BMW Group: Generation and Validation of Sensor Models **for Automated Driving Systems Using VIRES VTD**

By **Alexander Schaermann, Data Engineer, BMW Group**
Timo Hanke, Data Engineer, BMW Group

INTRODUCTION

Due to advancements in sensor technology and data processing algorithms over recent years, great progress has been made to enable automated driving systems to improve safety and comfort for the vehicle driver and occupants. Yet, due to the complexity of self-driving, one of the main challenges remains in ensuring and validating the safe conduct of the automated driving systems for public use.

Virtual worlds provide a suitable, safe and controlled environment to handle an important part of the required testing and validation efforts. A proper choice of scenarios as well as the generation of virtual sensor data that closely matches reality are among the central requirements for the success of the virtual development approach. Virtual sensor data is generated by means of sensor models that form a central component of the virtual environmental perception (Figure 1). This perception data constitutes one of the main input streams for the decision making algorithms of an automated driving system. Hence, the fidelity of the sensor model is a deciding factor for the viability and validity of virtual development and testing.

Generally speaking, there are two types of sensor models:

Sensor error models aim to reproduce the statistical characteristics of errors, i.e. deviations between the perceived and true values, of the measurement and perception performed by vehicle sensors.

Sensor measurement models, on the other hand, are based on a physical description of the measurement process, and they generate low-level measurement data based on the virtual scene. Models of this type are commonly used for a variety of sensors in robotics research, while the measurement models for automotive sensors are only emerging.

In this article, we introduce a sensor measurement model for an automotive LiDAR sensor. The model is based on a ray tracing approach for the simulation of the measurement process. This enables the real-time generation of a LiDAR Point Cloud within the framework of an automotive driving simulator. By directly comparing data from the real-world test drive to virtual data generated by the sensor model in a virtual environment, we are able to quantify the accuracy and validity of the sensor model using appropriate metrics.

SENSOR MEASUREMENT MODEL

A. Real-time Ray Tracing in a Driving Simulator

We consider the scanning type of LiDAR sensor, which is typically used in the automotive industry. This type of sensor determines distance by measuring the travel time of a laser pulse reflected by a target surface. Its angular resolution is achieved by means of scanning, i.e. by moving the transmitted laser beam as well as the selective field of view of the optical detector array successively over the sensor's complete field of view. Most commercially available systems at this time employ a mechanically rotating mirror for the scanning task. The operating principle of this type of sensor lends itself to a modeling approach using ray tracing techniques. The virtual environment for the proposed sensor model is provided by the Vires VTD driving simulation software (Figure 2), which offers a ray tracing framework based on the Nvidia OptiX ray tracing engine.

Figure 1. Virtual Sensor Models in VIRES VTD Environment

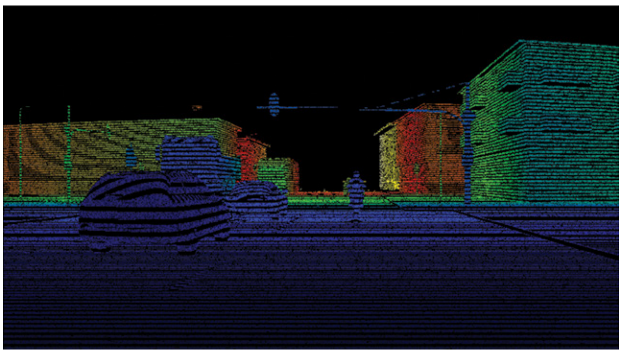

Figure 2. LiDAR Model Simulation in VIRES VTD

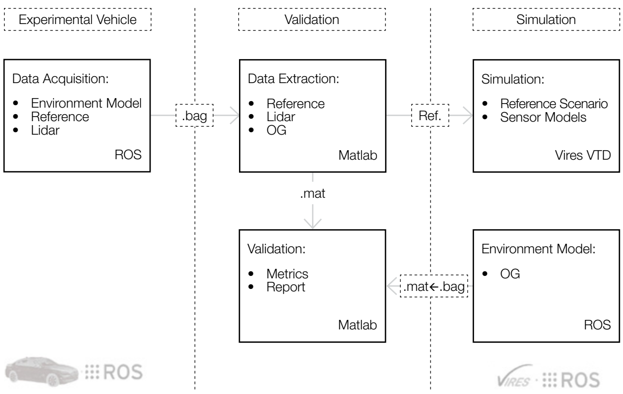

Figure 3. Tool chain for sensor model validation

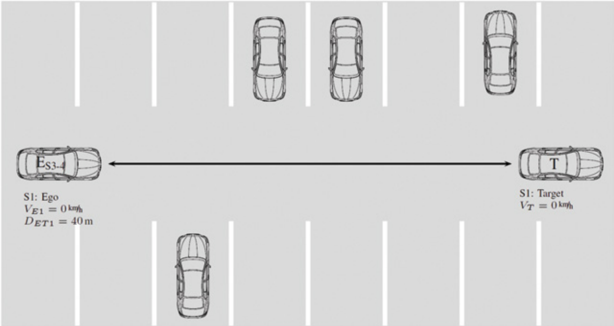

Figure 4. Static Validation scenarios with 40m distance between ego and target

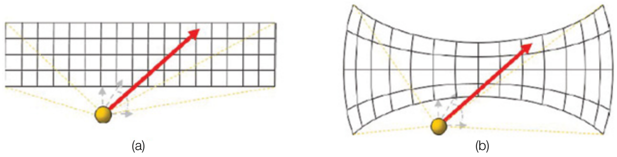

Figure 5. Sampling grids for ray tracer: (a) Cartesian (b) Spheric

B. Virtual Point Cloud Generation

To model the beam transmission, reflection, and detection of the LiDAR sensor, the camera program of the sensor measurement model generates a ray for each set of azimuth and elevation angles. That results in a Point Cloud, if a valid distance measurement is obtained (see reference 1 for more details).

SENSOR MODEL VALIDATION

A. Methodology for Validation

For the validation of the LiDAR sensor model described in section II, we propose the procedure shown in Figure 3. This method is based on the comparison of real and synthetic data. In the first step, real data is captured with an experimental vehicle equipped with LiDAR sensors, a differential global positioning system (DGPS) and environment model algorithms running in ROS (middleware) and including an occupancy grid implementation. Then, synthetic data is generated using the LiDAR sensor model described in section II and exactly the same occupancy grid implementation as used in the experimental vehicle, but provided with simulated data from the sensor model in VIRES VTD. For data exchange between the model and ROS, the Open Simulation Interface (OSI) is used. As soon as real and synthetic data are captured, we evaluate the data in the validation framework using Matlab in a two-step procedure. In the first step, the direct comparison of real sensor data and model output is taking place. In the second step, we compare occupancy grids generated with real and synthetic LiDAR data representing the static environment of the test vehicle.

B. Validation Premises

For the validation of the sensor model, a static scenario is evaluated (see Figure 4). The two vehicles, Ego (E) and Target

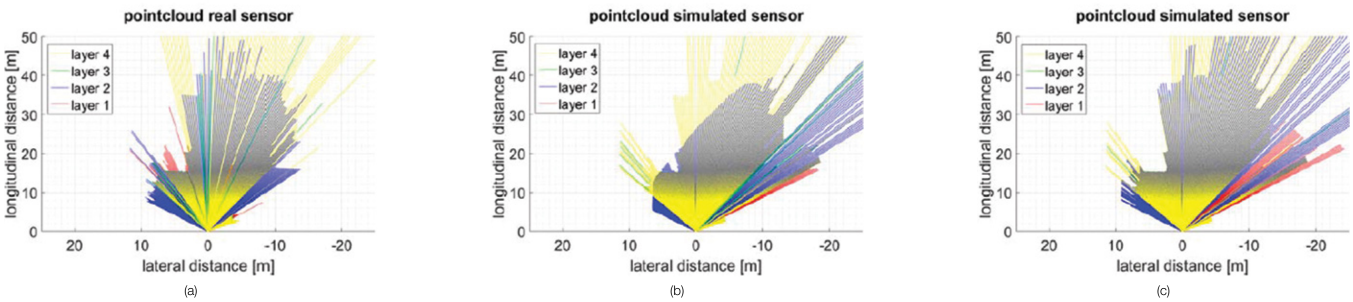

Figure 6. Visualization of Point Cloud: (a) real Point Cloud, (b) synthetic Point Cloud from SC1, (c) synthetic Point Cloud from SC2

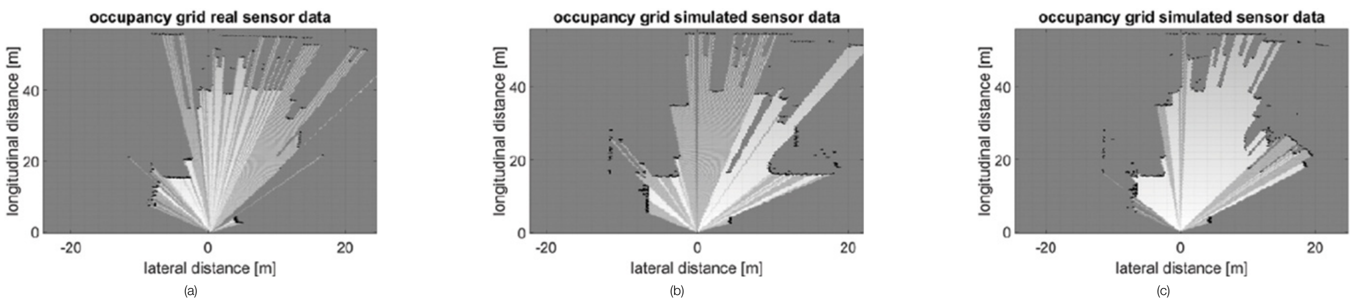

Figure 7. Visualization of occupancy grids: (a) real occupancy grid, (b) synthetic occupancy grid from SC1 (c) synthetic occupancy grid from SC2.

(T), shown in the schematic have an approximate distance of D1 = 40 m to each other. This area was modeled as a virtual 3D model for a simulator with particularly high fidelity requirements with respect to geometric dimensions and positions. Using this scenario, we show how the modeling of the LiDAR sensor's geometric configuration can be tested and how different configurations influence the generated Point Cloud and occupancy grids.

Two different sensor configurations SC1 and SC2 of the model approach are applied in this study. SC1 uses a Cartesian sampling grid for ray generation visualized in Figure 5(a), and SC2 uses a spherical grid shown in Figure 5(b). Usually for image generation, a linear distribution of rays generated with ray tracing is needed, so the SC1 approach would be the right choice.

Model State	Validation Level	Overall Error	Barons correlation	Pearson correlation
SC1	EDM PC	8729.2	0.733	0.824
	SG	1.0816×10^6	0.637	0.679
	OG	2.3668×10^6	0.602	0.677
SC2	EDM PC	8566.4	0.743	0.832
	SG	9.385×10^5	0.721	0.764
	OG	2.117×10^6	0.634	0.703

TABLE I: Calculated results for different validation metrics: overall error, Barons and Pearson correlations at different validation levels: Point Cloud, scan grid and occupancy grid.

However, since the geometry of the beam deflection of a LiDAR sensor leads to a conic shape of the point cloud, the spherical sampling grid is a more suitable choice for this purpose.

C. Data Evaluation

a. We start the investigation with a qualitative inspection of the captured Point Cloud shown in Figure 6. Visually observing the Point Cloud, it is obvious that the real Point Cloud is more similar to the Point Cloud generated from Sensor Configuration 2 (SC2) compared to Sensor Configuration 1 (SC1).
b. As mentioned before, occupancy grids are used as an abstraction level for sensor model validation. Here, we additionally use scan grids (SG) as a further abstraction level. Scan grids are single shot recordings of occupancy grids generated from a Point Cloud, whereas the occupancy grids are over time accumulated scan grids.

For evaluation of the environment model output, the real world scenario is re-simulated, and scan grids as well as occupancy grids are computed using generic Point Clouds from the two sensor configurations. The scan grid results are shown in Figure 7. Visually comparing the scan grid representations of the two sensor configurations with the real data, we can see a higher alignment between the real scan grids and the scan grids from the SC2. To quantify this observation, three metrics are applied and summarized in Table I. Similar to the quantitative results from the Point Cloud evaluation, these values show lower overall error and higher correlations between real scan grids and scan grids from SC2 compared to SC1.

SUMMARY AND FUTURE WORK

In this article, we propose a physically motivated sensor measurement model based on a ray tracing approach for an automotive LiDAR sensor. The model was employed to faithfully recreate the full sensor processing chain in a virtual environment with the help of VIRES VTD. Furthermore, a full processing chain in the virtual environment starting from low-level sensor data and ending with the first fusion stage of the whole automated driving system was reproduced in a virtual environment. With the presented setup, it is possible to evaluate real driving situations and reconstruct them in the simulation from high-fidelity data for static and dynamic scenarios. As sample use cases, we showed a static situation on a parking lot. We could quantify how closely the internal environment representation, i.e. the input to the automated driving function, matches between real world scenario and the simulation using a raw data LiDAR sensor model and appropriate validation metrics. The results represented in this paper show a higher correlation between real and synthetic data using the sensor model with a spherical ray tracer sampling grid.

Reference

1. "Generation and Validation of Virtual Point Cloud Data for Automated Driving Systems", T. Hanke, A. Schaermann, M. Geiger, K. Weiler, N. Hirsenkorn, A. Rauch, S-A. Schneider, and E. Biebl, IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), October 2017, Yokohama, Japan