus, the search space for prediction parameters was not expanded.

3.2 Ideal Measurements

In this scenario, HD map substantially outperformed the LoD2 map, achieving mean translational errors of $0.4\ mm$ and $0.2\ mm$ across $P1$ and $P2$, respectively, compared to the LoD2 map's mean translational errors of $137\ mm$ and $92\ mm$, respectively. Consequently, γ_t were calculated to be 342.5 for $P1$ and 460 for $P2$. The effects on velocity and orientation predictions were comparatively less pronounced, with γ_ω and γ_θ of 5.31 and 37 for $P1$, and 3.9 and 33.5 for $P2$, respectively. The significant discrepancies in the generalization factors across different estimated quantities can be attributed to the characteristics of the prediction and correction models used in this study. As detailed in Section 2.1, the prediction phase utilizes accelerometer and gyroscope readings, which are integrated over time to derive the necessary estimates. Given that acceleration is a third-order derivative, the first-order translational estimate is inherently damped by the covariance propagation and is less instantaneously susceptible to prediction errors when compared to velocity and orientation estimates. In contrast, the measurement model provides direct corrections for translation during the correction step. Therefore, the geometric simplification errors introduced by LoD2 in the measurement step have a more substantial impact on translation estimates compared to the orientation and velocity estimates.

To closely examine the specific sources of errors in our test scenario, we analyze the translational errors between the HD map and the LoD2 map along $P2$, as depicted in Figure 4. In the case of the HD map, noise is observed between the time periods of $24\ sec$ to $35\ sec$, $44\ sec$ to $48\ sec$, and between $58\ sec$ to $66\ sec$. These intervals correspond to situations where the vehicle underwent accelerations, stops, and turns. The primary source of these errors can be attributed to the limitations of our prediction model in perfectly capturing such maneuvers. Nonetheless, these errors are within millimeters and are hence tolerable. In contrast, the LoD2 map translational errors significantly increase when entering the large intersection in the south, around $30\ sec$. This can be explained by the reduction in surrounding building density, the added complexity of identifying the correct planes around corners, and the increased complexity of the building facades, which contain varying protrusions and irregular shapes. This explanation is supported by examining the average distance errors between points assigned to the LoD2 model and their true measured locations. Prior to entering the intersection, the average distance point-to-LoD2 model was approximately $0.025\ m$, while the mismatch between the LoD2-to-true locations was about $0.052\ m$. After entering the intersection, the average distance to the LoD2 model remained at $0.025\ m$, whereas the distance to the true locations increased significantly, averaging around $0.08\ m$.

3.3 Noisy Measurements

As expected, Table 4 shows that the introduction of noise results in increased errors across all estimated quantities, with LoD2 map exhibiting a greater magnitude of error increase compared to the HD map. However, the generalization factor is significantly reduced for all estimated quantities, a phenomenon attributed to the skewness described in Section 2.5. Figures 5 and 6 illustrate how in LoD2, noisy measurements introduced oscillations in pose estimates along $P1$, likely from broader point assignments (increased r) that incorporated non-LoD2 features such as balconies, window frames, bay windows, and air conditioning units. This is indicated by the increase of the average point-to-LoD2 plane from $2.5\ cm$ in noiseless scenario, to $0.1\ cm$ in noisy scenarios.

4. Discussion & Conclusion

Defining a required estimation accuracy for AVs poses significant challenges due to varying standards that depend on operating environments, vehicle types, regional regulations, and the available road infrastructure. For instance, (EUSPA, 2021) recommends a positional localization accuracy of $0.2\ m$ at 99.9% availability, while (5G-PPP, 2015) suggests an accuracy of $0.3\ m$. In contrast, the $0.1\ m$ accuracy requirement at 99.5% availability proposed by (Reid et al., 2019) aligns best with the specific demands of this study, as it is tailored to “local roads”

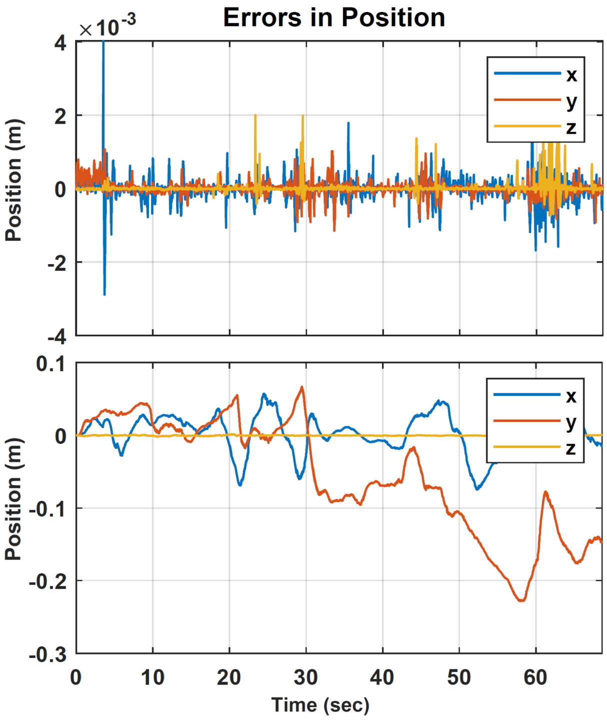


Figure 4. Error in position estimates along trajectory $P2$ with noiseless measurements, comparing HD (top), and LoD2 map (bottom)

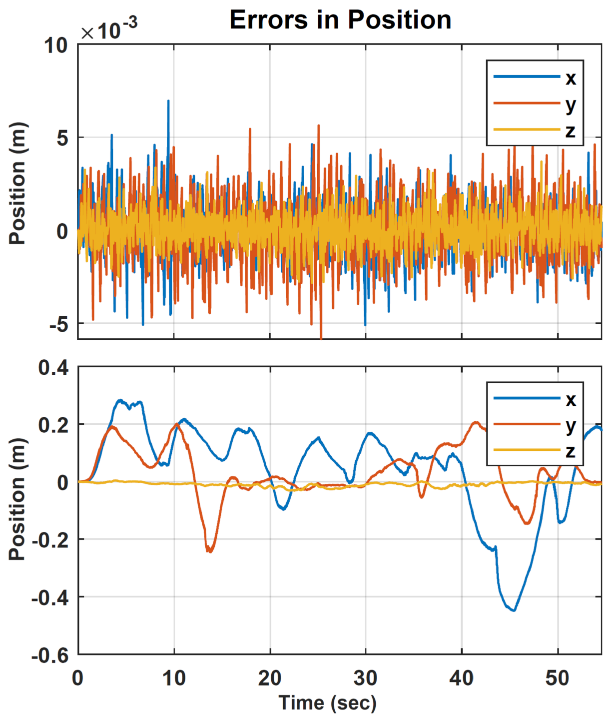


Figure 5. Error in position estimates along trajectory $P1$ with noisy measurements, comparing HD (top), and LoD2 map (bottom)

typically characterized by building density and limited GNSS availability. These constraints support the potential utility of LoD2 maps in such contexts.

Our study demonstrates that while current LoD2 map implementations may not yet enable full autonomy, they show significant promise as a lower-cost alternative to HD maps.Notably, the estimated positional accuracy remained within the alert limit of $0.29\ m$ for the majority of the time, and the estimated angular accuracy adhered to the $0.5°$ requirement suggested by (Reid et al., 2019). These results suggest that with error mitigation, LoD2 maps could be effectively utilized in AVs.

The primary sources of error for LoD2 maps stemmed from incorrect point-to-plane assignments during interactions with low-density or geometrically complex building facades, particularly during turns. This misalignment often arose due to LoD2's simplified geometry, which lacks the granularity found in HD maps. To mitigate these issues, we recommend the integration of advanced point assignment algorithms for better estimation and extraction of planer surfaces from LiDAR data. Additionally, incorporating auxiliary sensors, such as a wheel encoder, could enhance orientation estimation and improve the point assignments during complex maneuvers. The inclusion of cameras may also support feature rejection, further reducing errors by filtering out non-essential features.

In conclusion, while LoD2 maps exhibit considerable potential as a cost-effective solution for AV navigation, further research is necessary to enhance error mitigation strategies, particularly through optimizing point assignments and refining sensor fusion techniques. Such research would offer valuable insights into the capabilities of LoD2 maps and their role in the navigation paradigm, whether as a replacement for HD maps or as a complementary layer within them, aimed at reducing mapping requirements in urban environments.

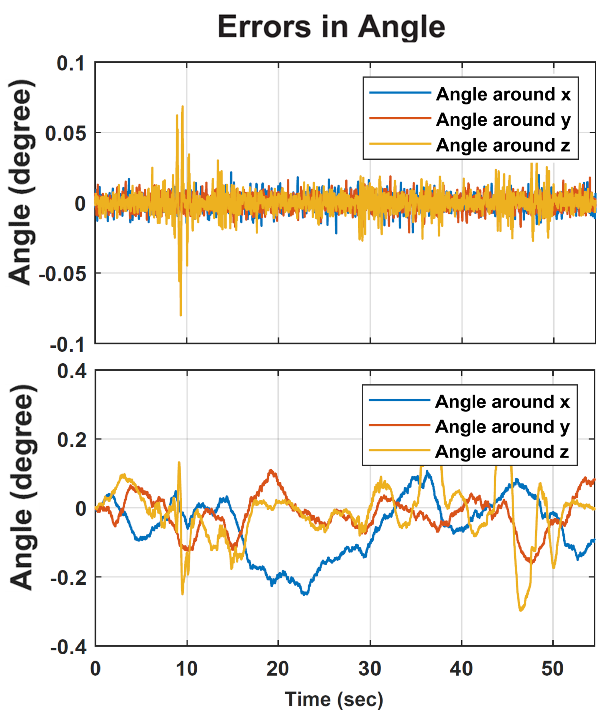


Figure 6. Error in orientation estimates along trajectory $P1$ with noisy measurements, comparing HD (top), and LoD2 map (bottom)

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-G-2025-915-2025 | © Author(s) 2025. CC BY 4.0 License.

References

3D-Gebäudemodelle LoD2 Deutschland, n.d. https://gdz.bkg.bund.de/index.php/default/3d-gebaudemodelle-lod2-deutschland-lod2-de.html (Accessed: 2 September, 2024).

5G-PPP, 2015. 5G Automotive Vision. https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf (Accessed: 11 September, 2024).

Abbeel, P., Coates, A., Montemerlo, M., Ng, A. Y., Thrun, S. et al., 2005. Discriminative training of kalman filters. *Robotics: Science and systems*, 2, 1.

Bureick, J., Vogel, S., Neumann, I., Unger, J., Alkhatib, H., 2019. Georeferencing of an Unmanned Aerial System by Means of an Iterated Extended Kalman Filter Using a 3D City Model. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 87(5), 229–247. https://doi.org/10.1007/s41064-019-00084-x.

Chen, X., Vizzo, I., Läbe, T., Behley, J., Stachniss, C., 2021. Range Image-based LiDAR Localization for Autonomous Vehicles. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 5802–5808.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An open urban driving simulator. *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16.

Elghazaly, G., Frank, R., Harvey, S., Safko, S., 2023. High-Definition Maps: Comprehensive Survey, Challenges, and Future Perspectives. *IEEE Open Journal of Intelligent Transportation Systems*, 4, 527–550.

Ernst, D., Vogel, S., Neumann, I., Alkhatib, H., 2023. Error state kalman filter with implicit measurement equations for position tracking of a multi-sensor system with imu and lidar. *2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 1–6.

Ernst, D., Vogel, S., Neumann, I., Alkhatib, H., 2024. Empirical uncertainty evaluation for the pose of a kinematic LiDAR-based multi-sensor system. *Journal of Applied Geodesy*, 18(4), 629–642.

EUSPA, 2021. Report on Road User Needs and Requirements. https://www.euspa.europa.eu/sites/default/files/uploads/road-annex7.pdf (Accessed: 11 September, 2024).

Ghallabi, F., Nashashibi, F., El-Haj-Shhade, G., Mittet, M.-A., 2018. Lidar-based lane marking detection for vehicle positioning in an hd map. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2209–2214.

Guo, C., Lin, M., Guo, H., Liang, P., Cheng, E., 2021. Coarse-to-fine semantic localization with hd map for autonomous driving in structural scenes. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1146–1153.

Han, S.-J., Kang, J., Jo, Y., Lee, D., Choi, J., 2018. Robust Ego-motion Estimation and Map Matching Technique for Autonomous Vehicle Localization with High Definition Digital Map. *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 630–635.

Javanmardi, E., Gu, Y., Javanmardi, M., Kamijo, S., 2019. Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area. *IATSS Research*, 43(1), 1-13.

Kassas, Z. Z. M., Maaref, M., Morales, J. J., Khalife, J. J., Shamei, K., 2020. Robust Vehicular Localization and Map Matching in Urban Environments Through IMU, GNSS, and Cellular Signals. *IEEE Intelligent Transportation Systems Magazine*, 12(3), 36–52.

Kolbe, T. H., Kutzner, T., Smyth, C. S., Nagel, C., Roensdork, C., Heazel, C., 2021. Modeling Guide for 3D Objects, Part 2: Modeling of Buildings (LoD1, LoD2 and LoD3). 3rd edn, OGC. Available at `https://docs.ogc.org/is/20-010/20-010.html`.

Li, Y., Hu, Z., Hu, Y., Chu, D., 2018. Integration of vision and topological self-localization for intelligent vehicles. *Mechatronics*, 51, 46–58.

Liu, R., Wang, J., Zhang, B., 2020. High Definition Map for Automated Driving: Overview and Analysis. *The Journal of Navigation*, 73(2), 324–341.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., Stoter, J., 2022. Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands.

Reid, T. G. R., Houts, S. E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., Pandey, G., 2019. Localization Requirements for Autonomous Vehicles. *SAE International Journal of Connected and Automated Vehicles*, 2(3), 173–190.

Rohde, J., Völz, B., Mielenz, H., Zöllner, J. M., 2016. Precise vehicle localization in dense urban environments. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 853–858.

SAPOS, 2015. https://zentrale-stelle-sapos.de/wp-content/uploads/2020/09/SAPOS-Broschuere-2015-eng.pdf (Accessed: 20 September, 2024).

Solà, J., 2017. Quaternion kinematics for the error-state Kalman filter.

Suhr, J. K., Jang, J., Min, D., Jung, H. G., 2017. Sensor Fusion-Based Low-Cost Vehicle Localization System for Complex Urban Environments. *IEEE Transactions on Intelligent Transportation Systems*, 18(5), 1078–1086.

Tao, Q., Hu, Z., Zhou, Z., Xiao, H., Zhang, J., 2022. SeqPolar: Sequence Matching of Polarized LiDAR Map With HMM for Intelligent Vehicle Localization. *IEEE Transactions on Vehicular Technology*, 71(7), 7071–7083.

Wan, G., Yang, X., Cai, R., Li, H., Zhou, Y., Wang, H., Song, S., 2018. Robust and Precise Vehicle Localization Based on Multi-Sensor Fusion in Diverse City Scenes. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 4670–4677.

Wysocki, O., 2024. OloOcki/awesome-citygml. https://github.com/OloOcki/awesome-citygml (Accessed: 8 September, 2024).

Yoneda, K., Hashimoto, N., Yanase, R., Aldibaja, M., Suganuma, N., 2018. Vehicle Localization using 76GHz Omnidirectional Millimeter-Wave Radar for Winter Automated Driving. *2018 IEEE Intelligent Vehicles Symposium (IV)*, 971–977.