Validation and Verification of Lidar System: AI-Generated Point Cloud and Object Sensor Model

Basem Abusaif
basem.abusaif@valeo.com

Ahmed Adham
ahmed.adham@valeo.com

Ahmed Elragaby
ahmed.elragaby@valeo.com

Ahmed Yousif
ahmed.yousif@valeo.com

Abstract: Modeling and verification of LiDAR sensor data and algorithms are crucial for robust product development. While previous research has explored LiDAR sensor modeling and phenomenological modeling, there has been limited focus on the fusion of point cloud generation with phenomenological modeling and their verification against real sensor data. This paper addresses this gap by proposing two models: a generative AI model for LiDAR point cloud generation and a phenomenological model for algorithmic tasks like object detection, tracking, and lane detection. Both models are rigorously validated against real sensor data. Our approach facilitates comprehensive end-to-end and module-specific testing, improving validation of perception algorithms. Verification results demonstrate high accuracy, with the phenomenological model achieving 95% similarity to real sensor data and the generative AI model 96%, ensuring robustness and reliability.

1 Introduction

LiDAR (Light Detection and Ranging) technology is crucial in fields such as autonomous vehicles, robotics, and environmental monitoring for creating precise 3D maps, making it vital to ensure the reliability and performance of LiDAR systems across diverse real-world scenarios. However, validating these systems through real-world testing is often expensive and limited in scope. To address these challenges, simulation-based validation offers a promising solution by allowing researchers to test LiDAR systems extensively under simulated real-world conditions, including environmental factors and sensor characteristics, without the high costs associated with field trials. In this paper, we introduce a novel approach to LiDAR validation using advanced simulation models that integrate phenomenological and generative AI to accurately replicate the spatial, intensity, and noise characteristics of LiDAR data. This method provides a scalable, cost-effective way to rigorously test and optimize LiDAR systems, facilitating their broader deployment across various industries.

Contribution: 2024 AutoTest Technical Conference | 16 - 17 October 2024 | Stuttgart

2 Related Work

[YE23, SRB21, PMD19] explored the development of a LiDAR phenomenological sensor model that integrates ground truth data from a simulation platform to generate detection lists, closely mimicking the performance of actual LiDAR systems. This method allows the simulation model to produce detection lists faster than real-time, facilitating quicker validation cycles and better integration with hardware-in-the-loop and software-in-the-loop testing methods. In contrast, [Nic22] investigated generating LiDAR point clouds using two diffusion models conditioned on generated images. [Ach18] focused on training point cloud autoencoders and fitting generative priors, while [LH21] developed two-stage models with a diffusion model for individual points in the second stage and a latent flow model or latent GAN in the first stage. [EAE$^+$19] uses a Unet architecture to generate point cloud echo pulse width and mimic the behavior of the noise profile of radial distance.

Our approach starts with ground truth bounding boxes as input, which serve as an object list for simulating perception outputs. Unlike traditional methods, we do not rely on computationally expensive ray tracing to mimic the behavior of an actual LiDAR sensor. Instead, our method leverages key performance indicators, LiDAR-specific phenomena, and point clouds representing radial distances. The model is trained to generate point cloud intensities that closely resemble real sensor data, learning from both point cloud artifacts (such as radial distance, intensity, and point-level annotations) and ground truth bounding boxes. This gives us a competitive advantage by significantly reducing computational overhead while maintaining high accuracy in sensor simulation.

3 Methodology

3.1 Overview

Validating and verifying LiDAR systems, including components like perception, point cloud generation, and LiDAR-specific phenomena, requires extensive and precise datasets. [VL09] Achieving system maturity demands rigorous testing to ensure reliable performance across diverse conditions. Due to the complexity of LiDAR technology and its integration into autonomous systems, real-world data alone is often inadequate because of the vast range of scenarios required for thorough testing. [KKK11]

To tackle these challenges, simulated data have become crucial for validating LiDAR systems. They offer a controlled environment for testing, allowing early detection of issues, speeding up development, and reducing research and development costs by minimizing extensive real-world testing. [Com22]

LiDAR simulation models are designed to address specific validation needs, with each model tailored to its purpose. For evaluating the accuracy and reliability of LiDAR perception systems, high-fidelity point cloud models are essential for replicating real-world scenarios and thoroughly testing perception algorithms [JHS22]. In hardware-in-the-loop (HiL) setups, real-time performance is critical, and OSI (Open Simulation Interface) point cloud models provide real-time data that mimics actual LiDAR sensor behavior, ensuring seamless interaction with other vehicle systems [HPK22]. For object-level fusion systems, phenomenological sensor models generate detection lists that simulate LiDAR outputs, facilitating the development and validation of fusion algorithms to ensure accurate object detection and tracking in complex scenarios [SRB21, RR19, PMD19].

Data input interfaces for simulation models vary by framework, but the open simulation interface (OSI) ensures compatibility across platforms. Complying with ISO 23150 standards for simulation scenarios [fS23], OSI offers a standardized framework that improves interoperability among different simulation environments.

Simulating a LiDAR sensor is complex due to the need to replicate its intricate principles. The model must accurately represent laser pulse characteristics, signal processing, receiver properties, and environmental effects. High-fidelity simulations require significant computational resources, typically involving parallel processing with multiple GPUs. [MBW$^+$23]

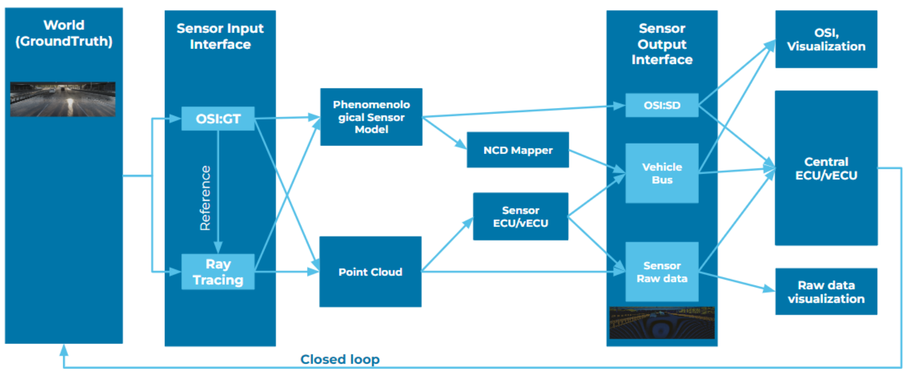

Figure 1: Simulation Pipeline Architecture

3.2 Phenomenological Sensor Model

The purpose of this LiDAR sensor model is to accurately simulate the sensor's response to a given scene. The input to the model includes ground truth data, comprising objects and lanes, along with their positions, velocities, and accelerations. The model processes this data through several stages: managing coordinate systems, filtering the field of view, handling occlusion of objects and lanes, tracking, and, finally, controlling key performance indicators (KPIs). The KPI controller is specifically designed to replicate the behavior of a real LiDAR sensor, ensuring that the model's output closely matches the statistical characteristics of the actual device. Figure 2 illustrates the main architecture of our model, and each of these stages will be discussed in detail [YE23].

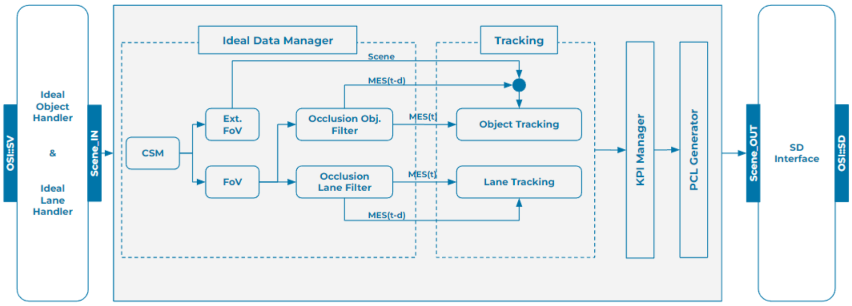

Figure 2: Phenomenological Sensor Model Component Diagram

3.2.1 Coordinate System Manager

As illustrated, the goal is to identify all objects, lanes, and visual cues relative to the LiDAR sensor's position. This step involves taking the input from the scene, which may be recorded relative to the vehicle or a map's reference frame, and performing the necessary transformations to determine their positions with respect to the LiDAR sensor. Figure 3 illustrates the different types of reference frames involved in this process.

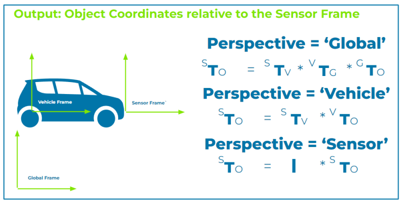

Figure 3: Coordinate System Manager

3.2.2 Field of View Filtration

At this stage, we have transformed all objects and lanes to be relative to the LiDAR sensor's position. With this information, along with the sensor's basic configuration—such as the maximum radial distance and the horizontal and vertical minimum and maximum angles—we can determine visibility. If any part of an object or lane falls within these defined regions, it is considered visible by the LiDAR, as we are still operating under ideal conditions. Figure 4 illustrates how we define the vertical and horizontal limits of the LiDAR sensor.

Figure 4: Field of view filtration

3.2.3 Occlusion

This module is designed to filter out objects and lanes that, although technically within the LiDAR's field of view, remain undetected due to occlusion. Occlusion occurs when one object blocks another, preventing the LiDAR sensor from recognizing the obscured object. To accurately simulate this, we apply a series of nonlinear transformations to the objects and lanes, projecting them into a canonical space. This process helps identify which objects are occluded. Figure 5 illustrates the impact of occlusion on objects and lanes.

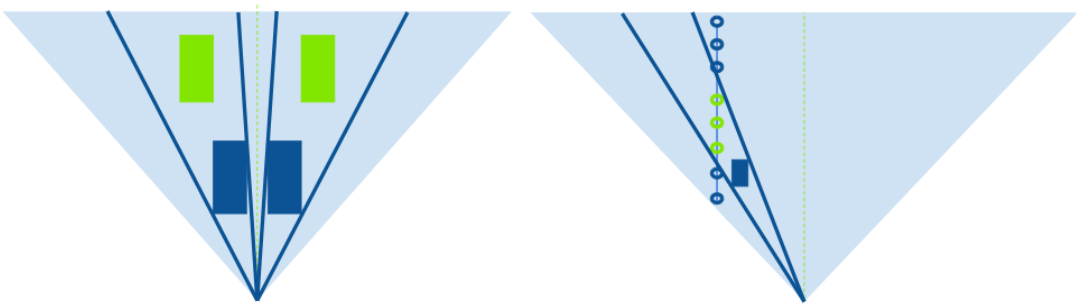

Figure 5: Objects and Lanes Occlusion

3.2.4 Tracking

This stage finalizes the ideal sensor model output, making it ready for noise injection. Our goal here is to replicate the tracking behavior found in a real sensor. For instance, if an object is visible but then moves out of the field of view or becomes occluded in subsequent frames, the sensor should still be able to predict its location for several frames. We achieve this by using Kalman filters, which predict the new location and match it with the true and ideal positions defined in the scene, while maintaining the correct object ID.

3.2.5 KPI Manager

The concept behind the KPI manager is to create a high-fidelity model that accurately mimics the behavior of a real sensor. To achieve this, we have collected extensive data, including the positions, orientations, velocities, and accelerations of objects and lanes. Additionally, we considered other factors, such as the type of detected object or lane. By comparing data from the real sensor with ground truth data, we were able to identify the sensor's error or noise in each parameter, such as velocity and position in the X and Y axes.

Using this information, we developed a discrete optimal controller to introduce noise into the ideal sensor model, replicating the deviations observed in the real sensor relative to the ground truth. This ensures that the distribution of each parameter matches that of the actual sensor. Figure 6 illustrates the purpose of the KPI manager, showing that the aim of this module is to generate deviations similar to those observed between the real sensor and the ground truth data.

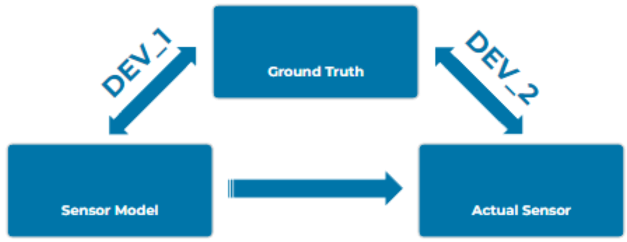

Figure 6: KPI Manager

3.2.6 PCL Generator Model

The PCL model fills the 3D bounding boxes of dynamic and static objects with point cloud data for each object within the LiDAR's field of view. This process involves segmenting the point cloud and assigning relevant points to their respective bounding boxes. These bounding boxes encapsulate the objects, with the points representing the surface geometry and features of each object within the LiDAR's field of view.

3.3 AI-based Intensity Model

AI is changing the automotive industry by making vehicles smarter and safer. One major use is in handling LiDAR data, which helps vehicles see and recognize their surroundings in 3D for tasks like detecting objects and avoiding collisions. By using the UNet architecture, originally for image processing, vehicles can now better understand these 3D images, leading to improved safety and efficiency in self-driving systems.

3.3.1 Polar Grid Map

The Polar Grid Map (PGM) is a 3D tensor-based representation of a full LiDAR scan, that encodes a point cloud of scan data. In this representation, each channel of the PGM is a 2D grid map where each row corresponds to a horizontal layer of the LiDAR data. Using the sensor as the reference point, each scan point is defined by its distance from the sensor, along with its azimuth and inclination angles. The rows and columns of the PGM correspond to the inclination and azimuth angles of the scan points, while the cell values store information about the specific points. In the first channel of the PGM, the cell value indicates the scan point's distance, and in the second channel, it means the scan point's class. This representation can be expanded by adding additional channels to store more information,although In our study we use 3 channels represented in ray tracing radial distance, object class ,and object material. An example of PGM representation in Figure 7.

Figure 7: Polar Grid Map

3.3.2 Intensity Model Architecture

In Figure 8 shows intensity model architecture primarily consists of multiple stages, starting with a simulation environment input that generates data for further processing. This input is then mapped into a Polar Grid Map (PGM), which organizes the LiDAR data into a structured format. Following this, a preprocessing step refines the data, preparing it for the core of the architecture a UNet-based deep neural network (DNN). The UNet DNN processes the data to predict the desired output with enhanced accuracy. After the DNN, a post-processing step is applied to further refine the results, leading to the final expected output. This structured approach ensures that the architecture efficiently handles the input data, leading to accurate and reliable predictions. Additionally, our integration with the Phenomenological sensor model will be facilitated through the PCL generator model that map the object level data into ray tracing point cloud, ensuring seamless data flow and enhanced object detection capabilities.

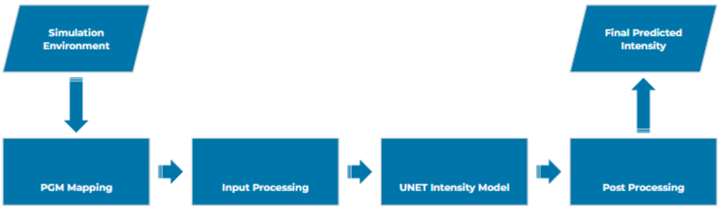

Figure 8: Intensity Model Architecture

3.3.3 Intensity DNN

Our intensity prediction model is built around a UNet-based deep neural network (DNN), which is well-suited for processing the structured data provided by the Polar Grid Map (PGM). The UNet architecture is ideal for this task due to its unique structure, which effectively captures both detailed and contextual information through its encoder-decoder design. The architecture consists of two main components: the contracting path (encoder) and the expansive path (decoder).

The contracting path captures the contextual features of the input data by downsampling it through a series of convolutional layers, each followed by max-pooling. This process extracts increasingly abstract features while reducing the spatial dimensions of the data.

The expansive path then works to restore the original resolution by upsampling the feature maps using transposed convolutions. To preserve spatial information and improve accuracy, skip connections are utilized between corresponding layers of the encoder and decoder, ensuring that the network retains both high-level context and fine-grained details.

In this model, the UNet processes the PGM input to predict the intensity values of each scan point. Accurate intensity prediction is crucial for tasks such as environmental perception and object detection, which rely on precise interpretation of LiDAR data.

The model is trained using a modified Mean Squared Error (MSE) loss function, enhanced with a regularization term to prevent overfitting. The MSE loss function is defined as:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} w_j^2$$

In this formula, y_i represents the true intensity values, $\hat{y}_i$ represents the predicted intensity values, n is the number of scan points, w_j denotes the model weights, and λ is a regularization parameter that controls the extent of weight penalization. The addition of the regularization term helps to reduce the model's complexity, encouraging it to generalize better to unseen data by discouraging overly complex models that may fit the training data too closely.

By minimizing this loss function, the model not only learns to make accurate predictions but also maintains robustness against overfitting, ensuring reliable performance in real-world scenarios. This combination of UNet architecture and a regularized MSE loss function allows the model to deliver precise intensity predictions critical for advanced automotive applications. The effectiveness of the model is demonstrated in Figures 9 and 10, where the predicted intensities closely align with the ground truth, showcasing the model's accuracy. The comparison between the two figures highlights the model's ability to replicate the true intensity values with minimal error.

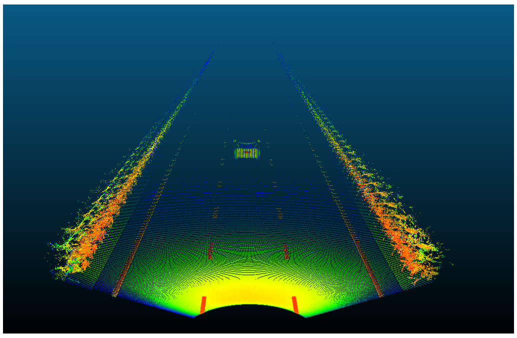

Figure 9: Ground Truth Intensities

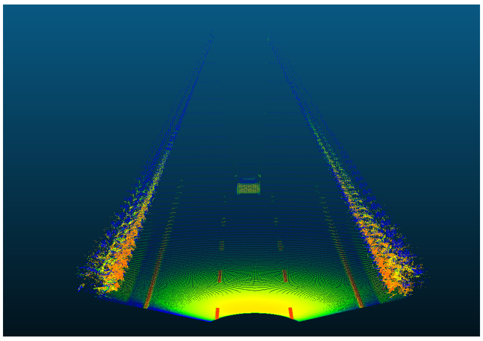

Figure 10: Predicted Intensities

4 Sensor Model Qualification

To ensure the developed model accurately simulates the real sensor, a sensor model qualification process is implemented. This involves comparing real recorded data from the vehicle's system with simulated data. [LS]

The qualification process, shown in Figure 11, involves recording real-world data and creating a corresponding digital twin scenario in the simulation environment. These scenarios are rigorously compared using key performance indicators (KPIs) and bug-to-bug testing in Hardware-in-the-Loop (HiL) or Software-in-the-Loop (SiL) setups. This method ensures accurate replication of the sensor's functionality, including any defects or bugs.

Point cloud intensity qualification evaluates the quality and accuracy of intensity values in a 3D point cloud, representing the reflectivity of surfaces captured by LiDAR sensors. This process is crucial for ensuring that the intensity data accurately reflects real-world conditions, aiding in tasks like object detection and environmental mapping. By validating the intensity values by calculate the deviation between the Point cloud intensity sensor model and the real sensor , we can improve the reliability of point cloud data for downstream applications.

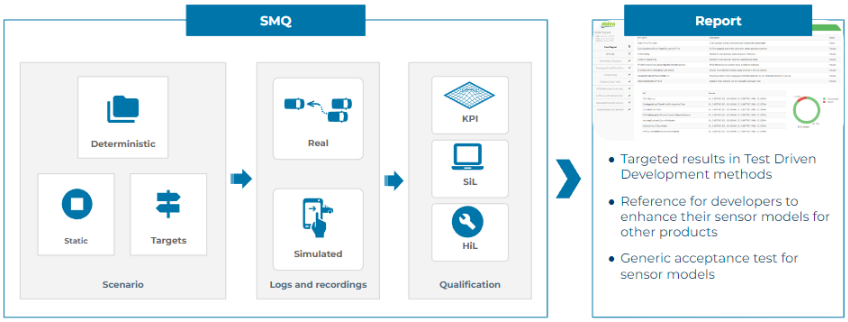

Figure 11: Sensor Model Qualification Components

5 Conclusion

In conclusion, this study highlights the importance of combining AI-generated point clouds with phenomenological sensor models to accurately validate and verify LiDAR systems. These models simulate real-world sensor performance, ensuring reliability. The AI model effectively generates point clouds that closely resemble those from actual LiDAR sensors, while the phenomenological model accurately mimics object detection and tracking. Together, these models allow for comprehensive testing of the LiDAR system and its components, enabling early detection of issues during development. These methods provide a cost-effective alternative to extensive real-world trials by offering detailed simulations across various conditions, ensuring the system's readiness for diverse scenarios. As technology advances, these approaches will be key in developing reliable autonomous systems across different industries.

References

[Ach18] P. et al. Achlioptas. Learning Representations for 3D Point Clouds, 2018.

[Com22] McKinsey & Company. Digital Twins: The Art of the Possible. https://shorturl.at/wAxZV, 2022. Accessed: 2024-08-20.

[EAE+19] K. Elmadawi, M. Abdelrazek, M. Elsobky, H. M. Eraqi, and M. Zahran. End-to-End Sensor Modeling for LiDAR Point Cloud. In *IEEE Int. Transportation Systems Conf.*, pages 1619–1624, Auckland, New Zealand, 2019.

[fS23] International Organization for Standardization. ISO 23239-1:2023 - Road Vehicles. https://www.iso.org/standard/83293.html, 2023. Accessed: August 21, 2024.

[HPK22] A. Haider, M. Pigniczki, and M. Köhler. High-Fidelity Automotive LiDAR Sensor Model. *Sensors*, 22(7556), 2022.

[JHS22] W. Jansen, N. Huebel, and J. Steckel. Physical LiDAR Simulation in Real-Time Engine. In *IEEE Sensors 2022*, 2022.

[KKK11] B. Khaleghi, A. Khamis, and F. Karray. Multisensor data fusion: A review. *Informat. Fusion*, 14(1), 2011.

[LH21] S. Luo and W. Hu. Diffusion Models for 3D Point Cloud Generation, 2021.

[LS] S. Ljungberg and F. Schalling. Benchmarking of LiDAR Sensor for Automotive Perception. Unpublished.

[MBW+23] Sivabalan Manivasagam, Ioan Andrei Bârsan, Jingkang Wang, Ze Yang, and Raquel Urtasun. Towards Zero Domain Gap: A Comprehensive Study of Realistic LiDAR Simulation for Autonomy Testing. In *ICCV*, 2023.

[Nic22] A. et al. Nichol. Point-E: Generating 3D Point Clouds, 2022.

[PMD19] Thomas Ponn, Fabian Muller, and Frank Diermeyer. Systematic Analysis of the Sensor Coverage of Automated Vehicles Using Phenomenological Sensor Models. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, June 2019.

[RR19] A. Rövid and V. Remeli. Raw Sensor Fusion in 3D Object Detection. In *IEEE 17th World Symposium on Applied Machine Intelligence and Informatics*, January 2019.

[SRB21] J. Schmitt, C. Robel, and B. Bäker. Measurement-Based LiDAR Sensor Model. In *21st Int. Stuttgarter Symposium*, pages 424–435, 2021.

[VL09] J. Van Laere. Challenges for IF Performance Evaluation. In *12th Int. Conf. on Information Fusion*, July 2009.

[YE23] A. Yousif and M. Elsobky. LiDAR Phenomenological Sensor Model. In *Mobility 4.0*, number 2023-01-1902, 2023.