International Journal of Automotive Technology, Vol. ?, No. ?, pp. ?–?(year)

DEVELOPMENT OF A LIDAR MODEL FOR THE ANALYSIS OF BORDERLINE CASES INCLUDING VEHICLE DYNAMICS IN A VIRTUAL CITY ENVIRONMENT IN REAL TIME

Rene Degen[1,4*)], Harry Ott[1,4)], Fabian Overath[1)], Dr. -Ing. Christian Schyr[2)], Florian Klein[3)], Prof. Dr. Mats Leijon[4)], Prof. Dr. rer. nat. Margot Ruschitzka[1)]

[1)] Faculty of Automotive Systems and Production, CAD CAM Center Cologne, Cologne University of Applied Sciences, 50679 Cologne, Germany
[2)]AVL Deutschland GmbH, Advanced Solution Lab, 76229 Karlsruhe, Germany
[3)]Hoersch und Hennrich Architekten GbR, HHVISION, 50827 Cologne, Germany
[4)] Department of Electrical Engineering, Uppsala University, 751 05 Uppsala, Sweden

(*Received date ; Revised date ; Accepted date*) * Please leave blank

ABSTRACT–Advanced driver assistance systems are an important step on the way towards the autonomous driving. However, there are new challenges in the release of increasingly complex systems. For the testing of those systems many test kilometers are necessary to represent sufficient diversity. Hence, the virtual testing of driver assistance systems brings new opportunities. In virtual environments, it is possible to run a much higher distance in a short time. Simultaneously, the complexity of the environment and the test scenarios are individually adjustable. It is possible to test scenarios that are not feasible in a real environment due to a risk of injury. A big challenge is the physical correct implementation of real vehicles and their components into the Virtual Reality. To enable a realistic virtual testing the vehicles surrounding sensors need to be modeled adequately. Thus, this paper presents an approach for the implementation of a Lidar model into a Virtual Reality. A physical Lidar model is combined with a real-time capable vehicle dynamics model to investigate the influence of vehicle movements to the sensor measurements. The models are implemented into a highly realistic virtual city environment. Finally, a test campaign shows the influence of the Lidars physics and the vehicle dynamics on the detection results.

KEY WORDS : Advanced Driver Assistance Systems (ADAS), Autonomous Mobility, Lidar Simulation, Vehicle Dynamics, Raytracing, Virtual Environment, Sensor Simulation

NOMENCLATURE

A_r : receiving lens area, m²
B_λ : electromagnetic bandwidth of the receiving unit, nm
C_0 : speed of light, m/s
D : receiving lens diameter, m
E_{Si} : radiation intensity of the sun ,W/m²/nm
E_{sun} : sun induced energy, J
f_{cl} : counter frequency, Hz
$\underline{i}$: incidence vector
I_D : dark current noise
IFOV : instantaneous field of view, rad
$\underline{n}$: surface normal
N_Z : number of counted timesteps
P_{DK} : dark current noise, W
P_r : received power, W
P_n : sum of noise powers, W
P_{sun} : sun induced noise power, W
P_t : transmitting power, W
Q_V : beam divergence, rad
R : distance to target, m
SNR : signal to noise ratio
t_{cl} : duration of one counter step, s
t_{of} : running time of the light, s
δ_t : discrete receiving timestep, s
η_{sys} : summarized system losses
θ_i : incident angle, rad
ρ_t : reflectance coefficient
τ : atmospheric transmission coefficient
$\Re_{max}$: maximum intensity of a photodiode

* *Corresponding author*. e-mail: rene.degen@th-koeln.de

1. INTRODUCTION

The simulation of environmental sensor data is a research issue that becomes more and more important due to a raising number of for Autonomous Driver Assistance Systems (ADAS) obstructed in new cars. In 2019 48 % of all new cars sold in Germany were equipped with a lane keeping assistant, 39 % have an autonomous emergency brake and 38 % were delivered with an adaptive cruise control (Statista, 2020a). In 2020 90% of the German car driver were of the opinion that ADAS increase the vehicle safety. 89% thought assistance systems make driving more pleasant (Statista, 2020b). Although the data refer to the German market, a similar result can be expected internationally. This leads to the expectation that the market for ADAS will continue to grow in the future. Besides the opportunities ADAS offer to the vehicle safety, they also increase the vehicles complexity and the testing effort. (Wachenfeld/Winner, 2015) give a theoretical approach for the approval of ADAS with regard to an autonomous highway pilot. On German motorways a deadly accident happens every 662 million kilometers. The authors say this situation needs to be reconstructed ten times to approve an ADAS system. This leads to a theoretical demand of 6.62 Billion test kilometers. This need cannot be met by test drives alone. Virtual testing offers advantages at this point. However, a distinction must be made between different levels of complexity. (Gadringer, et al., 2018) provide an approach for the ADAS testing in virtual environments in combination with a Hardware in the Loop (HiL) testbench for real vehicles. For the interaction of the car with the virtual surrounding artificial environmental sensors are needs. In the given examples, this is limited to the radar sensor technology. In addition to radar, Lidar sensors are being used more and more in automotive applications. Hence, also for this sensor technology accurate models are needed to allow the virtual ADAS testing. The literature contains a wide variety of approaches to simulating Lidar data. The simulation approaches can be roughly divided into two concepts, the semantic and the physical Lidar models. (Wang, et al., 2019) and (Gusmão, et al., 2020) are examples for semantic Lidar models. Their aim is not to simulate physical behavior of the Lidar, but to provide raw-data augmented with meta data of the virtual surrounding. Information like the color of objects or a classification are provided. These models are primary used for the training on deep learning algorithms and not suitable for testing purposes. On the other hand, there are the physical models, whose aim is to represent the physics of the Lidar. A physical Lidar model approach for Automotive applications can be found in (Castaño, et al., 2020). The authors use variations of the classical radar equation to compute the remaining power a Lidar receives of a sent signal, considering various sources of interference. The implemented model is largely based on (Kim *et al.*, 2013), a simulation model for aerospace applications. Due to that, the models has some inadequacies, which arise due to other requirements in the automotive sector. Furthermore the environment the model is implemented in is kept simple. The model neglects the changes in the Lidar vision due to vehicle dynamics. Even more compley physical models like (Byeon, Yoon, 2020) or (Muckenhuber, et al., 2020) are not taking this into account. Therefore, this publication presents a novel approach to simulate physical correct Lidar raw data under consideration of the effects of vehicle dynamics. Various sub models for the existing problems are introduced. This includes a physical Lidar model, a vehicle dynamics model and highly authentic virtual city environment. Subsequently a network of the sub models is implemented. Finally an exemplary investigation is carried out using the implemented model to evaluate the influence of vehicle dynamics on detection borderline cases in an inner-city scenario.

2. PHYSICAL LIDAR MODEL

As described above the physical correct modelling of virtual sensor data in artificial environments in realtime is an important topic for the testing of autonomous driving functions. The model described in the following sections is based on a raytracing approach. Virtual rays are used to simulate the signal of the emitted and reflected light beams, equivalent to a real Lidar system. For these both, the impinging powers and the disturbances acting on the sensor are taken into account. Section 2.1 will present the used methods and the underlying model. This is followed by the implementations strategy in a virtual reality in section 2.2. For this, the Unreal Engine of the developer Epic Games is used. At last, Section 2.3 will show the parametrization and verification tests of the described model.

2.1. Lidar principle

For the later mathematical description of a Lidar sensor, the basic working principles needs to be shown first. Automotive applications offer many different Lidar systems for the environmental detection. These systems differ in the way the surrounding is scanned. The system architectures can be divided into three mayor categories. Singlebeam scanning Lidar sensors use one laser beam, which is deflected by moving optics, for the discrete scanning of the environment. At each time step and thus each position of the optics, a light pulse is sent into the environment. The resolution of these systems is formed by the frequency of the emitted light pulses and the rotation speed of the deflection unit. In contrast, the multibeam principle uses diode arrays to cover the

detection field. For each direction to be detected there is an LED and an associated photodetector. With this system, all light beams are sent at the same time. The resolution results from the number of used array elements. The third Lidar architecture uses the multibeam sweep principle. It can be characterized as a hybrid form of the previous presented methods. The systems also uses a diode array. However, this array is not static. The sent beams are deflected, either by the mechanical movement of the array or as with the single beam technique by a moving optics. This enables the detection of a comparatively wide field depending on the current driving situation. For example, the beams can be swiveled when turning to detect passers. The advantage of this is a wide field of view (FOV) with a simultaneously high resolution. Comparatively complex algorithms are required to control this process, however their development is not the subject of this paper. For this reason, the first two approaches are considered in the following. (Winner *et al.*, 2016)

Regardless of the scanning method and thus the Lidar architecture, the principle of distance determination is similar for all Lidar sensors. The time of flight of the light signal is determined and used to compute the distance. For further considerations, a closer look is taken to the example principle of the *Time-to-Digital Converter* (TDC) to determine the time of flight. Figure 1 shows the principle of this method.

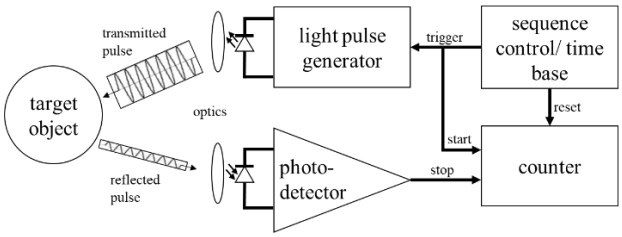

Figure 1 Schematic working principle of a Lidar Sensor according to (Kernhof *et al.*, 2018).

The basis of this method is a high-precision sequence control. It triggers a light pulse generator at a time step and a laser diode emits a light signal with a defined amplitude and length. At the same time, the trigger signal starts a counter. When the light beam hits a target object, it is reflected and a portion of the incident light is casted back to the sensor. The amplitude of the reflected pulse is depending on the targets reflection properties and the environmental conditions. Circumstances such as rain, snow or fog can massively reduce the signal strength. In a photo sensor, the reflected signal is received, amplified and transformed by an analog-to-digital converter (ADC). When the signal exceeds a certain threshold value, the counter is stopped and the number of counted time steps are read out and stored. Through the time between the transmission and the reception of the signal, a statement can be made about the distance of the target object. With this method, the accuracy of the result is significantly influenced by the time resolution of the counter. The TDC shows an example for a simple method of Lidar data analysis. More complex and moderns Lidar systems use the form and the amplitude of the reflected signal to generate further information about the environment and the target object. (Kernhof *et al.*, 2018) However, this is neglected in a first step, since in this work an evaluation is to be made whether a point is recognized. For that, it is necessary to have an accurate mathematic model of the Lidar with disturbances and losses. This model is introduced in the following section.

2.2. Mathematic Lidar Model

As described before, Lidar sensors use the runtime of the light to determine the distance of an object. Equation 1 shows this relationship, where d represents the distance to the target, c_0 stands for the natural constant of the speed of light and t_{of} is the duration the light takes.

$$d = \frac{c_0 \cdot t_{of}}{2} \tag{1}$$

$$t_{of} = N_Z \cdot t_{CL} \tag{2}$$

Based on the approach, that a TDC determines the value of t_{of} and thereby finds the target distance, the resolution of the counter needs to be considered. This is shown in Equation 2. The time of flight is computed by the number of counted time steps N_Z and the clock period of one counter step T_{CL}. By the usage of one time step the resolution of the distance as shown in Equation 3. Equation 4 shows the time step expressed by the clock frequency of the counter.

$$\Delta d = \frac{c_0 \cdot T_{CL}}{2} \tag{3}$$

$$T_{CL} = \frac{1}{f_{CL}} \tag{4}$$

In addition, measurement inaccuracies result from the amplification and the analog-digital conversion of the received signals. The resulting errors are dependent of the used hardware architectures. Pending on the system, oversampling methods and filters are used to reduce the measurement faults. Due to the large number of systems available, it is not sensible to find a mathematically universal description. In order to consider these errors, it is necessary to find a proper value and to implement it in the Lidar model.

As mentioned before, the Lidar model should not only represent the inadequacies of the distance measurements, but also give a decision whether a target point is recognized. The receiving power and the induced noise significantly determine the detection. A characteristic value for evaluation is the signal to noise ratio (SNR). The calculation of the SNR-Value is shown in Equation 5, where P_r is the received power and P_n is the sum of the induced noise powers.

$$SNR = \frac{P_r}{P_n} \tag{5}$$

The receiving power of the reflected light beams is influenced by various parameters. Optical properties of the target, environmental conditions, the target distance and the characteristics of the sensor itself, like the transmitting power, the beam divergence and the receiving lens area, have a major influence on the result. The literature provides different approaches for the mathematical description of the received power. Mostly variations of the classical radar equation are used. However, the exact implementation differs. Examples can be found in (Winner *et al.*, 2016), (Kim *et al.*, 2013) and (Kernhof *et al.*, 2018). Due to the different conversions, a generally valid equation for the received power must be determined for the further investigations. Equation 6, adopted from (Kim *et al.*, 2013), is used as a basis for this. The nomenclature is adapted to this paper.

$$P_r = \frac{P_t \cdot \rho_t \cdot D^2 \cdot \tau^2 \cdot \eta_{sys}}{4R^2} \tag{7}$$

In the equation P_t is the transmitting power, ρ_t is the reflectance coefficient of the target object, D is the diameter of the receiving lens, τ is the atmospheric transmission coefficient, η_{sys} is the sum of the system losses and R is the distance to the target. However, the beam divergence and the influence of varying incident angles are not regarded. This may be, because the equation is used in aerospace Lidar applications. The beam divergence can be assumed simplified as shown in Equation 8. The sinusoid is omitted due to a small angle approximation.

$$\sin(Q_v) \cdot R \approx Q_v \cdot R \tag{8}$$

Assuming that the laser beam hitting an object is reflected diffusely, the influence on the impinging angle can be expressed by the Lambert's cosine law. In real environments, not only diffuse reflections are to be expected. For the first implementation, the model offers sufficient accuracy. Finally, if the size of the receiving lens is expressed in terms of area rather than diameter, equation 9 is obtained.

$$P_r = \frac{\rho_t \cdot A_r \cdot \tau^2 \cdot P_t \cdot \eta_{sys} \cdot \cos(\theta_i)}{Q_V \cdot \pi \cdot R^3} \tag{9}$$

A_r represents the area of the receiving lens and θ_i indicates for the impinging angle of the laser beam. The equation represents an assumption for calculating the received power and needs to be verified later.

The next important parts to calculate the SNR are the noise powers acting on the Lidar system. According to (Kim *et al.*, 2013) two main noise sources can be found for Lidar applications, on the one hand the sun-induced noise and on the other hand the dark current noise. The sun induced noise results from the sunlight illuminating the same targets surface as the laser beam. In the source mentioned above, the energy impinging the sensor in a defined discrete time step δ_t is expressed as shown in Equation 10.

$$E_{sun} = E_{Si} \cdot B_\lambda \cdot \delta_t \cdot \rho_t \cdot A_r \cdot \tau \cdot IFOV^2 \cdot \eta_{sys} \tag{10}$$

E_{Si} represents the illumination intensity of the sunlight, B_λ is the electromagnetic bandwidth of the receiving unit and IFOV is the instantaneous field of view. In case the same optics for receiving and transmitting the signals is used, the IFOV corresponds to the beam divergence. As mentioned, the expression calculates an energy. However, for the SNR a power is needed. To obtain the power the time derivative has to be formed. In this way, δ_t is omitted. The result for the sun induced noise power is shown in Equation 11.

$$P_{sun} = E_{Si} \cdot B_\lambda \cdot \rho_t \cdot A_r \cdot \tau \cdot IFOV^2 \cdot \eta_{sys} \tag{11}$$

The method for calculating the dark current noise is adopted from (Mackowiak *et al.*, 2015). It is shown in Equation 12.

$$P_{DK} = \frac{I_D}{\mathfrak{R}_{\max}} \tag{12}$$

Although (Kim *et al.*, 2013) also provide a method for the computation of the dark current noise, the used computation is more suitable for the intended application. It uses the dark current I_D and the maximum sensitivity of a photodiode $\mathfrak{R}_{max}$ for the computation. These parameters can usually be found in the data sheets of photo elements.

Thus all necessary equations for the mathematical description of the Lidar are provided and the implementation of the model can be done.

2.3. Implementation in a Virtual Reality

The Lidar model is implemented in a Virtual Reality. As already mentioned, the Unreal Engine of the developer Epic Games will be used. It is a development environment originally designed for the production of computer games. In the last few years, the use of the engine was established in science and technology due to the increasing use of Virtual Reality. The programming can either be done graphically by using so called blueprint scripts or by textual C++ coding. The Lidar model will be implemented in C++ due to a higher performance and a lager set of functions. Later on, a highly realistic city environment will be used for the testing of the virtual Lidar sensor. For the first implementation, a basic test setting with simple geometries is used. A virtual character allows users to move the model within the scene by

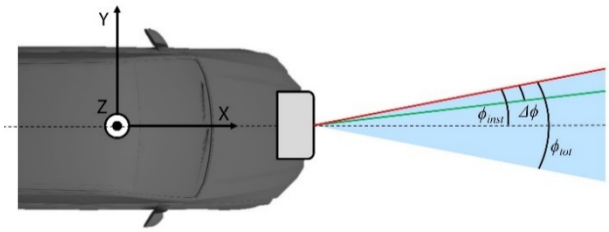

Figure 2 Visualization of the discrete scan process of the virtual Lidar by LineTraces.

The implementation of the Lidar model is based on raytracing, which is named Linetracing in the Unreal Engine. Virtual rays, representing the Lidars light beams, are shot into the artificial 3D Scene. As starting point for the rays the position of the virtual sensor is used. The end position is pending on the azimuth and the elevation angle of the respective ray. It is not possible to perform the LineTrace in infinite length, therefore a position needs to be found for the endpoint that represents the range of the Lidar and still limits the computational power. The virtual beams are emitted one after another. Figure 2 shows this process for the azimuth field. On each incrementation step, the actual azimuth angle ϕ_{inst} of the shot ray is incremented by $\Delta\phi$ until the whole azimuth range ϕ_{tot} is covered. After that, the elevation angle is incremented and the azimuth range is scanned again. This is done until the whole field of view of the virtual sensor is covered. The process needs to be done on every simulation step.

$$\theta_i = \cos^{-1}\left(\frac{\underline{i} \circ \underline{n}}{|\underline{i}| \cdot |\underline{n}|}\right) \tag{13}$$

If a virtual ray hits an object on its way from the starting point to the endpoint, the computation of the SNR according to Section 2.2 is initialized. To enable this the incidence angle of the ray on the object surface is needed. The value can be computed by Equation 13, where $\underline{i}$. the incident vector and $\underline{n}$ is the surface normal at the impact point. The incident vector results from the start point and the hitting point. The surface normal can be read directly from the LineTrace as a unit vector. Therefore, it is now possible to compute the SNR depending on the Lidar parameters. The results of the impinging power, the noise powers and the SNR are stored in arrays assigned to the respective light beam for further processing.

In addition to the calculated outputs, the actual output values of the Lidar sensor, the target distances, need to be recorded. The coordinates of the impact points can directly be read from the Linetraces. However, these values are provided in global coordinates. Hence, a coordinate transformation to the local axis system of the virtual sensor needs to be done. The coordinates are provided as floating point values without errors. As depicted in section 2.2, the output of real Lidar sensors is faulty due to the TDC, the amplification and the A/D-conversion. The large variance of systems does not allow a generally valid mathematical description of these errors. Hence, a substitute model needs to be implemented. Most Lidar manufacturers provide distance resolutions for their sensors. However, it is not practical to simply apply these to the measured values. Instead, a white noise with the given resolution as a standard deviation is added to the signal. This does not necessarily reproduce the real physical behavior, but it reproduces the resulting errors in sufficient quality. The local, by noise augmented impact points are also stored into arrays, according to the respective ray.

Finally, the determined points are displayed in color according to the calculated SNR in the virtual scene. This makes it possible to decide within Virtual Reality whether a point is potentially recognized or not. An example of the output in the simplified test scene for a Lidar sensor with a FOV of 60° in azimuth and 30° in elevation is shown in Figure 3.

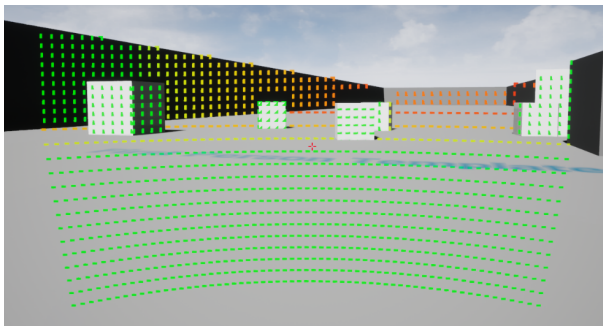

Figure 3 Visualization of the SNR value of recognized Lidar points within a simple Virtual Reality test scene.

2.4. Parameterization and Verification of the Lidar Model

The Lidar model described and implemented in the previous sections gives a theoretical approach for the simulation of the physical behavior of Lidar sensors. The verification of the model will be the topic of this section. Two tests will be carried out to estimate the validity of the model. Before carrying out the tests, reasonable parameters for the Lidar system must be found. The Lidar sensor *Valeo SCALA 3D Laser Scanner* is used as a typical reference for automotive Lidar applications. The datasheet for this sensor is provided in (Hexagon, 2021). A problem is that not all parameters required for the calculation are specified in this document. Hence, further assumptions have to be made. The missing parameters are adapted from (Weber, 2018), (Kim, et al., 2013) and (Hamamatsu, 2018). The values, used for the further considerations are summarized in Table 1.

In addition to the parameters for the Lidar sensor itself, suitable values for the environmental parameters have to be found. The atmospheric transmission coefficient τ can be assumed as 0.8 according to (U.S. Naval Academy, n.d.). At the wavelength of 950 nm, used by the described Lidar system, the Sun irradiance can be considered as 1.5 W/m²/nm corresponding to (Newport, n.d.). These values only represent a first assumption for the following tests, later on they need to be dependent on the illumination and the viewing conditions of the virtual environment. The reflectivity coefficient of the target surfaces is specified jet, as it is subject of the following tests. In the datasheet (Hexagon, 2021) different maximum detection distances are provided for the sensor under investigation, depending on the surface reflectivity coefficient and the true positive recognition rate (TPR). The given values are based on squared a Lambertian reflector with an edge length of 1 m. Table 2 summarizes these values.

Table 1 Identified parameters of the virtual Lidar sensor

	Symbol	Value	Unit
Azimuth Angle	ϕ_{tot}	145	deg
Elevation Angle	ϑ_{tot}	3.2	deg
Azimuth Resolution	$\Delta\phi$	0.25	deg
Elevation Resolution	$\Delta\vartheta$	0.8	deg
Distance Resolution	ΔR	0.1	m
Radiated Power	P_t	80	W
Lens Area	A_r	0.0007	m²
Beam Divergence / Instantaneous Field of View	Q_v / IFOV	0.003	rad
Electromagnetical Receiver Bandwidth	B_λ	2	nm
Dark Current	I_D	10	nA
Sensitivity Photo Diode	$\mathfrak{R}_{max}$	0.5	A/W
System Efficiency	η	0.9	-

Table 2 Maximum detection distances of the Valeo Scala Lidar (Hexagon, 2021)

Distance	Reflectivity	True Positive Rate
150 m	10 %	10 %
100 m	10 %	50 %
150 m	80 %	100 %
200 m	80 %	55 %

Based on this data the first test setup for the verification of the Lidar model is implemented. In the test scene a squared plate with a size of 1 m is modeled and placed in front of the virtual sensor. The plate is iteratively moved away from the actor in steps of 1 m. To reduce the data volume only one ray is shot on every simulation period. The computed SNR is stored for every distance of the plate. The test is carried out twice, once for a reflectivity of 0.8 and once for 0.2 of the plate. The result is plotted using MathWorks MATLAB as shown in Figure 3. The scale for the SNR is logarithmic.

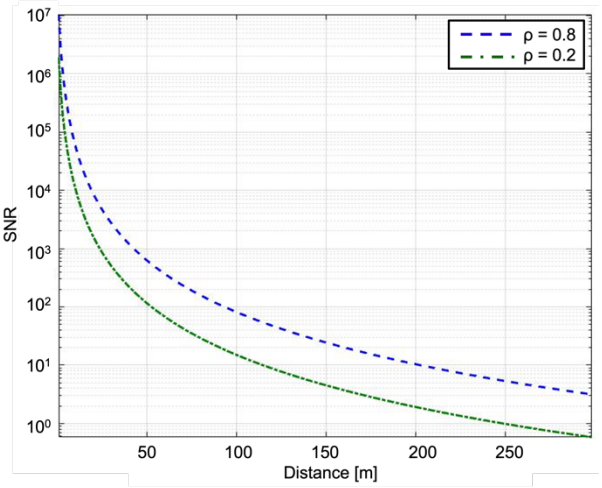

Figure 4 Logarithmic plot of the of the SNR values for increasing distances and different target reflectivities.

In order to compare the simulation results with the detection behavior of the real Lidar sensor, the computed SNR values are summarized in Table 3, according to the distances and true positive rates of Table 2. The results show a largely valid behavior of the virtual sensor. A TPR of 100 % corresponds to a SNR of 24.5. A TPR of 10 % is represented by an SNR of 4.5. These results are plausible. Only for the TPR values 50 % and 55 % other results would have been expected. For 55 % the according SNR is 10.4, for 50 % it is 15.1. This may be due to a slightly inaccurate parameterization of the model, as the exact test setup for the measurement of Valeo sensor is unknown. Nevertheless, the behavior of the sensor model can be considered valid for the carried out tests.

Table 3 SNR values read from Figure 4 according to the distances and reflectivities of Table 2

Distance	Reflectivity	SNR
150 m	10 %	4.5
100 m	10 %	15.1
150 m	80 %	24.5
200 m	80 %	10.4

In addition, a second test should allow an evaluation of the model behavior at different incident angles. The test setup is similar to the last one. A rectangular plate is placed at a distance of 1 m in front of the virtual sensor. However, the plate will not be moved away from the sensor, but turned around its vertical axis, starting with -90°, thus parallel to the virtual beam. On every simulation step, the object is turned by 0.5° up to 90°. The resulting SNR is stored and plotted by MATLAB. For the reflectivity coefficient, a value of 0.8 is chosen. The result is shown in Figure 5.

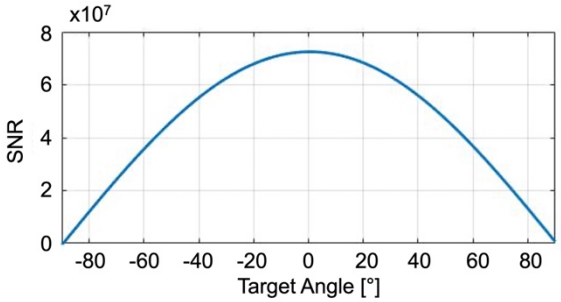

Figure 5 Plot of the SNR value depending on varying incident angles

This experiment also shows a plausible behavior of the Lidar model. For entrance angles approaching 90° and -90° the SNR value drops towards zero. At the incidence angle of 0° the SNR Value is at its maximum. This is consistent with the Lambertian reflection model, and thus the model can be considered valid for this experiment.

Both tests show realistic behavior for the sensor. Minor inaccuracies are caused by an inaccurate parameterization. Nevertheless, the model is used for further processing as described and parameterized. In future projects, the parameterization and validation of the sensor in comparison to the physical sensor could be taken up again.

3. COMPLEX VIRTUAL TEST ENVIROMENT

The main issue of this paper is the analysis of Lidar signals under consideration of the dynamics of a real vehicle. Therefore, in addition to the Lidar model, a realistic urban environment and a physically correct vehicle model are required. Moreover, a suitable communication is required for the exchange of data between the individual sub-models. Furthermore, the generated Lidar point clouds including the associated metadata need to be exported. The presentation of the sub-models and the communication strategy is provided in the following sections.

3.1. Virtual Urban Envorinment

The requirements for the virtual city model are high, since the Lidar models uses the surface normal of the geometries for the calculation. Furthermore, the objects placed in the virtual environment must be arranged realistically according to a natural city scene. These objects include buildings as well as vehicles, pedestrians, road signs, lane boundaries, trees and other street furniture. Since the influence of vehicle dynamics will also be a part of the study, the topography of the terrain is also an important criterion for the urban model. In order to represent as many of these objects as possible, the cityscape is modelled on the basis of a real role model. The city of cologne serves as a model. The scenery is based on the surroundings between Cologne's main railway station and the river Rhine. This scenery is well suited as it contains many pedestrian crossings and interactions between different road users. All measurements are adopted from the real environment. The dimensions as well as the optics of surrounding buildings are realistic. Curbs, lantern trees and bins are positioned as realistic as possible and their appearance is adapted to the real worlds objects. Even if the optical look is secondary for the computation of the Lidar model, as it is reduced to the objects reflectivity, the virtual test environment is enriched by highly realistic textures. This will also enable the testing of camera-based sensor technology in later work. The generated virtual environment is visualized in Figure 6.

A major feature of the environment are the pedestrian dummies. These are represented by high quality 3D scans of real people and also animals in natural, real life poses. Each 3D-model consists of 30,000 up to 200,000 faces, depending on the complexity of the geometry. This is sufficiently accurate to use the surface normals for the calculation of the Lidar model. Figure 7 shows an example for a pedestrian dummy represented by 100,000 faces with and without textures. It is obvious, that the model is not only realistic because of its textures, the geometry itself is in a high and smooth quality. The virtual city model thus has all necessary features for testing the implemented sensors. However, an accurate model for the vehicle movement is still missing. This will be the topic of the next section.

Figure 6 Visualization of the complex urban virtual test environment.

Figure 7 Comparison of a pedestrian dummy represented by 100,000 faces with and without texture.

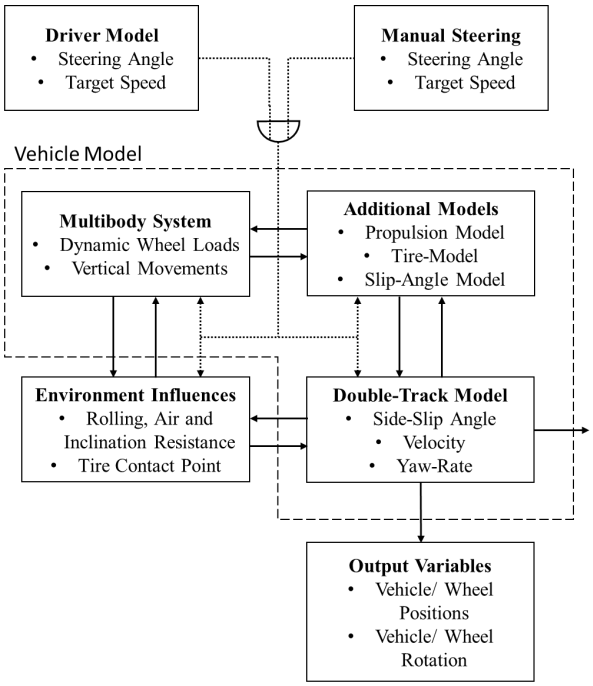

Figure 8 Representation of the vehicle model and its sub models

3.2. Vehicle Dynamics Simulation

For the testing the of virtual Lidar model the accurate simulation of the vehicle dynamics is an important issue. The vehicle movements like the pitching and rolling have a significant influence on the view of the sensors. Therefore, a vehicle model is needed that has sufficient accuracy to represent the realistic vehicle movements and still remains real-time capable. The implementation of the vehicle model takes place in MathWorks MATLAB Simulink. The vertical dynamics of the vehicle is modeled by a multibody system. It is composed of five bodies, representing the cars chassis and the four wheels. The lateral dynamic is simulated by a double-track model. Additional sub models represent the influences of driving and braking forces and of the tires depending on their slip angle. Moreover, a contact excitation is implemented at the wheel contact patches in order to be able to represent the influence of the topography in the virtual scenery. An overview of the model used given in Figure 8.

The model can be steered either manually by a user or by an automated driver model. The driver model is also implemented in Simulink. It uses target trajectories to follow a pre-defined path. These paths are created in the Unreal Engine and exported to Simulink by a defined number of waypoints, depending on the paths length. The algorithm is based on a pure pursuit path-tracking algorithm like presented in (Samuel, 2016). It uses look-ahead points in front of the car to follow the target trajectory permanently. The output of the controller is used as input for the vehicle steering. No further description of the vehicle model will follow at this point, since the elaboration of the model is not the topic of this work. It is only used to simulate the movement of the artificial vehicle. The vehicle is parametrized as a heavy SUV, as large body movements are to be expected in this vehicle class.

3.3. Model Communications

The previous sections show a detailed model for the simulation of Lidar data in a virtual environment. Additionally a complex virtual city model and a vehicle model to simulate the vehicles dynamics were presented. All these sub models are described individually so far. For fully comprehensive testing, an interaction between these models needs to be enabled. To be independent of the performance on a single computer system, the computations of the vehicle model and the virtual

environment are to be decentralized. The communication has to be bidirectional, since the vehicle model needs information from the scene and vice versa. Before a communication can be established, it must be clear which data volumes are to be transmitted. The expected data and their size are summarized in Table 4.

Table 4 Data to be transmitted between the Unreal Engine (UE) and MATLAB

Data	**Quantity**	**Direction**
Vehicle Coordinates	30 Float Values → 120 Bytes	MATLAB → UE
Contact Patch Coordinates	4 Float Values → 16 Bytes	UE → MATLAB
Lidar Results	Depending on Configuration	UE → MATLAB

It turns out that limited amounts of data are expected for the height coordinates and the vehicle positions. Only the amount of Lidar data to be send is not constant. For the Lidar sensor of Valeo, described in Section 2.4 an amount of 2,320 points can be detected at maximum. For every point, three coordinates and the SNR value need to be sent to MATLAB. If each value is stored and sent as a floating point variable, 37,120 bytes are used. The amount of data allows the usage of the User Data Protocol (UDP), since its maximum payload size is 65,507 bytes. UDP is a connectionless network protocol using the IP layer. It provides no guarantees for the delivery of data packages. Losses are possible. (Fairhurst, 2008) Since the sending on data is done in a much higher frequency, than the need to receive is, package losses are irrelevant in the given implementation. The protocol uses so-called sockets, defined by a computers IP-Address and a port for the communication. For the transmission of the data summarized in Table 4, it is practicable to define a UDP socket for each Datatype. Hence, three UDP sockets are used. The data transmission takes place in a worker thread, running at a higher frequency than the simulation of the model itself. Thereby it can happen that the same data is sent several times. However, this means that MATLAB and the Unreal Engine do not need to be synchronized. At the start of each simulation step, the vehicle coordinates received from MATLAB are used to position the vehicle as well as its wheels at the specified location within the Unreal scene. Depending on this new location, the computation of the Lidar Model is executed and the data are stored in arrays as described before. After that, the new height coordinates of the vehicles tire contact patches are determined and stored. As with the Lidar model, a Linetrace is used to obtain the height coordinates, starting at the wheel centers. The generated data are used for transmission until new data are provided with the next simulation step. A test confirms the error-free operation of the model network without major latencies in real time.

4. INVESTIGATION OF THE LIDAR DETECTION ON A SPEED BUMP

In the following sections, the previously developed and implemented model network is used for an exemplary analysis of the detection behavior of a Lidar sensor on a typical inner-city speed bump with an pedestrian crossing. In this scenario, large body movements of the test vehicle are to be expected. At the same time the permanent detection of pedestrians in mandatory. The investigation should clarify whether dangerous borderline cases of detection are to be expected with the combination of different driving speeds and pedestrian positions.

4.1 Test Setup

As mentioned before, the tests for the Lidar detection are to be carried out on a typical inner-city speed bump as visualized in Figure 9. However, the speedbump is not yet to be found in the scenario. As the influence of the driving dynamics on the Lidar is also tested, it is desirable that the topography of the terrain is not flat. Hence an artificial speedbump is placed into the scene. It is positioned before entering a roundabout on a road with a slight incline of 2.9 %. The elevation has a height of 200 mm and a length of 20 m at the top. The ramp of the bump has an angle of 10°. The alignment is adapted to the gradient of the road. For the investigation, 12 identical passenger dummies are placed on the bump. The measured test person is 1.75 tall and have a natural

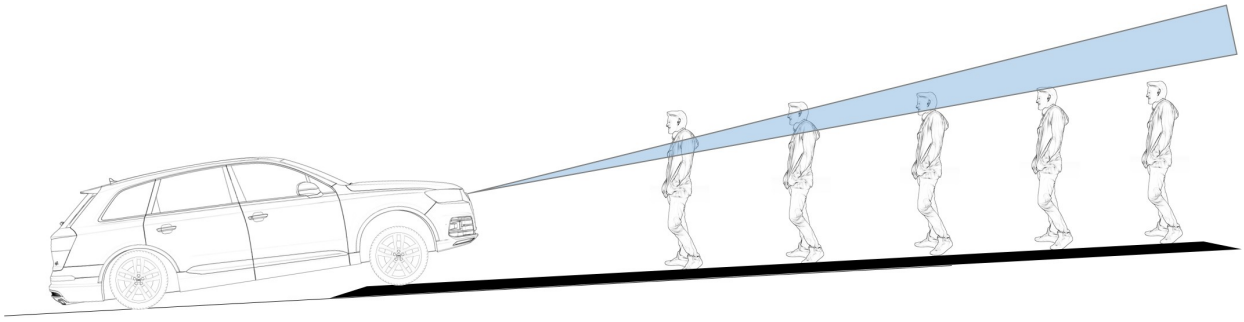

Figure 9 Testing concept for the investigation of the Lidar detection

pose found in the cityscape. In order to test not only the detectability of large pedestrians, there are also 12 dogs placed in the scenery next to each pedestrian with a height of 0.75 m. The dummys are positioned laterally offset at a distance of 2 m from each other. The height coordinated of the dummies are adjusted to the slope of the terrain so they stand on the roads surface. The final Test setup within the virtual environment is visualized in Figure 10.

Figure 10 Test setup within the virtual scene

4.2 Test Execution

For the execution of the test series, the Lidar model and the vehicle model are parametrized as described in Section 2.4 and Section 3.2 respectively. The setup represents the *Valeo Scala 3D Laser Scanner* fitted on a heavy SUV. For the execution of the tests a spline-path for the automated steering on the vehicle using the implemented pure pursuit controller is defined and exported to MATLAB. The vehicle will drive up the road towards the roundabout and pass the speed bump on its way. The recording of the signal is triggered two meters in front of the bump. The tests will be executed iteratively, starting with a vehicle speed of 5 km/h. The velocity is raised by 5 km/h on every cycle, up to 50 km/h. The results of the Lidar are stored with a frequency of 25 Hz, corresponding to the real Lidars scan frequency (Hexagon, 2021). The stored data are sent to MATLAB for further analysis and visualizations.

5. TEST RESULTS AND EVALUATION

Based on the recoded raw data the evaluation of the executed tests take place in the following section. First the way to achieve results is explained. This includes the data processing and the visualization on the results. After that, an analysis of the results is then carried out.

5.1 Data Processing and Visualization

As already mentioned, the data processing takes place in MATLAB. The starting point are four matrices, three for the determined coordinates of the Lidar and one that containing the corresponding SNR values. Taking into account the findings from section 2.4, a simple method is used to consider the SNR values. According to the Tables 2 and 3 all identified points with an SNR of 5 or less, corresponding with a TPR of 10 %, are neglected. For all points with a SNR between 5 and 20 half of the data points are deleted at random, to artificially simulate the TPR of 50 %. All data with a SNR of 20 and above are considered in their entirety. The processed data are plotted in a next step using a three-dimensional, rotatable Cartesian coordinate system. The visualization is designed to view the results for each step of the Lidar simulation individually. Figure 11 shows an example of the data-visualization at the trigger point with a velocity 50 km/h.

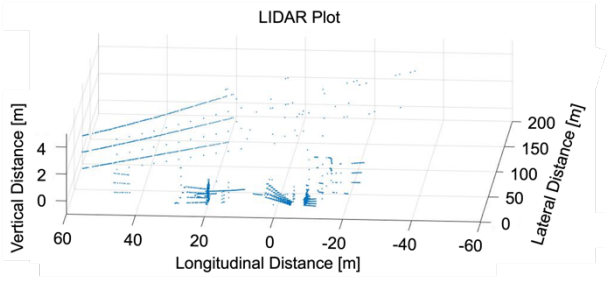

Figure 11 3D-Plot of the processed Lidar data at the Triggerpoint with 50 km/h.

In this Figure, almost all determined points are visualized. Many of the surrounding objects can be identified, including the pedestrians. The individual models, especially the dogs can hardly be distinguish. In order to analyze the area of interest, a plot reduced in size and viewed from the top suits the demands better. The results are shown in the Figures 12 to 15. The plots represent the signals of the Lidar received at a vehicle speed of 30 km/h at different frames. Each frame represents a time step of the simulation corresponding to the scan frequency of 25 Hz.

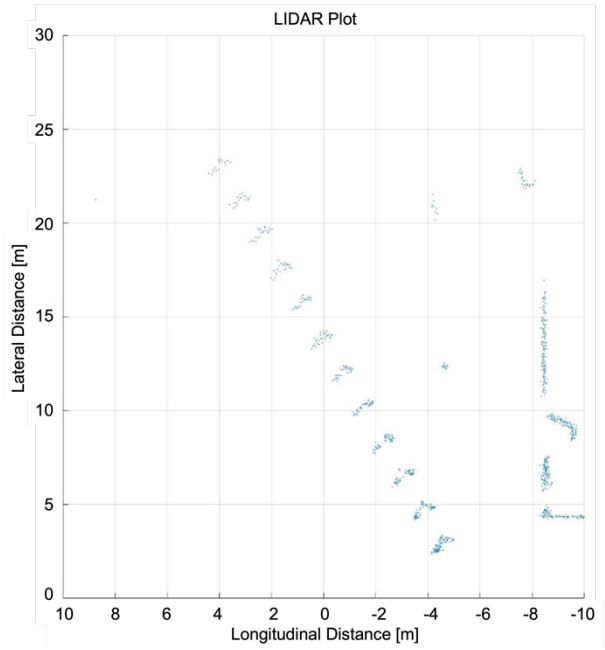

Figure 12 Plot of the processed Lidar Data at the trigger point with 30 km/h

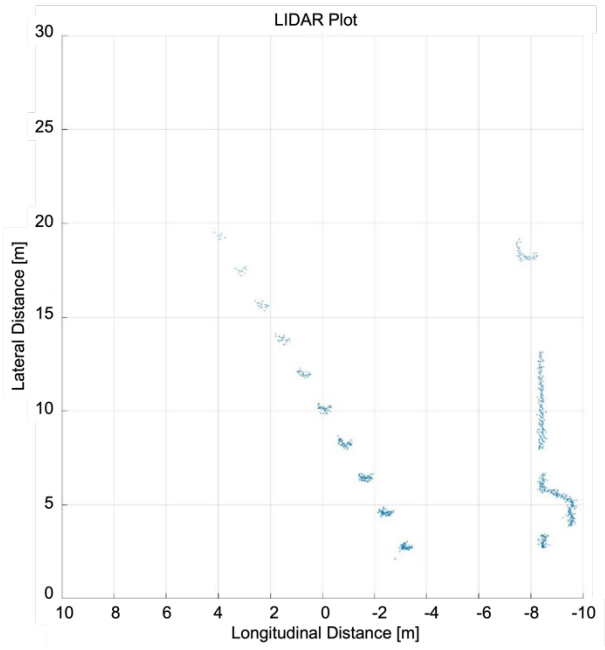

Figure 13 Plot of the processed Lidar Data at the simulation frame 13 with 30 km/h

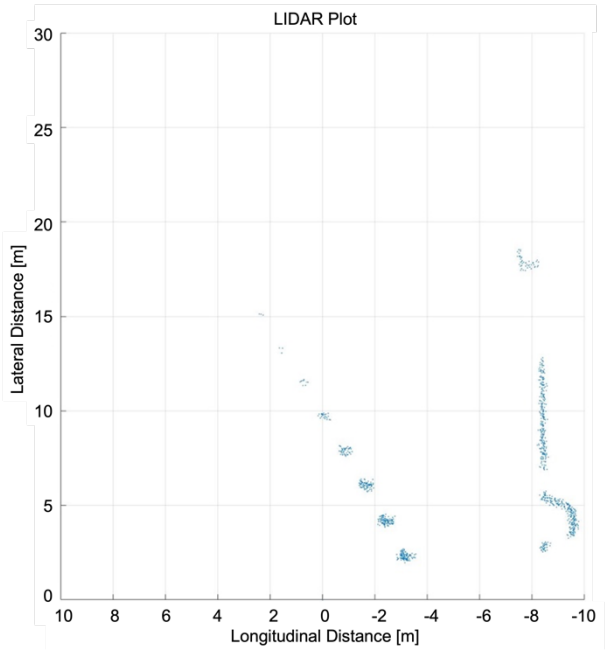

Figure 14 Plot of the processed Lidar Data at the simulation frame 15 with 30 km/h

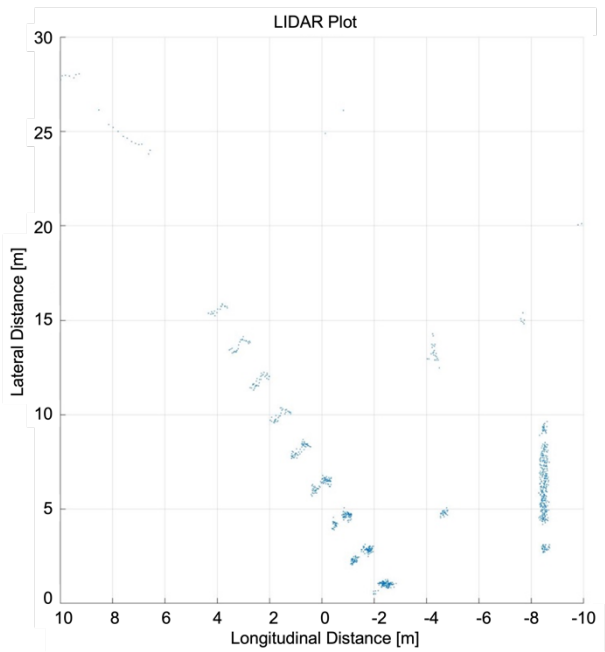

Figure 15 Plot of the processed Lidar Data at the simulation frame 23 with 30 km/h

Figure 12 shows, that all road user models can be detected and distinguish by the Lidar at the first frame, when the vehicle is two meters in front of the speed bump. At frame 13, visualized by Figure 13, the vehicle is at a position 4.3 Meters from the Triggerpoint. All dog models disappear from the field of view. The pedestrians are still visible. Two frames later, visualized in Figure 14, the pedestrian 11 and 12 also disappear. The number of points recognizing for the dummys 8 to 10 is significantly reduced. Figure 15 shows the data captured at the 23rd frame, when all disappeared objects are within the field of view again. The models one to three are not visible because the virtual car already passed them.

Since the figures only show the data for the test with a velocity of 30 km/h and it is not practical to show all determined plots at this point, the results of the data analysis are summarized in the Figures 16 and 17. The x-axis show the test velocities, the y-axis show the range the vehicle moves while the respective object is out of the view of the Lidar sensor. This value is computed using the vehicles velocity and the number of frames, the respective object is invisible. Only the results for the objects that disappear from the field of view of the Lidar in the meantime are shown.

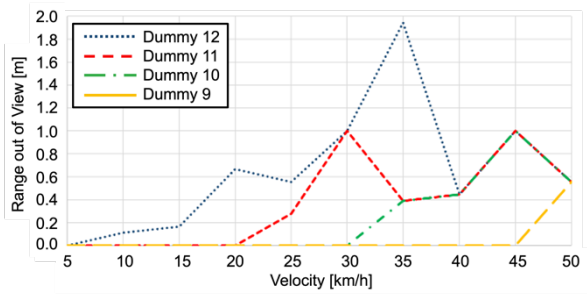

Figure 16 Visibility of the pedestrian dummies by the virtual Lidar for different test velocities

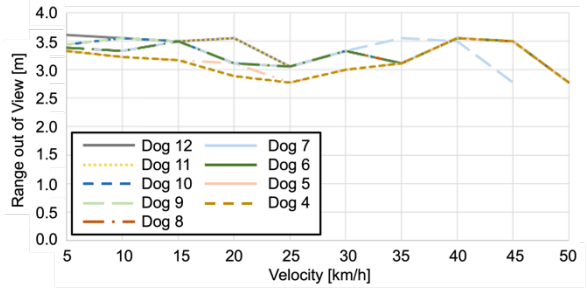

Figure 17 Visibility of the dog dummies by the virtual Lidar for different test velocities

5.2 Evaluation of Results

Now, based on the processed and visualized data of the last section, an evaluation of the results is carried out. Both, the physical behavior of the Lidar sensor as well as the vehicle dynamics and its influence on the road users visibility are discussed.

It is obvious that due to the limited size of the scenery under consideration, only a small fraction of the Lidar points determined is omitted. 0.1 % of the data points recoded within a range of 30 Meter in front on the test vehicle have a SNR smaller than five, 1.5% have a SNR smaller than 20. Nearly all removed points lie behind the area of interest where the dummies are located. The few points with a low SNR could be due to large entry angles of the Lidar beams on the wall surfaces in the scenery. A snapshotof the virtual reality, visualized in Figure 18, confirms the findings.

Figure 18 Visualization of the SNR values within the scenery

It is recognizable that all the detected Lidar points, representing the road users, are colorized in green. This confirms the high SNR values. Otherwise, the points would be colorized orange to red. This shows, that the influence of the physical behavior of the used virtual Lidar under the given test conditions do not have a major influence on the results. Further test could use the implemented model to analyze the detection behavior in harsh weather conditions like rain, fog and snow, by the manipulation of the atmospheric transmission coefficient.

If a closer look is taken at the simulation of driving dynamics, it is noticeable that its effects on the results are much greater. In the last section the vehicle movements have already been described. Based on the analysis of every recorded frame at all test velocities the Figures 16 and 17 were extracted. The results are now analyzed. A closer look at Figure 16 shows, that the influence of the vehicle dynamics on the Lidar signal is significant. At 5 km/h the pedestrian dummys are visible all the time. Even if the number of recognized data points for dummy 11 and 12 is reduced, since only their heads gets hit by the Lidar, they are still visible. With a raising velocity the

number of temporary invisible dummies increases. Also the distance the vehicle travels whilst the objects are invisible raises. This is due to the inertia of the vehicle, which causes it to move further upwards after driving onto the ramp. For dummy 12 the maximum is at a test speed of 35 km/h. After that, the value decreases again. With raising velocities, the rear axle hits the ramp earlier, hence the upwards movement is stopped earlier. This shows the great influence the vehicle dynamics have on the detection behavior of the Lidar. For the dog models the results are quite different. The values fluctuate much less and are at a fundamentally higher level of 3.5 meters for the vehicle movement. This is due to the low position of the dogs. To recognize them, the vehicles body and thus the alignment of the Lidar needs to be nearly parallel to the ground. Through this required parallelism, all objects are recognized at almost the same time. It can be seen that if the influence of driving dynamics had not been considered, this detection gap would not have been identified. In a real environment, this test situation could cause serious accidents. Even at a low vehicle speed of 5 km/h the car moves 3.5 meters without recognizing any of the dogs 4-12. If a dog or even a child with the same height would run in front of the car at the moment it enters the bump, it is questionable if the road user get detected. Further tests could be dedicated to this question. A dynamic scene with time controlled objects crossing the street could answer those questions. Also the influence of a potential sensor fusion with a camera could be tested. Automotive cameras are also bound to the vehicles body movements It is questionable if the camera also loses sight of the objects.

6. CONCLUSION

The paper shows that current Lidar simulation models have shortcomings when combining vehicle dynamics simulation and physical Lidar modelling. Hence, a raytracing based approach is presented to compute the fraction of a sent laser signal impinging on the sensor again. In addition, the noise powers acting on the system, consisting of the dark current and the sun-induced noise, are computed and used to form the SNR. This value makes it possible to make a statement whether a point is recognized. A verification of the model shows plausible results compared to the real sensor. Additionally a vehicle dynamics model is introduced. It uses a two track model and a multibody system to accurately simulate the vehicles movements in interaction with the realistic virtual environment. An investigation of a typical inner-city scenario shows, that this type of Lidar simulation makes sense. Different road users are out of view of the Lidar for up to 3.8 m, whilst the vehicles crosses a speed bump, depending on their position and the test vehicles velocity. At the same time it shows, that the influence of the physical modeling of the Lidars detection behavior is small. In the considered scenery, nearly all laser beams generate a sufficient receiving signal. Further investigations could take a closer look at the physical Lidar model. Influences of different environmental properties should be tested. It is to be expected that fluctuations in the atmospheric transmission due to rain, snow, fog or dust lead to a reduced receiving power. In addition, different sun intensities could be tested. The current model is based on the assumption that all surfaces are Lambertian reflectors. In real environments also specular reflecting objects like glass panes or lacquered surfaces can be found. Therefore, it makes sense to implement the case of specular reflections into the model. In addition, the connection of the vehicle and the Lidar model could be topic of further tests. The described scenario already shows a major influence on the results. Further scenarios could be cornering and turning with different speeds or the crossing over crests. Furthermore, dynamic scenarios would be interesting. Related to the bump it could be tested, if a child running in front of the car is detected. At this point, a look at the fusion of several sensor technologies would make sense. Finally, in order to enable a hardware in the loop testing, the generated data could be feed into evaluation algorithms or real Lidar systems.

ACKNOWLEDGEMENT– The Project is supported by the Ministry of Economic Affairs, Innovation, Digitization and Energy of North Rhine-Westphalia.

REFERENCES

Byeon, M, Yoon, S. W. (2020). Analysis of Automotive Lidar Sensor Model Considering Scattering Effects in Regional Rain Environments, *IEEE Access* **8, 1109**, 102669 - 102679

Castaño, F., Beruvides, G., Villalonga, A., Haber, R. E. (2020). *Computational Intelligence for Simulating a LiDAR Sensor* in van Driel, W. D., Pyper, O., Schumann, C. *Sensor Systems Simulations*, 149-178

Fairhurst, G (2008). *The User Datagram Protocol (UDP).* https://erg.abdn.ac.uk/users/gorry/course/inet-pages/udp.html

Gadringer, M. E., Schreiber, H., Gruber, M., Vorderderfler, M., Amschl, D., Bösch, W., Metzner, S., Pflügl, H. Paulweber, M. (2018). Virtual reality for automotive radars, *Elektrotechnik & Informationstechnik* **135, 4-5**, 335-343

Gusmão, G. F., Barbosa, C. R. H., Raposo, A. B. (2020). Development and Validation of LiDAR Sensor

Simulators Based on Parallel Raycasting. *Sensors* **20, 7186**.

Hamamatsu Photonics K.K. (2018). https://www.hamamatsu.com/resources/pdf/ssd/s12023-02_etc_kapd1007e.pdf

Hexagon Autonomy and Positioning (2021). https://autonomoustuff.com/products/valeo-scala

Kernhof, J., Leuckfeld,J. and Tavano, G. (2018). *LiDAR-Sensorsystem für automatisiertes und autonomes Fahren*, in: Thille, T. *Automobil-Sensorik 2*. Springer Vieweg. Berlin, Heidelberg.

Kim, S., Lee, I.and Kwon, Y. J.(2013). *Simulation of a Geiger-Mode Imaging LADAR System for Performance Assessment*. Sensors 13 ,7, 8460-8489

Mackowiak, V., Peupelmann, J., Ma, Y. and Gorges, A. (2015). *NEP – Noise Equivalent Power*. Thorlabs, Inc., Thorlabs GmbH.

Muckenhuber, S., Holzer, H., Bockaj, Z. (2020) Automotive Lidar Modelling Approach Based onMaterial Properties and Lidar Capabilities, *Sensors* **20, 3309**

Newport Corporation (n.d.). *Solar Simulator Sample Calculations.* https://www.newport.com/n/solar-simulator-sample-calculations

Samuel, M., Hussein, M. and Mohamad, M. B. (2016). *A Review of some Pure-Pursuit based Path Tracking Techniques for Control of Autonomous Vehicle. International Journal of Computer Applications* **135, 1**, 35-38

Statista (2020a) https://de.statista.com/statistik/daten/studie/1083873/umfrage/anteil-der-pkw-mit-fahrassistenzsystemen-in-deutschland/

Statista (2020b). https://de.statista.com/statistik/daten/studie/1108736/umfrage/meinungsumfrage-zu-assistenzsystemen-in-autos-in-deutschland/

U.S. Naval Academy (n.d.). *Atmospheric Transmittance.* https://www.usna.edu/Users/oceano/pguth/md_help/remote_sensing_course/atmos_transmit.htm

Wachenfeld, W., Winner, H. (2015) *Die Freigabe des autonomen Fahrens* in Maurer, M., Gerdes, J.Ch., Lenz, B. Winner, H. *Autonomes Fahren*, 439-464

Wang, G., Zhuang, Y, Gu, H., Hu, H. (2019). Automatic Generation of Synthetic LiDAR Point Clouds for 3-D Data Analysis. *IEEE Transactions on Instrumentation and Measurement* **68, 7,** 2671-2673

Weber, H., (2018). *Funktionsweise und Varianten von Lidar-Sensoren*. Sick AG.

Winner, H., Hakuli, S., Lotz, F. and Singer, C. (2016). *Handbook of Driver Assistance Systems*. 1st edn. Springer International Publishing. Cham.

.