TUM School of Computation, Information and Technology
Technische Universität München

Simulation of LIDAR sensors in real environmental conditions

Jose Roberto Vargas Rivero

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. habil. Dr. h.c. Alexander W. Koch

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Jia Chen
2. Prof. Dr. Abhinav Valada

Die Dissertation wurde am 03.07.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 08.05.2024 angenommen.

Abstract

As weather can suddenly change, future self-driving system have to be able to cope with adverse weather for at least as many seconds as needed for the driver to take over. Given that weather affects each of the main sensors used to accomplish the driving task, fast and accurate weather simulations are an important step towards the successful validation and testing of autonomous driving systems. From the main sensors used in autonomous driving, this thesis focuses on the LIDAR sensor. Given its high resolution and active distance measurement principle, the LIDAR provides valuable information for the driving task but simultaneously its performance is considerably affected by adverse weather. This thesis presents hybrid models for the simulation of weather related effects. These models are hybrid because some aspects of the models are physically based while others are phenomenological. The combination of both modes allows the proposed models to be able to run in real time while simultaneously been intuitive in their use. For the simulation of the sensor itself Raytracing is suggested due to its flexibility, broad adoption and speed.

The first model is related with the simulation of dirt and other substances that accumulate on the cover of the sensor. These substances change the optical properties of the cover. A general consideration of this phenomena is not possible as the way and type of substances that adhere to the sensor cover depend on the shape and material of the cover itself. To solve this problem a methodology is defined to collect real world samples and measure their optical properties. The obtained data can be used to create a texture database that can be easily included into a Raytracing based simulation.

The second model considers the effect of rain, fog, snow and changing background radiation. Similarly to the first model the development of a generic model for any LIDAR sensor is discarded, as a physically accurate simulation of the scattering caused by millions of drops or snowflakes would not be possible in real time. Additionally, usually not all the sensor internal parameters are known. For this reason a stationary measurement set up is proposed, collecting data from a sensor placed outdoors for a period of nine months. Using the extracted data is possible to build a stochastic sensor model that reproduces the distribution of real world detections caused by each of the considered weather types. It is also found that the effect of a changing background radiation can be simulated in real time by changing the noise level of the sensor.

The third model reproduces the effect of spray water on the LIDAR sensor using particle systems. These particles do not represent single water drops but a concentration of drops high enough to generate a detection on the LIDAR sensor. With this simplification the number of needed particles can be enormously reduced making real time simulations possible. For the collection of point clouds affected by spray water a set up is used with two vehicles. One vehicle generates the spray and the other captures the generated point clouds using a LIDAR, while simultaneously measuring the street water level using a specialized measuring device. A comparison of the proposed synthetic model and the corresponding real point clouds shows a Pearson correlation of 0.5 up to 0.73 of the distribution of detections in space.

Finally, the spray model is used to demonstrate how synthetic and real data combined form a powerful tool for the development of autonomous driving systems. The example used analyses the performance of a classifier for detection of solid objects on spray clouds. This classifier can be trained without having to collect real data for such a situation, which would be extremely dangerous and difficult. This is done by adding an object to the spray simulation, automatically extracting the synthetic detections generated by this object and using them to augment real spray point clouds. The resulting point clouds are a labeled and physically accurate representation of the desired scene.

This thesis shows that all the previously mentioned models can be related into a common framework for the simulation of weather related effects by dividing the path of the light emitted by the sensor into three parts: sensor cover, atmosphere and object emission. Object emission describing for example spray water but also something like sand or snow falling from a truck or water vapor from the exhaust pipe. Different simulation techniques adapt better to each of the regions but in general the model, weather it is a 2D texture an stochastic distribution or a particle system is sampled by a ray changing its intensity and/or direction. This framework is integrated with the main current standards for the simulation of Advanced Driver Assistance Systems (ADAS) and Self-Driving Systems (SDS) like OpenDRIVE and OpenSCENARIO. The combination of the standards and the framework allow the user to flexibly adjust the complexity of the simulation based on his needs and available data. In this way the developer of autonomous driving system can use simulation from the start of development until the end of the life of the product and even for new products (e.g new vehicle models). By doing this is possible to reduce costs, risks and time to market.

Zusammenfassung

Da das Wetter sich plötzlich ändern kann, müssen zukünftige selbstfahrende Systeme in der Lage sein, unter ungünstigen Wetterbedingungen für mindestens so viele Sekunden weiterzufahren, wie der Fahrer benötigt, um die Kontrolle zu übernehmen. Da das Wetter jeden der wichtigsten Sensoren beeinflusst, die für die Bewältigung der Fahraufgabe verwendet werden, sind schnelle und präzise Wettersimulationen ein wichtiger Schritt auf dem Weg zu einer erfolgreichen Validierung und Erprobung von autonomen Fahrsystemen. Von den wichtigsten Sensoren, die im autonomen Fahren verwendet werden, konzentriert sich diese Arbeit auf den LIDAR-Sensor. Aufgrund seiner hohen Auflösung und des aktiven Abstandsmessprinzips liefert der LIDAR wertvolle Informationen für die Fahraufgabe, ist aber gleichzeitig stark von ungünstigen Wetterbedingungen beeinträchtigt. Diese Arbeit präsentiert hybride Modelle für die Simulation von Wettereffekten auf einen automotiven LIDAR-Sensor. Die Modelle sind hybrid, weil einige Aspekte physikalisch modelliert sind, während andere phänomenologisch modelliert sind. Die Kombination aus beiden Simulationsarten ermöglicht es den vorgeschlagenen Modellen, in Echtzeit zu laufen, während ihre Nutzung intuitiv bleibt. Für die Simulation des Sensors wird Raytracing aufgrund seiner Flexibilität, breiten Akzeptanz und Geschwindigkeit vorgeschlagen.

Das erste Modell befasst sich mit der Simulation von Schmutz und anderen Substanzen, die sich auf dem Sensordeckel akkumulieren. Diese Substanzen ändern die optischen Eigenschaften des Sensordeckels. Eine allgemeine Betrachtung dieses Phänomens ist nicht möglich, da die Verteilung und Art der Substanzen, die auf dem Sensordeckel kleben, von der Form und dem Material des Deckels abhängig sind. Um dieses Problem zu lösen, wird eine methodologische Vorgehensweise vorgeschlagen, um reale Proben zu sammeln und ihre optischen Eigenschaften zu messen. Die so gesammelten Daten können verwendet werden, um eine Texturdatenbank zu erstellen, die auf einfache Weise in einer Raytracing-basierten Simulation verwendet werden kann.

Das zweite Modell betrachtet den Effekt von Regen, Nebel, Schnee und einer wechselnden Hintergrundstrahlung. Ähnlich wie beim ersten Modell wird von der Entwicklung eines generischen Modells abgesehen, das für jeden beliebigen LIDAR-Sensor gültig ist. Der Grund dafür ist, dass eine genaue physikalisch basierte Simulation der Streuung, die bei Millionen von Tropfen oder Schneeflocken erzeugt wird, nicht in Echtzeit berechnet werden könnte. Außerdem sind normalerweise nicht alle internen Sensorparameter bekannt. Aus diesem Grund wird ein statisches Messsetup vorgeschlagen, bei dem ein Sensor im Außenbereich platziert wird, um Daten zu sammeln. Die Sammlung findet über eine Periode von neun Monaten statt. Auf den extrahierten Daten basierend kann ein stochastisches Sensormodell gebaut werden, das die Verteilung von realen Detektionen für jede Wetterart reproduziert. In diesem Zusammenhang wird auch festgestellt, dass Änderungen in der Hintergrundstrahlung durch Anpassung des Rauschpegels des Sensors in Echtzeit simuliert werden können.

Das dritte Modell reproduziert den Effekt von Sprühwasser auf den LIDAR-Sensor mithilfe von Partikelsystemen. Diese Partikel entsprechen nicht einzelnen Wassertropfen, sondern einer Sammlung von Tropfen, die eine Reflexion des Laserlichts bewirken und stark genug sind, um eine Detektion auf dem LIDAR-Sensor zu verursachen. Durch die vorgeschlagene Vereinfachung kann die Anzahl der benötigten Partikel enorm reduziert werden, wodurch ein Echtzeit-Betrieb der Simulation möglich wird. Um die durch Gischt betroffenen Punktwolken zu sammeln, wird ein Messsetup verwendet, bei dem zwei Fahrzeuge eingesetzt werden. Das erste Fahrzeug generiert Gischt, während das zweite die entsprechenden Punktwolken speichert. In dem zweiten Fahrzeug wird ein Gerät zur Messung des aktuellen Wasserpegels auf der Straße montiert. Ein Vergleich der Raumverteilung zwischen synthetisch generierten und realen Punktwolken ergibt einen Pearson-Korrelationskoeffizienten von 0,5 bis 0,73. Schließlich wird das Gischtmodell verwendet, um zu demonstrieren, wie durch die Kombination von synthetischen und realen Daten ein mächtiges Tool für die Entwicklung von autonom fahrenden Systemen entsteht. Das verwendete Beispiel bewertet die Leistung eines Klassifikators, der in der Lage ist, feste Objekte in einer Gischtwolke zu detektieren. Dieser Klassifikator kann trainiert werden, ohne reale Daten sammeln zu müssen, was in diesem Fall unter extremer Gefahr und sehr selten möglich wäre. Um synthetische Daten zu erzeugen, wird ein Objekt in die Gischtsimulation hinzugefügt. Die Detektionen, die durch dieses Objekt generiert werden, werden automatisch extrahiert und verwendet, um eine reale aus Gischt generierte Punktwolke zu augmentieren. Die resultierenden Punktwolken sind eine beschriftete und physikalisch korrekte Repräsentation der gewünschten Szene.

Dieses Thesis zeigt, dass alle zuvor genannten Modelle in ein gemeinsames Framework eingeführt werden können, indem der Lichtweg in drei Bereiche unterteilt wird: Sensordeckel, Atmosphäre und Objektemission. Objektemission beschreibt zum Beispiel Sprühwasser, aber auch etwas wie Sand oder Schnee, der von einem Lastwagen fällt oder Wasserdampf aus der Auspuffrohre. Verschiedene Simulationstechniken passen sich besser an die einzelnen Bereiche an, aber im Allgemeinen wird das Modell, ob es eine 2D-Textur, eine stochastische Verteilung oder ein Partikelsystem ist, durch einen Strahl gesamplet, der dadurch seine Intensität und/oder Richtung ändert. In dem Framework werden die wichtigsten aktuellen Standards für die Simulation von autonomen Fahrsystemen und sicherheitskritischen Systemen, wie OpenDRIVE und OpenSCENARIO, integriert. Durch die Nutzung dieser Standards und des Raytracings wird es für den Nutzer möglich, die Komplexität der Simulation flexibel zu justieren, die auf seinen Bedarf und den verfügbaren Daten basiert. Somit wird es für die Entwickler von autonom fahrenden Systemen möglich, von Anfang der Entwicklung bis zum Ende der Lebensdauer des Produkts und sogar für neue Produkte (z.B. neue Fahrzeugmodelle) Simulationen zu nutzen. Auf diese Weise können Kosten, Risiken und Markteinführungszeit reduziert werden.

Contents

1 Introduction

1.1 Background

The amount of single lines of code present in vehicles is increasing very fast. For a modern vehicle, including driving and safety systems as well as infotainment and apps it is calculated to be around 100 million [1].

As every other software, this software has to be tested to validate it is working as expected. The testing requirements are based on the classification of the respective function based on its criticality. For example a breaking system that works by wire (electronically controlled) needs to have a much higher reliability and therefore testing requirements [2] as for instance an app used to pre-heat the vehicle. The high reliability requirements for safety-critical functions conduces to very high validation times. This problem can be understood by looking af the following equation [3]:

$$P_{sys}(t) = 1 - e^{-Ktp} \tag{1.1}$$

Were K is the number of inputs per unit time to a certain system for example a software module and p is the probability of failure per input which is assumed constant. This equations provides the probability of a system failure after a certain time using a Poisson approximation of a binomial distribution. Under this approximation the time it takes to find a failure in a system with a probability of failure of 10^{-7} to 10^{-9} in an hour is in the order of 10^{10} hours. This number can be reduced if simultaneously multiple copies of the same system are tested [3], by using formal proofs when possible and by using fault injection [4].

Additionally, testing needs to happen very early in the development process, a paradigm known as shift-left, in which software bugs should be found and corrected as soon as possible to reduce costs and decrease time to market [1].

Summarizing, there is a need for having software tests during the development of vehicle software. In general, but specially when considering safety-critical functions, these tests should be:

- Available early in the development process.
- Fast.
- Frequent.
- Parallelizable.
- Automatic.

- Cheap.

Considering now specifically the software required for autonomous driving functions, we can broadly divide it into: a perception module, followed by a planing module and finally an action module as shown in Figure 1.1. The environment, which includes other road users, roads, traffic signals, weather, etc. is the input to the perception module but also reacts to the actions taken by the vehicle. If we want to test these modules under the previously presented requirements we are forced to "emulate" the environment. This environment emulation may exist in different quality levels depending on the types of tests that are needed. For some of these tests the quality of the emulation needs to be high enough so that it should rather be called a simulation.

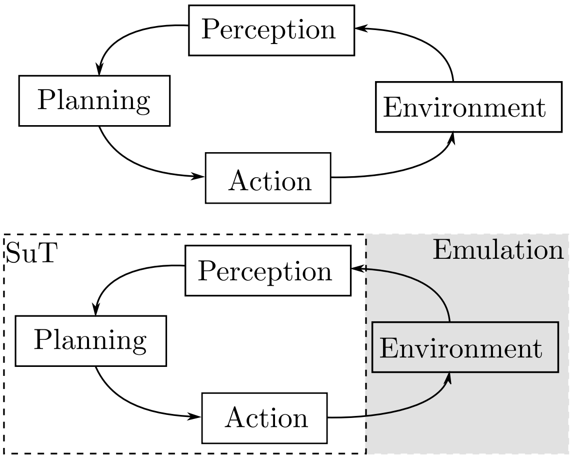


Figure 1.1: Close loop used in autonomous driving functions (up). For testing purposes the loop is broken into a System under Test (SuT) composed by the perception, planing and action modules and an environment which is replaced by a emulation / simulation (down)

We can for example start in the development process with a very simple model of the environment which is already enough to discover some bugs in the code. After those bugs are corrected a more complex environment is needed to discover new bugs. This process continous until a very accurate model of the environment is needed. Figure 1.2 shows an example in which each iteration uses an environment version that is good enough to reach a certain maturity (development cycle). For the next version a more advanced environment is needed, otherwise the performance of the function in the real world may diverge from the test results. This problem is known as the reality gap [5]. When a real prototype is available (dashed line) is possible to add the information obtained from real measurements into the simulation increasing its quality and reducing the reality gap. This has to be done carefully as prototypes may still considerably diverge in some aspects from the final hardware version. Another way of using the simulation is to guide the data collection efforts to focus on those aspects that are not well understood or are difficult to simulate.

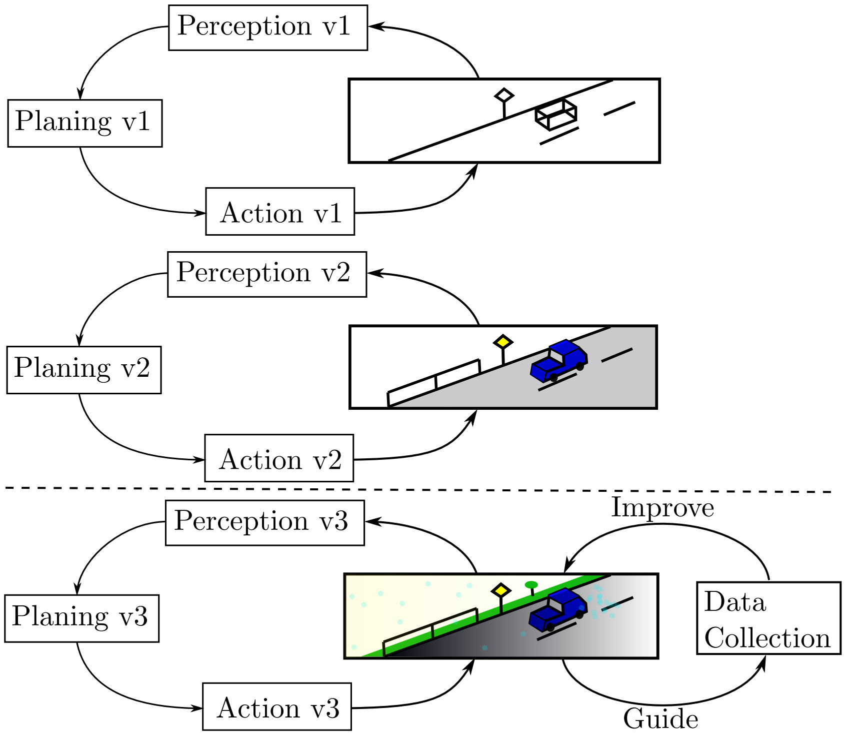


Figure 1.2: Sequential improvement of the environment model based on the needs of the SuT. The version number: v1, v2, v3 indicate how the SuT improves. An initial model can use simple synthetic geometries and materials. Later, complex geometries and textures can be used. Weather related effects like for instance the effect of spray water and sunlight can also be added. As soon as a real prototype is available (dashed line) is possible to improve the simulation using real data and hence to reduce the reality gap. The simulation can also be used to guide data collection campaigns.

In conclusion advanced environment models are a necessary part of the development and validation of autonomous driving functions. The development of these kind of models has been however impaired for three main reasons:

- There is no consensus about how to effectively transfer knowledge regarding weather related effects between real measurements and simulation and vice versa (improve and guide loops in 1.2).
- There is no standard approach for progressively including weather related effects into the simulation.
- The reality gap is not being measured or is measured in late development stages reducing confidence on simulation results.

This thesis proposes alternatives to deal with each of these problems. Solving these problems is an important step towards a cost effective use of simulation. When used correctly simulation can save costs over multiple vehicle generations because the knowledge and technology can be easily adjusted. As illustrated at the begining of the chapter, a cost effective simulation is a key enabler for autonomous vehicles. These kind of vehicles are expected to improve transportation by: increasing safety, comfort, accesibility and sustainability as well as reducing transportation costs [6].

1.2 Methodology

The focus of this thesis will be one of the sensors used for autonomous vehicles: the LIDAR sensor [7]. For this sensor the logical interface between the environment as presented in 1.1 and the SuT is taken to be the point cloud. Although smart sensors are able to provide higher abstractions, like pre-classified objects, the point cloud is a generic interface for many LIDAR sensors [8, 9]. Additionally, it will be assumed that there is a prototype available for the sensor and therefore that it is possible to make real measurements to improve the simulation (lower part of Figure 1.2). The interface to the action module is taken to be the 3D coordinates of the sensor axis and its orientation. In this way every kind of movement in 3D space can be simulated. The schema of the environment can hence be concretized as shown in Figure 1.3.

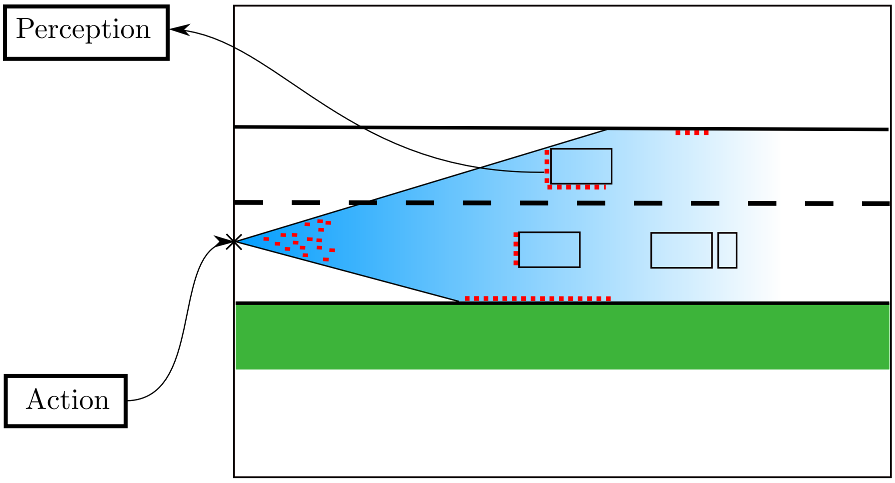


Figure 1.3: Environment as perceived by the LIDAR sensor. The image shows an eagle view of one layer of a highly simplified point cloud (red points) generated by a LIDAR sensor for a certain environment. The sensor's field of view is shown in blue. The generated point cloud is the input for the perception module. The position and orientation of the sensor is provided by the action module (position marked with a 'x'). The reflections in the near field of the sensor simulate extra detections caused by weather related effects or dirt on the sensor cover.

After having defined the logical interfaces, the question remains how the virtual environment and the sensor should be simulated. There have been different approaches to simulate the sensor as well as the environment. Sensor models can broadly be divided into [10]

- Low fidelity: is a fast but simple model of the sensor that considers mostly geometrical aspects.
- Medium fidelity: includes physical effects and detection probabilities. As it works on a phenomenological level data collection is required.
- High fidelity: is computationally very expensive but provides the highest level of realism. It works by solving the physics equations of the problem in consideration.

The environment can also be classified into three categories:

- Synthetic: generated manually, automatically or semi-automatically without having a equivalent in the real world.
- Measured: obtained by collecting aerial or terrestrial measurements of real environments.
- Hybrid: combines elements that are measured with elements that are generated.

The next section explain the differences of each category and introduces the sensor and environment type used in this thesis. Before doing that some of the more widespread standards used to simulate virtual environments in the context of autonomous driving are presented.

1.2.1 Standards

Proposing a method to simulate weather related effects is of lesser use if it is not compatible with the most popular simulation standards used currently in industry and academia. For this reason in this section the main standards are introduced. They are divided based on the type of information each of them describe.

1.2.1.1 Dynamic Behaviour

A way to implement the improve and guide loop from figure 1.2 is to use a scenario based approach. In this approach a scenario that describe a certain manouver involving one or more vehicles and other traffic participants is defined. The scenario is followed in reality and in simulation, the obtained results are compared and in case of relevant differences either actions to improve the simulation or actions to focus the data colletion efforts can be taken. In case no hardware exists the scenario can be tested in simulation until a certain maturity is reached. Another advantage of using scenarios is that a scenario library can constitute a common basis for industry and gouvernamental organizations to respectivelly test or verify that a SDS is working as it should [11].

There are two versions of OpenSCENARIO the version 1.x uses eXtensible Markup Language (XML) as serialization format. The advantage of XML being that multiple tools are available to validate, extract or add information to it. The scenario is composed by stories, which contain acts, which contain events and actions [11]. The events and actions define when something happens and what happens. The definition is however ambiguous Association for Standardization of Automation and Measuring systems (ASAM) so that different simulation tools may produce different results. The version 2 of OpenSCENARIO deal with this problem by defining constrains. If the solution fullfills the constrains then the scenario is considered as successfully pased. Instead of XML OpenSCENARIO 2 uses a Domain Specific Language (DSL), scenarios are seen as complex programs that can call each other serially or in parallel. The entry point for the program execution (begining of the scenario) is freely selectable [12].

Regarding weather phenomena OpenSCENARIO 2 defines a celestial light source that depends on the current weather, geodetic position and date-time. Using this light source, a sun and /or moon can be defined. Weather itself is a data structure containing variables for: precipitation intensity (either rain or snow), the visual range of fog and wind speed and direction between others.

Besides OpenSCENARIO there are other standards to define dynamic behavior. For example OpenODD is standard to define in a machine readable way the ODD. The International Organization for Standardization/ Working Draft (ISO/WD) 34501 up to ISO/WD 34504 standarize general terminology, define a scenario based framework for safety evaluation, define the ontology for the ODD and quality levels and categorizations for scenarios respectively [13].

1.2.1.2 Static Behaviour

The OpenDRIVE standard similarly to OpenSCENARIO 1.x uses XML as file format. In this case not for the descriptions of maneuvers but to describe the road geometry, road marks and traffic signs. To define the road a reference line is used. Road lanes can be added as needed to the left and right of this reference line. Increasing integer values mean extra lanes further left of the reference line while decreasing negative values mean extra lanes further right of the reference line. The beginning of the reference line defines the origin of a two dimensional reference system (relative to the reference line). Using either this 2D reference system or a global 3D coordinate system (whose origin is defined in the file) features like traffic signs or road marks can be placed. A whole road network is built by joining roads together. When more that two roads have to be joined a junction can be used. The junctions define how the lanes of the different roads are connected with each other. Road geometries can be adjusted by selecting different primitives to define the reference line, e.g. lines, spiral, arc or polynom between others. The lateral and elevation profile of the road can also be adjusted. Having all these degrees of freedom is possible to describe the complexity of real roads. This description can then be shared by multiple simulation tools making co-simulation possible. It can also be used as exchange format between industries and other organizations as happens with OpenSCENARIO [14]. Addtionally, as OpenDRIVE defines the exact location of the lanes, a driving simulator can be built that simply follows the middle of a lane. This is an advantage with respect to other formats to store geometry, like glTF, in which the mesh information is compresed [15]. In contrast however, as mentioned in the introduction section, glTF can be used to store not only the road and traffic signs information but also any other mesh, for instance those representing other traffic participants and including material properties. This drawback can be solved if the OpenDRIVE geometry is converted to glTF in order to display it in a 3D engine. The original OpenDRIVE file can still be used to move the vehicles on their respective lanes (see figure 1.5).

Other formats that can be used to store static data are OpenCRG, which can be used to store precise road information like a dense height profile or the friction coefficient in order to improve when needed upon the geometry defined using OpenDRIVE [13]. City Geography Markup Language (CityGML) is data model using XML for serialization with focus on storing 3D city and landscapes models. One advantage of CityGML is that the data model allows for sofisticated data hierarchies which make it a powerful representation in order to do data analysis. Navigation Data Standard (NDS) is also a popular standard with focus on storing high definition maps information [13]. Data stored in NDS can be converted into OpenDRIVE [16].

Finally, PLM XML [17] is a standard needed to close a hole in the description of the scene. For example none of the previously mentioned standards stores vehicle specific information, like the position of the rotation axis of the steering wheel, the position of the rotation axis of the doors, the position of the headlights, the rotation axis of each wheel, between others. This information is important, because otherwise is not easy to animate the movement of all those parts, which maybe needed on for example vehicle interior simulations.

1.2.1.3 Interfaces

In order to make the exchange of modules easy, for example the perception, action or environment modules in figure 1.1, some standards were created to concertize the interfaces that these modules should provide. Open Simulation Interface (OSI) is one of those standards, for example it defines a data structure called LidarSensorView that contains, between others, the mounting position of the sensor with respect to a reference coordinate system on the vehicle. It also contains a number of reflections based on the resolution of the sensor. Each reflection may contain values for time of flight, Doppler shift, signal strength and others. These values are relative to the sensor's mounting position. In this way the point cloud generated by certain LIDAR sensor can be converted into a generic data structure. OSI also defines an object based message passing mechanism that can be used to send and receive standard data [13]. FMI is another standard used to exchange simulation models. In order to do that it also standarizes interfaces but additionally it defines the way in which the models should be packed (content and type). Based on the standard interfaces an API is defined in order to start and run a simulation containing multiple modules [13]. The modules are assumed to contain integro-differential equations that have to be solved using a numeric solved. The option is given to the user to use a solver per module or a common solver. System Structure and parametrization (SSP) can be seen as an extension of FMI in which not the connection of modules but of systems (containing themselves modules) is specified. AUTomotive Open System ARchitecture (AUTOSAR) is another standard that takes the standarization one step further as OSI or FMI. Here the whole architecture is standarized. The standarized modules have access to the AUTOSAR API. In contrast to FMI or OSI this API provides a broad range of services that go beyond communication including for example diagnostics, log and trace between other services [13]. Finally, the International Organization for Standardization (ISO) 23150 standarizes the interface between environment sensors, like OSI does, and the output of a fusion unit that generates an environment model (from the vehicle perspective) using the sensor data [13]. The fusion unit is a concrete implementation of the perception module in figure 1.1

1.2.2 Sensor models

Although there are different alternatives to classify different sensor models. The fidelity based classification has the advantage of having a direct relation with how the model is used. A higher degree of fidelity does not mean that a model is in general better, each type has its purpose and use case.

1.2.2.1 Low fidelity

In this model the focus is to verify if an object, usually described by its 2D or 3D bounding box, is inside the field of view of the sensor. Occlusion between objects is also considered. Aspects like material properties, weather related effects and sensor specific effects are not taken into consideration. The field of view of the sensor may be defined precisely or be approximated by a simpler shape in order to reduce computation time. This kind of model is mostly used at the beginning of the project when requirements are defined [10], or as basis for medium fidelity models. Low fidelty models are usually created using the depth-buffer (z-buffer) generated per frame by the GPU when doing its normal rendering calculations. This buffer can be accessed using for example OpenGL [18]. In this case the virtual camera is placed in the position which the LIDAR sensor would occupy in the 3D scene. The obtained image can then be sampled based on the resolution of the sensor. The result is a 3D point cloud of the scene. Alternatively the output can be provided in form of an object list containing the objects that are visible for the sensor [10]. Another performant way to generate low fidelity sensor models is using raycasting. In raycasting a ray is cast from a source (the sensor) and only the first hit of this ray with an object is considered [19, 20]. It is not taken into consideration if the ray has a return path to the receptor (i.e. photodiode) in the Lidar. One advantage of using raycasting over z-buffer is that the effect of the scanning pattern of a scanning Lidar can be simulated.

The main advantage of these algorithms is that they can provide results deterministically and in very short times.

1.2.2.2 Medium fidelity

These models are usually built on top of ideal sensor models to increase the level of realism. They constitute a compromise between realism and computational costs. These models are based on real measurements from which the noise pattern of the sensor is learned [21, 22] or from physically based models [23, 24] from which a probability for the detection performance of the sensor is extracted. Neural networks have also been used to transform the point clouds generated by ideal sensor models into more realistic point clouds using domain adaptation [25, 26, 27]. The object description is usually refined from just a box to more complex shapes [10].

These models are usefull to simulate the effect of phenomena like rain, fog, snow [22] or dirt [28] on the performance of the sensor. An accurate physical simulation would take far too long and the phenomena described dont have strong locality, this means that the effect of snowflakes or raindrops can be described as a group using few variables without taken into consideration each single trajectory. The same considerations as in statistical physics [29]. For phenomena like spray water, locality plays a more important role as the LIDAR detections depend on the trajectory of the drops. This kind of phenomena is therefore better described using particle systems (as part of the environment, i.e. separate from the sensor model) [30].

These models are also usefull to simulate the effect of sunlight or other light sources containing the same wavelength as the LIDAR, which cause an increament in the noise level [31], or to simulate electrical noise [32].

1.2.2.3 High fidelity

Physically based sensor models are models where very few simplifications are made about the way in which light propagates and interacts with objects, about the objects themselves (material properties and shape) and about the internal workings of the sensor (e.g. optical properties, electronic characteristics, mechanical structure, temperature dependency). Based on our current physical understanding of each phenomena, the results obtained with this kind of sensors are known to be correct. This reduces the validation effort, increases the generality of the model [33] and makes easier the comparison between real measurements and simulations, as the same physical units are used in both cases. Usually some form of raytracing is used in this kind of models. In contrast to raycasting, in raytracing the whole path of light from sender to receiver is considered.

Although this model is very accurate it doesn't mean that it is warrantied that the simulation results will match real measurement. Some causes for a remaining reality gap are:

- Unknowns due to intellectual property protection.
- Missing theoretical models.
- Incorrect simulation parameters [34].
- Incomplete simulation models [34].
- Numerical errors [34].
- Non-simulability of the phenomena.

Besides, one big drawback of this type of model have been the very long simulation times [33, 35]. Recently however, the possibility of doing raytracing in real time

[36, 37, 38] has made the interest on more physically accurate models grow [39, 40]. This development goes hand in hand with preasure on the entertainment industry to provide more and more realistic experiences in real time (e.g. Virtual reality (VR) and Augmented reality (AR)) [41] and is therefore expected to remain or intensify in the future [42].

1.2.3 Virtual Environment models

The virtual environment constitutes a common ground truth for all the entities in the simulation. Its purpose is to be sampled by the different sensors and to display the effect of the actions taken by the simulation participants. While the virtual environment can be classified also based on its quality, in the context of this thesis a more useful classification is based on the way it is generated. The next sections present a broad classification of the available options. Commercial tools like Vires Virtual Test Drive (VTD) [43] and IPG CarMaker [44] where not included in this list as they contain workflows for each of the three categories. The methods proposed in this thesis can be included in any of these commercial tools as long as raytracing is available.

1.2.3.1 Synthetic virtual environment

Traditionally, tools like Blender [45] or Maya [46] have been used to manually create 3D geometries and textures which were then imported into a terrain. The terrain itself was either manually or automatically generated. The techniques used for the generation of the terrain are varied, for example using pink or Perlin noise, fractals, L-Systems, agent-based or AI-based [47, 48, 49, 50]. The terrain may also include automatically generated vegetation [51] and road networks [52] using the same techniques. Similar techiques are also used to automatically generate the required textures [53]. There are also new methods to automate or simplify the generation of 3D characters [54, 55] and to animate their movements [56, 57].

Regarding light sources, a skybox is usually used to illuminate 3D virtual worlds [58]. The skybox may also simulate variations on weather as weel as a day night cycle and atmospheric scattering [59]. In general these models are not physically accurate and are not conceived to be used by other sensors except by the camera. Physically accurate light sources are however possible by importing the IES profiles of real lamps or other light emmiting objects [60].

The main advantage of synthetic environments is that they can be generated automatically or semi-automatically, this means in a short time a portion of a city or a whole city can be created for testing. Additionally, every object is automatically labeled, i.e. every triangle is related to a known category. Doing this kind of labeling for real data for a whole city would be extremelly expensive. Techniques that allow automatic labeling [61] help reducing the costs but human verification is still needed. The main disadvantage of synthetic environments lie on the validation of the simulation results. As the environment is synthetic, comparing real measurements with the obtained simulation data may be difficult.

1.2.3.2 Measured virtual environment

Measured geometries and textures can be obtained by doing "drive by scanning" using for example high resolution LIDARs [62] usually combined with cameras [63] or using stereoscopic cameras [64, 65]. Similar methods exist using helicopters or fixed-wing aircraft [62] and satelites [66]

In order to run simulations using the measured data an alternative is to generate OpenDRIVE (Standard for the representation of road information) [67] files automatically [68] which can then be employed in a tools like VTD [43]. OpenDRIVE files can also be extracted from OpenStreetMap files [69] or from real accident videos [70]. Alternativelly, OpenStreetMap (OSM) files can be used [71] or even the 3D point clouds directly [72].

Although "x by scanning" can provide fast shape measurements it doesnt provide a very precise description of the optical properties of the scanned objects. A precise measurement of the optical properties of materials can be done using gonio-spectrometers to extract their Bidirectional Reflection Distribution Function (BRDF) [73]. This technique provides very accurate results and can be done simulataneously for visible and infrared wavelengths. Recently, inverse rendering methods have improved due to the use of neural networks and are able to extract the BRDF from the objects in a scene automatically [74, 75] using photos. Additionally, data driven methods are able to automatically define material properties as a composition of known BRDFs [76]. This is an interesting alternative as there are multiple databases containing Physically Based Rendering (PBR) materials. The use of these types of materials measured in the visible range to simulate LIDARs in the Near Infrared (NIR) range has to be verified based on the accuracy required for the simulation [77].

In contrast to synthetic virtual environments, the results obtained in measured environments should be easier to compare and validate. Additionally, because high resolution maps are often used as part of the self driving vehicle functions [78]. They can readily be used for simulation.

1.2.3.3 Hybrid virtual environment

Besides a purely synthetic and a purely measured virtual environment is also possible to have a combination of both. For instance using data augmentation techniques is possible to combine real and synthetic data. This approach combines the advantages of the previous two: as synthetic data does not have to be labeled and measured data can be easily validated. For instance is possible to use a measured static environment augmented with synthetic dynamic objects [79]. It is also possible to augment real good weather data with adverse weather data or adverse weather data with synthetic objects [30]. When augmenting a measured environment with synthetic data usually some adjustments are needed to match the two underlying data distributions, i.e reducing the domain shift. This can be done by simply adding a certain amount of noise [30] or using more sophisticated approaches like Generative Adversarial Network (GAN)s trained on real and simulated data [80]. Another alternative is to deliberately generate samples outside of the real data underlying distribution and use then as negative examples. These examples can be used to train a classifier about common pitfalls and help it avoid them [81]. Besides considering domain shift also tracking is important when placing synthetic object in a real environment or vice versa. This refers to the reference frame that will be used to locate the object. It may be a dynamic reference frame, for example relative to a vehicle [30] or a static reference frame. Additionally, the object may need to interact with other already existing objects in the simulation. Examples of these interactions are mutual occlusion but can evolve to more complex behaviors, for instance a synthetic object may need to bounce or break when hitting the road [30].

1.3 Raytracing based sensor model and hybrid virtual environment

Based on the initial discussion regarding the testing requirements for self driving functions, the model presented in figure 1.2 and the previous analysis of each of the sensor and environment models, the following strategy is proposed:

- Use Raytracing as basis for the development of the sensor model.
- Use a hybrid environment as basis for the virtual environment model.

Raytracing has as advantage that it can be evolved from raycasting (low fidelity sensor model) all the way up to bidirectional raytracing (high fidelity sensor model) making the simulation as physically accurate as required during development. This is presented in figure 1.4. Additionally, raytracing can be combined with conventional or neural network based statistical methods using as input the rays intensities and directions and generating as output new intensities and directions [23].

Regarding the environment, a hybrid environment is the best basis to implement the concept from figure 1.2. As with raytracing we can evolve the environment starting from a simple synthetic one and go all the way up to a photorealistic environment with measured object models. In order to do this however, we need to define the environment in a way in which replacing 3D models or materials or adding weather related effects is simple. Figure 1.5 presents the result of putting all these ideas together under a framework that can evolve from low fidelity up to high fidelity simulations including weather related effects. Here OpenDRIVE is used to represent the road together with Graphics Language Transmission Format (glTF)), and OpenSCENARIO is used to represent the scene. Product Lifecycle Management eXtensible Markup Language (PLM XML) is used to contain geometry related metadata. The vehicle geometry itself is stored in glTF [15]. Regarding weather related phenomena, in front of the sensor two regions are defined, the brown region simulates reflections in the near region of the sensor, these kind of reflections are mostly caused by substances on the sensor cover (dirt, water, etc.) [28]. The green region simulates reflections caused by rain, snow, fog [32] or hail. Close to the vehicle a third region is defined, this region corresponds to extra detections caused by spray water [30] or other object related emissions like water vapor from the exhaust pipe.

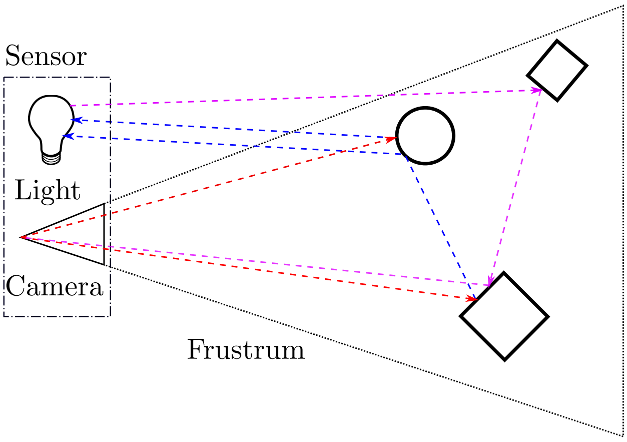


Figure 1.4: Using Raytracing as basis for the sensor model has as advantage that it can be evolved as needed during development. For example the sensor can start by doing simple Raycasting (red dashed line), later path tracing (blue dashed line), and finally if needed bidirectional path tracing (purple dashed line). The sensor is represented as a light source together with a camera (dash-dot line). Only the 3D objects in the frustrum (doted line) are detected by the sensor.

Additionally, the dotted line represents a sky box used to simulate the light generated by sun and moon (dashed lines, used to represent area lights. The moon is not relevant for the LIDAR sensor but it is important for the camera sensor). All these regions are controlled by the OpenSCENARIO description which defines date, time, weather and the relevant actions (e.g. speed). Although not shown in the picture, the noise level inside the sensor may also be adapted based on these values. Besides the geometry representation, glTF stores the material properties. In its version 2.0 glTF defines PBR materials [15]. It is important that the chosen material format supports PBR in order to cover the range from simple sensor models up to photorealistic models. Additionally, the standard interfaces defined by OSI are used to connect the environment simulation with the perception and action modules shown on figure 1.1 OSI, in that way the different modules can be replaced without much effort.

The use of the mentioned standards and regions allows for the desired flexibility. The XML descriptions may remain the same, while the geometries, materials and simulation models change or are replaced by real measurements. In this context, besides formalizing the setup from figure 1.4 and figure 1.5 under the requirements mentioned in section 1.1, the main contributions of this thesis are a new model for the spray region, techniques to extract the model parameters for each of the other regions using real measurements as well as a methodology to measure the reality gap that can be applied to all three regions.

1.4 Structure of the thesis

Based on the regions introduced in 1.5, this thesis can be grouped as follows:

1. Sensor cover: The sensor cover is a special optical element as it protects the sensor from the environment while being itself directly exposed to it. As it is the visible part of the sensor it influences the overall appearance of the vehicle and therefore its design is subjected to more constrains as just its size (e.g color, shape, weathering profile). Besides esthetical aspects, changes in the optical properties of the cover have an important influence on how the environment is perceived by the sensor. As the cover is part of the optical path of light, changes in its reflection, transmission or scattering characteristics may have a considerable impact on the sensor operation. In order to evaluate these changes and be able to include them in a simulation an experimental setup is proposed. Using the proposed setup is possible to measure all three parameters in a reproducible manner. To evaluate the proposed method the accumulation of dirt on the cover is characterized. Dirt is important because although there are cleaning mechanism they may not work under certain conditions [28]. The obtained transmission, reflection and scattering values can be used to create textures which when sampled by a raytracer can reproduce the corresponding effects in simulation. These textures can be used to create a database which can be shared as is done with other textures.

2. Atmosphere: It is well known that the presence of water droplets, snowflakes or hail on the atmosphere affects the perception of the LIDAR sensor by increasing the noise level and generating extra detections. What has not been deeply studied is the connection between those changes and the simultaneous change that these phenomena causes on the optical properties of other objects in the environment e.g change in the reflection pattern of the street when wet or when covered by snow. An accurate simulation of these phenomena has to include both effects. A measurement setup is proposed to simultaneously measure both changes for rain, fog and snow. From the measurements is possible to generate stochastic models which can be included in a raytracing based simulation to simulate the extra detections. They can also be used to adapt the material properties of objects in this case of the street.

3. Object emission: Objects themselves can also be a source of extra detections. One example that has been mostly ignored until recently is the generation of spray water. When driving at high speeds in wet roads, vehicles driving in front or next to the vehicle using a LIDAR sensor can impact its perception by producing clouds of spray. These spray clouds tend to float for enough time on the air to be confused with a real object and may cause phantom breaking. A simulation model for this kind of phenomena was missing and was developed during this thesis [30]. The model is based on particle systems which can be sampled by a raytracer generating similar spatial distributions as real spread clouds do. Additionally, a methodology is presented to extract the simulation parameters directly from real measurements as well as the corresponding measurement setup.

4. Background radiation: Although the sensors are protected by an interference filter, the presence of sun light increases the noise level reducing the average number of

detections [32]. This effect has to be considered in high fidelity simulations. A region is not assigned to it however because it can be simulated directly as part of the sensor model. The parameters required for simulating this effect can be extracted from the same measurement setup used for the atmosphere region.

5. Data augmentation: Having completed a model that is able to simulate the most relevant weather related phenomena one question remain: how can real and synthetic data be used together in order to solve concrete issues faced by self driving systems. A solution is proposed by showing a data augmentation workflow in which synthetic obstacles are added into real point clouds affected by spray. The synthetic obstacles are generated using the previously developed simulation and hence produce point clouds that are physically plausible under the corresponding weather conditions. The final augmented scenes are an example of data that would be almost impossible to collect in reality and that can be easily generated by combining real and synthetic data.

The document itself is thematically divided into the following sections:

- Chapter 2. Presents the fundamentals concepts and technologies discussed in this thesis. First, the classification of the ADAS and SDS functions is presented. This classification is important because it is the basis upon which the testing requirements, sensor set and operational domain are derived. When the automatization level grows the validation effort increases making the role of simulation increase. The LIDAR sensor is introduced, relating it to the automatization level required. The advantages of the LIDAR are contrasted with the problems generated by using the technology under adverse weather conditions. The argument is presented that deactivating the sensor under all adverse weather conditions would cause a drastic reduction of the operational domain of self driving functions. Finally, 3D engines are presented, this engines are focused on providing real time high quality 3D content and therefore constitute the technological basis used for the proposed simulation models.

- Chapter 3. Discusses the different options used for validating ADAS and SDS functions. The traditional ways to validate ADAS functions: proving ground and test drives are presented and it is discussed why these methods are not sufficient to validate highly autonomous systems. The new methods: shadow mode and simulation are described. Both have advantages and disadvantages and as proposed in figure 1.2 complement each other very well. The use of dedicated lanes for self driving vehicles is also discussed. As simulation is the focus of this thesis the topic is discussed more thoroughly including the current standards (some of which were mentioned in figure 1.5), the current trend towards co-simulation and the available alternatives to combine software and hardware as well as the use of cloud technologies. The main problem of simulation the "reality gap" is discussed at the end mentioning the methods used to quantify and if possible reduce it.

- Chapter 4. Based on the discussion of the previous two chapters, this chapter presents the concrete methods proposed to reduce the reality gap when simulating the effect of weather on the perception of an automotive LIDAR sensor using real time technologies. The proposed methods are divided into four parts based on the different components of a virtual environment and the corresponding weather effects. First, the effect of weather on materials and textures including changes on the optical properties of the sensor cover. Second, the mesh-based simulation of rain, fog and snow. Third, the particle-based simulation of spray water (extensible also to phenomena like exhaust gases). Fourth, the simulation of changes in background radiation. Finally, a concrete example is given in which the use of the proposed simulations allows to tackle a problem which would be almost impossible to solve using real data. Each of the sections in this chapter presents a brief description of the problem and solution. The full discussion can be found on the corresponding papers on the Appendix.

- Chapter 5. In this final section a summary of the results is presented relating them with the objectives formulated in the section 1.1. Additionally, suggestions are given to further improve and expand upon this work.

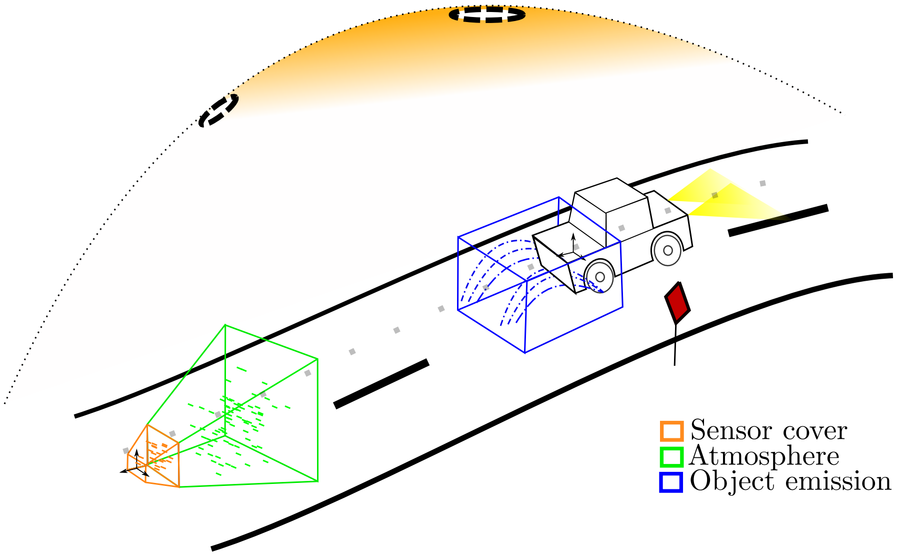


Figure 1.5: Flexibility in the conformation of the virtual environment is reached throw the use of standards. OpenSCENARIO defines the position of any active participant in the simulation over time as well as the date, time and weather. OpenDRIVE defines the road geometry, the lanes and the position and type of traffic signs. The geometry information may be converted to glTF for efficient storage. The OpenDRIVE file may still be used for example to extract the middle of the lane (dotted coarse line). OSI defines data structures for the exchange of sensor data for example the mounting position of the sensor and parameters of each single LIDAR detection. PLM XML is an example of a file containing metadata and geometry information about the vehicle. The metadata is necessary in order to animate the different parts of the vehicle, e.g. turning headlights on or off. In general textures and geometry information are stored in glTF. The fine dotted line represents the skybox containing two light sources (sun and moon, dashed lines). The skybox is controlled using the data contained in the OpenSCENARIO file. To simulate weather effects on the sensor perception three regions are proposed, one focused on reflections caused by the sensor cover (brown), one focused on reflections caused by the different types of precipitation (green) and a third region focused on spray related reflections (or other types of emissions). The reason for using different regions is to allow for different simulation methods e.g. mesh based in one vs particle based in other and to facilitate their independent activation/deactivation.

2 Fundamentals

2.1 Advanced Driver Assistance Systems and Self Driving Systems

The international standard SAE J3016 defines different levels of automation for on-road motor vehicles [82]. Six different levels are defined going from 0 to 5. From levels 0 to 2 the driver performs continously part of the driving task including monitoring the environment [83].

SAE Level	Name	Description
0	No automation	Driver performs all tasks
1	Driver assistance	Driver is assisted with steering or braking/acceleration
2	Partial automation	Driver is assisted with steering and braking/acceleration
3	Conditional automation	The system performs all tasks inside a certain ODD, the driver has to take over if requested
4	High automation	The system performs all driving tasks inside a certain ODD, the driver may not take over if requested
5	Full automation	The system performs all driving tasks

Table 2.1: Levels of automation as defined for on-road motor vehicles.

Starting with level 3 up to 5 the automated driving system performs the driving task and monitors the environment by itself [84, 85]. There is however a fallback mechanism that differentiates levels 3 and 4. In level 3, in case of failure or hazardous weather the driver is responsable of taking over the driving task. In level 4 the automated driving system is responsible of fullfilling the fallback task by itself [86]. In all levels up to 5 there is a Operational Design Domain (ODD) which defines certain conditions that are required for the driving function to be available. In level 5 the system is able to drive automatically under all conditions. For levels lower than 5 the vehicle manufacturer is free to choose for which ODD which automation level applies. One ODD that is considered appropriate for level 3 systems is traffic jams and stop-and-go traffic on highways at low speeds. This kind of traffic is quite tiring and monotonous for a human driver increasing the risk of accidents. Simultaneously due to the low speeds, physical separation of the highway and the possibility of extending already existing Adaptive Cruise Control (ACC) systems is considered technologically approachable. Another ODD in which some Original Equipment Manufacturer (OEM)s are working on is level 3 and level 4 highway driving up to the maximum allowed speed [84]. Additionally, some companies are working on level 4 systems known as robotaxis which are able to transport passengers along certain routes in cities at low speeds. In general most ODDs on which companies are currently working on are not isolated from weather and therefore have to cope in some way or another with it. The next section will consider the fallback mechanism under adverse weather conditions.

2.1.1 Fallback mechanism and weather

In order to ask the driver to take over, a level 3 system will have to identify that the weather has changed, or is expected to change, enough for it to be outside its ODD. After the take over request the driver may need up to 30s to take over depending on the mental load of his previous task and the complexity of the driving situation [87]. Additionally, due to its finite time predictability barrier [88] weather can not be predicted with enough precision to precalculate the Take over request (TOR). Alternatives like using Vehicle to Everything (V2X) are also affected by weather [89], besides, although the first vehicles that encounter adverse weather may be able to inform other vehicles, those first vehicles will have to use their own sensors in order to measure the current weather and initiate their TORs. This means that a vehicle may have to cope with adverse weather for around 30s. After the adverse weather has passed the vehicle will have to evaluate again if a re-engagement is possible when requested by the driver. A similar analysis applies to level 4 systems with the difference that level 4 systems may have to cope with adverse weather for even longer intervals.

Reasons why weather has to be considered in level 3 systems:

- Evaluate current weather in order to start a TOR.
- Cope with adverse weather for as long as the driver needs to take over.
- Evaluate current weather when the driver request re-engagement.

Reasons why weather has to be considered in level 4 systems:

- Evaluate current weather in order to start a maneuver to reduce the risk.
- Cope with adverse weather until ODD is breached or driving task is completed.

2.2 The LIDAR sensor

The following table shows the wavelength and the broad influence that weather has on each of the main sensors used in autonomous vehicles [90] [91] [92]. The table shows that the smaller the wavelength used by the sensor the higher the effect weather has on its environment perception. At the same time it is known from the physics of imaging systems (Rayleigh criterion [93]) that the smaller the wavelength the higher the details that can be extracted from the reflected light. The short wavelength used in RGB cameras and LIDAR sensors is specially important for localization and scene interpretation.

For example only the RGB camera is able to read traffic signs while the LIDAR sensor is ideal for doing real time matching between the measured 3D pointclouds and a high definition map of the area [94]. The improved localization provided by the LIDAR operates in conjunction with the one provided by the GPS system and is a fallback when the GPS signal is not available. The advantages provided by the LIDAR relative to its cost make it relatively uncommon in level 2 systems [95] but usually present in level 2+ (hands free driving), level 3 [96] and level 4 systems [97].

Sensor	Wavelength	Influence of weather	Main advantage
LIDAR	850-950nm and 1550nm	High	High resolution 3D information
RGB Camera	400-750nm	High	Same wavelength as human eye
Thermal camera	$2 - 10\mu$m	Medium	Object detection at low light
Radar	1.2cm, 4mm 2.5mm	low	Direct measurement of relative speed
Ultrasonic	7.5 - 14km	low	Low cost, short range distance measurement
GPS	19cm	low (local), medium (atmospheric)	Low cost localization

Table 2.2: Influence of weather on each of the main sensors used in ADAS and SD.

While the radar and the camera have been used for many years in level 1 vehicles, the LIDAR has been introduced only relatively recently as part of the sensor set in vehicles of level 2+ automation and higher [98]. Due to its novelty and short wavelength the LIDAR is a perfect candidate for investigating the effect of weather on a sensor's perception. The resulting simulation framework is nevertheless general enough to be applied with small adjustments to any of the sensors listed in table 2.2. This is due the high flexibility provided by raytracing, which can be used to simulate the propagation of electromagnetic as well as mechanical waves [99] [100].

2.2.1 Working principle

LIDAR is a sensor technology which compared with the others sensors used in vehicles: RADAR, camera, ultrasound and GPS is relatively new. For this reason there are still multiple alternatives currently contending for the market. This thesis will however focus on the technology used during the measurements which is a Time of flight (TOF) LIDAR using a 905nm wavelength LASER, a rotating mirror and an Avalanche Photo Diode (APD) as receiver [101]. There is a similar variant mostly used in robotaxis [102] in which the whole optics are rotated. The two options are shown in the following figures for comparison. The used wavelength, receiver technology and LIDAR type correspond to the currently most commonly used configuration in autonomous vehicles [103].

Instead of rotating the whole optical unit, the used LIDAR rotates only the mirror reducing the horizontal field of view. The vertical field of view is also reduced by confining the possible tilt values of the mirror to only two: using a slightly different tilt for each side of the mirror. These simplifications allow to reduce energy consumption, costs and size.

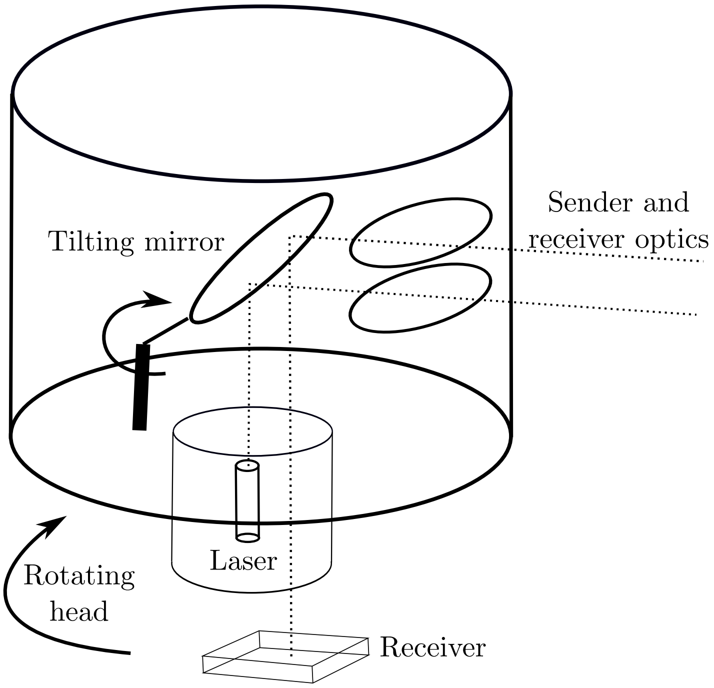


Figure 2.1: Simplified schema of a mechanical LIDAR in which the whole head containing the optics rotates providing a 360° horizontal field of view. The vertical field of view is obtained by mechanically tilting the mirror [104]. This kind of set up provides a low frame rates due to the weight of the rotating head in the range of 5 to 15 Hz [105]. Notice how sender and receiver path share the same mirror but are separated from each other in a biaxial configuration.

The following table presents the most important characteristics of the used sensor. Notice that its size, package and field of view constrain the possible mounting positions in the vehicle. The mounting position as well as the sender and receiver technologies have a significant impact on the way in which adverse weather affects the sensor. Additionally the used sensor has the characteristic of providing multiple echos per pixel, considering a pixel the minimal unit of resolution of the sensor. Up to three echos can be detected this means that as long as an object does not block the whole pixel other objects in the sensor range can be detected. This is specially important for non-solid objects like small droplets in which case depending on the concentration other objects can still be detected.

Now we precede to discuss each technology aspect in detail.

2.2.1.1 Wavelength

The wavelength is the distance at which the shape of a wave repeats [93]. The wavelength is equivalent to the period of the wave but measured in space instead of in time. It is more convenient to use as frequency in order to characterize light sources and receivers at very high frequencies as the ones used in LIDAR systems. Due to the presence of molecules on

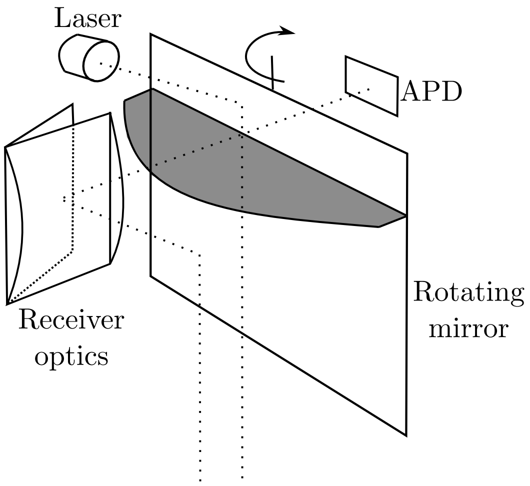


Figure 2.2: Simplified schema of the Valeo Scala LIDAR sensor used in the measurements presented in this thesis [98]. In this mechanical set up only the mirror rotates while the sender and receiver optics remain fixed. Here also a biaxial configuration is used for the sender and receiver optical path. The two sides of the rotating unit contain mirrors with a sightly different tilt, which allows to increase the vertical field of view. The mirror rotates with a frequency of 25 Hz. Due to the mirror tilting not all layers are sampled in each rotation. From the 4 layers of the sensor only the middle ones are updated with this frequency. The lowest and highest layers are updated with 12.5 Hz [101].

the atmosphere there are certain windows in which the transmitted light is more or less absorved. For the application needed in automotive it is desired to avoid absorption on the atmosphere. Therefore the regions from 0.85µm to 0.95µm NIR and around 1.55µm Short-Wave Infrared (SWIR) are usually used [106]. Additionally, it is important that the used laser light doesn't cause harm to other traffic participants. The IEC-60825-1 defines this concretelly as class 1 or 1M LASER systems [107]. This regulation limits the maximum power level depending on the wavelength, which limits the maximum range. The price of LASERs and detectors for the respective wavelengths are also an important factor, for instance considering the use of the same or similar lasers or receivers in other areas like telecommunications. Another factor in the selection of the wavelength is the effect of sunlight on the sensor. Sunlight has a broad spectrum including the wavelengths used in LIDAR sensors, in the infrared region however the intensity is much lower than in the visible region and reduces further with longer wavelengths. This makes the 1.55µm more convenient in order to reduce the effect of sunlight. Finally, the sizes of the available senders and receivers are important as they influence the final size of the sensor which is a very important factor for vehicle manufacturers. With all these factors taken into consideration the industry is currently mainly using the 905 nm wavelength [106]. Considering the 905 nm wavelength the different particles in the atmosphere scatter light in different ways which are summarized in table 2.4. Except for snow and hail it is assumed that the shape of the particles is approximately spherical. Symmetric

Parameter	Value
Wavelength	905 nm
Horizontal field-of-view	145°
Vertical field-of-view	3.2°
Layers	4 (0.8° each)
Data refresh time	40/80 ms
Distance resolution	$< 0.1m$
Typical range	150 m
Typical mounting position	40 to 80 cm height at vehicle front

Table 2.3: Relevant parameters of the used LIDAR sensor.

indicates that the forward and the backwards scattering are similar. In the case of snow and hail the spherical approximation is not valid anymore and therefore the scattering profile depends on the exact shape.

Type	Radius (µm)	Scattering profile	Concentration
Air molecules	0.0001	Symmetric	$< 3x10^{25}m^{-3}$
Haze, smoke, dust	0.01-1	Symmetric	$10^5 - 5x10^{10}m^{-3}$
Fog	1-20	Slightly forward	$10^6 - 5x10^9m^{-3}$
Rain	100-10000	Mostly forward	$10 - 10^3m^{-3}$
Snow	1000-5000	Depending on shape	$10^{-2} - 10^3m^{-3}$
Hail	5000-50000	Depending on shape	$10^{-2} - 1m^{-3}$

Table 2.4: Usual ranges for the size and concentration of droplets/flakes based on weather type.

2.2.1.2 Measurement type

Time of flight scanning LIDAR: The most simple approach to measure the distance of an object to the sensor is to send a pulse of light which is then reflected by an object and sent back to the sensor. The receiver optics then identify the pulse intensity over the noise level as shown in figure 2.3. Knowing the time in which the pulse was sent and recording the time in which the pulse returns gives the traveled distance after conversion based on the speed of light. Besides its simplicity, the scanning pulse approach has as advantage a higher tolerance to noise as the other methods because the LASER energy is concentrated in time and space. One disadvantage however is the peak-to-peak power variation of the LASER which maybe around 10% and that is temperature dependent [108]. Additionally, the distance resolution is limited by the temporal resolution of the receiver optics and generates resolutions that are usually in the order of centimeters. This technology is nevertheless the most commonly used approach for automotive applications [103] [102].

2.2.1.3 Sender technology

Mechanical: This type of technology uses rotating or oscilating mirrors or prisms to produce the scanning pattern in one of the two dimensions. To increase the field of view in the other dimension a tilting mirror can be used, multiple lasers or receiver arrays.

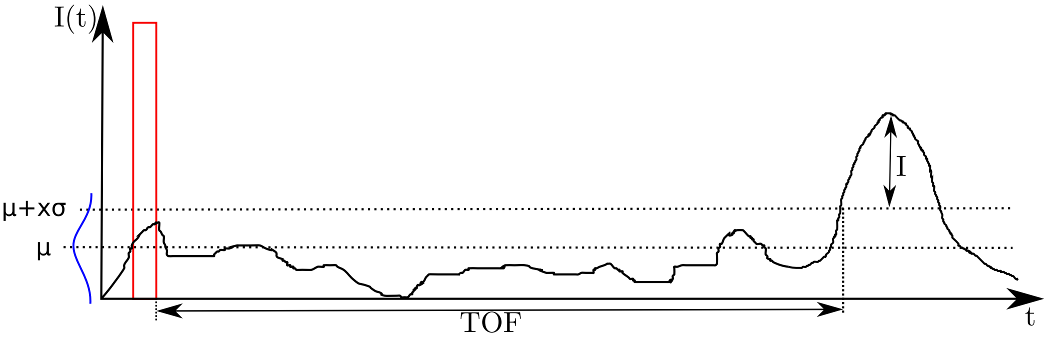


Figure 2.3: Working principle of the time of flight LIDAR. Here idealized the sender pulse in red shown simultaneously with the signal in the receiver. After sending the pulse the time is measured until a pulse is received with an intensity higher than a certain factor 'x' of the standard deviation of the noise. When the pulse arrives the time measurement is stooped and is translated to a distance

In order to reduce costs however, this method usually provides a large field of view in one direction and a small one in the other. Due to the inertia of the moving parts: mirrors, prisms or whole optical units (fig 2.1) this kind of system usually consumes higher amounts of energy, is bigger and heavier as systems with micromechanical or non-moving parts [102].Additionally due to its bulky nature these LIDARs are suceptible to miss-calibration under vibration or shock [103]. Despite the disadvantages this scanning technology is the most common in current LIDARs used in autonomous vehicles [109] as it still provides a higher range as the alternatives [102].

2.2.1.4 Receiver technology

In order to process the incoming light the receiver contains a photodiode (or a photodiode array) that make use of the photoelectric effect in order to convert photons into electrons. The photodiode is usually a combination of a 'p' and 'n' doped layers with an 'i' undoped layers. The used sensor employs an avalanche photodiode array of 3 layers on the detector side. To better understand the APD is worth to introduce first the p-i-n photodiode (also PIN photodiode) [110].

PIN Photodiode: This diode is a photodiode with an 'i' region between a 'n' and a 'p' regions, in which the applied reverse bias voltage is not very high. This has as consequence that the diode when hit by photons produces a current with a gain of 1. This limits its use to applications in which the amount of light hitting the photodiode is high. One advantage of these diodes is that they can operate up to a very high bandwidth [103]. This is due to their low noise level caused by their operation in the linear region of the V-I curve [110].

Avalanche Photodiode: In this case the applied reverse bias voltage is higher than for the PIN diodes. The gain becomes proportional to the applied inverse voltage linearly up to the break-down voltage of the diode, at which point it becomes non-linearly dependent. When working in the non-linear region the APD is said to work in Geiger-mode. The name avalanche comes from the effect that a photon has when hitting the 'i' region. This causes an avalanche of electrons and therefore a higher current as for the PIN diode. Although the avalanche effect makes the APD very sensitive it also produces more noise as in a PIN photodiode due to the gain fluctuations produced during the avalanche. For this reason is usually prefered to use a gain lower as the maximum possible [103]. APDs can provide gains of up to 1000 [111]. For the 905nm wavelength silicon is used to create the junctions [106]. When APDs are built into an array there is a second source of noise caused by the crosstalk between each cell [112].

2.2.1.5 Optical System

As shown in figure 2.2 the used LIDAR uses two different parallel axis for the sender and receiver path. This type of configuration is called biaxial and has consequences related with the influences on weather on the sensor. In this section the biaxial configuration is contrasted with the coaxial one showing the characteristics of each.

Coaxial: In this case sender and receiver use the same axis. Internally a beam splitter can be used to redirect part of the light into the receiver optics for example while the sender path is left unchanged (except for the losses caused by the beam splitter) [113]. In contrast to the biaxial setup in this setup the field of view of the receiver can be reduced to match the size of the sender field of view and the same aperture can be used for both. This reduces the amount of background noise. Additionally, objects can be detected that are very close to the sensor which is not the case for the biaxial setup in which the first objects that can be detected are those at the distance at which the sender and receiver field of view first overlap. Another reason for using coaxial systems is that the allignment of the sender and recevier path in a biaxial system can be very complicated [114].

Biaxial: The sender and receiver axis in this configuration are usually parallel to each other and are separated by a certain gap. As a consequence the sender and receiver path only start to overlap after a certain distance. This may help to reduce the noise if detections very close to the sensor are not important. This is usually the case in an automotive LIDAR system, for instance when rain drops or snow flakes accumulate on the sensor cover is useful to avoid overlapping of the sender and receiver field of view for a distance of at least 50cm (considering multiple reflection) [32]. The fact that the receiver field of view has to be designed to cover the sender field of view increasing background noise in this setup may be compensated by the losses that would otherwise originate by including the beam splitter.

2.3 Phenomena affecting the perception of LIDAR sensors

As mentioned in the previous section the power of the laser beam is limited by the laser class. This limit on the power of the laser has as consequence that the only parameter that can be adjusted to reach a certain range is the receiver. A very sensible receiver is therefore employed. A sensible receiver paired with the short wavelength of the NIR light causes a high sensibility to weather phenomena. The effects caused by these phenomena on the perception of the LIDAR are discussed in the next sections.

2.3.1 Rain, fog, snow

As mentioned in table 2.4 when the laser light hits the rain, fog or snow particles (drops or ice crystals) the photons may be scattered, transmitted or absorved depending on the wavelength of light and the particle size and refractive index. Photons scattered in directions different to the receiver of the LIDAR or absorved reduce the intensity of the beam and hence the range. However, photons that are scattered in the direction of the receiver either directly or throw multiple reflections become a source of noise and false detections [115, 32]. The spurious detections caused by each of these phenomena tend to concentrate radially up to a maximum distance from the sensor after which the intensity of the reflections is not high enough to generate a detection [32] [116]. These spurious detections tend to have the following characteristics when compared to the detections caused by solid objects[117] [32]:

- Higher density.
- Higher level of randomness in their spatial distribution.
- Lower intensity.

These characteristics of the precipitation related detections allow the construction of outliers removal filters with accuracies higher than 90% [116], or of weather classifiers [32].

Besides the spurious detections, all these three precipitations may cause changes in the reflection of the objects in the environment. Not only the reflection value but the directions (BRDF) may change, making objects with a diffuse reflection pattern become more specular in their radiation. The opposite is also possible if we consider for example a metalic object that is covered by snow [32]. The extra water or snow left after the precipitation is also relevant, water on the road can generate spray [30] and snow can mix with dirt and salt in winter soiling the sensor [28].

Summarizing these different precipitations affect the performance of the LIDAR in three main ways [32]:

- Increase the noise level reducing the total number of detections.
- Increase the number of spurious detections.
- Change the optical properties of objects in the environment.

2.3.2 Sunlight and other light sources

The presence of sunlight or in general the presence of any light source with a wavelength similar or equal to that of the LIDAR sensor and directed towards it is a source of noise and perhaps also of false detections. As a reference in a sunny day the sunlight intensity may vary between 10 to 130 klx [118]. As mentioned in the introduction section LIDARs usually have an interference filter covering the receiver that rejects light of wavelengths different to those of the sender [31, 28]. However this rejection is not perfect and the sun produces light in a very broad spectrum including NIR and SWIR. To counteract this effect the LIDAR may automatically detect the higher noise level and accordingly increase the minimum detection intensity. For this reason during the night the average number of detections in the same location is higher [32]. This mechanism is specially useful against the average increment in background radiation. Direct or indirect sunlight, for example by reflecting on a metallic object may still cause spurious detections [119]. Regarding other light sources it is necessary to consider that in the future many cars may use a LIDAR sensor and that the probability is high that the wavelength of those sensors is the same making interference very common. A technique to reduce this kind of interference is to introduce random delays on the time in which the light pulse is emitted. By informing the receiver about these delays and assuming that the scene remains the same between two consecutive scans, is possible to reject returning pulses that do not correlate with the expected temporal patterns [120].

2.3.3 Spraywater

Due to rain and depending on how good is the drainage, a thin water film can accumulate on the street surface. This water enters into the treads of the vehicle wheels and depending on the speed of the wheels is sling in a trajectory that depends on the size of the spray droplets. The average drop size and standard deviation are themselves dependent on the rotation speed of the wheel as shown in figure 2.4. In addition to spray, also splash water is generated. Splash generate usually bigger droplets as spray that fall close to the wheels and are therefore usually not problematic for the LIDAR. Additionally, a portion of the water remains due to capillary adhesion in the tread all the way until the wheel has done a half rotation. This water is then blown away by the wind creating extra spray. The smaller the particles are, the more important are the wind forces so the final spray cloud is a combination of bigger particles following a parabolic trajectory with smaller particles following the wind trajectories and particles in between following a mix trajectory. Very small droplets can also remain on the wind for long times creating an effect similar to light rain or fog [32, 30]. The use of spray suppression devices common in trucks is not common in other vehicles and doesn't completely remove spray.

The following points summarize the effect of spray on the LIDAR sensor:

- Some small droplets remain floating on the air on a random distribution and cause an effect similar to fog or fine rain. They are perceived as spurious detections with a high density close to the sensor and with a fast radial reduction desapearing almost completelly after a certain distance (5m, for the use sensor [30]).

- Depending on the wind speed some droplets may follow a turbulent trajectory and cause groups of spurious detections at random locations if a high concentration of particles is formed not very far from the LIDAR sensor (20m, for the used sensor [30]).

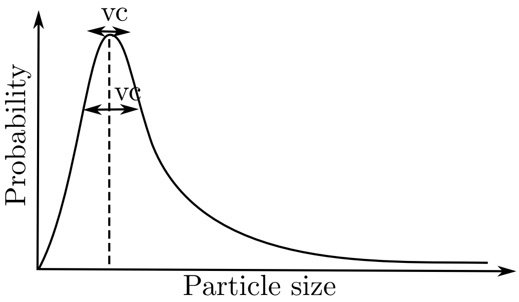


Figure 2.4: Probability distribution of the particle size depending on the rotation speed of the wheel v_c. Higher speeds generate particles of smaller diameters in average and at the same time reduce the possible size range

- At lower wind speeds some droplets follow an inertial (parabolic) trajectory and cause groups of spurios detections close to the wheels of the vehicle generating the spray.
- As happens with rain and fog droplets may be deposited on the LIDAR cover.

2.3.4 Substances and deformations on the sensor cover

The LIDAR as well as the camera are very sensible to substances or objects covering their field of view. The RADAR as well, but mostly due to the precense of wetness which can be reduced using appropriate radome shapes and hidrophobic films [121]. As already mentioned, the LIDAR laser power is limited and therefore any object placed infront of it may considerably reduce its range. As a consequence placing it behind the windshild to protect it from dirt as is done with some cameras is usually not an option [122]. Placing the LIDAR close to the vehicles' roof may also be inconvenient either esteticaly or due to a limited vertical field of view of the sensor. There are also mechanisms to clean up the sensor [123], or to free it from snow [124], these mechanism are however not perfect and some depend on water or cleaning liquid which may be exhausted. The effect of hydrophobic films sinks with time and doesn't protect from all forms of dirt. It is therefore necessary to consider the effect of substances on the sensor cover, some examples are:

- Snow: either accumulated when parking or sling perhaps in combination with salt or dirt by other vehicles.
- Insects.
- Water: caused by rain or spray coming from other vehicles, combined perhaps with dirt or salt.
- Dust: accumulated when parking or sling by other vehicles.

As happens to objects under the effect of rain, fog or snow these substances change their optical properties over time so for instance the dust initially combined with water may dry out leaving small dirt patches on the sensor. As rain, fog and snow these substance may cause a reduction of the range, false detections and/or higher noise [28].

In addition to substances, the optical properties of the cover may also change due to rock impacts, temperature, the use of abrasives, or the effect of UV light [32].

2.4 Creating low cost real time simulations

2.4.1 3D Engines

In order to create a game or in general a 2D or 3D virtual interaction software, some parts are needed over and over again. For example collision detection, artificial intelligence, a renderer and more recently certain networking capabilities. Those parts were usually developed separately by each game development studio. Recently however, the high costs and the short deadlines present in game development make the use of games engines more and more common. Figure 2.5 shows the idea of a game engine. A game engine or, due to their use outside of games, a 3D engine is a layer of code that implements much of the functionality needed to develop a game. The functionality is provided to the user in form of modules that abstract away a big part of the inherent complexity. In addition to reducing the amount of work over multiple games, the engine reduces also the amount of work required to port a game to different platforms.

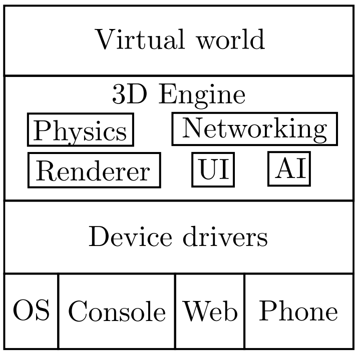


Figure 2.5: In order to avoid developing each time the same functionality. A 3D engine delivers modules covering the requirements of most games or in general 3D virtual interactive software. The low level programming required to comunicate with the device drivers is done in the engine once and can be used for multiple projects in different industries. Porting the code to other platforms is also simplified by using the engine (Based on [125]).

Besides game studios other industries in need of 3D content have discovered game engines. For instance the visual effects industry for film and television that used to have specialized software is recently moving towards 3D engines [126]. The same is happening with architecture and design companies [127] as well as the automotive industry [128]. These industries and others interested in VR, AR, the Metaverse and Digital Twins provide high amounts of resources towards the development of 3D engines, making it difficult for specialized software to compete, except in certain niches. The scientific use of game engines have been limited by the gray or black box aproach of certain companies developing the engines. Recently however, open source game engines are available like Godot [129] and Open 3D engine (O3DE) [130]. The researcher can hence use, verify and modify the code and in that way obtain reliable results that other researchers or institutions can reproduce. An alternative to 3D engines is to use specialized libraries for each of the modules mentioned in figure 2.5. This is a good option if only for example the renderer and the physics engine are needed and not the rest. Tools like Blender [45] are also a good option having a production quality open source path tracer [131] and a physics engine.

2.4.2 Rendering, materials, and light sources

As mentioned in the introduction section the use of Raytracing is recommended for the simulation of the LIDAR sensor. Raytracing has the advantage that it can be very fast if just the first intersections of the rays are calculated (Raycasting) or it can be very precise but also slow (although modern graphics card make this less of a problem) if required. Raytracing in general tries to solve an equation called the light transport equation. This is an integral equation that defines the equilibrium radiance distribution in a scene. This equation can not be solved analitically for most cases and has to be solved numericaly. Conventional numerical integration methods like quadrature require a computation time proportional to the number of dimensions of the integral, for that reasons other methods are prefered like Monte Carlo integration whose computation time is dimension independent. In broad terms the method consists on sending rays randomly from the camera pixels and calculating their trajectories until eventually they hit a light source, having the intensity of the light source and the losses caused by each reflection and free space propagation the intentisity reaching the pixel can be calculated. The disadvantage of the Monte Carlo techniques is that they requires a certain amount of rays per pixel to smooth out otherwise the variance is high which is perceived as noise in the image. Tracing rays takes time for that reason a number of techniques have been developed in order to obtain an image with low variance while tracing a few rays as possible [132]. The Cycles renderer used in this thesis is an open source path tracer that uses some of these techniques [131].

Regarding materials and textures the use of some kind of principled BRDF [133] is common. The advantage of this kind of material definition is that the material can be improved as needed. For instance if only the base color is known or needed the other parameters can be ignored. If more information is available like roughness, specular and metalic they can be added succesively increasing the level of realism. The PBR material definition in glTF 2.0 is also compatible [134]. This is important because glTF will probably become the default standard for storing 3D assets.

As with the other parts of the simulation the type of light source depends on the type of data needed for testing. For example point light sources are very fast to sample but using them to simulate the laser light would ignore the scanning pattern. For a non flash LIDAR it may or may not be an issue depending on how far the objects are, how fast the vehicle is moving, etc. If the scan pattern is important a better technique would be to use raycasting from the light source with multiple rays per pixel [135], this technique however ignores multiple reflections and is therefore good for objects that are perpendicular to the sensor.

2.4.3 Physics engine

Similarly to what happened with the creation of game engines. Physics simulations in games were initially programed over and over again as needed for each game. As the desire for higher levels of realism increased and the deadlines became shorter the necesity appeared of using a general and reusable approach to the simulation of physics in games. In this way the physics engine was born. Broadly speaking, the role of the physics engine is to take the current positions and velocities of objects and apply the equations of motion to update these values considering the effect of all the forces currently acting on each object. Additionally, the physics engine need to identify collisions and add the effect of the forces generated by those collisions [136]. The equations of motion are integro-differential equations that have to be solved numericaly usually during the interval provided by the refresh rate of the screen. Due to this real-time requirement some simplifications and approximations may be made which may produce results that are not physically correct. This is however not a reason for discarding the use of physics engines for software testing. One option to overcome this issue is to validate the results for the needed range using a realiable solver. Another alternative, if the code of the engine is open source, is to directly verify the code. Some examples of open source physics engines are: MuJoCo [137], Bullet [138] and ODE [139]. The main advantage of using this kind of engines is their real time capability and low cost (or no cost) compared to conventional tools.

2.4.3.1 Mesh-based vs Particles-based methods for fluid simulation

Fluid simulations are an important part of the physics engine. Although it is possible to simulate aerodynamic forces or bouyancy without having to solve the Navier-Stokes equations [136], many games include water, fire, smoke and other fluids in some way in their gameplay, for which solving the fluid equations is necesary. There are two approaches to solve the fluid equations: Eulerian and Lagrangian. The first one uses a mesh (2D or 3D), the second one uses particles. It is also possible to combine both, in this case particles are used to store variables that are not fundamental for the fluid simulation like the position of boubles or foam [140]. In the mesh approach the variables that characterize the fluid are stored in the mesh and are update by numericaly solving the Navier-Stokes equations [140]. The fluid is then rendered using volumetric rendering techniques that sample the content of the mesh. In the particles approach the information is stored in the particles [141], in the same way as previously discussed for the physics engine. The position of the particles are updated each frame and it is the particles themselves that are rendered. In this case normal surface rendering techniques can be used. Fluid simulation if done in a big mesh or using a high number of particles are not warranted to finish before the next frame and are therefore calculated offline and stored. In general particle based fluid simulations are faster than their grid counterparts and if the number of particles is limited can run in real time [141]. An open source solver like mantaflow [142] contains solvers for both methods and is integrated in Blender.

2.5 Conclusions

- The LIDAR as sensor for autonomous driving is still in a development phase. There are multiple technologies competing in the market with different advantages and disadvantages. In general a trend towards the use of solid state technologies is desired for robustness, size and cost reduction.
- The maximum power that the LIDAR sensor can emit is limited by safety restrictions. There is certain leeway regarding the wavelength but this is also limited by commercial and technological reasons. These two restriction make difficult to develop a sensor that is not affected by adverse weather as much as for example the RADAR sensor. This disadvantage is compensated by the high resolution, depth measurement and ability to work under low or no light.
- Simulation models of the effect of weather on the sensor are needed to monitor the ODD and correctly assets the risks. This is relevant for level 3 and level 4 systems as in level 3 systems the vehicle needs to decide when to start the fallback mechanism.
- Creating the kind of simulations needed to validate the performance of self driving system has become easier due to the existence of open source game engines and related tools which strongly reduce the amount of code needed to create 3D interactive software.

3 Testing of ADAS/SDS functions

Depending on the level of automation there are different alternatives to test autonomous systems . Classically a combination of proving ground tests, laboratory experiments and test drives in public roads was enough to verify that a system was working as expected. Starting with advanced level 2 systems new ways were introduced to complement the initial methods. Some of these methods are shadow mode and simulation. Dedicated lanes although normally not considered a testing method can be seen as a public road test where human drivers as factor are removed [143] and are therefore included in the discussion.

3.1 Proving ground

The core idea behind the proving ground is to test scenarios that are safety critical but which would be unsafe or unethical to tests on public roads. These scenarios are usually part of homologation or quality tests required for the SuT to be approved [144]. Besides these core scenarios, the proving ground may also contain scenarios that are representative of the situations that the SuT may find on public roads compresed in a small scale [145]. New proving grounds may also include areas specifically for autonmous vehicles for example a small portion of a smart city in which the SDS is by itself or interact with other SDSs, human drivers or vulnerable road users [146]. This new kind of proving ground starts to be very similar to geographycaly limited public road tests. One difference however is that the humans taking part on the experiments are employed to do so. In addition to real humans some road users or obstacles may exist only virtually making use of simulations [44].

3.2 Laboratory experiments

Laboratory tests are usefull for component and integration tests [146]. For example is possible to check that the communication between the different parts of the system work using Hardware In the Loop (HIL) tests. For each sensor, besides durability tests, the sensor's perception can be verified using target simulators [147, 148] in Vehicle In the Loop (VIL) setup to test the sensor front end. The effect of weather can be analized with a static outdoors test setup [32] or using for example fog chambers [22]. Using driving simulators or virtual reality is possible to check how humans interact with the system (Driver In the Loop (DIL)). Software In the Loop (SIL) tests can be considered also in this category although both HIL and SIL tests make an increasingly use of simulation.

3.3 Public roads

Tests in proving ground and laboratory are done in an area to which the public has no access. The following methodologies remove this restriction.

3.3.1 Shadow mode and automatic data collection

In shadow mode the next version of the autonomous driving software runs its perception and planing algorithms but does not actively controls the vehicle [149]. The current software version drives the vehicle while a human driver supervices the operation. If a situation happens in which the driver has to intervene, also known as a disengagement, the action of the human driver is compared with the action that the software in shadow mode would have taken. If the two differ the data is stored in order to train and improve the software. Besides triggering data collection based on disengagements other triggers can be used, for example novelty or anomaly triggers can be used to automatically identify content in the data that significantly deviates from the previously observed data. One way to do this is using autoencoders which are able to automatically find a compressed data representation. The data can then be decompresed and by comparing it with the original data a quality metric can be calculated. If the value of this metric decreases the existance of an anomaly can be infered [150]. Another possibility is to trigger data collection based on the presence of a certain label in the data. Simulation plays a role in shadow mode as it has to be understood what would have happened if the human driver would not have taken action. This question is solved using counter factual, also known as what-if simulations. Knowing what would have happened helps to confirm if the driver acted correctly and if that is the case a new scenario can be created to test and improve the SDS [151].

3.3.2 Test drives with driver in vehicle or remote

Test drives with a trained driver sitting on the driver seat and paying atention to the behavior of the SDS are a traditional way to verify that the ADAS or SDS systems are working as expected. A new variation of this approach is the use of remote drivers. In this case when the fallback procedure starts the control is not taken by a driver in the vehicle but a remote one. The remote driver handles the adverse situation and when possible returns control to the SDS. In this case it is assumed that the fallback mechanism would be rarely required and therefore a remote agent can supervice multiple vehicles simultaneously, reducing costs. The remote driver controls the vehicle using the already existing sensors. This information is streamed to the remote driver and the driving actions are streamed back to the vehicle. This introduces multiple safety, quality and latency requirements over the connection [152]. Additionaly, compresion techniques are needed to be able to transfer the sensor information in real time [153].

3.3.3 Geographically limited zone

This is a public road or portion of a city in which there is an agreement with the community to test SDSs. The public should be informed that SDSs are being tested in the area and the SDSs are not allowed to leave the region [44]. As other public road tests, this kind of tests are only possible once the SDS has reached a very high maturity level. Its main advantage been the high level of realism (stochasticity), although the geographical limitation translates into a limitation over possible weather types.

3.3.4 Dedicated lanes and parking garages

There are two types of dedicated lanes. Those that are physically separated from other driving lanes used by human drivers (by a divider, tunnel, etc.) and those which are not isolated. In the case of lanes that are not isolated the driver can activate the autonomous driving mode and the vehicle will move to this lane and keep driving on it until the driver decides to take back control or there is an unexpected problem [143]. Both types are good oportunities for testing the SDS without having the influence of human drivers, additionally the dedicated lanes may include extra sensors to help the SDS fullfill its tasks. Parking garages may also be reserved for autonomous vehicles only, in this case the driver would get out before the vehicle goes into the building. As in the case of the dedicated lanes the garage may include extra sensors to help the vehicle park [154, 155]. One difference in the parking lot case is that no humans are present in the area reducing risks. In tunnels or parking garages besides human factors also weather effects are removed, the developer of the SDS can hence focus on testing the remaining aspects.

3.4 Simulation

In contrast to other testing methods, simulations can be done from the very begining of the development of the autonomous driving software. As discussed in the introduction section and shown in figure 1.2 even without any hardware prototype a simulation model can be built to progresively test the perception algorithms, verify how other traffic participants react to the actions of the autonomous vehicle and verify if communication with the infraestructure (V2X) work as needed. Some of these simulations are relatively new and complement traditional ones like testing and development of vehicle dynamics [146]. In addition, simulation is becomming an important input for other testing methods. As shown in figure 3.1 each of the other testing methods can profit from a simulated environment. For example, as mentioned before, the proving ground can use virtual traffic participants synchronized with the real ones placed appropiately in 3D space. These participants can substitute real objects, persons or animals [156]. SIL, HIL, VIL and DIL tests make an increasing use of virtual environments and virtual traffic and all other methods can make use of counter factual (what-if) simulations when a disengagement happens.

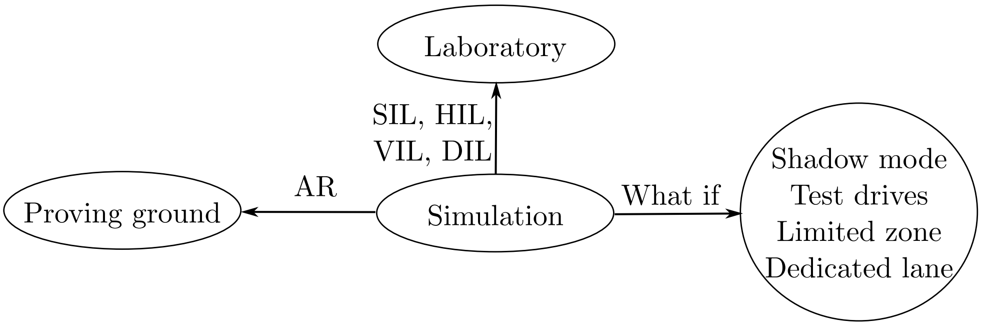


Figure 3.1: In addition of being an important testing method by itself, simulation complements the other testing methods. Proving ground tests can be expanded using virtual traffic participants or objects (AR techniques). In laboratory tests simulation can provide the virtual environment for SIL, HIL, VIL and DIL test benches and in all other cases it can be used to do counter factual analysis.

3.4.1 Software/Hardware Co-simulation

Some of the interface standards mentioned before provide also an API for message passing between modules and to start and configure the simulation. There are additional tools specificaly to achieve these tasks. Using these tools is possible to create X-in-the-loop test benches combining hardware and software modules as needed that run automatically. For example Functional Engineering Platform (FEP) is an open source framework and Software Development Kit (SDK) that allows the creation of simulation systems consisting of multiple modules that comunicate with each other in a time triggered or data triggered way [157]. In the time triggered mode FEP allows to define a module which is responsible for the time. This module generates time update events that are distributed to the other modules generating a common time basis for the simulation. Alternatively, the timing-slave modules can synchronize continuously with the timing master clock. FEP provides also mechanisms to start, configure and finalize the simulation system and each of its modules. In order to do this consistently a state machine is used that controls the initialization and deinitialization of each module. Robot Operating System (ROS) is another group of libraries and tools with the focus of helping in the development of robots [158], at its core there is also a message passing system between nodes (modules). As in FEP there are tools to control single nodes or systems of nodes and different timing configuration possibilities. While FEP and ROS use other protocols like Data Distribution Service (DDS) to transport messages between modules Distributed Co-simulation Protocol (DCP) defines itself a protocol, a data model and a finite state machine. As with ROS and FEP the result is a way to define, configure and run a co-simulation system in this case being independent of other tools [159]. Finally, X-In the Loop (XIL) is different to the previous standards as it defines how the communication between the test automation tools (e.g. ControlDesk, ECU-Test, TPT) and the test bench should take place. In order to do that XIL defines a port-based API. Using these ports is possible to configure, control and record data from the test bench. Changing the test bench is then simple as the test definition remains the same.

3.4.2 Scaling the simulation using cloud technologies

While co-simulations involving hardware or other modules that have real time requirements are better done using dedicated test benches, other types of simulation that do not have this requirement can be done using cloud technologies. The advantage of running the simulations on the cloud is that thousands of scenarios can be simulated parallely. The previously presented algorithms and tool used for co-simulation are well adapted for use in the cloud. For example, one or more modules can run in a container. A simulation system can then be formed by using multiple containers. If the network interface is configured so that the modules in a system can communicate with each other [160], protocols like DDS would allow message passing between them [161] and other services like establishing a common simulation time. Multiple simulations systems with different parameters can be created, configured and started using the tools provided by FEP or ROS. The simulation results can then be used for example for training of neural networks directly in the cloud, for validation by comparing them with real data also stored in the cloud or simply as testing stage in the development of a SDS.

3.4.3 Quantifying the reality gap

As mentioned in the introduction section there are multiple reasons for simulation results to diverge from real measurements. Some of these reasons can be corrected by improving the simulation model and adjusting its parameters. While others can not be solved in reasonable time frames and can hence be considered unsolvable in the context of an industry project. The problem however is that usually is not clear what is the acceptable reality gap, i.e. how much and in which aspects can simulation and measurements diverge so that the simulation data remains useful. Defining useful is important here. How useful the simulation results are depends on how they are used. In the context of this thesis simulation is used to generate point clouds which are consumed by a perception module. The question is then how the perception module is implemented. If we assume a conventional object-based sensor fusion algorithm then it is important to know how objects hypothesis are created based on the LIDAR detections. The generation of objects has usually as first step the segmentation of the point cloud. Segmentation can be done in different ways, one of them is using Random Sample Consensus (RANSAC) [162]. In RANSAC there is a parameter called the noise threshold, if for example a line is fitted then the detections outside the noise threshold measured from the line are considered outliers. We could then define an acceptable reality gap as equal to the noise threshold because two point clouds with detections that differ a distance (nearest neighbor) lower than this threshold will generate similar segmentations and objects. The reality gap itself can be measured based on how different position, size and rotation are between real and synthetic object for the same scene. This approach however reaches its limits when something like raw data sensor fusion using deep neural networks is used. As the classification results are based on multiple non-linearly interrelated parameters an acceptable reality gap has to be defined in a differnt way. One approach to do this is by comparing the performance of a network trained using real data and one trained using synthetic data in classifying unseen real data. The performance difference is caused by the reality gap (also known as domain gap in this case) [163]. An acceptable reality gap is then the one that produces an acceptable performance difference considering the costs and risks of collecting real data.

If simulation is used to do counter factual analysis the reality gap can be calculated by measuring how well the model can predict the position and direction of the traffic participants into the future using replayed real data. An acceptable reality gap is then determined by the tolerable difference between real and projected trajectories over time.

Once an acceptable reality gap is determined and the current reality gap is measured a cause analysis can be done. In this way is possible to define if the needed simulation quality can be reached for the phenomena in consideration, otherwise it may be better to guide the data collection campaigns towards capturing the needed data (see figure 1.2). The next chapter will focus on techniques for reducing the reality gap on LIDAR point clouds considering weather related effects.

3.5 Conclusions

- Traditional methods for testing ADAS have been evolving to deal with the demands introduced by SDS. The proving ground is growing adding more interaction with other traffic participants either real or synthetic and with the infrastructure. Laboratory tests add in all their forms: SIL, HIL, DIL, VIL increasingly new aspects of the real environment into a virtual replica. Innovations in public road tests include the use of automatic or manual trigger mechanism for data collection, as well as infrastructure adjustments in order to isolate human traffic participants or assist autonomous vehicles with their task reducing risks.

- Simulation is becoming a central part of the validation of SDS as it can be used to complement each other testing method. Either to generate the virtual environment needed for laboratory test, to augment the proving area with virtual elements or to do counter factual analysis.

- Due to the importance of having a common source of truth for the virtual environment in all testing methods, multiple standards and tools to define the static and dynamic parts of the virtual environment, as well as to connect simulation models with each other, have appeared. Together with the use of cloud technologies these standards and tools provide a very flexible framework to build simulations that can be scaled as needed.

- Measuring the reality gap and knowing what is the acceptable level is fundamental for correctly using simulation data. There are different techniques for measuring the reality gap depending on the algorithms that are consuming the data. Once it is measured if found bigger than the acceptable level the roots for the difference have to be considered. Some may be unsolvable in the time frames of the project but some may need only small adjustments in the simulation. Even in the first case the analysis is valuable because it can be used to guide data collection.

4 Effect of weather on radar and camera sensors

This chapter shows how the effect of weather on the radar and camera sensors can also be organized on the basis provided by the framework introduced in figure 1.5. This framework divides the effect of weather into three different regions: sensor cover, atmosphere and object emission. This division is useful because it allows to organize the different approaches used in simulating the effect of weather on each sensor. It also increases flexibility as the regions can be activated or deactivated separately, which also facilitates testing. Comparing models also becomes easier. Aditionally, it is important to show that the proposed framework can be used for other sensors, as the fusion of multiple sensors is a common method in autonomous vehicles [164]. The simulation framework must be able to provide a common ground truth for weather related phenomena in the same way as it does for the environment and traffic participants. Sensors whose main role is localization and not perception like GPS and Inertial Measurement Unit (IMU) are not included in this discussion, as well as short range sensors like the ultrasonic sensors. Internal adjustments of the sensor like for example increasing or reducing the noise level or failure injection are also not mentioned as these effects can be added directly into the sensor model. Simulation models based on GANs are not mentioned but can be easily integrated into the framework. In this case the GAN generated image or point cloud can be augmented with the extra pixels or detections generated by the models in each of the regions. The same method applies for real images or point clouds [165] as shown in the following figure.

4.1 Sensor cover

This region includes all soling effects. Soiling affects all sensors even the ones behind a windshield as the cleaning system doesn't act automatically, doesn't clean perfectly, there may not be enough water or there may be some defect. This region also include the effect of any other substance or particle that accumulates on the sensor as well as deterioration of the surface due to weather, sunlight, abrasives, rock impacts, etc.

4.1.1 Camera

Due to its small wavelength and high resolution the camera sensor is highly affected by soiling or droplets covering it [167]. Besides reducing the visibility this substances may also alter the perception by causing scattering or lensing effects. In order to classify and simulate the effect of soiling on the camera sensor a tile based approach is useful [167]

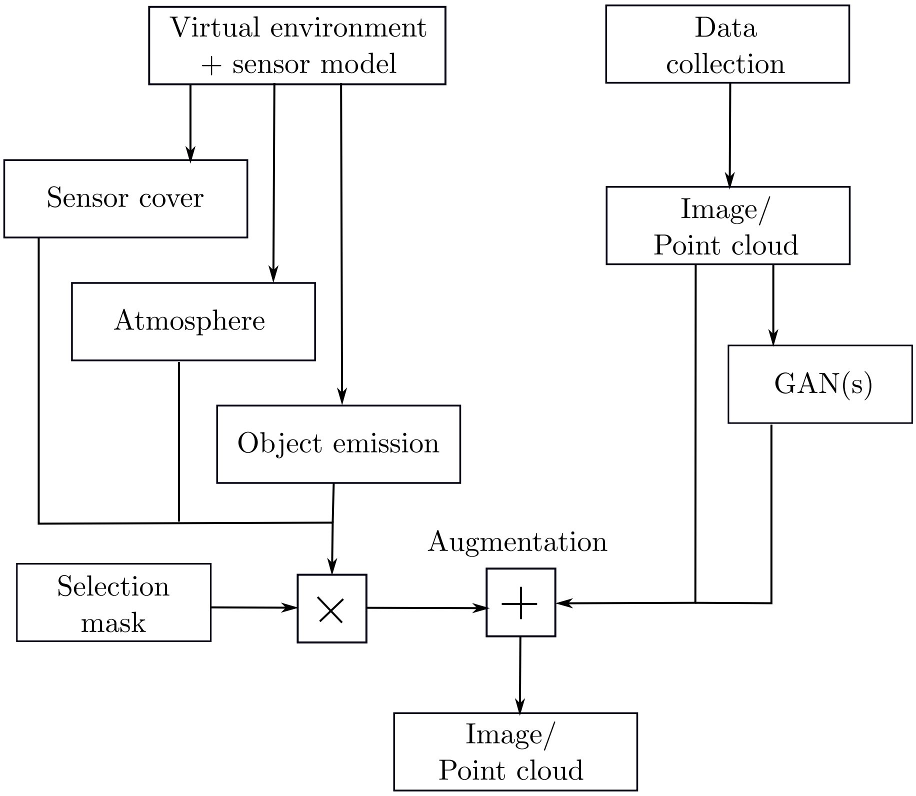


Figure 4.1: Real images and point clouds acquired with data collection as well generated using GANs can be combined with the results generated by each of the regions. This can be done by applying a selection mask in order to extract the needed pixels or detections which are then applied in the original images or point clouds. The augmentation process may require multiple steps to adjust the original and augmented data into a consistent whole. For example adjusting occlusion [165] or in the case of images reconstructing the 3D scene in order to correctly place the augmented data [166].

[168]. This approach is similar to the one suggested in this thesis for the LIDAR sensor in which the cover is divided into small regions from which the transmission values are measured [32]. In contrast to a LIDAR in the case of a camera is not possible to remove the sensor cover in order to measure its optical properties. For this reason a manual classification is suggested [167] (although GANs have also been used to automatically generate soiled images from real images [168]). The following classes were used: clean, opaque and transparent, depending on how strongly the light is blocked. Transparent tiles are those that are affected by some kind of soling but where the visibility is not zero, in opaques ones the visibility is zero and in clean ones it is 100%. The tiles are choosen with a size of 64x64 pixels and are then used to train a classifier to be able to automatically activate a cleaning system [167]. To create this texture the classified tiles could be used translating the label into a percentage. In this case a higher number of

classes would be desirable. This method can also be used to simulate the accumulation of snow that blocks certain parts of the sensor. Besides soiling, to simulate droplets caused either by rain or fog that accumulate on the windshield or lens photo realistic rendering can be employed. Droplets are simulated using spherical caps distributed around the lens or windshield [169]. This method provides a high level of realism. In case a high level of realism is not needed a texture can be used containing an approximate image of the drops.

4.1.2 Radar

The accumulation of a thin water, dirt or ice films on the radome of the radar sensor changes its transmission and reflection characteristics [170]. There are equations available that quantify the thickness of a water film formed in a spherically shaped radome [171]. Similarly as occurs with the LIDAR cover [32] a multi-layer structure is formed in this case from air, radome, water and air again. Besides absorbing part of the energy of the wave, the water-air interface generate an extra reflection that interacts with the already present reflection generated by the air-radome interface [170]. In contrast to the camera or LIDAR in which multiple samples on the cover are necessary to characterize or simulate the phenomena, for the radar most researchers use a single value [170] [172] [173] corresponding to the thickness of the film to calculate a single new value for transmissivity and reflectivity which is frequency dependent. After calculation these values can be easily integrated into the simulation using a texture as is done with the other sensor types. This covers the possibility of new radars having a higher resolution and requiring a finer grid.

4.2 Atmosphere

This region includes all scattering or absorption effects caused by weather phenomena. For some use cases these effects can be simulated directly by adjusting the sensor model for example to homogeneously reduce the range. There are nevertheless use cases, for example when multi-path effects are important, in which is necessary to simulate the effects using this region. Another motive could be that the effect of forces like wind or gravity upon the drops or snowflakes has to be simulated accurately.

4.2.1 Camera

Rain, fog and snow reduce contrast and visibility in the scene. They may also scatter the light from existing light sources in the direction of the sensor difficulting further the interpretation of the scene, for example in the case of headlights. Although snow, fog and rain can be simulated using textures [174] this means that depth and an accurate representation of the dynamics of the drops or flakes are lost [166]. An alternative is to use particles, these particles can be distributed in a certain volume based on physically accurate probability distributions for their size and concentration. The effect of each of the relevant forces can be simulated and then the scene can be rendered. Each particle can consist of a simple texture with the shape of a snowflake in order to increase rendering time. The size of the region in which the particles are located and their number can be controlled depending on the allowed rendering time and the level of realism needed. Rain can be simulated in a similar way by adjusting the corresponding probability distributions for the particles in this case based on real rain data. For each particle either textures or spheres can be used. The scene is then rendered under the same considerations as for snow. In the case of fog, single particle simulation is neither needed nor practicable for this reason volumetric rendering techniques are used [166].

4.2.2 Radar

At 77Ghz which is the usual frequency used in automotive radar systems fog appears to cause only a very small attenuation of the propagating radar signal [173] [175] and is usually not considered from the perspective of been able to generate extra detections. Something similar happens with snow as ice has a lower dielectric constant as water. The attenuation caused by snow although higher than for fog is still relatively small as well as its back-scattering value. This changes in case of wet snow in which case the values increase and become similar to that of water [176]. The main issue for radar sensors is rain specially at high rainfall rates [173]. Besides attenuation the higher back-scattering value of rain translates into a bigger back-scattering cross section when a whole volume is considered. This cross section can be big enough to generate extra detections in case the beam of the radar is narrow enough [175]. This is the case for Long range radar (LRR) which use a narrow beam to be able to detect object at high distances. Another factor in rain is a reduction of the accuracy in the perception of other objects in the scene as Radar Cross-section (RCS) of the objects becomes unstable. In order to simulate radar sensors a variation of raytracing can be used called Shooting and bouncing ray method (SBR). In the SBR method as is done to generate LIDAR point clouds the lengths of the rays are added as they advance in this case from the transmitting antenna until reaching the receiver antenna [175] [30], with this information the range but also the Doppler velocity and angle of each object in the scene can be calculated. One important difference is that to simulate the Doppler effect multiple temporal samples have to be acquired in order to calculate a single sensor frame [177]. In order to simulate the extra detections caused by rain particles can be used, as suggested for the LIDAR sensor a phenomenological model can be employed in which the particles do not correspond en number to the number of drops but to the number of detections that those drops generate reducing the simulation time [30].

4.3 Object emission

This region includes any weather phenomena that has a direct relation with an external emitter. Considering for example spray this region can be used to simulate the extra detections caused by particles following an inertial trajectory from the wheels. Very fine spray droplets that remain floating on the air would rather be simulated using the atmosphere region and the ones deposited on the sensor cover would be simulated using the sensor cover region.

4.3.1 Camera

Experiments considering the effect of spray on RGB camera images report a reduction of the contrast around the vehicle generating the spray due to the scattering of light produced by the spray droplets. Additionally, the field of view of the camera is blocked by the droplets that adhere to the windshield. Although the windshield wipers reduce this effect new droplets keep accumulating. The reduced contrast make it difficult to identify the edges of vehicles leading to miss-classifications. Additionally, during the night the reduced contrast caused by spray difficults the identification of rear lights [178]. In order to reconstruct these effects in simulation a method similar to the one proposed in this thesis for the LIDAR sensor [30] has been used for the camera [179]. The calculation of the droplets size are alike as well as the trajectory equations for the droplets. The rendering is more complicated due to the high resolution of the camera sensor in comparison with the LIDAR. The reflection of other objects in the scene on the spray drops is also taken into consideration. As the authors use case is to augment real images, they reconstruct the scene in 3D using the distance information provided by other sensors in the used dataset and project the pixels in the original image into their corresponding positions in 3D [179]. If the virtual environment is synthetic this step is not needed, all other steps remain identical.

4.3.2 Radar

Experiments related with measuring the attenuation caused by spray on the radar signal show a level of attenuation of 0.1 dB/m in average for a system using a 76 Ghz frequency [121]. This would correspond to around 1 dB losses for a 10 m spray cloud and is considered unlikely to prevent the detection of a vehicle. Other studies report similar results indicating that the detection of the radar remains stable under the precense of spray [178]. Although currently no information was found, it would be interesting to investigate how stable remains the detection with and without spray. As happens with rain it is possible that the RCS becomes unstable. Additionally, specifically narrow beam radars will have to be considered as in that case spurious detections are more likely. The secondary effects of spray for instance the accumulation of fine particles on the air are expected to cause only a small attenuation as they should be similar to fog or light rain. In contrast to LIDAR and camera a detailed particle simulation is in the case of radar not needed. Possible instabilities of the RCS can be simulated using error injection methods.

5 Journal Publications - Summaries

As mentioned in the previous chapter this thesis proposes 3 different regions in which each relevant weather related phenomena is simulated. In this chapter the concrete models to simulate each of the regions for a concrete LIDAR sensor are presented. The only assumption is that raytracing is used to sample each of the regions. Although the way in which each region is simulated as well as their sizes and locations can change depending on the sensor used for the simulation, the methodology proposed should remain applicable for other LIDAR sensors or even other sensor types (see chapter 4). The examples shown here have the focus on being able to run fast, use open source software and have a quantified reality gap. The proposed simulation models are able to reproduce the most important adverse weather effects on a LIDAR sensor. As shown in the previous chapter 4 the results from each of the regions can be directly used for testing and development of SDS or can be combined with real data, an example of which is also shown here. Although the next sections present the main results you may refer to the corresponding papers in the attachments for more information.

5.1 Characterization and Simulation of the Effect of Road Dirt on the Performance of a Laser Scanner

This region is the first region hit by the rays after leaving the sensor. For this reason the rays still contain a high intensity and changes in the reflection values of the cover can generate strong interference. Additionally, it makes the spurious detections generated in this region more difficult to filter out due to multiple reflections between the cover and the sender / receiver mirror used in mechanical LIDARS, which can cause the detections to be perceived at a distance further away than the physical limits of the cover. This thesis proposes to simulate these effects using a single texture which can cover most scattering, absorption and reflection phenomena. Extra detections due to multiple reflections can be integrated directly into the atmosphere region. As the following case study shows is possible to create the proposed textures using real measurements. The measurements can be highly automatized making the generation of textures a fast process, assuming data collection happens simultaneously. The resolution of the textures can be controlled based on the diameter of the measuring light source after collimation if the measurement is done in transmission or the size of the homogenizing rod if the measurement is done in reflection. Figure 5.1 exemplifies a measurement setup in transmission.

In order to be able to collect real samples, sensor covers are placed in the grill region of a vehicle doing a 12 days test drive. The roads chosen are normal roads where dirt accumulation is considered likely. The samples are placed in two different grill positions

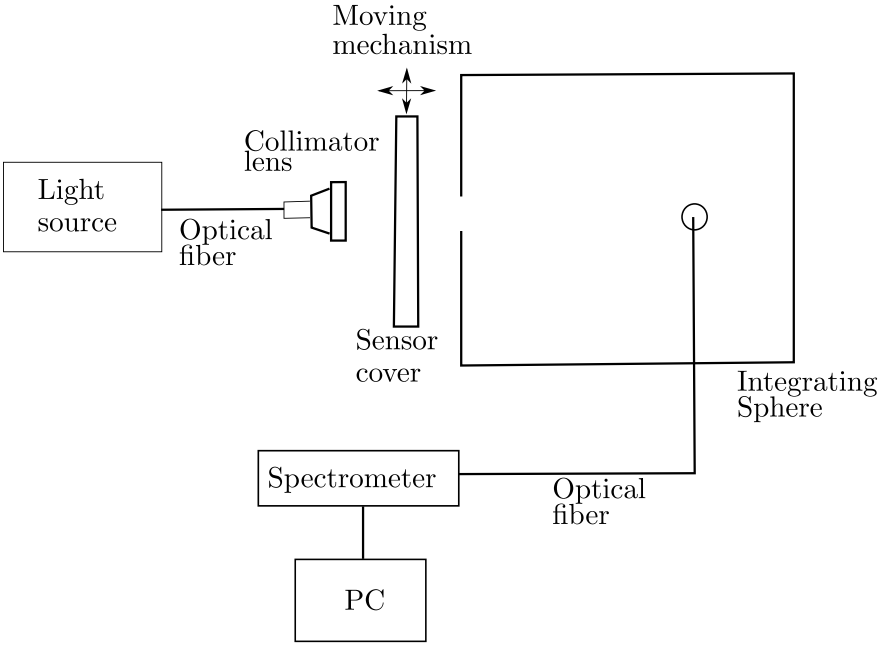


Figure 5.1: The sensor cover is sampled using a moving mechanism. The simplicity of this mechanism can vary depending on the shape of the sensor cover as ideally the laser light should hit the surface perpendicularly so the effect measured is due to changes in transmission and not due to variations in shape. The mechanism can be as simple as a linear motor or as complex as a robot arm. The rest of the setup corresponds to a conventional spectrometric measurement. For more details see the corresponding paper in the attachments section [28]

and are replaced every two days, documenting the weather conditions during those days with labels: dry, wet/sleet and sun/rain. The advantage of using only the covers is that the physical properties that influence dirt accumulation are the same as for the complete sensor. Additionally, the obtained covers can be replaced on a working sensor to verify how its performance is affected. To quantify changes on the optical properties transmission, reflection and scattering measurements are made for each of the collected samples (dirty covers) with respect to a clean sample. Performance measurements are made by measuring the position of a static object for multiple frames and comparing how the standard deviation of the measured distance changes for each sample with respect to a clean one. The measurements show a maximum variation in the transmission of 34.6%. Measurements done using the reflection setup show a maximum change of 37.5% with respect to a clean sample. The variation on the standard deviation of the distance measurements has a maximum value of 81.8% and is generated by the same cover with the largest variation in transmission and reflection values. This cover contains a combination of heavy and fine dirt and was generated during sunny/rain changeable weather. Using this kind of approach is possible to create a texture database for different types of dirt or other substances. In contrast to most textures in computer graphics that

are sampled in reflection these textures should be sampled in transmission. As shown in figure 1.5 the texture(s) would be placed in front of the sensor and would cause that some rays are absorbed, scattered or reflected. Changes in other optical properties like phase can also be added if needed (for example to simulate other LIDAR types for which phase differences are important). One disadvantage of this approach is that the dynamical changes of the optical properties are not captured. One way to do this would be to create a miniaturized reflection measurement system. This system could run periodic measurements of the cover. In simulation the dynamic changes in materials can be reproduced by exchanging textures or interpolating between them. In order to switch the textures the material name can be send using the data types defined by OSI. While the texture itself would be stored in a glTF format. The textures selected should depend on the weather related properties defined in OpenSCENARIO or / and on the road conditions defined in OpenDRIVE.
The author of this dissertation was responsible or co-responsible for conceptualization, methodology, validation, investigation, resources, writing of original draft preparation, writing of review and editing and visualization.

5.2 Weather Classification Using an Automotive LIDAR Sensor Based on Detections on Asphalt and Atmosphere

This region can reach from the sensor cover up to many meters radially. As the laser light propagates due to dispersion its intensity reduces. For this reason the probability of finding extra detections caused by atmospheric effects also reduces with distance. After a certain distance further back-scattering reduces the amount of light available for detecting other objects in the scene but is not able to create spurious detections anymore. This is an advantage from the perspective of simulation as it reduces the resources needed to simulate such complex phenomena. The amount of computing needed remains nevertheless high. In case the effect of forces like gravity and wind on the particles are not relevant for the simulation using volumetric scattering can be a good approach to simulate this region. Otherwise particles can be employed, in this case a second simplification can be used regarding the number of particles employed. The number of particles can be reduced to an amount similar to the number of spurious detections rain, fog or snow generate instead of the actual drops or snowflakes. For a sensor with a resolution in the thousands of pixels and considering that only a percentage of those pixels detect spurious reflections a reduction of one to two orders of magnitude in the number of particles needed can be reached. The remaining absorption and scattering effects can be treated as an homogeneous reduction of range dependent on rain intensity.

Collecting information about different weather types in a moving vehicle introduces multiple factors that complicate the analysis of an already complex interaction between light, particles and other objects in the environment. Therefore in this paper it was decided to use a static set up. Based on the information obtained using this set up it is possible, in a next step, to add the effects caused by the vehicle movement. Additionally, the information obtained is relevant for LIDARs placed as part of the road infrastructure.

For example inside of a road side unit on a pole [180]. Three different weather types were considered: rain, snow and fog, as well as the effect of a changing background radiation which will be discussed later. The measurement was done by placing the sensor outdoors for 9 months. Two factors were considered for the placement of the sensor. First it was placed high enough to reduce the effect of nearby buildings on the local weather and second the it was placed at an angle in which it had as a target an asphalt region of a parking lot. It is important to have a target as weather causes changes in the atmosphere but also changes in the material properties of objects in the environment. For a complete simulation of the phenomena these two changes have to be taken simultaneously into consideration and therefore they are also measured simultaneously. The measurements show that in the atmosphere all three weather types generate extra detections up to a distance of 5m. These detections are distributed radially over the whole field of view of the sensor. The form of the distribution of the detections is a decreasing exponential one from 0 to 50 cm. These detections are mostly caused by multiple reflections on the sensor cover. After 50 cm the detections tend to be distributed more uniformly with a peak at the point in which the sender and receiver fields of view of the used biaxial sensor start to overlap. The exact shape of the distribution depends on the weather type. The paper formulates the problem as a classification problem but simultaneously shows which parameters are relevant to generate a distribution (e.g. using particle systems or volumetric scattering) that simulates the detections caused by each weather type. The same findings apply to the detections on the asphalt region. The distribution of detections in this case can be used to parameterize the BRDF of the material used in the simulation. One important difference in the asphalt region is the material accumulation caused by snow. Due to the accumulation of snow on the street the detection distance changes. To introduce this change in a simulation the surface normals could be adjusted but eventually the mesh itself will have to be modified. The weather related configuration parameters provided by OpenSCENARIO can be easily mapped into the atmosphere region simulation model. In the case of fog a conversion from visibility into concentration could be used [181]. For rain and snow the precipitation intensity can be used directly. From the perspective of classification this case study shows that it is possible to classify weather using 10s intervals into: clear, fog, rain and snow with a F1 score higher than 80%. The intervals may refer to the detections caused either on the atmosphere or due to an object in the environment exposed to changing weather (although the paper uses a big horizontal object: asphalt region). The two classification methods can be used independently of each other or simultaneously. Between the most important classification parameter for both regions are the maximum number of detections as well as the average intensity of the reflections.

As indicated in the author contributions the author of this dissertation was responsible or co-responsible for conceptualization, methodology, software, validation, investigation, resources, data curation, writing of original draft preparation, writing of review and editing and visualization.

5.3 The Effect of Spray Water on an Automotive LIDAR Sensor: A Real-Time Simulation Study

As mentioned in the fundamentals section part of the effect caused by spray is the generation of spray clouds around the vehicle wheels. These spray clouds move together with the vehicle. From a 3D modeling perspective they can be considered as parented with the object that represents the vehicle in the simulation. Spray is not the only phenomena in which this kind of parenting relation is present. Exhaust gases as well as snow falling from the roof of a vehicle or sand failing from a truck are all phenomena in which extra detections are generated by an emitter which move together with the vehicle generating it. The consequence of moving together with the emitting vehicle is that the phenomena acquires new temporal dynamics. In contrast to atmospheric phenomena that are homogeneous in space, object emission phenomena are relevant for a time interval dependent on the actions taken by the emitter. When the emitter is behind or sufficiently far away no extra detections are generated but if it overtakes or remain in an adjacent lane it may generate spurious detections during long time intervals. The following case study shows how a spray model can be generated and how to evaluate its reality gap.

In order to collect real spray data four different measurement campaigns were done. One of them was done on a proving ground using two vehicles, one generating the spray and driving in front and a second one capturing the generated point clouds using a LIDAR sensor. Both vehicles were driven at a constant speed and distance from each other. The vehicle in front was driven on a tile covered road which was floated with water. Multiple measurements were done at speeds going from 39 kmh up to 60 kmh at a distance of $17.4 \pm 0.3m$. A device was also mounted on the vehicle having the LIDAR which is able to measure the water level on the road. The other three measurements were done on public highways during rainy days at an average speed of $89 \pm 4kmh$. The measurement set up was the same in all four cases and is shown in figure 5.2. As it is not possible to maintain a constant distance between the two vehicles in a public highway the measurements were classified in this case based on the distance between the vehicles going from 15.5 m up to 32.6 m.

The proposed simulation model is a semi-phenomenological one, as some aspects are physically based while others are extracted directly from the measurements. The advantage of the model however is that it is able to run on real time. The particles were inserted at specific portions of the wheels. Their number being determined from the measurements and depending on the vehicle speed and water level on the road. Their longitudinal trajectory being determined by their mass, the wheel rotation speed and the longitudinal wind force, whose value depends on the vehicle speed. Their transversal trajectory is defined by the effect of lateral wind forces which are not calculated but are adjusted in intensity and direction based on the measurements. This is similar to the method used in the Reynolds-Average Navier-Stokes (RANS) equations in which a time dependent and a time independent components are used [182, 183, 184]. Similarly to what was done for the atmosphere the reality gap of the proposed spray model was

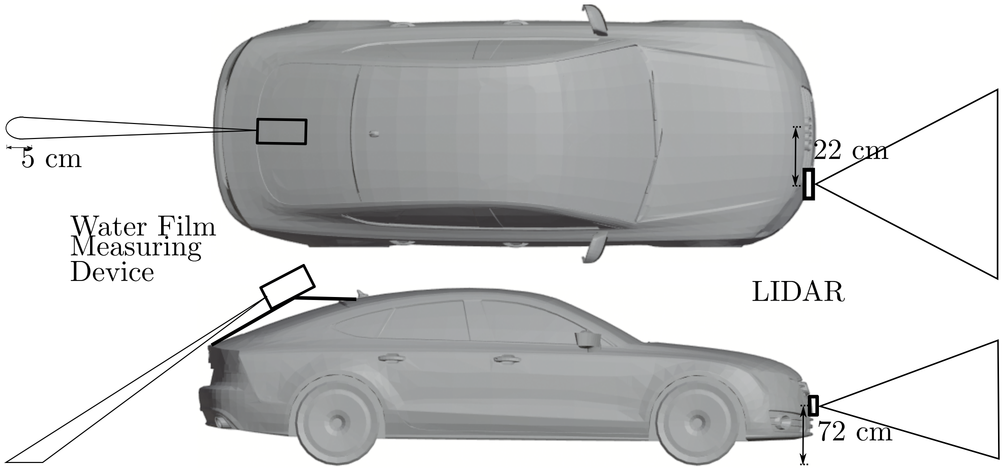


Figure 5.2: Eagle and side view of the measurement setup used for collection of spray data.

calculated by creating a histogram in this case in 2D of the whole spray region. The real and simulated distributions were compared using different metrics. The paper suggest the use of the Pearson correlation coefficient but Jaccard distance or Hamming distance could also be used. The correlation reduces from lower speeds in which it is around 0.7 to higher speeds in which it reaches the minimum of 0.5. As shown in the figure 1.5 a particle based model similar to the one proposed here can be integrated into the suggested simulation framework by inserting the particles into the simulation at the correct position on the wheel meshes. This position can be determined if the wheel axis and wheel size are known. The street water level can be saved on the OpenDRIVE file. The wind forces can be placed on the scene based on the the wind speed and wind direction parameters from OpenSCENARIO. Instead of particles a mesh based model (see chapter Fundamentals) could be used. In this case the volume containing the boundaries of the mesh would be parented to the moving vehicle.

5.3.1 Background radiation

For the used LIDAR sensor it was found that the presence of sunlight causes a reduction in the total number of detections [32]. This may vary for other LIDARs depending on the algorithm the manufacturer uses to deal with an increased noise level. For this reason a phenomenological simulation model is suggested, which also has as advantage to be able to run in real time. The simulation parameters can be extracted from a static setup, similar to the one suggested in the weather classification use case, by counting the total number of detections in the scene while measuring the radiation level normal to the sensor surface. The variations in the number of detections can them be added directly into the sensor model.

The author of this dissertation was responsible or co-responsible for conceptualization, methodology, software, validation, investigation, resources, writing of original draft preparation, writing of review and editing and visualization.

5.4 Data Augmentation of Automotive LIDAR Point Clouds under Adverse Weather Situations

While each of the previous regions provide a model of a certain weather effect that can be activated or deactivated as needed in a simulation, it is also important to consider how these results can be combined with real data or with data generated using GANs. Being able to combine simulation, real and GANs generated data provides the maximum flexibility and allows for the framework to be usable to test and validate ADAS functions in all their maturity levels. The following case study shows how the selection mask and augmentation block in figure 4.1 can be implemented in a concrete example. Additionally, the selected example shows how simulation can reduce the complexity and risk of the data collection task while simultaneously being able to provide realistic results.

As mentioned in the fundamentals chapter a human driver may require up to 30s in order to understand the driving situation and be able to take over from a SDS. In a highway at high speeds this time could translate into a long traveled distance. Assuming it is raining and other vehicles are generating spray the SDS will have to clearly differentiate between detections caused by spray and possible solid objects on the highway, for example objects that fell off from other vehicles. Considering a situation in which there is a real solid object on the street it is important for the SDS to be able to detect this object(s) and activate a safety mechanism, for instance: slightly reducing speed and activating the seat belt pretensioner of the passengers. Collecting real data for this use case would be extremely difficult and dangerous even in a proving ground. Simulation can be used in this case in order to generate training data for a classifier, which would then be able to trigger the safety mechanism. The proposed method is to use the previously discussed spray and rain models and add a solid object (a box for simplicity) placed at different positions on the road. The reason for using the spray and rain models in the simulation is to match the changes in echo number and number of detections that otherwise would make the obtained object point cloud less realistic. The synthetic object position, size, rotation and reflection values are changed and the scene is rendered obtaining the corresponding point cloud. From the point cloud the detections caused by the synthetic object can be easily extracted as the renderer (Cycles in Blender) creates a separate mask for each object in the scene. Figure 5.3 shows the simulation setup used.

The point cloud of the object is extracted and is used to augment real spray point clouds. After recalculating object occlusion noise is added to the synthetic detections based on the noise profile of the point cloud generated around real moving vehicles. The final result is a plausible and labeled point cloud including the desired solid object.

This method has three main advantages:

- The classifier can be thoroughly evaluated, as simulation allows the modification of all possible characteristics of the solid object.
- Although real spray point clouds have to be collected, the dangerous aspect: the solid object(s), doesn't have to be present.

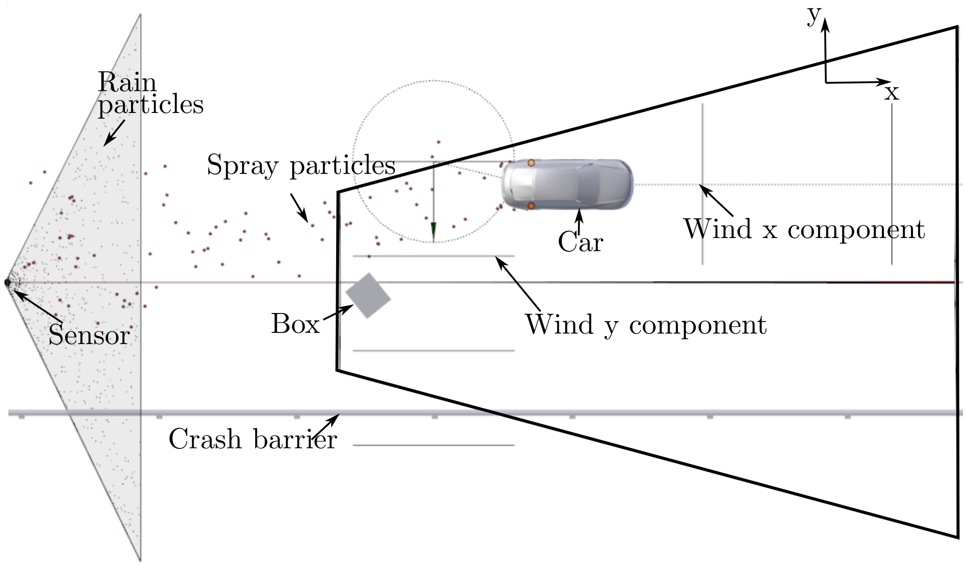


Figure 5.3: This figure shows the simulation setup used to generate the synthetic object point clouds. The spray particles are injected from the vehicle wheels and follow a trajectory based on their mass and the wind forces. The size of the particles and their number are selected to match the average number of detections that real spray clouds generate. The region marked with the black line corresponds to the region of interest. This region is determined based on the presence of spray and the minimum time required to activate safety actions. The light gray region simulates the effect of rain on the sensor. For more information refer to the corresponding paper on the attachment section [165].

- The degree of realism is high as only the object is synthetic and it is rendered in a way in which the resulting detections match those that real detections would cause.

As indicated in the author contributions the author of this dissertation was responsible or co-responsible for conceptualization, methodology, software, validation, investigation, resources, data curation, writing of original draft preparation, writing of review and editing and visualization.

5.5 Conclusions

- The use of phenomenological models together with the tools provided by 3D engines allow the creation of simulation models for weather related effects that are able to run in realtime. Additionally, these models have a measured reality gap that can be calculated based on the correlation of their 1D or 2D histograms with those of real point clouds.

- Particle systems are very flexible and can be adjusted interactively to the presence of new forces making their use more intuitive and modifications of the simulation simpler for the user. When the number of particles exceed a certain value in which real time simulation is not possible or when the effect of forces is not relevant volumetric scattering models can be used.

- In order to reduce the number of needed particles the proposed spray model simulates the minimum necessary number of them in order to match the detections observed in real point clouds. The particles in the simulation correspond therefore to a high enough concentration of real drops to generate a reflection and not to a single drop. The same approach can be used to simulate rain or snow.

- Each of the proposed models corresponds to a region along the path followed by the light emitted by the sensor. Having separate regions allow to combine the models between them and also together with real data or data generated using GANs.

6 Summary & Outlook

A methodology is presented and validated for the simulation of weather related effects on an automotive LIDAR sensor. The proposed methodology has as characteristic that it makes use of simulation techniques that are able to run in real time. In order to achieve this, a hybrid sensor model is proposed that combines as much as possible from the known physics of the corresponding phenomena with a phenomenological representation of the effect of water drops, snowflakes and other substances on the sensor perception. Using this simplification is possible to synthetically generate point clouds that highly correlate with measured ones while reducing the number of used particles to a minimum. Due to their phenomenological nature the proposed models require a certain amount of real data collection either in measurements campaigns or in the laboratory. This is however not a disadvantage as the reality gap of the simulation has always to be measured. Because the proposed models use established simulation standards and Raytracing they can be included in early development phases when no real sensor prototype exists. The different sections simulating the effect of weather on the sensor cover, on the near region and on the spray region can be deactivated until enough information is available. When enough information is available the regions can be activated independently as needed. Differences between the synthetic and real point clouds can be used to guide data collection campaigns until the reality gap is reduced to the required level. In addition to its flexibility and real time capability, the proposed model has also advantages regarding costs as it can be implemented using existing open source tools. Costs savings also express in the reduction of the amount of data that needs to be collected and of the risks incurred by collecting it.

6.1 Summary

While developing the proposed methodology a number of partial achievements was reached. These achievements are listed below.

1. A method is defined to collect samples of dirt accumulated on the cover of a LIDAR sensor as generated on conventional road trips. The optical properties of these samples are analyzed. Methods are proposed to measure the transmission, reflection and forward scattering caused by each dirt type as well as the effect on the distance resolution. The experimental results are quantized in a way in which they can be easily included into a Raytracing based simulation.

2. A nine months measurement is done placing a LIDAR sensor outdoors and collecting point clouds affected by rain, snow, fog and changing background radiation

(sunlight). The sensor is placed in a position so that it has as target an asphalt region of a parking lot. Using the collected data the effect of each weather type is analyzed for two regions: the atmosphere region (extra detections in the near field of the sensor) and the street region (target area). The analysis are based on changes on the statistical properties of the histogram obtained by accumulating the detections for each respective region. A relation is established between the statistical changes and the optical properties changes that caused them. Besides showing that an automotive LIDAR sensor is able to classify different weather types, the results also show the main parameters to consider when simulating these type of phenomena and how to validate the simulation results.

3. Measurements were made on a proving ground and on the highway (three different rainy days) in order to collect point clouds affected by spray. In the proving ground a street was used that can be fluted with few millimeters of water. In both cases two vehicles were used. One producing spray and the other capturing the LIDAR data. Additionally, an extra sensor was used in order to measure the water level on the street. All this information was analyzed based on the physical background of the spray generating process and the working principle of the LIDAR sensor. A simulation model was proposed and validated using particle systems and forces to simulate the detections caused by spray. Cycles [131] was used as path tracer to render the image. The material properties were adjusted in order to obtain per pixel the distance that the rays traveled in a set up similar to the one shown in figure 1.4. The simulated point clouds show a high correlation with the real ones, the simulation times go from 40 ms to 240 ms per frame.

4. The proposed spray simulation model is used in a problem in which collecting real data would be extremely difficult. Here a function is proposed to protect the vehicle passengers by activating certain safety measures in case a solid object is seen on the current highway driving lane. It is however assumed that the street is wet and that there are other vehicles generating spray. Considering only the LIDAR sensor the question is asked what would be the performance of a classifier trained to detect solid objects on the spray cloud. This question is solved by generating physically accurate synthetic objects using the spray model. The point cloud generated by those objects is used to augment real point clouds containing spray. The results show in which areas the classifier would struggle and what kind of performance is to be expected. The results also show the importance of the improve / guide loop in figure 1.2.

5. The different models for each of the regions: cover, near and spray, are integrated into a complete framework based on Raytracing and established simulation standards. These three characteristics: the use of regions, the use of Raytracing and the use of standards, make the resulting framework flexible enough to be used in all stages of development of the autonomous driving function. The simulation can in this way evolve or devolve as needed.

6.2 Outlook

The simulation framework introduced in figure 1.5 constitutes a basis upon which collaboration between different companies and institutions developing autonomous driving system can flourish. It will be upon this collaboration that robust simulation models, best practices and further standards will emerge. The following list mentions improvement options in the development of weather related simulations for LIDAR and other sensors.

1. It would be desirable and it is likely that over time the industry converges to one or two main LIDAR technologies for use in autonomous vehicles. This convergence will free resources that are currently focused in the many different alternatives into the winning ones. When this happens it will be possible to build large databases using data collected with those sensors. Something like an open access dirt (and other substances) database will then be possible. This database could contain as proposed a detailed measurement of the transmission and scattered caused by different substances accumulated on the LIDAR cover. These textures can then be shared for use in simulation. If possible, the database will also contain a measurement of the environment by the sensor been affected by the corresponding substance. In this way the simulation results can be validated.

2. The proposed models work well with the used sensor but the question remains how well they would work with a different LIDAR sensor of the same or different technology. Using two different LIDARs for data collection could allow to better differentiate the aspects that are valid in general based on the physics of the phenomena and those that are sensor or vendor specific. The use of other sensor types like high resolution RADAR could also help to improve the quality and generality of the proposed simulation models. The use of Raytracing in industry and academia for the simulation of many of the sensor used in autonomous driving helps in comparing the obtained results.

3. The reliable measurement of the level of water on the street is an important aspect for a correct simulation of spray water. This measurement is however challenging using moving platforms. The broad adoption of sensors to measure this quantity as part of the street infrastructure would be an important step towards the development of autonomous driving vehicles. It helps with simulation and provides online information to the SDS while driving. In a closed loop the simulation results and the currently measured data can be directly compared.

4. While for each of the proposed simulation models a brief description was given about how it can be integrated into the current main simulation standards, a better integration would be desirable in the future. Ideally the standards would define some default sensor models for each weather phenomena and find a parametrization that is valid for all of them. The user could use the default models or use his own based on the defined parametrization. Similarly to the texture database for dirt,

a database with "ready to use" weather models would highly increase the use of simulation and strongly reduce the time required to develop it. This is to be expected however, when the convergence in technologies mentioned in the first point happens.

5. The existence of a common database for industry wide exchenage of sensor data would highly help to improve the proposed models and in general the use of simulation. Projects like GAIA-X [185] could provide companies with a lot of data to validate their simulation models and create new ones. This common database also helps in the development of best practices and standards.

Acknowledgments

The author would like to thank Jia Chen and Thiemo Gerbich for their valuable advice during the years that took to complete this thesis. Additionally, many thanks to the colleges Hayko Tobias, Andreas Sonntag, Olaf Schubert, Mario Berk, Christoph Glassl, Boris Buschardt, Hans-Martin Kroll and Thomas Maag for their help and valuable comments while collecting the required data and writing the articles.

Acronyms

ADAS	Advanced Driver Assistance Systems.
AI	Artificial intelligence.
AR	Augmented reality.
ASAM	Association for Standardization of Automation and Measuring systems.
AUTOSAR	AUTomotive Open System ARchitecture.
BRDF	Bidirectional Reflection Distribution Function.
CityGML	City Geography Markup Language.
DCP	Distributed Co-simulation Protocol.
DDS	Data Distribution Service.
DGR	Detection generating region.
DIL	Driver In the Loop.
DSL	Domain Specific Language.
FEP	Functional Engineering Platform.
FMI	Functional Mock-up Interface.
GAN	Generative Adversarial Network.
glTF	Graphics Language Transmission Format.
HIL	Hardware In the Loop.
IIHS	Insurance Institute for Highway Safety.
ISO	International Organization for Standardization.
ISO/WD	International Organization for Standardization/ Working Draft.
LASER	Light Amplification by Stimulated Emission of Radiation.
MEMS	Micro Electromechanical Mirror.
MPPC	Multipixel Photon Counter.

NDS	Navigation Data Standard.
NIR	Near Infrared.
O3DE	Open 3D engine.
ODD	Operational Design Domain.
OPA	Optical Phased Array.
OSI	Open Simulation Interface.
OSM	OpenStreetMap.
PBR	Physically Based Rendering.
PLM XML	Product Lifecycle Management eXtensible Markup Language.
RANSAC	Random Sample Consensus.
ROS	Robot Operating System.
SDK	Software Development Kit.
SDS	Self-Driving Systems.
SIL	Software In the Loop.
SiPMT	Silicon Photomultiplier.
SNR	Signal to Noise Ratio.
SSP	System Structure and parametrization.
SuT	System under Test.
SWIR	Short-Wave Infrared.
V2X	Vehicle to Everything.
VIL	Vehicle In the Loop.
VR	Virtual reality.
VTD	Virtual Test Drive.
XIL	X-In the Loop.
XML	eXtensible Markup Language.

List of Figures

Bibliography

[1] D. K. Oka, T. Fujikura, and R. Kurachi. Shift left: Fuzzing earlier in the automotive software development lifecycle using hil systems. In *Proc. 16th ESCAR Europe*, pages 1–13, 2018.

[2] P. Koopman. A case study of toyota unintended acceleration and software safety. *Presentation. Sept*, 2014.

[3] R. W. Butler and G. B. Finelli. The infeasibility of quantifying the reliability of life-critical real-time software. *IEEE Transactions on Software Engineering*, 19(1):3–12, 1993.

[4] P. Koopman and M. Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.

[5] J. Collins, D. Howard, and J. Leitner. Quantifying the reality gap in robotic manipulation tasks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6706–6712. IEEE, 2019.

[6] J. Meyer, H. Becker, P. M. Bösch, and K. W. Axhausen. Autonomous vehicles: The next jump in accessibilities? *Research in transportation economics*, 62:80–91, 2017.

[7] J. Liu, Q. Sun, Z. Fan, and Y. Jia. Tof lidar development in autonomous vehicle. In *2018 IEEE 3rd Optoelectronics Global Conference (OGC)*, pages 185–190. IEEE, 2018.

[8] T. Hanke, N. Hirsenkorn, C. van Driesten, P. Garcia-Ramos, M. Schiementz, S. Schneider, and E. Biebl. A generic interface for the environment perception of automated driving functions in virtual scenarios. *Internet: https://www. hot. ei. tum. de/forschung/automotive-veroeffentlichungen*, 2019.

[9] C. Linnhoff, P. Rosenberger, M. F. Holder, N. Cianciaruso, and H. Winner. Highly parameterizable and generic perception sensor model architecture. In *Automatisiertes Fahren 2020*, pages 195–206. Springer, 2021.

[10] B. Schlager, S. Muckenhuber, S. Schmidt, H. Holzer, R. Rott, F. M. Maier, K. Saad, M. Kirchengast, G. Stettinger, D. Watzenig, et al. State-of-the-art sensor models for virtual testing of advanced driver assistance systems/autonomous driving functions. *SAE International Journal of Connected and Automated Vehicles*, 3(12-03-03-0018):233–261, 2020.

[11] ASAM. Asam openscenario, May 2022. URL: `https://www.asam.net/standards/detail/openscenario/`.

[12] ASAM. Openscenario v2.0.0-prc.1, June 2022. URL: `https://asam-ev.github.io/public_release_candidate/asam-openscenario/2.0.0/introduction.html`.

[13] ASAM. Asam sim: Guide, June 2021. URL: `https://www.asam.net/asam-guide-simulation/`.

[14] ASAM. Asam opendrive, August 2021. URL: `https://www.asam.net/standards/detail/opendrive/`.

[15] R. W. Lentz, D. P. Brutzman, and M. Kamburelis. X3d and gltf model differencing for conversions, comparison and conformance testing. In *Web3D*, pages 6–1, 2021.

[16] N. association members. Nds tools: Providing our members with everything they need, October 2019. URL: `https://nds-association.org/nds-tools/`.

[17] L. Ding, A. Ball, J. Matthews, C. McMahon, and M. Patel. Product representation in lightweight formats for product lifecycle management (plm). In *4th International Conference on Digital Enterprise Technology*, page 47. Citeseer, 2007.

[18] M. E. O'Brien and D. G. Fouche. Simulation of 3d laser radar systems. *Lincoln Laboratory Journal*, 15(1):37–60, 2005.

[19] F. Wang, Y. Zhuang, H. Gu, and H. Hu. Automatic generation of synthetic lidar point clouds for 3-d data analysis. *IEEE Transactions on Instrumentation and Measurement*, 68(7):2671–2673, 2019.

[20] G. F. Gusmão, C. R. H. Barbosa, and A. B. Raposo. Development and validation of lidar sensor simulators based on parallel raycasting. *Sensors*, 20(24):7186, 2020.

[21] S. Hasirlioglu and A. Riener. A model-based approach to simulate rain effects on automotive surround sensor data. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2609–2615. IEEE, 2018.

[22] T. Yang, Y. Li, Y. Ruichek, and Z. Yan. Performance modeling a near-infrared tof lidar under fog: A data-driven approach. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[23] M. Berk, M. Dura, J. V. Rivero, O. Schubert, H.-M. Kroll, B. Buschardt, and D. Straub. A stochastic physical simulation framework to quantify the effect of rainfall on automotive lidar. *SAE International Journal of Advances and Current Practices in Mobility*, 1(2019-01-0134):531–538, 2019.

[24] R. H. Rasshofer, M. Spies, and H. Spies. Influences of weather phenomena on automotive laser radar systems. *Advances in Radio Science*, 9(B. 2):49–60, 2011.

[25] B. Wu, A. Wan, X. Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018.

[26] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019.

[27] S. Zhao, Y. Wang, B. Li, B. Wu, Y. Gao, P. Xu, T. Darrell, and K. Keutzer. epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3500–3509, 2021.

[28] J. R. V. Rivero, I. Tahiraj, O. Schubert, C. Glassl, B. Buschardt, M. Berk, and J. Chen. Characterization and simulation of the effect of road dirt on the performance of a laser scanner. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[29] K. Huang. *Introduction to statistical physics*. Chapman and Hall/CRC, 2009.

[30] J. R. Vargasrivero, T. Gerbich, B. Buschardt, and J. Chen. The effect of spray water on an automotive lidar sensor: A real-time simulation study. *IEEE Transactions on Intelligent Vehicles*, 2021.

[31] C. Jianguang, N. Xuxiang, Y. Bo, and Y. Huimin. Analysis of detection probability performance of sipm lidar under sunlight. *Opto-Electronic Engineering*, 48(10):210196–1, 2021.

[32] J. R. Vargas Rivero, T. Gerbich, V. Teiluf, B. Buschardt, and J. Chen. Weather classification using an automotive lidar sensor based on detections on asphalt and atmosphere. *Sensors*, 20(15):4306, 2020.

[33] P. Rosenberger, M. Holder, M. Zirulnik, and H. Winner. Analysis of real world sensor behavior for rising fidelity of physically based lidar sensor models. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 611–616. IEEE, 2018.

[34] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

[35] X. Yang, Y. Wang, T. Yin, C. Wang, N. Lauret, O. Regaieg, X. Xi, and J. P. Gastellu-Etchegorry. Comprehensive lidar simulation with efficient physically-based dart-lux model (i): Theory, novelty, and consistency validation. *Remote Sensing of Environment*, 272:112952, 2022.

[36] B. Bitterli, C. Wyman, M. Pharr, P. Shirley, A. Lefohn, and W. Jarosz. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (TOG)*, 39(4):148–1, 2020.

[37] P. Moreau, M. Pharr, and P. Clarberg. Dynamic many-light sampling for real-time ray tracing. In *High Performance Graphics (Short Papers)*, pages 21–26, 2019.

[38] C. Barré-Brisebois, H. Halén, G. Wihlidal, A. Lauritzen, J. Bekkers, T. Stachowiak, and J. Andersson. Hybrid rendering for real-time ray tracing. In *Ray Tracing Gems*, pages 437–473. Springer, 2019.

[39] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman. Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6941–6948. IEEE, 2019.

[40] J. Peddie. *Ray Tracing: A tool for all*, volume 5. Springer, 2019.

[41] A. Alhakamy and M. Tuceryan. Real-time illumination and visual coherence for photorealistic augmented/mixed reality. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[42] P. E. Kivi, M. J. Mäkitalo, J. Žádník, J. Ikkala, V. K. M. Vadakital, and P. O. Jääskeläinen. Real-time rendering of point clouds with photorealistic effects: A survey. *IEEE Access*, 10:13151–13173, 2022.

[43] K. Franke. Volkswagen group: Leveraging vires vtd to design a cooperative driver assistance system. *Engineering Reality Magazine, Winter*, pages 10–14, 2018.

[44] K. Gangel, Z. Hamar, A. Háry, Á. Horváth, G. Jandó, B. Könyves, D. Panker, K. Pintér, M. Pataki, M. Szalai, et al. Modelling the zalazone proving ground: a benchmark of state-of-the-art automotive simulators prescan, ipg carmaker, and vtd vires. *Acta Technica Jaurinensis*, 14(4):488–507, 2021.

[45] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL: `http://www.blender.org`.

[46] D. Derakhshani. *Introducing Autodesk Maya 2013*. John Wiley & Sons, 2012.

[47] J. Olsen. Realtime procedural terrain generation. 2004.

[48] W. L. Raffe, F. Zambetta, and X. Li. A survey of procedural terrain generation techniques using evolutionary algorithms. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2012.

[49] N. Shaker, J. Togelius, and M. J. Nelson. *Procedural content generation in games*. Springer, 2016.

[50] C. Beckham and C. Pal. A step towards procedural terrain generation with gans. *arXiv preprint arXiv:1707.03383*, 2017.

[51] J. Togelius, N. Shaker, and J. Dormans. Grammars and l-systems with applications to vegetation and levels. In *Procedural Content Generation in Games*, pages 73–98. Springer, 2016.

[52] R. M. Smelik, K. J. De Kraker, T. Tutenel, R. Bidarra, and S. A. Groenewegen. A survey of procedural methods for terrain modelling. In *Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)*, volume 2009, pages 25–34. sn, 2009.

[53] J. Dong, J. Liu, K. Yao, M. Chantler, L. Qi, H. Yu, and M. Jian. Survey of procedural methods for two-dimensional texture generation. *Sensors*, 20(4):1135, 2020.

[54] A. Potdar and M. Ghotgalkar. 3d character generation from images using convolutional neural networks and 3d-character factory. 2019.

[55] Y. Oliveira. A primer on procedural character generation for games and real-time applications. In *Game Dynamics*, pages 115–132. Springer, 2017.

[56] A. Akman, Y. Sahillioglu, and T. M. Sezgin. Generation of 3d human models and animations using simple sketches. 2020.

[57] Z. Bhatti, A. Shah, and F. Shahidi. Procedural model of horse simulation. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pages 139–146, 2013.

[58] M. Bonner. The world-shaped hall on the architectonics of the open world skybox and the ideological implications of the open world chronotope. *Game— World— Architectonics*, page 65, 2021.

[59] M. MIYAMOTO and R. DRIANCOURT. Rendering techniques of final fantasy xv. 2016.

[60] M. Scorpio, R. Laffi, A. Teimoorzadeh, G. Ciampi, M. Masullo, and S. Sibilio. A calibration methodology for light sources aimed at using immersive virtual reality game engine as a tool for lighting design in buildings. *Journal of Building Engineering*, 48:103998, 2022.

[61] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3050–3057. IEEE, 2014.

[62] N. Haala, M. Peter, J. Kremer, and G. Hunter. Mobile lidar mapping for 3d point cloud collection in urban areas—a performance test. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, 37:1119–1127, 2008.

[63] N. Paparoditis, J.-P. Papelard, B. Cannelle, A. Devaux, B. Soheilian, N. David, and E. Houzay. Stereopolis ii: A multi-purpose and multi-sensor 3d mobile mapping system for street visualisation and 3d metrology. *Revue française de photogrammétrie et de télédétection*, 200(1):69–79, 2012.

[64] D. HOLDENER. Konzeption und realisierung eines neuen portablen 360-stereokamerasystems.

[65] B. Zhang, S. Miller, S. Walker, and K. DeVenencia. Next generation automatic terrain extraction using microsoft ultracam imagery. In *Proceedings of the ASPS 2007 Annual Conference*, 2007.

[66] H. Arefi, P. d'Angelo, H. Mayer, and P. Reinartz. Automatic generation of digital terrain models from cartosat-1 stereo images. ISPRS, 2009.

[67] M. Dupuis, M. Strobl, and H. Grezlikowski. Opendrive 2010 and beyond–status and future of the de facto standard for the description of road networks. In *Proc. of the Driving Simulation Conference Europe*, pages 231–242, 2010.

[68] K.-W. Chiang, H.-Y. Pai, J.-C. Zeng, M.-L. Tsai, and N. El-Sheimy. Automated modeling of road networks for high-definition maps in opendrive format using mobile mapping measurements. *Geomatics*, 2(2):221–235, 2022.

[69] H. Shi. Automatic generation of opendrive roads from road measurements. Master's thesis, 2011.

[70] Z. Xinxin, L. Fei, and W. Xiangbin. Csg: Critical scenario generation from real traffic accidents. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1330–1336. IEEE, 2020.

[71] F. Grazioli, E. Kusmenko, A. Roth, B. Rumpe, and M. von Wenckstern. Simulation framework for executing component and connector models of self-driving vehicles. In *MODELS (Satellite Events)*, pages 109–115, 2017.

[72] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W.-C. Ma, and R. Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11167–11176, 2020.

[73] A. M. Rabal, A. Ferrero, J. Campos, J. L. Fontecha, A. Pons, A. M. Rubiño, and A. Corróns. Automatic gonio-spectrophotometer for the absolute measurement of the spectral BRDF at in- out-of-plane and retroreflection geometries. *Metrologia*, 49(3):213–223, feb 2012. URL: `https://doi.org/10.1088/0026-1394/49/3/213`, `doi:10.1088/0026-1394/49/3/213`.

[74] Z. Chen, S. Nobuhara, and K. Nishino. Invertible neural brdf for object inverse rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[75] K. Zhang, F. Luan, Z. Li, and N. Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5565–5574, 2022.

[76] V. Cooper, J. Bieron, and P. Peers. Estimating homogeneous data-driven brdf parameters from a reflectance map under known natural lighting. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[77] S. Muckenhuber, H. Holzer, and Z. Bockaj. Automotive lidar modelling approach based on material properties and lidar capabilities. *Sensors*, 20(11):3309, 2020.

[78] H. G. Seif and X. Hu. Autonomous driving in the icity—hd maps as a key challenge of the automotive industry. *Engineering*, 2(2):159–162, 2016.

[79] J. Fang, D. Zhou, F. Yan, T. Zhao, F. Zhang, Y. Ma, L. Wang, and R. Yang. Augmented lidar simulator for autonomous driving. *IEEE Robotics and Automation Letters*, 5(2):1931–1938, 2020.

[80] N. Jaipuria, X. Zhang, R. Bhasin, M. Arafa, P. Chakravarty, S. Shrivastava, S. Manglani, and V. N. Murali. Deflating dataset bias using synthetic data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 772–773, 2020.

[81] A. Sinha, K. Ayush, J. Song, B. Uzkent, H. Jin, and S. Ermon. Negative data augmentation. *arXiv preprint arXiv:2102.05113*, 2021.

[82] S. O.-R. A. V. S. Committee et al. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE Standard J*, 3016:1–16, 2014.

[83] I. Barabas, A. Todoruţ, N. Cordoş, and A. Molea. Current challenges in autonomous driving. In *IOP conference series: materials science and engineering*, volume 252, page 012096. IOP Publishing, 2017.

[84] B. C. Zanchin, R. Adamshuk, M. M. Santos, and K. S. Collazos. On the instrumentation and classification of autonomous cars. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2631–2636. IEEE, 2017.

[85] I. R. Dariani. State of the art in autonomous driving. 2017.

[86] Y. Emzivat, J. Ibanez-Guzman, P. Martinet, and O. H. Roux. Dynamic driving task fallback for an automated driving system whose ability to monitor the driving environment has been compromised. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1841–1847. IEEE, 2017.

[87] A. Eriksson and N. A. Stanton. Takeover time in highly automated vehicles: noncritical transitions to and from manual control. *Human factors*, 59(4):689–705, 2017.

[88] E. Lorenz. The butterfly effect. *World Scientific Series on Nonlinear Science Series A*, 39:91–94, 2000.

[89] S. Dimce, M. S. Amjad, and F. Dressler. Mmwave on the road: Investigating the weather impact on 60 ghz v2x communication channels. In *2021 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS)*, pages 1–8. IEEE, 2021.

[90] J. Vargas, S. Alsweiss, O. Toker, R. Razdan, and J. Santos. An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors*, 21(16), 2021. URL: `https://www.mdpi.com/1424-8220/21/16/5397`, `doi:10.3390/s21165397`.

[91] Y. Zhang, A. Carballo, H. Yang, and K. Takeda. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:146–177, 2023.

[92] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar. The impact of adverse weather conditions on autonomous vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE vehicular technology magazine*, 14(2):103–111, 2019.

[93] E. Hecht. *Optik*. Walter de Gruyter GmbH & Co KG, 2018.

[94] K. Yoneda, N. Suganuma, R. Yanase, and M. Aldibaja. Automated driving recognition technologies for adverse weather conditions. *IATSS research*, 43(4):253–262, 2019.

[95] X. Li, K.-Y. Lin, M. Meng, X. Li, L. Li, Y. Hong, and J. Chen. Composition and application of current advanced driving assistance system: A review. *arXiv preprint arXiv:2105.12348*, 2021.

[96] K. Min, S. Han, D. Lee, D. Choi, K. Sung, and J. Choi. Sae level 3 autonomous driving technology of the etri. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 464–466. IEEE, 2019.

[97] M. Anderson. The road ahead for self-driving cars: The av industry has had to reset expectations, as it shifts its focus to level 4 autonomy-[news]. *IEEE Spectrum*, 57(5):8–9, 2020.

[98] Audi. Mission accomplished: Audi a7 piloted driving car completes 550-mile automated test drive, 2016. URL: `https://www.audiusa.com/newsroom/news/press-releases/2015/01/550-mile-piloted-drive-from-silicon-valley-to-las-vegas`.

[99] S. O. Wald and F. Weinmann. Ray tracing for range-doppler simulation of 77 ghz automotive scenarios. In *2019 13th European Conference on Antennas and Propagation (EuCAP)*, pages 1–4. IEEE, 2019.

[100] P. Degerman, J. Pohl, and M. Sethson. Ultrasonic sensor modeling for automatic parallel parking systems in passenger cars. Technical report, SAE Technical Paper, 2007.

[101] J. F. Tamm-Morschel. Erweiterung eines phänomenologischen lidar-sensormodells durch identifizierte physikalische effekte. Master's thesis, Technische Universität Darmstadt, 2020.

[102] R. Roriz, J. Cabral, and T. Gomes. Automotive lidar technology: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6282–6297, 2021.

[103] S. Royo and M. Ballesta-Garcia. An overview of lidar imaging systems for autonomous vehicles. *Applied sciences*, 9(19):4093, 2019.

[104] K. Wenzl, H. Ruser, and C. Kargel. Decentralized multi-target-tracking using a lidar sensor network. In *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 2492–2497. IEEE, 2012.

[105] V. Lidar. Hdl-64e. *online],[Retrieved Jul. 15, 2020]. Retrieved from Internet (10pgs)*, 2016.

[106] Y. Li and J. Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020.

[107] N. Heussner, R. Reppich, and A. Frederiksen. Considering the movement of the laser source for classification in iec 60825-1. In *International Laser Safety Conference*, volume 2017, pages 280–284. Laser Institute of America, 2017.

[108] Y. Fan, H. Zhang, J. Yang, P. Wang, W. Zhang, Y. He, X. Tian, T. Fu, G. Zhang, C.-e. Zah, et al. 905nm line-beam laser module with high-reliability for automotive lidar applications. In *High-Power Diode Laser Technology XXI*, volume 12403, pages 216–222. SPIE, 2023.

[109] J. Lambert, A. Carballo, A. M. Cano, P. Narksri, D. Wong, E. Takeuchi, and K. Takeda. Performance analysis of 10 models of 3d lidars for automated driving. *IEEE Access*, 8:131699–131722, 2020.

[110] B. Doherty. Micronotes: Pin diode fundamentals. *Watertown, MA: Microsemi Corp., MicroNote Series*, 701, 1998.

[111] Q. Hao, Y. Tao, J. Cao, and Y. Cheng. Development of pulsed-laser three-dimensional imaging flash lidar using apd arrays. *Microwave and Optical Technology Letters*, 63(10):2492–2509, 2021.

[112] M. A. Albota, B. F. Aull, D. G. Fouche, R. M. Heinrichs, D. G. Kocher, R. M. Marino, J. G. Mooney, N. R. Newbury, M. E. O'Brien, B. E. Player, et al. Three-dimensional imaging laser radars with geiger-mode avalanche photodiode arrays. *Lincoln Laboratory Journal*, 13(2):351–370, 2002.

[113] Y. Wang, P. Tang, L. Shen, and S. Pu. Lidar system using mems scanner-based coaxial optical transceiver. In *2020 IEEE 5th Optoelectronics Global Conference (OGC)*, pages 166–168. IEEE, 2020.

[114] P. Wilhelm, M. Eggert, J. Hornig, and S. Oertel. High spatial and temporal resolution bistatic wind lidar. *Applied Sciences*, 11(16):7602, 2021.

[115] M. S. Awan, L. Csurgai-Horváth, S. S. Muhammad, E. Leitgeb, F. Nadeem, and M. S. Khan. Characterization of fog and snow attenuations for free-space optical propagation. *J. Commun.*, 4(8):533–545, 2009.

[116] M.-H. Le, C.-H. Cheng, D.-G. Liu, and T.-T. Nguyen. An adaptive group of density outlier removal filter: Snow particle removal from lidar data. *Electronics*, 11(19):2993, 2022.

[117] N. Charron, S. Phillips, and S. L. Waslander. De-noising of lidar point clouds corrupted by snowfall. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 254–261. IEEE, 2018.

[118] S. Gnecchi, C. Barry, S. Bellis, S. Buckley, and C. Jackson. Long distance ranging performance of gen3 lidar imaging system based on 1× 16 sipm array. In *Proceedings of the International Image Sensors Society (IISS) Workshop, Snowbird, UT, USA*, pages 23–27, 2019.

[119] C. Linnhoff, K. Hofrichter, L. Elster, P. Rosenberger, and H. Winner. Measuring the influence of environmental conditions on automotive lidar sensors. *Sensors*, 22(14):5266, 2022.

[120] L. Carrara and A. Fiergolski. An optical interference suppression scheme for tcspc flash lidar imagers. *Applied Sciences*, 9(11):2206, 2019.

[121] E. G. Hoare and R. Hill. System requirements for automotive radar antennas. In *IEE Colloquium on Antennas for Automotives (Ref. No. 2000/002)*, pages 1–1. IET, 2000.

[122] F. Verbiest, M. Proesmans, and L. V. Gool. Modeling the effects of windshield refraction for camera calibration. In *European Conference on Computer Vision*, pages 397–412. Springer, 2020.

[123] M. Trierweiler, T. Peterseim, and C. Neumann. Automotive lidar pollution detection system based on total internal reflection techniques. In *Light-Emitting Devices, Materials, and Applications XXIV*, volume 11302, pages 135–144. SPIE, 2020.

[124] M. Kutila, P. Pyykönen, M. Jokela, T. Gruber, M. Bijelic, and W. Ritter. Benchmarking automotive lidar performance in arctic conditions. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[125] M. Lewis and J. Jacobson. Game engines. *Communications of the ACM*, 45(1):27, 2002.

[126] H. Schoenau-Fog and B. A. Larsen. Creating interactive adaptive real time story worlds. In *International Conference on Interactive Digital Storytelling*, pages 548–551. Springer, 2018.

[127] J. Moloney and R. Amor. Stringcve: Advances in a game engine-based collaborative virtual environment for architectural design. In *Proceedings of CONVR 2003 conference on construction applications of virtual reality*, pages 156–168. Citeseer, 2003.

[128] P. Ekströmer, R. Wever, P. Andersson, and J. Jönsson. Shedding light on game engines and virtual reality for design ideation. In *Proceedings of the Design Society: International Conference on Engineering Design*, volume 1, pages 2003–2010. Cambridge University Press, 2019.

[129] J. Linietsky and A. Manzur. online, 2014. URL: `https://godotengine.org/`.

[130] A. games. Open 3d engine. online, 2021. URL: `https://www.o3de.org/`.

[131] Blender-Foundation. Cycles: Open source production rendering. URL: `https://www.cycles-renderer.org/development/`.

[132] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation.* Morgan Kaufmann, 2016.

[133] B. Burley and W. D. A. Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, volume 2012, pages 1–7. vol. 2012, 2012.

[134] B. . Manual. gltf 2.0. online, June 2022. URL: `https://docs.blender.org/manual/en/latest/addons/import_export/scene_gltf2.html`.

[135] A. E. Johnson, A. R. Klumpp, J. B. Collier, and A. A. Wolf. Lidar-based hazard avoidance for safe landing on mars. *Journal of guidance, control, and dynamics*, 25(6):1091–1099, 2002.

[136] I. Millington. *Game physics engine development: how to build a robust commercial-grade physics engine for your game.* CRC Press, 2010.

[137] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. `doi:10.1109/IROS.2012.6386109`.

[138] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. URL: `https://pybullet.org/wordpress/`.

[139] R. L. Smith. Open dynamics engine. URL: `http://ode.org/wiki/index.php/Main_Page`.

[140] R. Bridson. *Fluid simulation for computer graphics*. AK Peters/CRC Press, 2015.

[141] C. Braley and A. Sandu. Fluid simulation for computer graphics: A tutorial in grid based and particle based methods. *Virginia Tech, Blacksburg*, 2010.

[142] T. Thuerey, N.; Pfaff. Mantaflow, 2018. URL: `http://mantaflow.com`.

[143] F. Hou, Y. Zhang, S. Wang, Z. Shen, P. Mao, and X. Qu. Influencing factors of the length of lane-changing buffer zone for autonomous driving dedicated lanes. *Applied Sciences*, 12(10):4923, 2022.

[144] S. Duleba, T. Tettamanti, Á. Nyerges, and Z. Szalay. Ranking the key areas for autonomous proving ground development using pareto analytic hierarchy process. *IEEE Access*, 9:51214–51230, 2021.

[145] R. Chen, M. Arief, W. Zhang, and D. Zhao. How to evaluate proving grounds for self-driving? a quantitative approach. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5737–5748, 2020.

[146] Z. Szalay. Structure and architecture problems of autonomous road vehicle testing and validation. In *15th Mini Conference on Vehicle System Dynamics, Identification and Anomalies-VSDIA*, pages 229–236, 2016.

[147] A. Diewald, C. Kurz, P. V. Kannan, M. Gießler, M. Pauli, B. Göttel, T. Kayser, F. Gauterin, and T. Zwick. Radar target simulation for vehicle-in-the-loop testing. *Vehicles*, 3(2):257–271, 2021.

[148] S. Grollius, M. Ligges, J. Ruskowski, and A. Grabmaier. Concept of an automotive lidar target simulator for direct time-of-flight lidar. *IEEE Transactions on Intelligent Vehicles*, 2021.

[149] N. Kalra. *Challenges and approaches to realizing autonomous vehicle safety*. RAND Santa Monica, 2017.

[150] A. Stocco and P. Tonella. Towards anomaly detectors that learn continuously. In *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 201–208. IEEE, 2020.

[151] M. Schwall, T. Daniel, T. Victor, F. Favaro, and H. Hohnhold. Waymo public road safety performance data. *arXiv preprint arXiv:2011.00038*, 2020.

[152] L. Kang, W. Zhao, B. Qi, and S. Banerjee. Augmenting self-driving with remote control: Challenges and directions. In *Proceedings of the 19th international workshop on mobile computing systems & applications*, pages 19–24, 2018.

[153] R.-T. Juang. The implementation of remote monitoring for autonomous driving. In *2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 53–56. IEEE, 2019.

[154] R. Kummerle, D. Hahnel, D. Dolgov, S. Thrun, and W. Burgard. Autonomous driving in a multi-level parking structure. In *2009 IEEE International Conference on Robotics and Automation*, pages 3395–3400. IEEE, 2009.

[155] A. Ibisch, S. Stümper, H. Altinger, M. Neuhausen, M. Tschentscher, M. Schlipsing, J. Salinen, and A. Knoll. Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded lidar sensors. In *2013 IEEE intelligent vehicles symposium (IV)*, pages 829–834. IEEE, 2013.

[156] J. Font and A. Brown. Investigating the effects of roadside cover on safe speeds for autonomous driving in high-risk deer-vehicle collision areas. *Advances in transportation studies*, (SI 2):97–112, 2020.

[157] Cariad. Fep 3 sdk, September 2021. URL: `https://github.com/cariad-tech/fep3_sdk`.

[158] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[159] M. Krammer, M. Benedikt, T. Blochwitz, K. Alekeish, N. Amringer, C. Kater, S. Materne, R. Ruvalcaba, K. Schuch, J. Zehetner, et al. The distributed co-simulation protocol for the integration of real-time systems and simulation environments. In *Proceedings of the 50th Computer Simulation Conference*, pages 1–14, 2018.

[160] V. M. Vilches, A. H. Cordero, A. B. Calvo, I. Z. Ugarte, and R. Kojcev. Robot_gym: accelerated robot training through simulation in the cloud with ros and gazebo. *arXiv preprint arXiv:1808.10369*, 2018.

[161] M. M. Madden and P. C. Glaab. Distributed simulation using dds and cloud computing. In *Proceedings of the 50th Annual Simulation Symposium*, pages 1–12, 2017.

[162] E. Grilli, F. Menna, and F. Remondino. A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:339, 2017.

[163] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3752–3761, 2018.

[164] J. Kocić, N. Jovičić, and V. Drndarević. Sensors and sensor fusion in autonomous vehicles. In *2018 26th Telecommunications Forum (TELFOR)*, pages 420–425. IEEE, 2018.

[165] J. R. Vargas Rivero, T. Gerbich, B. Buschardt, and J. Chen. Data augmentation of automotive lidar point clouds under adverse weather situations. *Sensors*, 21(13):4503, 2021.

[166] A. Von Bernuth, G. Volk, and O. Bringmann. Simulating photo-realistic snow and fog on existing images for enhanced cnn training and evaluation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 41–46. IEEE, 2019.

[167] A. Das. Soildnet: Soiling degradation detection in autonomous driving. *arXiv preprint arXiv:1911.01054*, 2019.

[168] M. Uřičář, P. Křížek, G. Sistu, and S. Yogamani. Soilingnet: Soiling detection on automotive surround-view cameras. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 67–72. IEEE, 2019.

[169] A. Von Bernuth, G. Volk, and O. Bringmann. Rendering physically correct raindrops on windshields for robustness verification of camera-based object recognition. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 922–927. IEEE, 2018.

[170] N. Chen, R. Gourova, O. Krasnov, and A. Yarovoy. The influence of the water-covered dielectric radome on 77ghz automotive radar signals. In *2017 European Radar Conference (EURAD)*, pages 139–142. IEEE, 2017.

[171] M. Kurri and A. Huuskonen. Measurements of the transmission loss of a radome at different rain intensities. *Journal of Atmospheric and Oceanic Technology*, 25(9):1590–1599, 2008.

[172] F. Norouzian, R. Du, M. Gashinova, E. Hoare, C. Constantinou, M. Lancaster, P. Gardner, and M. Cherniakov. Signal reduction due to radome contamination in low-thz automotive radar. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–4. IEEE, 2016.

[173] A. Arage Hassen. *Indicators for the Signal Degradation and Optimization of Automotive Radar Sensors Under Adverse Weather Conditions*. PhD thesis, Technische Universität, 2007.

[174] N. Wang and B. Wade. Rendering falling rain and snow. In *ACM SIGGRAPH 2004 Sketches*, page 14. 2004.

[175] R. Gourova, O. Krasnov, and A. Yarovoy. Analysis of rain clutter detections in commercial 77 ghz automotive radar. In *2017 European Radar Conference (EURAD)*, pages 25–28. IEEE, 2017.

[176] F. Norouzian, E. Marchetti, E. Hoare, M. Gashinova, C. Constantinou, P. Gardner, and M. Cherniakov. Experimental study on low-thz automotive radar signal attenuation during snowfall. *IET Radar, Sonar & Navigation*, 13(9):1421–1427, 2019.

[177] C. Schüßler, M. Hoffmann, J. Bräunig, I. Ullmann, R. Ebelt, and M. Vossiek. A realistic radar ray tracing simulator for large mimo-arrays in automotive environments. *IEEE Journal of Microwaves*, 1(4):962–974, 2021.

[178] S. Walz, M. Bijelic, F. Kraus, W. Ritter, M. Simon, and I. Doric. A benchmark for spray from nearby cutting vehicles. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 188–195. IEEE, 2021.

[179] A. Von Bernuth, G. Volk, and O. Bringmann. Augmenting image data sets with water spray caused by vehicles on wet roads. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3055–3060. IEEE, 2021.

[180] Y. Zhang, A. Carballo, H. Yang, and K. Takeda. Autonomous driving in adverse weather conditions: A survey. *arXiv preprint arXiv:2112.08936*, 2021.

[181] M. Mohan and S. Payra. Aerosol number concentrations and visibility during dense fog over a subtropical urban site. *Journal of Nanomaterials*, 2014, 2014.

[182] F. Toja-Silva, C. Pregel-Hoderlein, and J. Chen. On the urban geometry generalization for cfd simulation of gas dispersion from chimneys: Comparison with gaussian plume model. *Journal of Wind Engineering and Industrial Aerodynamics*, 177:1–18, 2018.

[183] F. Toja-Silva, T. Kono, C. Peralta, O. Lopez-Garcia, and J. Chen. A review of computational fluid dynamics (cfd) simulations of the wind flow around buildings for urban wind energy exploitation. *Journal of Wind Engineering and Industrial Aerodynamics*, 180:66–87, 2018.

[184] F. Toja-Silva, J. Chen, S. Hachinger, and F. Hase. Cfd simulation of co2 dispersion from urban thermal power plant: Analysis of turbulent schmidt number and comparison with gaussian plume model and measurements. *Journal of Wind Engineering and Industrial Aerodynamics*, 169:177–193, 2017.

[185] A. Braud, G. Fromentoux, B. Radier, and O. Le Grand. The road to european digital sovereignty with gaia-x and idsa. *IEEE Network*, 35(2):4–5, 2021.

Publications

Journal

1. Weather Classification Using an Automotive LIDAR Sensor Based on Detections on Asphalt and Atmosphere. Vargas Rivero, Jose Roberto and Gerbich, Thiemo and Teiluf, Valentina and Buschardt, Boris and Chen, Jia. Sensors, vol. 20, Art. no. 15, 2020.

2. The Effect of Spray Water on an Automotive LIDAR Sensor: A Real-Time Simulation Study. Vargas Rivero, Jose Roberto and Gerbich, Thiemo and Buschardt, Boris and Chen, Jia. IEEE Transactions on Intelligent Vehicles, 2021.

3. Data Augmentation of Automotive LIDAR Point Clouds under Adverse Weather Situations. Vargas Rivero, Jose Roberto and Gerbich, Thiemo and Buschardt, Boris and Chen, Jia. Sensors, vol. 21, Art. no. 13, 2021.

Conference

1. Characterization and simulation of the effect of road dirt on the performance of a laser scanner. Vargas Rivero, Jose Roberto and Tahiraj, Ilir and Schubert, Olaf and Glassl, Christoph and Buschardt, Boris and Berk, Mario and Chen, Jia, in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6.

Supervised Works

1. Hongquan Wang: Simulation of light scattering by spray water from vehicles (Sep.2017 - Apr.2018)

Attachments

The following papers are attached to this document:

1. Characterization and simulation of the effect of road dirt on the performance of a laser scanner © [2017] IEEE. Reprinted, with permission, from [Rivero, Jose Roberto Vargas and Tahiraj, Ilir and Schubert, Olaf and Glassl, Christoph and Buschardt, Boris and Berk, Mario and Chen, Jia], Characterization and simulation of the effect of road dirt on the performance of a laser scanner, march/2018]. The author confirms that the accepted version of the paper is used.

2. Weather Classification Using an Automotive LIDAR Sensor Based on Detections on Asphalt and Atmosphere. Vargas Rivero, Jose Roberto and Gerbich, Thiemo and Teiluf, Valentina and Buschardt, Boris and Chen, Jia. Sensors, vol. 20, Art. no. 15, 2020. Published under the license CC BY 4.0.

3. The Effect of Spray Water on an Automotive LIDAR Sensor: A Real-Time Simulation Study © [2021] IEEE. Reprinted, with permission, from [Vargas rivero, Jose Roberto and Gerbich, Thiemo and Buschardt, Boris and Chen, Jia], The Effect of Spray Water on an Automotive LIDAR Sensor: A Real-Time Simulation Study, march/2021]. The author confirms that the accepted version of the paper is used.

4. Data Augmentation of Automotive LIDAR Point Clouds under Adverse Weather Situations. Vargas Rivero, Jose Roberto and Gerbich, Thiemo and Buschardt, Boris and Chen, Jia. Sensors, vol. 21, Art. no. 13, 2021. Published under the license CC BY 4.0.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Technical University of Munich's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` to learn how to obtain a License from RightsLink.

Characterization and Simulation of the Effect of Road Dirt on the Performance of a Laser Scanner

Jose Roberto Vargas Rivero, Ilir Tahiraj, Olaf Schubert, Christoph Glassl, Boris Buschardt
Development Radar/Laser Sensors Automated Driving
Audi AG
Ingolstadt, Germany

[1]Mario Berk, [2]Jia Chen
[1]Engineering Risk Analysis Group
[2]Department of Electrical and Computer Engineering
Technische Universität München
Munich, Germany

***Abstract*—Automotive environment sensors such as LIDARs (light detection and ranging) are the backbone of automated driving. It has to be made sure that the performance of these sensors is sufficient, even under adverse environment influences like accumulated dirt on a sensor cover. This paper describes the collection and analysis of real world road dirt samples, accumulated on the plastic cover of a LIDAR sensor. The effect of the dirt on the sensor performance is experimentally quantified by measuring the transmission and reflection properties of the dirty sensor plastic cover. Moreover, an analysis of the sensor's uncertainty in raw data position measurements is presented. Different alternatives to include these effects in a virtual simulation of the sensor are discussed. While the sample size of the dirty plastic covers is too small to representatively describe the effect of dirt on LIDAR performance, this study shows that an abstract effect on sensor performance such as dirt can be described, quantified and eventually be represented in a virtual simulation.**

Keywords—LIDAR; Laser scanner; dirt; sensor simulation; automated driving vehicle.

I. Introduction

Highly automated driving is a development goal of almost every major car manufacturer and highly automated concept cars have already been realized in different projects [1-5]. While these projects show the technical feasibility of highly automated driving, its safety for real traffic scenarios still has to be demonstrated. One important aspect for safe automated driving is a detailed and reliable knowledge of a vehicle's environment. This knowledge is provided by sensors such as radar, lidar and camera. A major challenge for the reliable environment perception is however the influence of environmental conditions such rainfall and dirt on the sensor cover. It has to be validated that the environment perception of the sensors is reliable despite these influences. Statistical validation of an automated driving vehicle's safety would require extensive testing as it is argued that a highly automated car should have a smaller accident probability than an average human driver [6]. Generally this also holds for a safety validation of the environment sensing [7], making field operational testing through real-world driving as a main test method economically unviable. Therefore, virtual simulation and laboratory tests are put forward as feasible alternatives in the validation process of highly automated cars. In a virtual world, it is possible to artificially generate a large number of permutations of the situations in which a wrong behavior of a function or sensor under test could occur. The main challenge with virtual simulations however is that the objects and the sensors in the simulation need to behave in a reproducible and realistic way. Only then, the required approval process for a certain function can be based on virtual testing.

The purpose of this study is to examine the effect of dirt on the performance of a LIDAR with the aim of including this effect in a virtual simulation. As no cleaning mechanism is able to achieve a perfect cleaning under all conditions and the amount of cleaning water is limited, the characterization of the dirt effect is an important step towards a realistic virtual simulation. We collect samples of dirty LIDAR sensor plastic covers under real world driving conditions and analyze the covers' transmission and reflection properties. Further, the effect of dirt on the sensor's position measurement accuracy in terms of the position standard deviation is analysed. This study is structured as follows: First, the problem of validating the safety of an automotive LIDAR sensor in the context of automated driving is further discussed in section II. In section III, the transmission and reflection measurement method is outlined with a presentation of the measurement results. The effect of dirt on a LIDAR's position measurement accuracy is analyzed and experiments are conducted to indentify potential forward scattering caused by the dirt. In section IV it is discussed how the evaluated influence of dirt on sensor performance can be incorporated in a virtual simulation. Conclusions are provided in section V.

II. Background: Safety Validation of Automotive LIDAR Sensors

Based on the automated driving function specifications and existing car safety regulations, a number of requirements have

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

to be met by the implemented sensor set. Some of these requirements are for example the use of redundant components, different measurement principles, and ensuring that even under adverse conditions a safe state is reached [8] which all help to achieve reliable environment sensing. To fulfill the environment perception requirements for highly automated driving (SAE level 3 and BASt level 3 [9]) such as sensor redundancy with independent physical measurement methods, range and resolution, LIDAR sensors are being developed to complement the use of the already common camera and radar sensors. On the one hand, a LIDAR has advantages compared to a radar regarding resolution and compared to a camera due to its direct distance measurements. On the other hand, as a disadvantage, a LIDAR in comparison with a radar is more sensitive to weather phenomena in terms of its environment perception quality [10]. The laser light of a LIDAR, which is in the near infrared spectrum, is easily absorbed and scattered by particles in the atmosphere or dirt on the plastic cover of the sensor because of its short wavelength. The different factors that influence the propagation of a LIDAR's Laser beam through scattering and attenuation are illustrated in Fig. 1. As is illustrated, attenuation and scattering occur on the plastic cover, on particles in the atmosphere (rain, snow, fog…) and on the objects that are to be detected.

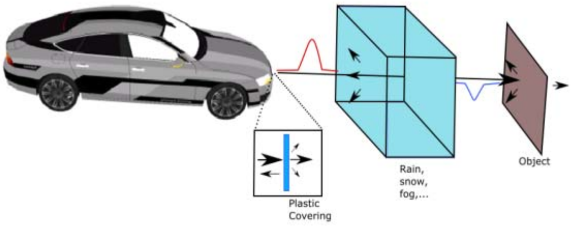


Fig. 1. Factors that cause scattering/ attenuation of the laser rays.

The mean attenuation of gases, haze, fog, rain, snow and dust for a certain (laser) wavelength and the increment of ground level noise in the laser scanner receiver due to solar radiation have been studied and simulated in different publications [11-14]. These effects are usually described in terms of their mean values, can be predicted and partially corrected by measures implemented during the sensor design phase. In contrast to existing studies, we are not interested in the atmospheric attenuation but in the unwanted attenuation and scattering of the laser rays on the sensor plastic cover due to dirt. Also, from the perspective of a safety validation, we are interested not in the mean values of those effects but rather in the fluctuations and safety relevant extreme values which can occur.

Due to the dirt cover, the visibility and range of a sensor can degrade considerably. Moreover, the accuracy (e.g. the standard deviation of the distance measurements) might decrease. The performance degradation due to dirt correlates well with the presence of rain which facilitates the dirt accumulation through spray water but it depends also on the conditions of the specific street, the time of the year, the speed of the car, the shape, temperature and material of the sensor and the environment temperature.

It is pointed out that not only the sensor range is reduced by dirt but also the form of the field of view might be affected. For example, dirt can scatter and reflect the light to regions which previously were not visible. This effect is shown in Fig. 2. The blue area represents the original field of view. Under the influence of dirt, the laser beams are scattered, changing the original field of view (blue) to the field of view under dirty conditions (red). This effect is physically caused by forward scattering by the dirt particles.

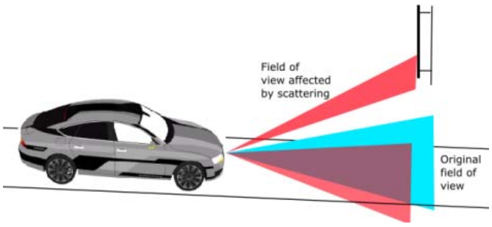


Fig. 2. Possible effect of forward scattering caused by some kinds of dirt. The original field of view (blue area) is altered by dirt on the sensor to the red area.

To summarize, there are four main ways in which dirt can affect the perception performance of a sensor:

-Reducing the range.

-Partly blinding a sensor, leading to decreased detection capabilities.

-Changing the field of view.

-Increasing the uncertainty of physical measurement quantities such as position measurements.

To analyze the presented effects of dirt on LIDAR performance, real world samples of dirty plastic covers are collected which is explained in the next section.

III. Analizing the Effect of Dirt on Automotive LIDAR Sensing Performance

In this section we first describe the experimental setup to collect dirty sensor cover samples and to measure the transmission and reflection properties of these covers. Further, the effect of dirt on the sensor performance is analyzed.

A. *Experiment setup and data collection: Transmission and reflection of dirty sensor covers*

It is very difficult to realistically replicate the varying types and degrees of dirt accumulation on the sensors in a labor environment as the environment conditions and processes that cause dirty sensor covers are highly variable. Therefore, we conducted experiments in field tests under real-world conditions. For this purpose, two sensor plastic covers were placed on the grill of a test car as shown in Fig. 3.

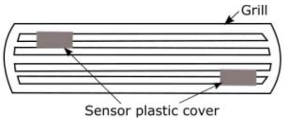


Fig. 3. Position of the plastic covers on the radiator grill.

The driving route was chosen in a way that high levels of dirt were reached with a total travelling time of approximately 2 days for each plastic cover. In total 12 samples (i.e. dirty sensor plastic covers) are obtained of which 6 correspond to the upper position (≈ 72 cm over the floor, 22 cm to the left) and 6 to the lower position (≈ 41 cm over the floor, 29 cm to the left). In the following, the samples are named using a number 1-6 and a letter u (upper position) and l (lower position).

Fig. 4 exemplarily illustrates two of the sensor plastic cover samples with the accumulated dirt. One mechanism of dirt accumulation consists of dust adhering to the plastic cover in combination with water (e.g. dirty spray water) which later evaporates leaving dust and dirt chunks. Alternatively a fine layer of dust is formed on the plastic cover on which bigger dust particles adhere.

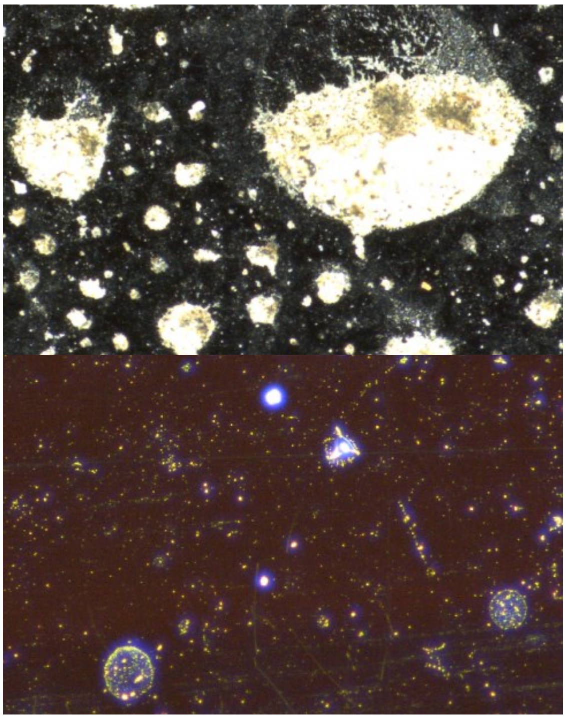


Fig. 4. Microscope pictures of the accumulated dirt on the sensor plastic covers. The top picture shows big chunks of dust formed due to the combination of fine dust particles with water (sample 5l with 2 times magnification). The bottom picture shows a relativelly clean cover where micro scratches can be seen (sample 1l with 6 times magnification).

To study the effect of the dirt on LIDAR performance with the aim of implementing it in a virtual simulation, Transmission T [%] and reflection R [%] measurements were conducted with the collected samples. Fig. 5 and Fig. 6 show the setups for the transmission and reflection measurements, respectively. As is visible in Fig. 5, a robotic arm is used to place the sample close and parallel to an integrating sphere. The measurements were conducted in the near infrared region corresponding to the wavelength of the LIDAR under test. A total of 72 measurement values (4 layers of 18 measurement points on a plastic cover) of transmission are obtained for each dirty plastic cover sample. The spacing between the measurement points was selected such that the central region of the plastic cover had a smaller distance between two measurement points (see also Fig. 7).

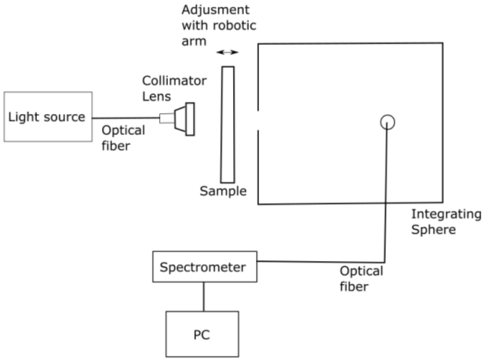


Fig. 5. Transmission measurement setup for the dirty sensor covers.

Similarly, for the reflection measurement in Fig. 6 a robotic arm was used to place the sample close and parallel to the homogenizing rod respectively. In contrast to the transmission measurements only one reflection measurement value is collected due to the big size of the homogenizing rod in comparison to the plastic cover.

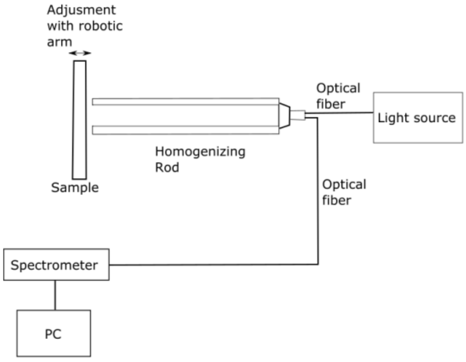


Fig. 6. Reflection measurement setup for the dirty sensor covers.

In principle, information on either transmission or reflection is enough for a virtual simulation. The advantage of measuring the reflection is that the plastic cover in a real sensor wouldn't have to be removed in order to perform the measurement. Here however, compared to the reflection measurements, the transmission measurements are more accurate because of the small diameter of the beam from the collimator lens (the beam diameter is comparable with the real laser source). The spatial distribution of the dirt is with the given experiment design better accounted for by the transmission measurement.

Additionally, a slightly modified version of the transmission measurement (Fig. 5) is performed to estimate the amount of forward scattering that each dirt samples causes. For this measurement series, the overall distance between the collimator lens and the integrating sphere is increased. The robot moves the dirty plastic cover sample between the collimator lens and the integrating sphere. This procedure leads to different measured intensities for each position of the sample. The resulting curves of transmissions (indirectly the intensities) over position can then be used to estimate the forward scattering that the dirt generates.

In another experiment, we study the influence of dirt on the uncertainty of the LIDAR's position measurements. For this purpose, a test target with reflectivity of around 50% was located at a distance of 19 m from the LIDAR sensor. Measurements were conducted for 30 s utilizing all the dirty plastic covers, i.e. the covering plastic of the sensor under test was for each measurement replaced with that corresponding to the different dirt samples. For each plastic cover, a 30 s time series of data was collected which contains the point cloud corresponding to the test object. The position measurement variability of the points corresponding to the test object are analyzed and expressed in terms of their standard deviation.

B. Results: Transmission and reflection measurements

The reflection measurements reveal a ripple in the spectrum which is due to a thin film coating of the plastic cover. This effect can be characterized by the thin film equation with the approximation that the substrate is infinite [15]:

$$d = \frac{1}{4(\nu_{j+1} - \nu_j)\sqrt{n^2 - sin^2\,\varphi}} \qquad (1)$$

Where d is the thin film thickness, ν_j and ν_{j+1} are the wavenumbers of two consecutive extremes (peak to valley) in the ripple profile, n corresponds to the refractive index of the coating and φ is the angle of incidence of the beam. With Eq. (1) it is checked if the ripple pattern in the measurements are due to the thin film. Normal incidence is assumed ($\varphi = 0$) although in reality the rays will be distributed in a cone given by the radius of the homogenizing rod. With $\nu_{j+1} = 1/887$ nm , $\nu_j = 1/899$ nm and $n \approx 1.45$ Eq (1) coincides with the expected thickness d of the thin film. This interference pattern is a limitation of the reflection measurement because in principle, each plastic cover can have a different reflection value at a certain wavelength depending on how this pattern is distributed. Therefore, it is difficult to select a representative reflection value. In the transmission measurement this effect is not present because the amount of light which reaches the receiver is much bigger than the 'noise' generated by the interference pattern. This means that the measured transmission information is here preferred for a virtual simulation.

Fig. 7 exemplarily shows the distribution of the measured transmission values on a part of one of the plastic cover samples. It can be seen that measured transmission values for this particular sample vary between 64.0% and 77.5%.

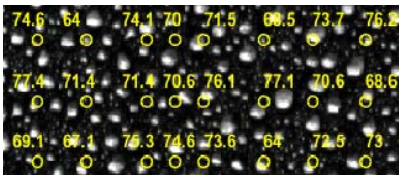


Fig. 7. Exemplary results of a transmission measurement (near infrared) for one dirty plastic cover samples.

Fig. 8 exemplarily illustrates the distribution of the measured transmission values for the samples 1u, 1l, 3u, 3l, 5u and 5l. The variability of the transmission for one specific plastic cover is due to the spatially non-uniform distribution and composition of the dirt on the cover.

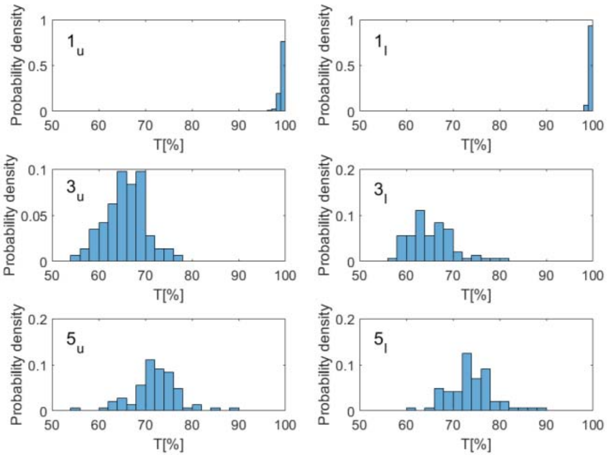


Fig. 8. Histogram showing the probability densitiy of the measured transmissions for the samples 1u,1l,3u,3l,5u and 5l.

It is seen that for samples 1u and 1l, the most likely transmission is close to 100% with the smallest measured values around 98%. The variability here might thus be neglected. From samples 3u, 3l, 5u and 5l it is however clear that this does not hold in general. For these samples the transmission is substantially smaller and also the variability is large. This (spatial) variability should in a simulation be taken into account with adequate statistical models and methods.

The reflection $\bar{R}$ and transmission $\bar{T}$ measurements (averaged over the whole area of the sensor plastic cover) as well as the relative decrease or increase $|\Delta T\ |$ and $|\Delta R|$ in transmission and reflection are presented in Table I. For the transmission and reflection values, a clean plastic cover is the reference.

TABLE I. COMPARISON: TRANSMISSION AND REFLECTION MEASUREMENTS.

Sample number	$\bar{T}$ [%]	$\bar{R}$ [%]	$\|\Delta T\ \|$ [%]	$\|\Delta R\|$ [%]
Reference	100	100%	0	0
1u	99.5	101.5	0.5	1.5
1l	99.8	98.5	0.3	1.5
2u	95.8	91.5	4.3	8.5
2l	94.5	89.4	5.6	10.6
3u	65.8	65.7	34.2	34.3
3l	65.4	62.5	34.6	37.5
4u	96.1	94.1	3.9	5.9
4l	94.1	98.8	6.0	1.2
5u	72.0	92.1	28.0	7.9
5l	73.9	92.9	26.1	7.1
6u	95.4	92.6	4.6	7.4
6l	95.6	89.5	4.4	10.5

It is assumed that the absorption remains constant during the measurement and that it can be neglected. Therefore, it is expected that the relative change in transmission equals that of reflection. In Table I it can however be seen that the relative change in reflection $|\Delta R|$ for almost all samples exceeds the relative change in transmission $|\Delta T\ |$. This is due to the much bigger beam diameter of the homogenizing rod compared with the collimator lens (see Fig. 5 and Fig. 6). The different beam diameters and consequently spot areas on the plastic covers of the transmission and reflection measurements make it difficult to compare both values. Due to the before mentioned problem with the ripple pattern in the reflection measurements and the here found discrepancy between the relative change in transmission and reflection, it is concluded that the transmission measurements are here preferable for a virtual simulation. Improvements to the measurement setup might allow to get better reflection information.

For most samples the averaged transmission is bigger than 90% (see Table I) which indicates that the LIDAR performance will only slightly be influenced by dirt. However, for four of the samples, the transmission is between 60-75% which means the performance such as detection capability of the LIDAR might be severely impaired. It can also be noticed that the difference in the mean transmission $\bar{T}$ for the upper and lower position on the grill (l and u) is negligible whereas the difference between the samples (i.e. 3 and 6) seems to matter more. From this observation it is concluded that the composition and type of dirt has a larger influence on the transmission than the position of the sensor on the grill.

C. *Results: Forward scattering due to dirt on sensor cover*

A first qualitative evaluation of forward scattering (see Fig. 2 for an illustration of this effect) for each sample of dirty plastic covers can be obtained by measuring the transmission over an increasing distance between the lens and the integrating sphere. If the difference in transmission between dirty and clean plastic cover changes with increasing distance to the integrating sphere, then some form of forward scattering occurs due to dirt. Fig. 9 shows the difference in transmission (substracting the transmission corresponding to a clean plastic cover from the measured transmission) at different distances of the sample plastic cover from the integrating sphere (see Fig. 5).

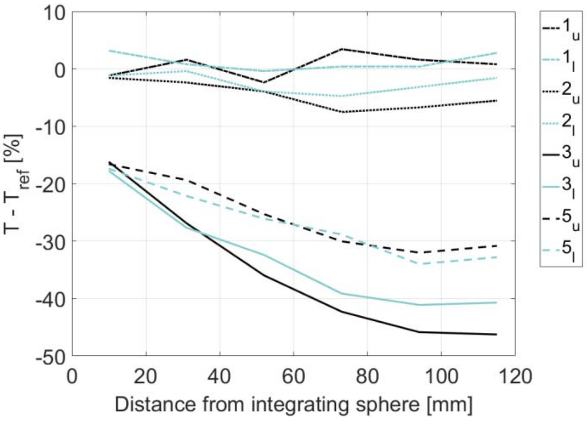


Fig. 9. Difference in transmission of the sample and clean plastic covers at varying distances from the integrating sphere. A constant difference in transmission indicates no forward scattering due to dirt.

As is visible in Fig. 9, samples 3 and 5 generate a larger forward scattering of the light beam in comparison with the samples 1 and 2. Not illustrated in the figure are the samples 4 and 6 which show similar behavior as samples 1 and 2. This is in accordance with Table I where samples 3 and 5 show the strongest influence on the transmission and the position standard deviation.

D. *Results: Effect of dirt on sensor position measurement performance*

The last sections quantify the effect of dirt on the beam propagation through the plastic cover in terms of transmission and reflection. The reduction of transmission and the increase in reflection leads to a decreased sensor performance which is here analyzed in terms of position measurement uncertainties in the raw data, i.e. the point cloud. Table II presents the relative difference between the standard deviation of the position measurements for each dirt sample with respect to the clean plastic cover.

TABLE II. RESULTS: INFLUENCE OF DIRT ON THE POSITION MEASUREMENT STANDARD DEVIATION RELATIVE TO A CLEAN SENSOR.

Sample number	**Conditions**	***Dirt Description***	$(\sigma - \sigma_r)/\sigma_r$ **[%]**
Reference	Laboratory	Clean	0
1u	Dry	Very light	26.7
1l	Dry	Very light	-2.1
2u	Wet, Sleet	Light	17.6
2l	Wet, Sleet	Light	17.1
3u	Changeable (Sun/Rain)	Heavy, fine	72.2
3l	Changeable (Sun/Rain)	Heavy, fine	81.8
4u	Changeable (Sun/Rain)	Medium	59.4
4l	Changeable (Sun/Rain)	Medium	55.1
5u	Changeable (Sun/Rain)	Heavy, coarse	65.8
5l	Changeable (Sun/Rain)	Heavy, coarse	39.6
6u	Changeable (Wet/Dry)	Light	78.6
6l	Changeable (Wet/Dry)	Light	55.1

It can be seen that the sample 3, which also corresponds to the highest reduction in transmission (see Table I), leads to the largest increase in distance measurement uncertainty. However, also samples 4 and 6 show a large increase in the uncertainty, even though the mean transmission for these samples is still around 95 %. This means that the mean transmission itself is not sufficient to describe the effect of the dirt on the LIDAR performance. Instead, the whole transmission matrix including scattering and electro-mechanical effects need to be integrated into a virtual simulation in order to better account for the statistical variation. To account for additional variation statistical methods can be used. It is pointed out that the absolute values of the standard deviation underlying Table II are low even for the dirty plastic covers.

IV. Integrating the results in a simulation

The previous section quantified the effects of dirt on LIDAR performance. This section discusses how to incorporate these findings into virtual simulations to reproduce the dirt effects.

The (potential) reduction of the range – which was not evaluated here in detail – is caused by an increased absorption and scattering and should according to the Lidar equation [10] be similar to the relative change in transmission squared (considering two way losses). Therefore with the transmission measurements, one is able to account for the range reduction. Using the measured values from each plastic cover sample, it is possible to calculate the new range values for the whole field of view of the sensor by multiplying the transmission matrix – a matrix with values between 0 to 1 representing the transmission properties at each sampled location on the plastic cover – with the previous form of the field of view (corresponding to a clean sensor). To account for the spatial variability illustrated in Fig. 7 and summarized in Fig. 8, one could model the spatial distribution of the transmission values on a plastic cover with a random field. As is apparent from the different samples shown in Fig. 8, the distribution parameters of the random field (i.e. the mean and standard deviation) itself are subject to uncertainties and should be modeled for instance with a hierarchical statistical model. The application of such models to performance analysis of automotive environment sensing is discussed in [16].

One possibility to integrate the forward scattering effect (Fig. 9) into a real time simulation is using a virtual material. An example is Nvidia MDL[17] which is a render independent material definition language that can be used to define a level of diffuse, specular reflection and transmission of a layer located in front of a ray tracer in a way that the rays that hit this layer are reflected, transmitted, absorbed or scattered as occurs with a real type of dirt. An alternative to calibrate the parameters of the simulation is to reconstruct the measurement shown in Fig. 5 using a render like OptiX [17] to simulate the beams (as spectral light source) and integrate the intensity on the receiver, simulating the working principle of the spectrometer for the required wavelength.

To account for increased measurement uncertainties (as illustrated in Table II) in a simulation, one could utilize a random generator based on Monte Carlo simulation. This is equivalent to adding noise to the values of the corresponding 3D coordinate of the points in the cloud of a virtual simulation. A similar approach can be followed for the detection capabilities and false-negatives.

V. Conclusions

The influence of dirt on LIDAR performance is analyzed by measuring its effect on the transmission and reflection properties of (dirty) sensor covers and by studying its effect on distance measurement uncertainties. Samples of dirty sensor covers are collected through real world driving to perform this analysis. The dirt effect for two different positions of the plastic cover on the grill was found to be similar, while different samples (containing different real world dirt profiles) showed considerable variability. A physical simulation of the dirt effects on the sensor performance needs to include absorption and scattering in order to successfully replicate the effects observed in real road conditions. Future tasks include the direct measurement of the changes in range for each type of dirt and the effect over other uncertainties like the probability of detection.

References

[1] Audi, “Mission accomplished: Audi A7 piloted driving car completes 550-mile automated test drive”, https://www.audiusa.com/newsroom/news/press-releases/2015/01/550-mile-piloted-drive-from-silicon-valley-to-las-vegas, 2016.

[2] Google, “Google Self-Driving Car Project”, http://static.googleusercontent.com/media/www.google.com/en/us/selfdrivingcar/, 2016.

[3] Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., Herrtwich, G., “Making Bertha See,” 2013 IEEE International Conference on Computer Vision Workshops (ICCVW), Sydney, Australia:214-221, 2013.

[4] Aeberhard, M., Rauch, S., Bahram, M., Tanzmeister, G., Thomas, J., Pilat, Y., Homm, F.,Huber, W., Kämpchen, N., “Experience, Results and Lessons Learned from Automated Driving on Germany's Highways”, *IEEE Intell. Transport. Syst. Mag*. 7(1):42-57, 2015, doi:10.1109/MITS.2014.2360306.

[5] Jens Mazzega, Hans-Peter Schöner, “Wie PEGASUS die Lücke im Bereich Testen und Freigabe von automatisierten Fahrzeugen schließt”, http://www.pegasus-projekt.info/files/tmpl/pdf/PEGASUS_Tagung_Methodenentwicklung%20Folien.pdf, 2016.

[6] Markus Maurer, J.Christian Gerdes, Barbara Lenz, Hermann Winner. “Autonomes Fahren: Technische, rechtliche und gesellschaftlich Aspekte”,2015.

[7] Berk, M., Kroll, H., Schubert, O., Buschardt, B., Straub, D., "Bayesian Test Design for Reliability Assessments of Safety-Relevant Environment Sensors Considering Dependent Failures," SAE Technical Paper 2017-01-0050, 2017, doi:10.4271/2017-01-0050.

[8] Thomas Haller, “Funktionale Sicherheit in der Praxis”, https://rg-muenchen.gi.de/sites/default/files/13-02-04a.pdf, 2014.

[9] Hermann Winner, Walther Wachenfeld, Philipp Junietz M, “Safety Assurance for Highly Automated Driving – The PEGASUS Approach validation and releasing the autonomous vehicle”, 2016.

[10] R. H. Rasshofer, M. Spies, H. Spies, “Influences of weather phenomena on automotive laser radar systems”, Advances in radio science, 9, p 49–60, 2011.

[11] Michael E. O'Brien, Daniel G. Fouche, “Simulation of 3D Laser Radar Systems”, 1st edn, 2005.

[12] Çaglar Kavak, “GPU based infrared signature modeling and scene simulation”, Master Thesis, 2014.

[13] Matt Pharr, Greg Humphreys, “Physically based rendering: From Theory to Implementation”, Second Edition. MK, 2010.

[14] Ristorcelli Thomas . “Evaluation de l'apport des visées multi-angulaires en imagerie laser pour la reconstruction 3D des couverts végétaux”,Thesis, 2013.

[15] Olaf Stenzel, “The physics of thin films optical spectra, an introduction”, Springer, p. 111, 2005.

[16] Berk, M., Kroll, H., Schubert, O., Buschardt, B., Straub, D., “Eine stochastische Methodik zur Berücksichtigung von Umgebungseinflüssen am Beispiel von LiDAR Sensoren”, VDI/VW Gemeinschaftstagung Fahrerassistenz und automatisiertes Fahren, p.455–475, 2016.

[17] Detlef Röttger, “Implementing physically based ray tracing with OptiX and MDL”, GPU Technology conference, http://on-demand.gputechconf.com/gtc/2016/presentation/s6244-roettger-optix-mdl.pdf, 2016.

RightsLink

Sign in/Register

Characterization and simulation of the effect of road dirt on the performance of a laser scanner

Conference Proceedings:
2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)

Author: Jose Roberto Vargas Rivero

Publisher: IEEE

Date: October 2017

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

sensors

MDPI

Article

Weather Classification Using an Automotive LIDAR Sensor Based on Detections on Asphalt and Atmosphere

Jose Roberto Vargas Rivero [1,2], Thiemo Gerbich [1], Valentina Teiluf [3], Boris Buschardt [1] and Jia Chen [2,4,*]

[1] Audi AG, Auto-Union-Str., D-85057 Ingolstadt, Germany; jose-roberto.vargas-rivero@audi.de (J.R.V.R.); thiemo.gerbich@audi.de (T.G.); boris.buschardt@audi.de (B.B.)
[2] Electrical and Computer Engineering, Technical University of Munich, Theresienstr. 90, D-80333 München, Germany
[3] Department of Electrical, Electronic and Communication Engineering, Friedrich-Alexander University of Erlangen, Schloßplatz 4, D-91054 Erlangen, Germany; valentina.teiluf@fau.de
[4] Institute for Advanced Study, Technical University of Munich, Lichtenbergstraße 2 a, D-85748 Garching, Germany
* Correspondence: jia.chen@tum.de

Received: 29 June 2020; Accepted: 29 July 2020; Published: 1 August 2020

Abstract: A semi-/autonomous driving car requires local weather information to identify if it is working inside its operational design domain and adapt itself accordingly. This information can be extracted from changes in the detections of a light detection and ranging (LIDAR) sensor. These changes are caused by modifications in the volumetric scattering of the atmosphere or surface reflection of objects in the field of view of the LIDAR. In order to evaluate the use of an automotive LIDAR as a weather sensor, a LIDAR is placed outdoor in a fixed position for a period of 9 months covering all seasons. As target, an asphalt region from a parking lot is chosen. The collected sensor raw data is labeled depending on the occurring weather conditions as: clear, rain, fog and snow, and the presence of sunlight: with or without background radiation. The influence of different weather types and background radiations on the measurement results is analyzed and different parameters are chosen in order to maximize the classification accuracy. The classification is done per frame in order to provide fast update rates while still keeping an F1 score higher than 80%. Additionally, the field of view is divided into two regions: atmosphere and street, where the influences of different weather types are most notable. The resulting classifiers can be used separately or together increasing the versatility of the system. A possible way of extending the method for a moving platform and alternatives to virtually simulate the scene are also discussed.

Keywords: autonomous driving; automotive LIDAR; weather conditions; sunlight; classification; atmosphere; street

1. Introduction

One of the main challenges for the safety validation of autonomous driving vehicles lies on the influence of weather phenomena [1,2]. As each of the main sensors, namely LIDAR, radar and cameras increases its sensitivity in order to detect smaller objects faster and hence be able to drive autonomously at higher speeds, the possible influence of environmental perturbations on their perception increases. On the one hand, those perturbations could cause false positives (the term false positives in this paper refers to a false detection in the point cloud before segmentation and classification are done), confuse self-calibration algorithms and reduce the sensor range [3,4]. On the other hand, they could constitute a source of valuable information if the dependencies are known and properly characterized to better evaluate and predict road conditions or adapt its operation mode, for example, [5].

In this paper, we focus on the influence of rain, fog and snow on a LIDAR sensor. Previous results are expanded by considering not only absorption and reflection [2] as well as changes in the reflection characteristics of the target [3], but also the simultaneous influence of a changing ambient illumination in outdoor conditions.

With that objective in mind, a LIDAR was placed outdoors in a static position. As target, an asphalt region of a parking lot was used. The collected point cloud data was separated between detections on the atmosphere and street. The detections on those two areas were analyzed using a classifier in order to identify if, in a static scenario, reliable information about the current weather could be extracted based on the information provided by an automotive LIDAR. Finally, possible applications of the results are shown, including a way of extending the method for use in a moving platform (car/bus/truck) in which sensor and targets move. For the virtual simulation, the use of physically-based rendering is suggested in order to include the effects of a changing background illumination together with changing reflection properties of targets in a reproducible way.

The next section introduces the current state of the art regarding the influence of weather on the LIDAR performance and the use of weather classification algorithms with focus on LIDAR data. Section 3 presents the experimental setup and Section 4 the results for the atmosphere and street regions.

2. State of the Art

2.1. Influence of Weather on a Performance

One of the main disadvantages of optical sensors (camera and LIDAR) in comparison to radar is their higher performance degradation under the presence of rainfall, fog and snow [6]. In general, the influence of adverse weather on the performance of a LIDAR sensor can be divided as follows:

- Changes in the mechanical, optical and chemical properties of the LIDAR cover like: change in the transmission caused by water absorption [7]. Deformation and change in the refractive index caused by changes in temperature. Changes in its chemical composition due to constant exposure to ultraviolet light [8].
- New layers formed on the LIDAR cover such as: dirt layers [9], water layers (or ice layers) deposited due to the presence of rain, fog, insects and spray from other vehicles.
- Scattering or absorption caused by rain drops, exhaust gasses or other pollutants [10] and dust [11].
- Changes in the optical properties of the target [2] which can be: wet, covered with snow, covered with dirt, etc.
- Changes in the background illumination.

Previous attempts to simulate the impact of weather on the performance of a LIDAR make use of the LIDAR equation [3], which calculates the absorption and scattering coefficients of rain, snow or fog based on the drop size distribution [12]. Given the relationship between drop size distribution and rain intensity, it is possible to use Monte Carlo simulations to calculate a false positives rate caused by the drops [13] for a given rain intensity. Alternatively, the detections can be filtered out using a Kalman filter or dynamic radius outlier removal filters [14,15].

The effect of rain on the reflectance of different targets has also been studied [2,7]. In this case, the absorption and scattering caused by rain is characterized simultaneously with the change in target reflectance by measuring the range, intensity and number of reflected points using a LIDAR sensor for targets placed outdoors. It is found that rain causes a decrement in the intensity and number of detected points for all targets tested, which included materials such as metal, asphalt, concrete and a retro reflective surface, each one placed at a different distance [2]. The authors mention that the change in the reflection characteristics of the target seem to generate an intensity change bigger than the one generated by absorption and scattering caused by rain [2]. The material with the maximum reduction in number of reflected points is found to be asphalt.

This paper will focus on the case in which the LIDAR sensor and target are both placed outdoors. In that way, the effect of a changing background illumination and the change in the optical properties of the LIDARs cover are also included. As target, an asphalt portion of a parking lot is used. Besides physical effects, the influence of sensor internal algorithms that dynamically adapt the performance of the sensor like automatic gain control [16], for example, are left active.

2.2. Effect of Water on a Surface's Transmission and Reflection

For analysis purposes, the effect of water upon the transmission and reflection of a porous material can be divided into three stages: first, the pores of the material are filled with water. When the material reaches saturation, a thin water film forms on top of it. As the water level increases, the characteristics of the material itself lose importance and the reflection starts to depend mostly on the optical properties of water.

The first stage can be understood by considering the change in the optical path length of the material when its pores are filled with water instead of air. This implies a higher probability of absorption of the photons and hence a reduction in its transmission. Regarding reflection, the scattering of the surface changes to a more forward scattering. Consequently, as can also be seen in the visible spectral range, the surface becomes more opaque [7]. A comparison of different dry and wet materials shows an average reflectivity reduction of around 10% for a wavelength of 900 nm and of around 30% for a wavelength of 1.5 µm [1]

The second stage can be analyzed assuming a thin and homogenous water layer. In this case, considering the plastic cover of a LIDAR, for example, a multilayer interface is built consisting of air, plastic or glass, a coating, and a water layer. The Fresnel equations can then be used to calculate the reflectance. In most cases, though, the layer will not be homogeneous. In that case, the surface can be characterized in transmission and reflection by measuring its bidirectional transmittance distribution function (BTDF) and bidirectional reflectance distribution function (BRDF) [17] for the required wavelength.

For further increments in the water level, the reflection can be analyzed by using the bathymetric LIDAR equation [16].

This equation uses the parameters as illustrated in Figure 1. It is defined as:

$$P_r = \frac{cP_T\rho cos^2\theta}{(n_w H + D)^2}\, exp(-2\alpha D \sec\phi) \tag{1}$$

where P_r corresponds to the received power, P_T to the transmitted power, ρ to the reflectance of the bottom material, θ to the incident angle of the transmitted laser beam in air, ϕ to the refracted beam angle in water, H to the altitude of the LIDAR above the water, D to the distance between the water surface and the bottom material, n_w the refraction index of water, α which combines stretching and attenuation of the pulse, and c is a constant containing sensor-related values [16]. This equation corresponds to the conventional LIDAR range equation for an extended target, with an added exponential decay in a scattering medium corresponding to the Beer-Lambert law.

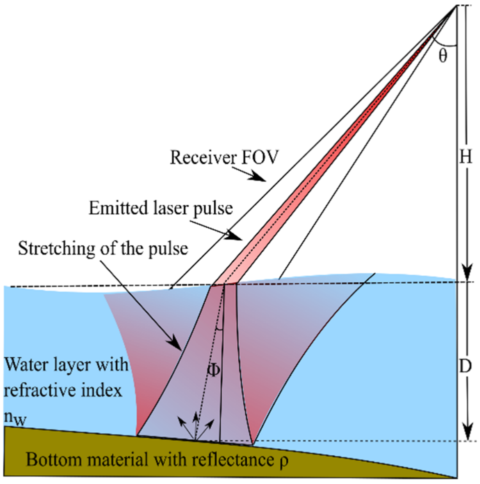

θ
H
Receiver FOV
Emitted laser pulse
Stretching of the pulse
Water layer with refractive index n_W
Φ
D
Bottom material with reflectance ρ

Figure 1. Illustration of the parameters used in the bathymetric LIDAR equation. Although the used LIDAR is not a bathymetric LIDAR the equation is used to show the dependencies present when the amount of water on top of a certain surface is high. These dependencies are used for the analysis of the street region when covered by water in Section 4.

2.3. *Effect of Ambient Light on LIDAR Measurements*

Since at the end, the information obtained from the LIDAR is based on detections over the noise level, the APD (avalanche photo diode) equation also needs to be considered:

$$SNR = \frac{P_r}{N} \tag{2}$$

where P_r can be calculated as shown in Equation (1) and N corresponds to the noise which is a combination of the shot noise N_S, the background noise N_B, which is proportional to the background light optical power collected by the detector, and the thermal noise N_T [1].

$$P(N(t) = n) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{n-\mu}{\sigma})^2} \tag{3}$$

$$I(t_d) = \mu + x\sigma \tag{4}$$

$$TOF = t_d - t_p \tag{5}$$

Finally, the time of flight (TOF) is defined as the time difference between the start of the sent pulse t_p and the moment when the detected intensity reaches a certain minimum value t_d (5). Assuming a Gaussian distribution for the noise (3), a SNR of 3σ (0.27% probability that a detection is caused by noise) with respect to the mean noise level can be used, for example, in order to define a valid detection (4). If the average noise level increases from μ_1 to μ_2 as shown in Figure 2, and the shape of the echo pulse remains the same, there is no detection. Correspondingly, the total number of reflected points is reduced. This fact is used for the analysis in the measurement section.

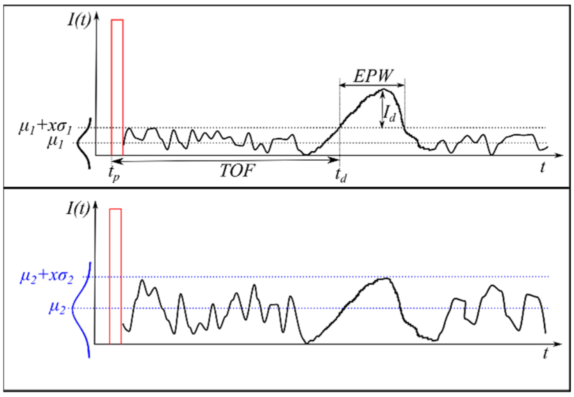


Figure 2. Relation between TOF and intensity for a received pulse. If the noise level increases (blue) no detection is registered. The sent pulse (idealized) is shown in red and the received signal in black.

As can be deduced from Figure 2 the intensity of the detection I_d, its distance, which is calculated using the TOF, and the EPW (Echo pulse width) value also change depending on the noise level. This kind of walk error, which can also be caused by a change in the form of the echo pulse, can in some cases be compensated but not completely avoided [18]. The EPW, which is measured in meters [19], corresponds to the width of the pulse above the noise level. It is proportional to the reflection of the object.

2.4. Light Scattering and Absorption by Particles in the Atmosphere

Particles in the atmosphere scatter and absorb the laser light depending on their shape, size and complex index of refraction [20]. Spherical particles with a size parameter $\alpha = 2\pi r/\lambda$ smaller than 0.1 (Rayleigh regime) tend to have a symmetric forward/backwards scattering. Particles with α values between 0.1 and 50 (Mie regime) have a bigger forward as backward scattering lobe, while particles with α values bigger than 50 (Geometric regime) have a very large forward scattering lobe and almost no backwards scattering [20]. In case of snow, the analysis is more complicated and depends on the exact shape of the crystal, which depends on the temperature [3]. For multiple particles, the scattering coefficient is also proportional to particle concentration and size [21].

Table 1 shows the average radius and size parameter for typical particles in the atmosphere. The size parameter is based on a laser with a wavelength of 905 nm, which is in the range of wavelengths typically used for autonomous driving cars [22,23]. The refractive index for both water and ice at this wavelength has an imaginary part in the order of 10^{-7} [24] and hence the single scattering albedo (SSA) for both drops and crystals in the geometric regime approximates 0.53 [25,26]. For fog, it approximates 0.8 [26].

Table 1. Typical atmospheric scattering particles with their average size parameter for a wavelength from 905 nm and concentration [21,24,27–29].

Type	Radius (µm)	Size Parameter	Concentration
Air molecules	0.0001	0.0007	$<3 \times 10^{25}\ m^{-3}$
Haze, smoke, dust	0.01–1	0.07	10^5–$5 \times 10^{10}\ m^{-3}$
Fog	1–20	7–139	10^6–$5 \times 10^9\ m^{-3}$
Rain	100–10000	694–69427	10–$10^3\ m^{-3}$
Snow	1000–5000	6943–34714	10^{-2}–$10^3\ m^{-3}$
Hail	5000–50000	34714–347137	10^{-2}–$1\ m^{-3}$

Although the parameters are distributed over a broad range, not all the sizes occur with the same probability, with common sizes for snowflakes around 1 mm [29], for rain drops around 0.2 mm and for fog droplets around 3 µm (Chu Hogg) and 18 µm (Advection) [3].

2.5. Weather Classification Using LIDAR

There are different alternatives to evaluate current weather and road friction in the vicinity of a car. One alternative is using the information provided by the vehicle, for example from windshield wipers, fog lights, torque and speed of engine and tires, anti-lock braking system (ABS), electronic stability control (ESC) and traction control system (TCS) intervention events, temperature, global navigation satellite system (GNSS) position, steering wheel angle and breaking signal [30]. Another alternative is to use sensors specific for road surface analysis like polarization cameras or short distance multi-wavelength IR sensors. These sensors use the change in the amount of vertically polarized light or its resonance frequency caused by the different phases of water to classify between ice, snow and mixtures [31]. A third alternative, and the focus of this paper, is to use advanced driver assistance systems (ADAS) sensors like visible spectrum (VIS) cameras, ultrasound, radar or LIDAR whose main purpose is the detection of static and moving objects but whose performance is affected by weather [32]. All these techniques can be used by themselves or combined to provide different levels of classification accuracy [30]. Additionally, the information provided by other cars or sensors can be included using vehicle-to-everything (V2X) technologies [33].

LIDAR sensors have been used to classify aerosols on the atmosphere using the difference between the extinction cross section and backscatter cross section caused by the different types of aerosols. The differences in the linear depolarization ratio and the frequency differences caused by inelastic scattering (Raman LIDAR systems) are also used [34,35]. These kinds of systems are able to provide the type, size and concentration of the different aerosols. A drawback regarding automotive LIDAR systems is that they usually use monochromatic unpolarized light and measure only elastic scattering effects. This is compensated in some systems by providing multi target detection (multiple echoes), which facilitates the differentiation of detections caused by rain, fog or snow from those caused by a solid objects [36,37].

Regarding the types of classifiers used, in the context of removal of detections caused by fog [38], support vector machine (SVM) and K-nearest neighbor (KNN) were used reporting a classification accuracy of heavy fog vs. solid objects higher than 90% (F-Score) with SVM and 79.4% (F-Score) with KNN. The room in which the experiment took place had a size of 5 m by 4 m. A recent study [32] with a focus similar to the present paper used a Velodyne LIDAR puck (VLP) and a SCALA sensor to classify between clear, rain, and fog. They reported a true positive rate higher than 95% for all three classes using the VLP sensor. The SCALA sensor had a true positive rate (TPR) of 99.78% for fog, but it fell to 84.92% for rain and 83.19% for clear using SVM. It is important to mention that the measurements were done in a climate chamber where the visibility values for the fog and rain intensity were kept within relatively constant ranges. The measurements are reported in a region with a size of 10 m by 25 m. A second set of measurements was performed on the road using only the VLP sensor. In this case, only two classes were used: clear and rain. A TPR of 92.45% for rain and 97.60% for clear is reported using KNN. To train the classifier, a parameter vector based on the position of the detection in Cartesian and spherical coordinates as well as the echo number and the intensity (for the VLP sensor) or echo-pulse width (for the SCALA sensor) is used.

In a recent paper [39], the measurement distance was reduced to a region close to the LIDAR cover (<1.6 m). In this case, the classes used were: clean, salt, dirt type 1, dirty type 2 and frost. Images where constructed from two different views: front view image, with layer number on the y axis, horizontal angle on the x axis and echo-pulse width as color and top view image with vertical angle on the y axis radial distance in the x axis and layer number as color. These two image types were used to train a deep neural network reaching a classification accuracy of 77.98%. If only classifying between clean, salt and frost, the accuracy increased up to 95.41%. Although in this case the focus is not weather classification, the state of the LIDAR cover provides an important hint about it.

3. Experiment

As mentioned in the introduction, the sensor was placed outdoors and the recorded data was extracted for two regions: atmosphere region (distance from sensor <5 m) and street region (distance from sensor between 33 m and 37 m).

The measurement setup consisted of a LIDAR sensor, which was installed on the roof of a building, as illustrated on the left side of Figure 3 (see Appendix A for detailed setup information). The reflectance of water for non-polarized light coming from air is almost constant and close to 2% up to an angle of around 45° to the surface normal, and increases exponentially [40] above this angle. Hence, an angle close to 45° or less would be convenient in order to preserve a dependency on ρ (see (1)) in the results for thin water layers. In our test setup, an angle of 46° is used.

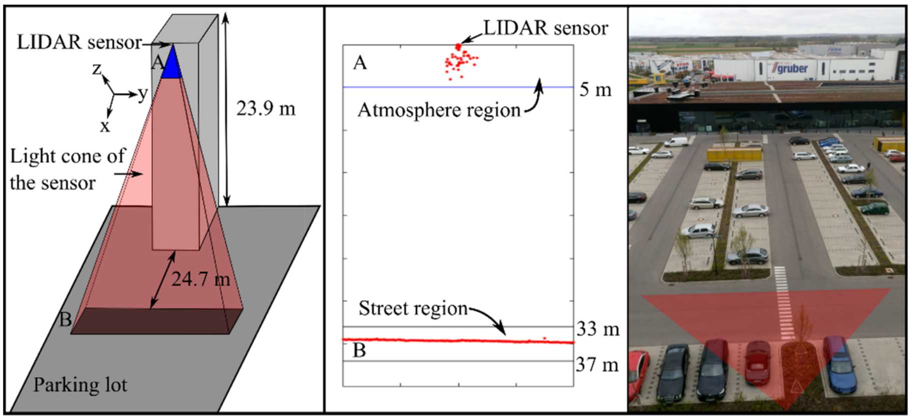


Figure 3. Depiction of the measurement setup. Left: schematic front view of the LIDAR sensor position and its effective field of view (A: atmosphere region, B: street region), the used axis corresponds to the LIDAR coordinate system. Middle: example of the obtained point cloud (birds eye view). Right: Picture from top of the building. The field of view of the sensor is illustrated by the red triangle.

The recorded data was obtained by activating and deactivating the laser scanner at predefined times, from April to December 2017. In total, 639 h were recorded, during which it was raining 74 h (12%), foggy 45 h (7%) and snowing 9 h (1%).

For the evaluation, the recorded hours were discriminated in four weather classes: Clear, Rain, Fog and Snow. The recorded data was classified using information from a weather station from the Bavarian Environmental Agency [41]. As fog is not reported, we performed the fog measurements manually started and stopped them. The precipitation values are reported with an interval of five minutes and for snow and background radiation each hour (see Appendix B for detailed weather information). The weather station was located at a distance of 7 km from the measurement setup.

Due to an automatic gain control system, which will be further explained in the Results section, only two values were used for background radiation: (1) with background radiation and (2) without background radiation. For each sample, the following procedure was followed:

- It was verified that no people, cars or other objects were passing by during the measurement.
- For Fog, Rain and Snow samples it was verified that the weather remained the same during a period of at least 5 min before and after.
- Clear samples were taken from days were the weather station did not report rain or snow and no fog was seen.
- The samples were randomly chosen over different days in order to increase the variability.

From each sample, which had a duration of 10s, the single frames were extracted. This resulted in a total of 6130 frames without background radiation and 1930 with background radiation for each weather type. The reason for having a lower number of frames during the day is the difficulty of recording snow data on the street region during that time. The total number of frames is nevertheless within the range used for similar classifiers [32,39].

Each frame contains possibly up to thousands of detections distributed on the cover of the sensor, the atmosphere and the street region. Section 4.1 explains the features used for the parameter vector upon which the classification takes place. These features are based on the analysis of each region:

- Region A (Section 4.1.1) corresponds to changes in the optical characteristics of the atmosphere and the LIDAR cover.
- Region B (Section 4.1.2) corresponds to changes in the reflection of the street. The term region is used instead of surface or plane because it includes reflections coming from water drops splash which may be a few cm away from the street surface or also from a snow cover.

In both figures, the distances are measured horizontally across the axis of symmetry of the sensor (x-direction; Cartesian coordinates were used instead of polar coordinates to facilitate the extraction of the detections corresponding to the street region).

4. Results

In the first step, the influence of a changing background radiation was investigated. Background radiation is measured in watts per square meters and gives a numeric value for the brightness. During the night, its value is zero, the biggest value measured during the day and also in the group of selected samples was 779 W/m^2.

The number of scan points shown in Table 2 corresponds to the average of 3800 frames during clear weather. The frames are separated based on their echo number: n_{e1} for the first echo and n_{e2} for the second. It can be seen that with no background radiation, the mean number of scan points is on average 35% higher for the first echo and for the second echo it is 2.3 times higher. This is caused by the reduction on the noise level (as explained in Figure 2) which also causes the increment of the EPW (~30%). When comparing the number of detections in the atmosphere region with and without background radiation (Section 4.1.1) a reduction in the number of detections can also be seen.

Table 2. Influence of background radiation on the total number of echoes and EPW for 1900 frames during the day and 1900 during the night including both regions during clear weather. The results show mean value ± standard deviation.

W/m^2	n_{e1}	n_{e2}	EPW [m]
Global radiation $= 0$	257 ± 19	66 ± 67	1.21 ± 0.39
Global radiation > 0	190 ± 110	20 ± 25	0.93 ± 0.27

4.1. Characterization of the Distributions in the Atmosphere and Street Region

In order to characterize each of the obtained distributions, a group of seven different parameters were used as shown in Figure 4. The same parameters apply to the atmosphere and street regions with and without background radiation, as in each case they capture the main changes in the shape of the different distributions.

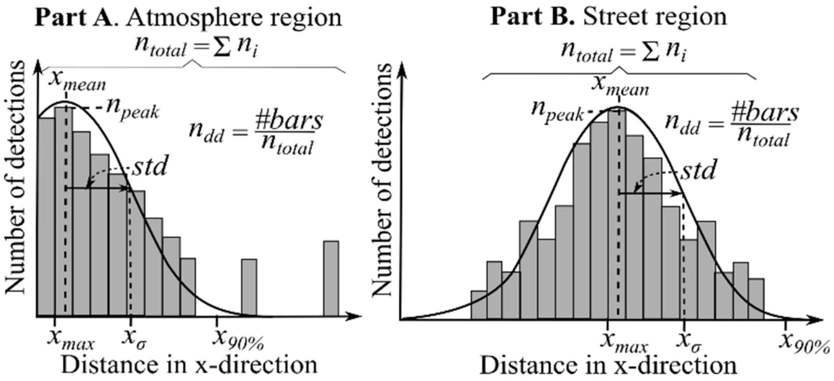


Figure 4. Parameters used to characterize the distributions in the atmosphere and street region.

The meaning of each parameter is as follows:

(1) The number of detections in the maximum of the histogram (n_{peak}).
(2) The position of the maximum of the histogram in meters (x_{max}).
(3) Total number of detections in the region (n_{total}).
(4) Mean detection distance in x-direction (x_{mean}).
(5) Standard deviation of the detection distance in x-direction (x_{σ}).
(6) Distance at which 90% of the total number of detections in the region is reached. The points are accumulated per bin from left to right starting at 0 m or 33 m, respectively ($x_{90\%}$).
(7) Number of discrete distances (n_{dd}) where detections take place relative to the total number of detections. Especially useful when there is background radiation. This parameter is proportional to the number of bars in the histogram.
(8) Number of first echo detections (n_{e1}).
(9) Number of second echo detections (n_{e2}).
(10) Number of third echo detections (n_{e3}).
(11) Mean value of the echo-pulse width (EPW_{mean}).
(12) Standard deviation of the echo-pulse width (EPW_{std}).

As will be examined in the discussion section, not all parameters are equally relevant for both regions. For this reason, Tables 3 and 4 summarize the most relevant parameters for each. Furthermore, they constitute the bases for the presented analysis, which tries to relate the theoretical background about the variation in the optical properties of the LIDAR cover, the street and the atmosphere as presented in Section 2 with the change in the distribution of the detections.

Table 3. Most important parameters for the atmosphere region (0–5 m)*. (w/o) Without background radiation, (w) with background radiation. With n_{total}: total number of detections in the region, n_{peak}: number of detections in the maximum, x_{σ}: standard deviation of the detection distance, n_{e1}: number of first echo detections and *EPW*: echo-pulse width. The results show mean value ± standard deviation.

	Clear		**Fog**		**Rain**		**Snow**	
	w	**w/o**	**w**	**w/o**	**w**	**w/o**	**w**	**w/o**
n_{total}	86 ± 34	244 ± 16	114 ± 14	313 ± 55	238 ± 161	661 ± 113	137 ± 13	344 ± 57
n_{peak}	55 ± 18	113 ± 19	74 ± 14	124 ± 34	159 ± 124	287 ± 60	79 ± 14	126 ± 21
x_{σ} [cm]	5 ± 11	10 ± 8	3 ± 2	7 ± 5	3 ± 3	15 ± 8	57 ± 22	84 ± 23
n_{e1}	86 ± 34	244 ± 16	114 ± 14	313 ± 55	238 ± 161	659 ± 112	135 ± 12	324 ± 46
EPW [m]	0.75 ± 0.29	0.95 ± 0.35	0.82 ± 0.29	0.94 ± 0.34	0.86 ± 0.30	1.00 ± 0.35	0.86 ± 0.32	1.02 ± 0.38

* In contrast to Figures 5 and 6, in which multiple frames are used and the parameters are calculated once for all of them, in this table as well as in Table 4 the parameters are calculated per frame and compared.

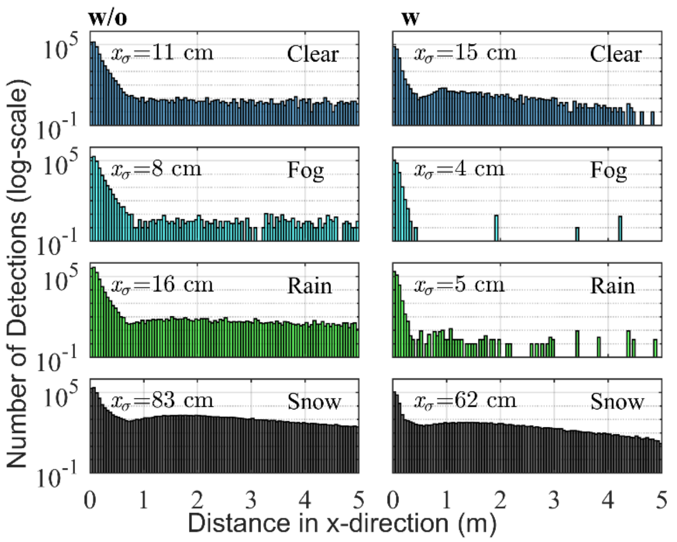


Figure 5. Distribution of the detections for each weather class (1900 frames each) in the atmosphere region (0-5 m) (w/o) without and (w) with background radiation. With x_σ: standard deviation of the detection distance in x-direction in cm.

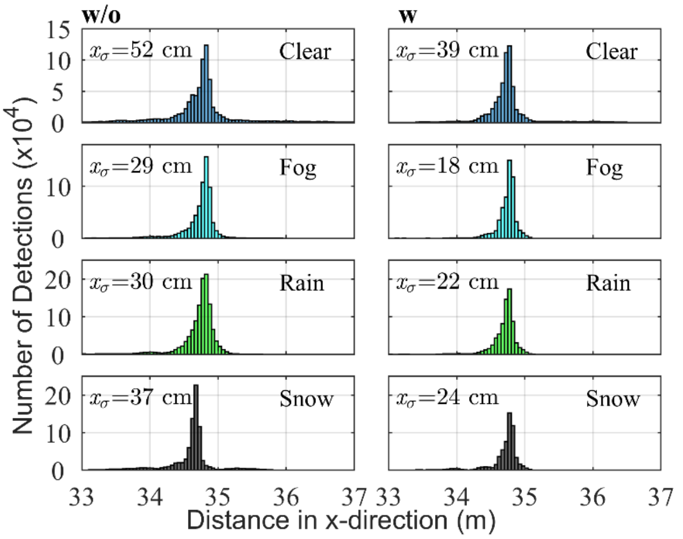


Figure 6. Distribution of the detections for each weather class (1900 frames each) in the street region: (w/o) without and (w) with background radiation. With x_σ: standard deviation of the detection distance in x-direction in cm.

Table 4. Most important parameters for the street region (33–37 m). (w/o) Without background radiation, (w) with background radiation. With n_{peak}: number of detections in the maximum, x_{mean}: mean detection distance, x_σ: standard deviation of the detection distance, n_{e2}: number of second echo detections and *EPW*: echo-pulse width. The results show mean value ± standard deviation.

	Clear		**Fog**		**Rain**		**Snow**	
	w	**w/o**	**w**	**w/o**	**w**	**w/o**	**w**	**w/o**
n_{peak}	77 ± 39	45 ± 37	88 ± 30	72 ± 26	97 ± 39	125 ± 33	89 ± 30	132 ± 33
x_{mean} [cm]*	74 ± 15	96 ± 27	73 ± 02	74 ± 03	69 ± 03	71 ± 03	70 ± 05	60 ± 03
x_σ [cm]	27 ± 22	68 ± 30	17 ± 5	24 ± 5	21 ± 5	31 ± 7	22 ± 8	37 ± 9
n_{e2}	40 ± 21	131 ± 20	37 ± 11	128 ± 21	52 ± 19	300 ± 52	51 ± 9	193 ± 32
EPW [m]	1.10 ± 0.24	1.48 ± 0.23	1.02 ± 0.25	1.09 ± 0.30	1.01 ± 0.26	1.22 ± 0.33	1.03 ± 0.26	1.63 ± 0.30

* This value is shown after subtracting the same value (34 m) from each class in order to show the differences in cm.

4.1.1. Atmosphere Region

In order to facilitate the interpretation of the data, the results are shown using a histogram with a bin size of 5 cm. For each class, 1900 frames are used.

Figure 5 shows the detections in the atmosphere region without and with background radiation. A logarithmic scale is chosen to point out the differences between the different weather types. The relevant numerical parameters extracted from each distribution are presented in Table 3.

The reflections in the region from 0 to 0.5 m are caused by a combination of multipath reflections on the LIDAR cover and detections on drops or snowflakes. The presence of water or dirt on the cover caused by the different weather types also influences the form of the distribution.

Regarding the region from 0.5 m to 5 m for Clear, some detections are seen. An explanation is that in this case the noise level is at its minimum and hence the sensibility is high. Therefore, some of these detections could have been caused by other particles in the air like pollen, dust or insects.

For Snow, the total number of detections is larger than for Fog, but smaller than for Rain. In comparison with raindrops, the snowflakes tend to be bigger. However, their concentration tends to be smaller (Table 1) and given that the SSAs are similar, it is reasonable to assume that the number of detections is lower. In contrast to Clear and Fog, the distribution for Snow is not homogeneous and shows a maximum around 1.8 m (also slightly visible in the rain distribution w/o), which most probably depends on the optical design of the sensor.

In the case of Fog, it is interesting to see that the total number of detections is lower than for Rain. Fog drops tend to have a higher SSA than snow and raindrops and since they have a bigger backscattering lobe, their concentration is also higher. On the other hand, their size is much smaller which seems to cause - together with the increment in the noise level - that more detections fall under the minimum required voltage (Figure 2) in comparison with raindrops or snowflakes. The increased noise level also explains why the x_σ value is the lowest of all classes.

As already mentioned with background radiation, the number of detections generally smaller (Table 2) and therefore the differences between the distributions for each weather type (Figure 5 w, compared to Figure 5 w/o) are reduced. This influences the classification accuracy as will be seen in Table 5. Besides that, most of the characteristics already described remain valid. It is interesting to see that even with a higher noise level due to sunlight, the number of detections while snowing maintains a value over distance higher than the other classes. As a result of this, the value of x_σ is much higher. This effect is caused by the bigger size, on average, of the snowflakes.

Table 5. Confusion matrix for the atmosphere region without background radiation (left, 6130 samples per class) and with background radiation (right, 1930 samples per class) (classifier: Weighted KNN).

		Predicted Class							
		Without Background Radiation				**With Background Radiation**			
		Clear	**Rain**	**Fog**	**Snow**	**Clear**	**Rain**	**Fog**	**Snow**
Actual Class	Clear	5284	1	844	1	1670	32	188	40
	Rain	3	6055	32	40	0	1877	43	10
	Fog	980	16	5132	2	58	80	1785	7
	Snow	2	19	1	6108	0	50	42	1838
	F-Score	84.97	99.04	84.36	99.42	91.42	94.74	89.78	96.23

4.1.2. Street Region

Measurements of the optical properties of the street provide direct information about road friction, which are relevant for self-driving cars to identify situations like aquaplaning. As proof of concept, it is hence interesting to know if the same method can be used for this region. In comparison with the atmosphere, measurements in the street region have a higher degree of uncertainty. Especially during the day and during winter, the use of de-icing or cleaning agents as well as the change in surface temperature could influence measurements. Other factors are the presence of dew, ice and the inhomogeneous distribution of water on the street surface. Nevertheless, the distributions for each weather type (Figure 6) are distinct enough to allow for a useful interpretation and classification. As in Figure 5, the bin size is 5 cm.

Figure 6 w/o shows the detections on the street without background radiation and Figure 6 w with background radiation. In this case, a linear scale is preferable. As for the atmosphere region, the relevant numerical parameters extracted from each distribution are presented in Table 4. However, n_{total} is not present and its place is taken by x_{mean}, additionally n_{e1} is replaced by n_{e2} the reason being that for the street region, the new parameters allow for a better separation between the classes as will be analyzed in the Discussion section (feature ablation study).

Considering Figure 6 w/o as well as the information provided by Table 4 first, the EPW (the analysis is based on the mean value of each variable) value is lower as would be expected from the reflection of the different surfaces going from Snow to Clear to Rain. This is reasonable given that the street had a snow cover of a few cm [41], which has a higher albedo (0.8) [42] than dry asphalt (0.12) [43] or wet asphalt (0.03 at 46°). Fog has the lowest EPW value, which contrasts with the results reported using a climate chamber [32] in which the EPW increased proportionally with fog density beyond the values for Rain and Clear. We believe this may apply to surfaces perpendicular to the laser beam in which the backscattering caused by fog particles and the reflection of the surface have the same main direction; in our case, due to the angle of 46° between the laser beam and the surface normal, the net effect is a reduction of the EPW. Besides having the highest EPW, Snow is also characterized by having the smallest x_{mean} value due to the presence of a snow cover.

The class Clear has a x_{σ} and x_{mean} value higher than all other classes. This is caused by the more Lambertian reflection lobe of dry asphalt. The presence of humidity or water—as is the case for Fog and Rain—reduces this lobe and increases the forward reflection, as was discussed in Section 2.2. Clear is also characterized by a low number of second echo detections (n_{e2}) in comparison with Rain and Snow where the laser beam hits many drops or snowflakes first before hitting the street surface.

Rain is characterized by a relatively low EPW. The number of second echo detections is the highest in all the classes. This, as already mentioned for the atmosphere region, is caused by the higher concentration of drops over snowflakes and their bigger size in comparison with fog droplets. More first echo detections on the drops cause more second echo detections on the street.

Fog has the lowest x_{σ} value as happened for the atmosphere region. As mentioned before, this hints to an increment of the average noise level which causes weak detections to vanish (Figure 2). The number of second echo detections is similar to Clear.

Considering Figure 6 w, the EPW for Snow is reduced to a value similar to that of Fog and Rain. This could have been caused by melting of part of the snow cover and the presence of footsteps and wheel marks, which reduced the average albedo of the surface.

Clear has the highest x_{σ} and Fog the smallest, as happens without background radiation. Regarding x_{mean} it remains the highest for Clear but now Fog has the second highest value. Those for Snow are similar. This also indicates a partial melting of the snow cover and the reduction of any extra humidity caused by fog on the street surface.

Rain and Snow become very similar with Rain having a higher n_{peak} value. The value of n_{peak} increases when n_{e2} increases and therefore is higher for Snow and Rain and decreases with x_{σ}. This causes that the n_{peak} values for Clear and Fog with background radiation are higher than without. In general, though, an increase in the noise level due to sunlight causes in both atmosphere and street regions a reduction of the mean values of all the parameters. The next section presents the classification results and relates them with the analysis presented in this section.

4.2. Classifier

The results show that there is enough variability in the data for classification. However, a single sample by itself cannot be accurately classified, especially with background radiation when the number of detections is low. For that reason, all frames are used to train a classification algorithm. The inputs for the classifier are the parameters shown in Figure 4, the most important of which were discussed in the previous section: number of detections in the maximum of the histogram (n_{peak}), mean detection distance in x-direction (x_{mean}), standard deviation of the detection distance in x-direction (x_{σ}), number of second echo detections (n_{e2}) and mean value of the echo-pulse width (EPW_{mean}).

For the classification, KNN was employed due to its good performance and use in previous studies [32,38]. To balance the number of samples per class random under sampling was used. With the aim of allowing for comparison and avoiding bias, the same classifier was used in all cases. Five-fold cross validation was used; meaning that four fifths of all data is used for training and one fifth for validation. The process is repeated five times with different partitions of the data, the result is then averaged. The confusion matrix shows the number of samples which where correctly classified in its diagonal. A gray scale is used to facilitate the interpretation going from white for zero frames to black for the maximum number of frames (6130 without and 1930 with background radiation). The F-score is presented as metric. Being the harmonic mean of the precision and recall metrics [44], it provides a good evaluation of the classifier's performance.

4.2.1. Atmosphere Region

As shown in Table 5, the F-Score for Snow and Rain for the atmosphere region is higher than 94% in both cases. Without background radiation, it is higher than 99%. This is due to the characteristically high x_{σ} value of Snow and the high amount of total detections n_{total} and first echo detection n_{e1} of Rain in comparison with the other two classes.

Clear and Fog have a higher F-Score with and without background radiation. The reduction of the average x_{σ} values for Fog during the day to values smaller than those for Clear reduces the number of misclassifications between the two classes. This, on the other hand, has as disadvantage that more frames are incorrectly classified between Rain and Fog as the similarities between the two classes increase.

4.2.2. Street Region

The street region (Table 6) shares some similarities with the atmosphere region. As before, Snow and Rain have the highest classification values. The values for Clear and Fog are slightly better, as fewer frames are confused between the two classes. This happens mostly because the variations between the x_{σ} values are higher. Misclassifications between Rain and Fog are also far less common. This coincides with previous results [2] in which it is mentioned that changes in surface reflection are the ones that cause the most notable effects on the detections.

Table 6. Confusion matrix for the street region without background radiation (left, 6130 samples per class) and with background radiation (right, 1930 samples per class) (classifier: Weighted KNN).

		Predicted Class							
		Without Background Radiation				**With Background Radiation**			
		Clear	**Rain**	**Fog**	**Snow**	**Clear**	**Rain**	**Fog**	**Snow**
Actual Class	Clear	5462	0	667	1	1761	32	112	25
	Rain	0	6078	52	0	5	1912	1	12
	Fog	315	6	5809	0	13	0	1900	17
	Snow	11	0	3	6116	0	0	13	1917
	F-Score	91.62	99.45	91.67	99.89	95.04	98.71	96.11	98.31

5. Discussion

In this section, the obtained results are compared with known weather classification results using LIDAR. Furthermore, feature ablation is used to reduce the number of features to the most relevant ones while simultaneously increasing in some classes the classification accuracy. Finally, the possibility of using the same classifiers in a moving platform like a car or truck, for example, is briefly discussed as well as ways to improve accuracy and robustness. The section finishes with a short note about simulation.

5.1. Atmosphere Region

Compared with previous results mentioned in Section 2.5, the classifier for Fog has a slightly lower F-Score: 84.4 (w/o BR (Background radiation)) and 89.8 (w BR) than a classifier trained specifically to distinguish between fog an solid objects (F-Score: 90.1) [38]. When compared to the results obtained using a climate chamber, our classifier has a TPR for Fog of 83.7% (w/o BR) and 92.5% (w BR) versus 99.8% [32]. The slightly lower results are mostly a consequence of the innate variability of the outdoor measurement as the parameters and algorithm used for the classification are similar. In case of Rain and Clear, this variability could have been an advantage as in both cases the TPRs are higher. For Rain, our classifier has a TPR of 98.8% (w/o BR) and 97.2% (w BR) compared to 84.9% in a climate chamber [32]. For Clear, the values are 85.0% (w/o BR) and 91.4% (w BR) compared to 83.2% in a climate chamber [32].

5.2. Street Region

For this region, the results are better. Consequently, only the TPR for Fog is lower when comparing it with the result obtained in a climate chamber [32]: 94.8% (w/o BR) and 98.4% (w BR) versus 99.8%. In all other cases, our classifier provides better results.

5.3. Feature Ablation Study

Some features provide a very small separation between the classes and in some cases may even reduce the classification accuracy. For that reason, a feature ablation study is done. The results are presented in Table 7 for the atmosphere region and Table 8 for the street region. In each case, the classifier is trained with all features minus one, the results are compared with the original result using all features. The F1-score is used as evaluation parameter as was done in Tables 5 and 6.

Table 7. Results of the feature ablation analysis for the atmosphere region using the F1-Score as evaluation parameter. In comparison with the score using all features, the features are marked with ✓ when the F1-Score for two or more classes decreases more than 0.4 (marked in grey), with ✗ when increases more than 0.4 (marked in blue) and with **-** in other cases (All-EPW_{std} is marked as neutral because the F1-Score of Clear increases).

Feature	Atmosphere w/o BR (F1-Score)				Atmosphere w BR (F1-Score)				Evaluation
	Clear	Rain	Fog	Snow	Clear	Rain	Fog	Snow	
All	84.97	99.04	84.36	99.42	91.42	94.74	89.78	96.23	
All-n_{peak}	85.11	98.94	84.48	99.36	91.10	94.23	89.15	96.05	✓
All-x_{max}	84.83	99.11	84.34	99.41	91.54	94.76	89.65	96.02	-
All-n_{total}	84.50	99.04	83.85	99.40	91.04	94.65	89.16	96.10	✓
All-x_{mean}	84.94	99.11	84.24	99.48	91.60	94.54	89.74	96.11	-
All-x_{σ}	84.71	99.02	84.01	99.31	91.40	94.38	89.12	95.17	✓
All-$x_{90\%}$	84.76	99.09	84.18	99.47	91.36	94.66	89.49	96.16	-
All-n_{dd}	85.62	99.12	85.20	99.38	91.25	94.67	89.64	96.13	✗
All-n_{e1}	84.43	99.04	83.84	99.42	91.20	94.86	89.26	96.18	✓
All-n_{e2}	85.21	99.09	84.50	99.49	91.37	94.43	89.55	96.02	-
All-n_{e3}	85.25	99.11	84.63	99.44	91.50	94.59	89.63	96.21	-
All-EPW_{mean}	82.01	98.09	80.71	98.79	91.55	92.51	88.22	95.48	✓
All-EPW_{std}	84.86	98.92	84.21	99.47	91.93	93.59	88.55	96.42	-

Table 8. Results of the feature ablation analysis for the atmosphere region using the F1-Score as evaluation parameter. In comparison with the score using all features the features are marked with ✓ when the F1-Score for two or more classes decreases more than 0.4 (marked in gray), with ✗ when increases more than 0.4 (marked in blue) and with **-** in other cases.

Feature	Street w/o BR (F1-Score)				Street w BR (F1-Score)				Evaluation
	Clear	Rain	Fog	Snow	Clear	Rain	Fog	Snow	
All	91.62	99.45	91.67	99.89	95.04	98.71	96.11	98.31	
All-n_{peak}	91.19	99.51	91.42	99.81	95.18	98.71	96.23	98.15	✓
All-x_{max}	91.62	99.37	91.66	99.85	94.94	98.79	95.89	98.18	-
All-n_{total}	91.63	99.44	91.67	99.90	94.89	98.76	95.93	98.23	-
All-x_{mean}	91.58	99.23	91.42	99.90	94.09	98.63	94.80	98.10	✓
All-x_{σ}	91.67	99.50	91.77	99.89	94.70	98.50	95.94	97.87	✓
All-$x_{90\%}$	91.44	99.46	91.58	99.82	94.90	98.66	96.04	98.31	-
All-n_{dd}	91.57	99.62	91.75	99.89	95.98	98.91	96.67	98.39	✗
All-n_{e1}	91.35	99.37	91.32	99.89	94.76	98.61	95.91	98.05	-
All-n_{e2}	91.32	99.42	91.33	99.91	93.76	97.86	95.31	96.51	✓
All-n_{e3}	91.55	99.45	91.68	99.77	96.15	98.87	96.95	99.07	-
All-EPW_{mean}	91.07	99.27	90.99	99.86	94.00	97.91	95.49	97.67	✓
All-EPW_{std}	92.05	99.55	92.08	99.95	95.79	99.07	96.61	98.59	✗

Half of the features seem to have a very small impact on the classification. The number of discrete distances (n_{dd}) when removed mostly improves the classification results while, for example, removing the mean value of the EPW (EPW_{mean}) notably reduces the classification values. Something similar happens for the street region. In this case, besides the (n_{dd}) the standard deviation of the EPW (EPW_{std}) also seems to have a detrimental impact on the classification.

Based on these results the number of parameters used for the classification was reduced with minor reduction in the F1-Scores and in half of the cases with minor increases. The final classification results are presented in Tables 9 and 10.

Table 9. F1-Score per class for the atmosphere region when using only the parameters: n_{total}, n_{peak}, x_{σ}, n_{e1}, EPW_{mean}.

Atmosphere w/o BR (F1-Score)				Atmosphere w BR (F1-Score)			
Clear	**Rain**	**Fog**	**Snow**	**Clear**	**Rain**	**Fog**	**Snow**
85.30	98.82	84.82	99.27	92.54	93.27	89.01	96.86

Table 10. F1-Score per class for the street region when using the parameters: n_{peak}, x_{mean}, x_{σ}, n_{e2}, EPW_{mean}.

Street w/o BR (F1-Score)				Street w BR (F1-Score)			
Clear	**Rain**	**Fog**	**Snow**	**Clear**	**Rain**	**Fog**	**Snow**
91.29	99.43	91.42	99.76	96.67	99.09	96.81	98.86

While for the atmosphere region: Clear, Fog and Snow (w BR) improve, all classes with background radiation improve for the street region.

5.4. Extension for a Moving Platform (Street)

In a dynamic environment, the surface of the road doesn't remain constant with respect to the sensor, additionally other objects block dynamically the field of view of the sensor, the road itself changes and the effect of the wind becomes more important. Nevertheless, it should be possible to use a ground plane classification algorithm to select a region on the road as long as it is not blocked. If the measurement is done for sufficient time in order to compensate for the surface deviations, a similar analysis as done for the street before could be used. The change in the reflection of the road would also need to be considered.

5.5. Increasing Classification Accuracy and Robustness

Much more snow data needs to be collected with as many different types of snow as possible. Specifically for the street, the measurement should be repeated using different types of asphalt; additionally, a direct measurement of the state of the surface needs to be done using a reference sensor in order to reduce the uncertainty of the measurements. There are already special sensors that can accomplish this task, but they are in most cases not designed for automotive applications. This, together with the advantage of using a sensor, which is already installed in various car models, are the main motivations for further research.

With enough data it may be possible to return a probability per class instead of a binary classification, this may increase the robustness and usefulness of the method allowing for weather combinations. Furthermore, the acquisition of a weather station which can be placed next to the sensor would increase accuracy.

Finally, the use of the same method with other LIDAR sensors would provide important information about the influence of internal sensor calibration algorithms and architecture upon the most important classification parameters and the best size for the atmosphere region.

5.6. Virtual Simulation

Some form of ray tracing is usually used for the simulation of LIDAR sensors [45,46]. Regarding the atmosphere, it is possible to use volumetric scattering or particle systems. Particles would have the advantage of being able to interact with forces. For example, the effect of wind could be simulated, but the computational cost would be higher. The change in the background illumination can be simulated using a lamp placed at infinity (parallel rays without distance falloff). For the street, different BRDFs and textures can be used to simulate the reflection properties and roughness.

6. Conclusions

This paper shows that it is possible to use an automotive LIDAR sensor to differentiate between four different weather types: Clear, Rain, Fog and Snow. Two alternatives were presented, using the detections in the atmosphere region or using the detections on asphalt. Additionally, the presence of background radiation was taken into consideration. For the atmosphere region the F1-Scores were: Clear (85.30 w/o BR, 92.54 w BR), Rain (98.82 w/o BR, 93.27 w BR), Fog (84.82 w/o BR, 89.01 w BR) and Snow (99.27 w/o BR, 96.86 w BR). For the street region the F1-Scores were: Clear (91.29 w/o BR, 96.67 w BR), Rain (99.43 w/o BR, 99.09 w BR), Fog (91.42 w/o BR, 96.81 w BR) and snow (99.76 w/o BR, 98.86 w BR)

Different parameters were defined and used as input for the classifier based on the changes in the distribution of the detection distances for each weather type as well as the number of echoes and the EPW.

For the atmosphere region, the average particle size and density seem to be the most important physical parameters that influences the total number of detections. For the street region, the surface albedo and scattering profile are the most important parameters.

The presence of background illumination increases the noise and hence reduces the total number of detections and the EPW.

The classification based on the atmospheric region does not depend on the road surface or the angle of the LIDAR sensor and hence can be directly used in a moving car. The classification results can be combined with the information provided by other sensors in the car or cooperative with other cars [47] in order to increase the confidence level.

An algorithm like the one presented can help to improve the evaluation of road friction and be an input for other sensors or semi-/autonomous functions whose performance depends on local weather [48,49].

Author Contributions: Conceptualization, J.R.V.R. and T.G.; methodology, J.R.V.R. and T.G.; software, V.T. and J.R.V.R.; validation, J.R.V.R., V.T. and T.G.; investigation, J.R.V.R.; resources, J.R.V.R. and B.B.; data curation, V.T. and J.R.V.R.; writing—original draft preparation, J.R.V.R. and V.T.; writing—review and editing, J.R.V.R., T.G., J.C. and B.B.; visualization, J.R.V.R.; supervision, T.G., J.C. and B.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Acknowledgments: The authors would like to thank Olaf Schubert, Thomas Maag and Mario Berk from the Development Radar/Laser Sensors Automated Driving Department of Audi AG, Ingolstadt, Germany for helpful recommendations (O.S, T.M) and support with the statistical evaluation (M.B).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ABS	Anti-lock braking system
ADAS	Advanced driver assistance systems
BR	Background radiation
EPW	Echo pulse width
ESC	Electronic stability control
GNSS	Global navigation satellite system
KNN	K-nearest neighbor
LIDAR	Light detection and ranging
SVM	Support vector machine
TCS	Traction control system
TPR	True positive rate
V2X	Vehicle-to-everything
VIS	Visible spectrum (340 to 740 nm)
VLP	Velodyne LIDAR puck

Appendix A

Placement of the LIDAR sensor.

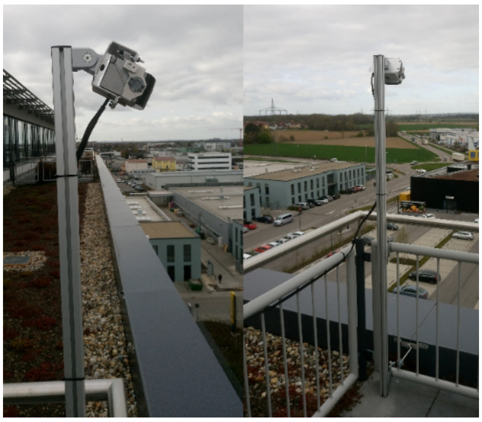


Figure A1. Measurement setup showing the placement of the LIDAR sensor. The location was chosen to avoid the influence of dust caused by cars and local weather variations caused by nearby high buildings or other structures.

Appendix B

Relevant probability distributions of weather-related variables.

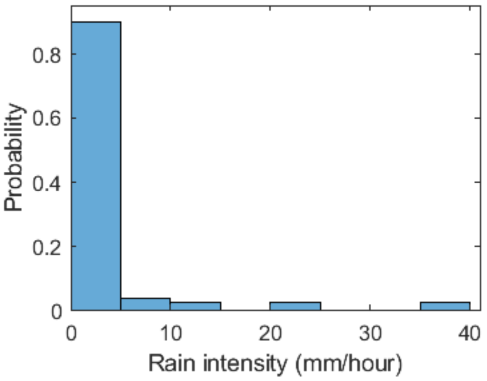


Figure A2. Histogram of the rain intensity for the Rain samples. In most cases the rain intensity was less than 5 mm/h.

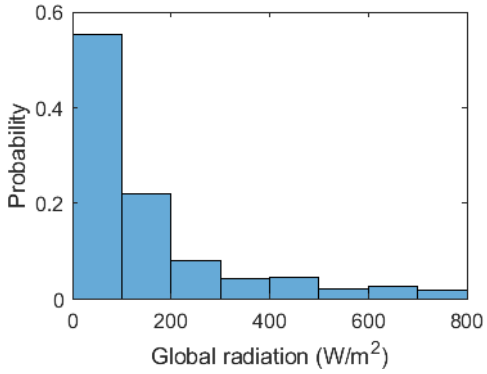


Figure A3. Histogram of the global radiation for all samples with background radiation. The average value varies depending on the class: 343 W/m^2 (Clear), 104 W/m^2 (Rain), 86 W/m^2 (Fog), 13 W/m^2 (Snow).

References

1. Wojtanowski, J.; Zygmunt, M.; Kaszczuk, M.; Mierczyk, A.Z.; Muzal, M. Comparison of 905 nm and 1550 nm semiconductor laser rangefinders' performance deterioration due to adverse environmental conditions. *Opto-Electronics Rev.* **2014**, *22*, 183–190. [CrossRef]
2. Filgueira, A.; González-Jorge, H.; Lagüela, S.; Díaz-Vilariño, L.; Arias, P. Quantifying the influence of rain in LiDAR performance. *Meas.* **2017**, *95*, 143–148. [CrossRef]
3. Rasshofer, R.H.; Spies, M.; Spies, H. Influences of weather phenomena on automotive laser radar systems. *Adv. Radio Sci.* **2011**, *9*, 49–60. [CrossRef]
4. Schöner, H.P. The role of simulation in development and testing of autonomous vehicles. In Proceedings of the Driving Simulation Conference, Stuttgart, Germany, 7 September 2017.
5. Wood, M.; Knobel, C.; Garbacik, N.; Wittmann, D.; Liu, S.; Syguda, S.; Wiltschko, T.; Weast, J.; Dornieden, B. Safety first for automated driving. Available online: https://newsroom.intel.com/wp-content/uploads/sites/11/2019/07/Intel-Safety-First-for-Automated-Driving.pdf (accessed on 5 July 2019).
6. Reif, K. *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*, 1st ed.; Wiesbaden: Vieweg+ Teubner: Wiesbaden, Germany, 2010.
7. Shimano, M.; Okawa, H.; Asano, Y.; Bise, R.; Nishino, K.; Sato, I. Wetness and Color from a Single Multispectral Image. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–27 July 2017; pp. 321–329.
8. McKeen, L.W. *The Effect of UV Light and Weather on Plastics and Elastomers*; William Andrew: Norwich, NY, USA, 2013.
9. Rivero, J.R.V.; Tahiraj, I.; Schubert, O.; Glassl, C.; Buschardt, B.; Berk, M.; Chen, J. Characterization and simulation of the effect of road dirt on the performance of a laser scanner. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6. [CrossRef]

10. Gaudio, P.; Gelfusa, M.; Malizia, A.; Parracino, S.; Richetta, M.; De Leo, L.; Perrimezzi, C.; Bellecci, C. Detection and monitoring of pollutant sources with Lidar/Dial techniques. *J. Physics: Conf. Ser.* **2015**, *658*. [CrossRef]
11. Ryde, J.; Hillier, N. Performance of laser and radar ranging devices in adverse environmental conditions. *J. Field Robot.* **2009**, *26*, 712–727. [CrossRef]
12. Guo, J.; Zhang, H.; Zhang, X.-J. Propagating Characteristics of Pulsed Laser in Rain. *Int. J. Antennas Propag.* **2015**, *2015*, 1–7. [CrossRef]
13. Dura, M. Modeling the effect of precipitation on automotive LIDAR detection capability. Master's Thesis, Technical University of Munich, Munich, Germany, 2017.
14. Dannheim, C.; Icking, C.; Mader, M.; Sallis, P. Weather Detection in Vehicles by Means of Camera and LIDAR Systems. In Proceedings of the Sixth International Conference on Computational Intelligence, Communication Systems and Networks, Tetova, Macedonia, 3–5 June 2014; pp. 186–191.
15. Charron, N.; Phillips, S.; Waslander, S.L. De-noising of Lidar Point Clouds Corrupted by Snowfall. In Proceedings of the15th Conference on Computer and Robot Vision (CRV), Toronto, ON, Canada, 9–11 May 2018; pp. 254–261.
16. Kashani, A.G.; Olsen, M.J.; Parrish, C.E.; Wilson, N. A Review of LIDAR Radiometric Processing: From Ad Hoc Intensity Correction to Rigorous Radiometric Calibration. *Sensors* **2015**, *15*, 28099–28128. [CrossRef] [PubMed]
17. Bartell, F.O.; Dereniak, E.L.; Wolfe, W.L. The Theory and Measurement of Bidirectional Reflectance Distribution Function (BRDF) and Bidirectional Transmittance Distribution Function (BTDF). In Proceedings of the Huntsville Technical Symposium, Radiation Scattering in Optical Systems, Huntsville, AL, USA, 3 March 1981; pp. 154–160.
18. He, W.; Sima, B.; Chen, Y.; Dai, H.; Chen, Q.; Gu, G. A correction method for range walk error in photon counting 3D imaging LIDAR. *Opt. Commun.* **2013**, *308*, 211–217. [CrossRef]
19. Holder, M.; Rosenberger, P.; Bert, F.; Winner, H. Data-driven Derivation of Requirements for a Lidar Sensor Model. In Proceedings of the Graz Symposium Virtual Vehicle, Graz, Austria, 15–16 May 2018.
20. Wallace, J.M.; Hobbs, P.V. *Atmospheric science: An introductory survey*, 2nd ed.; Elsevier: Amsterdam, The Netherlands, 2006.
21. Kim, I.I.; McArthur, B.; Korevaar, E.J. Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications. In Proceedings of the Information Technologies 2000, Boston, MA, USA, 6 February 2001; pp. 26–37.
22. Thakur, R. Scanning LIDAR in Advanced Driver Assistance Systems and Beyond: Building a road map for next-generation LIDAR technology. *IEEE Consum. Electron. Mag.* **2016**, *5*, 48–54. [CrossRef]
23. Kidono, K.; Miyasaka, T.; Watanabe, A.; Naito, T.; Miura, J. Pedestrian recognition using high-definition LIDAR. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 405–410.
24. Carn, S.A. Scattering: Fundamentals of Remote Sensing. Available online: http://pages.mtu.edu/~{}scarn/teaching/GE4250/scattering_lecture.pdf (accessed on 26 April 2018).
25. Liou, K.N.; Yang, P. *Light scattering by ice crystals: Fundamentals and applications*; Cambridge University Press: Cambridge, UK, 2016.
26. Moosmüller, H.; Sorensen, C. Small and large particle limits of single scattering albedo for homogeneous, spherical particles. *J. Quant. Spectrosc. Radiat. Transf.* **2018**, *204*, 250–255. [CrossRef]
27. Awan, M.S.; Horwath, L.C.; Muhammad, S.S.; Leitgeb, E.; Nadeem, F.; Khan, M.S. Characterization of Fog and Snow Attenuations for Free-Space Optical Propagation. *J. Commun.* **2009**, *4*, 533–545. [CrossRef]
28. Mohan, M.; Payra, S. Aerosol Number Concentrations and Visibility during Dense Fog over a Subtropical Urban Site. *J. Nanomater.* **2014**, *2014*, 1–6. [CrossRef]
29. Tokay, A.; Bringi, V.; Huang, G.; Schoenhuber, M.; Basho, P.; Wolff, D.; Hudak, D.; Skofronick-Jackson, G.; Petersen, W. Snowflake Size Distribution Measurements in South. Central Ontario, Canada. Available online: https://pmm.nasa.gov/sites/default/files/document_files/parsivel_Tokay_c3vp_agu.pdf (accessed on 26 April 2018).
30. Ditze, M.; Golatowski, F.; Laum, N.; Várhelyi, A.; Gustafsson, S.; Geramani, K. A survey on intelligen vehicle safety systems for adverse weather conditions. In Proceedings of the FISITA World Automotive Congress, Budapest, Hungary, 30 May–4 June 2010; pp. 1491–1498.

31. Andersson, M.; Bruzelius, F.; Casselgren, J.; Hjort, M.; Löfving, S.; Olsson, G.; Rönnber, J.; Sjödahl, M.; Solyom, S.; Svendenius, J.; et al. Road Friction Eestimation, Part II: IVSS Project Report. Available online: http://fudinfo.trafikverket.se/fudinfoexternwebb/Publikationer/Publikationer_001101_001200/Publikation_001109/IVSS_RFEII_Slutrapport.pdf (accessed on 10 November 2010).
32. Heinzler, R.; Schindler, P.; Seekircher, J.; Ritter, W.; Stork, W. Weather Influence and Classification with Automotive Lidar Sensors. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; p. 7.
33. Dannheim, C.; Mader, M.; Loewenau, J.; Icking, C.; Massow, K. A novel approach for the enhancement of cooperative ACC by deriving real time weather information. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), Hague, The Netherlands, 6–9 October 2013; pp. 2207–2211.
34. Groß, S.; Esselborn, M.; Weinzierl, B.; Wirth, M.; Fix, A.; Petzold, A. Aerosol classification by airborne high spectral resolution lidar observations. *Atmos. Chem. Phys. Discuss.* **2013**, *13*, 2487–2505. [CrossRef]
35. Papagiannopoulos, N.; Mona, L.; Amiridis, V.; Binietoglou, I.; D'Amico, G.; Guma-Claramunt, P.; Schwarz, A.; Alados-Arboledas, L.; Amodeo, A.; Apituley, A.; et al. An automatic aerosol classification for earlinet: Application and results. *EPJ Web Conf.* **2018**, *176*. [CrossRef]
36. Koskinen, S.; Peussa, P. Friction: Final Report. Available online: https://trimis.ec.europa.eu/sites/default/files/project/documents/20130411_151442_58182_FRICTION_FinalReport_D13.pdf (accessed on 26 June 2009).
37. Bijelic, M.; Gruber, T.; Ritter, W. A benchmark for lidar sensors in fog: Is detection breaking down? In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018.
38. Shamsudin, A.U.; Ohno, K.; Westfechtel, T.; Takahiro, S.; Okada, Y.; Tadokoro, S. Fog removal using laser beam penetration, laser intensity, and geometrical features for 3D measurements in fog-filled room. *Adv. Robot.* **2016**, *30*, 1–15. [CrossRef]
39. James, J.K.; Puhlfürst, G.; Golyanik, V.; Stricker, D. Classification of LIDAR Sensor Contaminations with Deep Neural Networks. In Proceedings of the Computer Science in Cars Symposium (CSCS), Munich, Germany, 13–14 September 2018; p. 8.
40. Mobley, C.D. *Light and Water: Radiative Transfer in Natural Waters*; Academic Press: New York, NY, USA, 1994.
41. Bayerisches Landesamt für Umwelt, Gewässerkundlicher Dienst Bayern. Available online: https://www.gkd.bayern.de/de/meteo/niederschlag/kelheim/hepberg-200106/download (accessed on 30 July 2020).
42. Markvart, T.; McEvoy, A.; Castaner, L. *Practical Handbook of Photovoltaics: Fundamentals and Applications*; Elsevier: Amsterdam, The Netherlands, 2003.
43. Pavement Albedo. Available online: https://web.archive.org/web/20070829153207/http://eetd.lbl.gov/HeatIsland/Pavements/Albedo/ (accessed on 18 October 2019).
44. Sasaki, Y. The truth of the F-measure. Available online: https://www.researchgate.net/publication/268185911_The_truth_of_the_F-measure/citation/download (accessed on 18 October 2019).
45. Fink, C.; Russ Moulton Jr, J.; Bybee, D.; George, K. GPU Raytracing for real-time sensor-band phenomenology modeling. In Proceedings of the IMAGE Society, Dayton, OH, USA, 22–23 June 2012.
46. Kavak, Ç. GPU Based Infrared Signature Modeling and Scene Simulation. Master's Thesis, Middle East Technical University, Ankara, Turkey, 2014.
47. Jalalmaab, M.; Pirani, M.; Fidan, B.; Jeon, S. Cooperative Estimation of Road Condition Based on Dynamic Consensus and Vehicular Communication. *IEEE Trans. Intell. Veh.* **2018**, *4*, 90–100. [CrossRef]
48. Cheng, G.; Wang, Z.; Zheng, J.Y. Modeling Weather and Illuminations in Driving Views Based on Big-Video Mining. *IEEE Trans. Intell. Veh.* **2018**, *3*, 522–533. [CrossRef]
49. Brunker, A.; Wohlgemuth, T.; Frey, M.; Gauterin, F.; Alexander, B. Odometry 2.0: A Slip-Adaptive EIF-Based Four-Wheel-Odometry Model for Parking. *IEEE Trans. Intell. Veh.* **2018**, *4*, 114–126. [CrossRef]

The Effect of Spray Water on an Automotive LIDAR Sensor: A Real-Time Simulation Study

Jose Roberto Vargas Rivero, Thiemo Gerbich, Boris Buschardt, and Jia Chen, *Member, IEEE*

***Abstract*—This paper presents the first of its kind real-time simulation of the effect of spray water on an automotive LIDAR sensor. The simulation is based on physically measurable quantities in order to facilitate its validation and extension. Both the sensor and the environment are simulated using open source software tools: Blender and Cycles, with the objective of facilitating standardization and broader adoption. The parameters required to construct and calibrate the simulation are extracted from real measurements done on a test area and highway covering different water film thicknesses and vehicle speeds. The simulation is validated in different stages, first by comparing the trajectories of the particles with a regular solver and secondly, by the spatial distribution of the virtual and real point clouds with the measurement results. The resulting framework can be easily expanded to cover scenarios that are more complex and other phenomena counterproductive to the performance of a LIDAR sensor like dirt, exhaust gases, snow, rain and fog. The virtually generated point clouds can be used to validate the performance of the software required for the segmentation, classification and tracking of objects generated by the LIDAR sensor, thereby reducing development costs and increasing software quality.**

***Index Terms*—LIDAR sensor, self-driving car, real time simulation, spray water, virtual point cloud, ray tracing, particle system.**

I. Introduction

TRADITIONALLY, the testing of self-driving (SD) and ADAS (Advanced Driver Assistance System) functions has been done open-loop, reprocessing data captured with test vehicles for each new software version.

The performance of the software (SD or ADAS function) is verified against collected data in a series of tests evaluating each successive abstraction level, going from the perception to the decision-making modules. Although this approach provides the highest level of realism, it is also limited in practice not only due to the nature of the open loop testing but also due to the increasing costs, complexity and time required for driving all the required scenarios in reality [1]–[3].

An alternative is to use simulation to expand the real data by introducing thousands of variations to a real scenario and creating synthetic scenarios based on known weaknesses of the system, which are difficult or impossible to test in reality.

Nevertheless, the use of simulation has been limited in some cases due to the "reality gap". The reality gap refers to the differences between simulation and physical system. These differences can in some cases be ignored but in some cases, they are fundamental and can cause a simulation to diverge completely from reality [4], [5].

In order to reduce the reality gap, there is a tendency towards the use of a photorealistic- and physics-based simulation in academic and commercial products [6]–[8]. Physically based sensors and actuators can potentially reduce the validation effort by using models known to be correct. However, their use has been limited by the significant hardware resources that an accurate reproduction of the physics of the problem requires, especially under the constraint of having to generate results in real time or faster than real time in order to speed up the validation process.

Recently, however, ray tracing has reached a level in which is possible to trace millions of rays in few milliseconds using GPUs. With this technique it is possible to simulate: LIDAR, Radar [9], ultrasonic sensors [10] and cameras [8], [11]. Besides ray tracing, which is done by the render engine, the movement and interaction of the different objects in the environment, can be simulated using a physics engine [12]–[15] like Pybullet [16] or PhysX [17] or can be animated. The animation can be done manually or data driven [18].

Based on the previously mentioned developments, this paper proposes the use of the Cycles [19] physically-based render for the simulation of a LIDAR sensor together with the physics and animation engine in Blender [20]. These tools provide the basis for a testing framework in which a broad range of complex scenarios can be simulated. The reality gap is reduced by adjusting the simulation based on real measurements and validating the relevant parts of the physics engine using MATLAB [21]. Specifically, the effect of spray water on the detections of a LIDAR sensor is simulated. Spray water was chosen because it generates extra detections [22] on different positions of the LIDAR field of view depending on multiple factors including wind forces. These spray detections can be critical in a driving situation like the one illustrated in Fig. 1. In this case, the segmentation and classification algorithm detects an 'L' shape on the spray cloud and creates an object hypothesis. In the following frames, the hypothesis is not refuted and an unknown object is

Manuscript received June 29, 2020; revised November 14, 2020; accepted March 7, 2021. Date of publication March 23, 2021; date of current version April 21, 2022. *(Corresponding authors: Jose Roberto Vargas Rivero and Jia Chen.)*

Jose Roberto Vargas Rivero is with Audi AG, Ingolstadt 85045, Germany (e-mail: jose-roberto.vargas-rivero@audi.de).

Thiemo Gerbich and Boris Buschardt are with Audi AG, Ingolstadt 85045, Germany (e-mail: thiemo.gerbich@audi.de; boris.buschardt@audi.de).

Jia Chen is with the Technische Universität München, Umweltsensorik und Modellierung, München 80333, Germany (e-mail: jia.chen@tum.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2021.3067892.

Digital Object Identifier 10.1109/TIV.2021.3067892

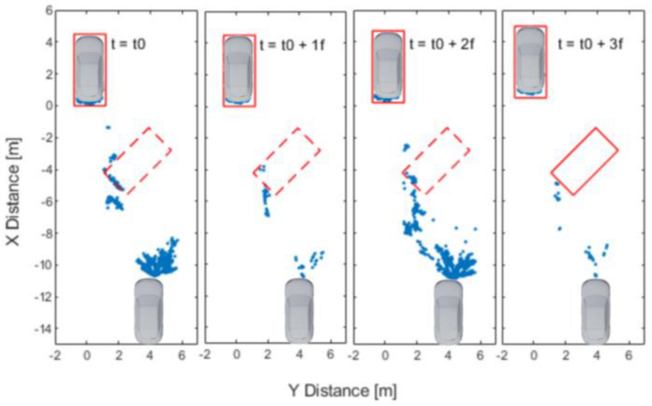


Fig. 1. Sequence illustrating the creation of a ghost object due to LIDAR detections caused by spray water. Initially, the points are accidentally ordered in a way in which a partial ‘L’ shape similar to the one for a real car is formed. Consecutive frames do not totally contradict the hypothesis and hence a ghost object is created after three frames. Although this object exists for a very short time, it could cause problems for the function using the sensor data.

created. Although the duration of this object can be only a couple of frames, it increases the risk of phantom breaking, especially if a second sensor like a radar or camera accidentally validates its existence.

To reduce the risk in a scenario with a large amount of spray, the following alternatives exist:

- Deactivate the function and ask the driver to take control.
- Degrade the function by reducing the speed, for example.
- Filter out the detections/objects caused by the spray cloud.

From the driver’s perspective, the third alternative is the most comfortable but it requires that the system correctly identifies which detections/objects are being caused by the spray. This could be achieved using supervised or unsupervised learning on collected point cloud data [23]. As this data needs to be captured for different vehicle speeds, rain intensities, car and asphalt types, etc., it would constitute a highly time and cost consuming task. Another alternative would be to improve the accuracy of the object generation algorithms (segmentation and classification) by adding noise to “spray free” point clouds [24], [25]. Although this option can be useful to simulate detections caused by rain or snow, spray has the characteristic of generating regions with a high number of detections (as shown in Fig. 1). These regions are not randomly distributed but depend on the forces that act upon the spray drops. A third alternative is to inject a false object hypothesis into the sensor fusion algorithm and improve it in order to reject those objects [26]. This alternative, however, has the same disadvantages as the first two depending on how these extra objects are generated.

Therefore, the alternative followed in this paper is to combine the data collection efforts with a partially physically based simulation model. Instead of simulating real scattering, particles are used to generate the required number of detections. Their trajectories are adjusted based on the real forces that affect the movement of spray drops. The obtained model generates similar point clouds as the ones collected and is versatile enough to generate with simple variations a vast amount of new data for the validation of SD and ADAS functions.

The theoretical basis for the construction of the simulation is presented in the next section. Section III presents the methodology used for data collection and Section IV presents the obtained results. The Results section is divided into four parts: parts A and B present real measurements from spray clouds collected in a proving ground and highway. Part C combines the theoretical background and the results presented in the previous two parts to construct the simulation. Finally, part D compares the simulation results with the measurements.

II. Background

In the path of the laser beam from the sensor and back, there are different stages in which its intensity and direction can be changed. The first is the cover of the sensor. The changes on the material properties of the cover and any possible solid or liquid deposited on it greatly influences its performance [27]. Going further are the effects caused by rain, snow, fog and exhaust gases, which generate extra detections mostly in the near region [28]–[30]. Spray is more complicated as the detections are distributed depending on multiple factors like [31], [32]:

- Vehicle shape.
- Vehicle load.
- Ambient wind.
- Type of asphalt.
- Water film thickness.
- Street topology.
- Wheel type.
- Asphalt temperature.
- Target and ego car velocities and relative position.
- The presence of spray suppression devices.

The water film thickness can be related to the rainfall intensity using [32]:

$$\boldsymbol{h} \ = \boldsymbol{k_s} \ \cdot \boldsymbol{T_s}^{0.3} \cdot (\boldsymbol{L_s I_R})^{0.5} \cdot \boldsymbol{S_s}^{-0.3} \tag{1}$$

where h is the water film thickness, T_s is the texture depth which characterizes the porosity of the material, L_s and S_s correspond to the drainage length and surface slope of the street, I_R is the rainfall intensity and k_s is a constant [32]. The values of T_s can go from 0.208 [mm] for smooth concrete to 1.644 [mm] for porous asphalt. Although this formula ignores evaporation effects, together with (2) and (3) it shows the main variables that at the end define the average size of the spray drops.

The next section will cover the spray generation from the wheel and its dispersion; further dependencies important for the simulation will be discussed in the Results section.

A. Spray Generation

A useful distinction is to separate the spray phenomenon into splash and spray. Splash is the name given to large droplets, which move mostly following an inertial trajectory (large Stokes number), whereas spray is the name given to smaller particles that remain suspended in the air and follow its streamlines (small Stokes number) [31], [33].

A systematic classification from the spray/splash contribution from the different parts of the wheel without any spray suppression device is shown in Fig. 2 [31]. The bow and side waves

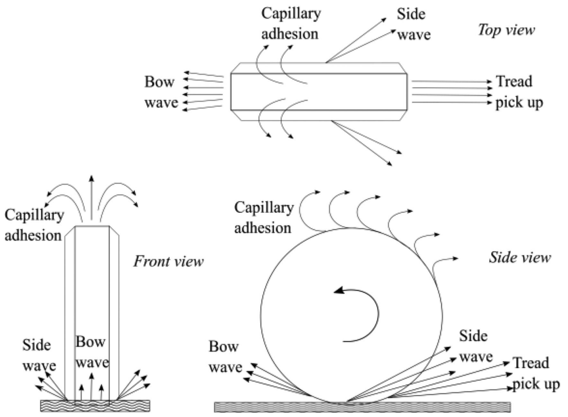


Fig. 2. Different ways in which splash and spray are generated from a wheel [31]. The bow and side wave are considered splash because of the large droplet size. The capillary adhesion and the tread pick up generate smaller drops which are considered spray [34].

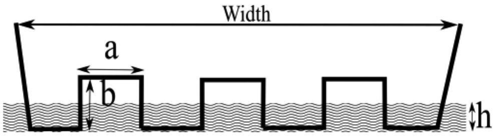


Fig. 3. Parameters used to approximate the amount of water captured by the wheel treads, a and b correspond to the width and height of the tread while h corresponds to the water film thickness [31].

generate primarily splash. The water captured in the tread is mostly ejected in the direction marked as tread pick up. The water remaining due to thin film capillarity is blown away by the wind when it reaches the top of the wheel [31].

A simple model of the amount of water that is picked up by the treads for water levels smaller or equal to 3 [mm] uses the tread height *b*, width *a* and water level *h* as shown in Fig. 3 [31].

The amount of water picked up per second is:

$$\dot{\boldsymbol{w}}_{\boldsymbol{tread}} = \frac{\boldsymbol{h}}{3} \cdot \boldsymbol{v_c} \cdot \boldsymbol{n_t} \cdot \boldsymbol{a} \cdot \boldsymbol{b} \tag{2}$$

where besides *a*, *b* and *h,* the tire velocity v_c is also considered. The width is indirectly represented by the parameter: n_t which corresponds to the number of grooves in the wheel [31].

The average size of the drops generated by the tread pick up was found to be inversely proportional to the square root of the Weber number [35] which relates the fluid inertia with its surface tension [36].

$$\boldsymbol{D_{mean}} = \frac{\boldsymbol{k}}{\boldsymbol{v_c}} \sqrt{\frac{\boldsymbol{\sigma}}{\boldsymbol{\rho b}}} \tag{3}$$

where D_{mean} is the average drop size diameter, v_c and *b* are as defined before, σ is the surface tension for water against air, ρ is the density of water and *k* is a proportionality constant [35], [36].

B. Spray Dispersion

A precise calculation of the trajectories of the droplets can be done using CFD (Computational Fluid Dynamics) software, frequently used to predict the possible surface contamination (“soiling”) pattern in a vehicle [33], [34], [37]. Nevertheless, as mentioned in the introduction section, the proposed simulation should speed up the development of self- driving functions and hence run ideally in real time. For this reason, approximations are used that although less exact, roughly replicate the trajectories of the drops, or more specifically, the detections that those drops cause on the LIDAR sensor. Concretely, the wind speed and particle size are assumed to be constant. The effect of the gravity, drag and wind forces upon the particles trajectory can hence be calculated using a fast numerical integration method. The injection velocity of the particles is calculated based on the wheel rotation speed, as will be shown in Section IV-C.

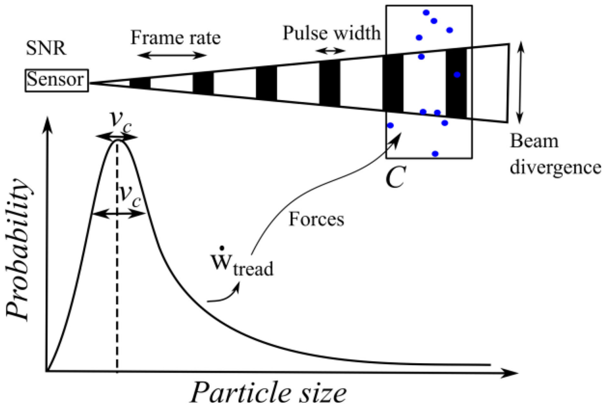


Fig. 4. Parameters that influence the detection of spray by a LIDAR sensor. The wheel rotation speed v_c influences particle size distribution. The wheel type and water level influence the amount of water picked up per second $\dot{w}_{tread}$ and hence the number of particles. The different forces change the particles trajectories and generate a certain concentration C. Considering a pulsed laser depending on its frame rate, pulse width, divergence and SNR a detection will take place or not [35], [36], [41].

Additionally, turbulence is assumed, as in the Pasquill-Gifford dispersion relation [38], [39], to be caused by small eddies that move the particles laterally [31]. These eddies are simulated using the curl of a Perlin noise texture [12].

C. Spray Detection

Based on the trajectories of the particles a certain spatial distribution is created which is sampled by the sensor at regular intervals. A detection takes place when the concentration of particles with a certain size scatter an amount of energy back in the direction of the receiver that is higher than the current SNR (Signal-to-Noise Ratio) [40]. As the SNR is variable due to a changing background illumination [30], for example, the point at which a detection takes place will change dynamically over time. Fig. 4 shows the relation of the main parameters that influence whether a detection takes place or not. The wheel speed v_c influences the shape and mean value of the particle size distribution. The detection itself happens when a certain concentration C is reached for a certain SNR value. The concentration depends on the trajectories of the particles and correspondingly on the forces that act upon them. Additionally, the concentration depends on the water picked up per second by the wheel.

D. Simulation

In this section, the tools and concepts used for the simulation are introduced.

1) Blender and Cycles: As will be discussed in section IV-C in order to reconstruct a scene as shown in Fig. 1 in simulation, two main tools were needed: a physics engine and a physically based render. Both capabilities are available in Blender. Blender is an open source 3D creation suite which has been already used for robotics [42] and sensor simulations [43]. It includes a physically based path tracer called Cycles [44].

As a 3D creation suite, the use of Blender in scientific simulations has the disadvantage that it has to be adjusted and validated for the specific use case [45]. Depending on the required accuracy however, these kind of tools can provide very high frame rates and therefore their use for the generation of synthetic data has increased [46]–[48]. The proposed method is not limited to Blender. Some alternative physics engines besides those already mentioned (Pybullet [16], PhysX [17]) are: ODE [49] and Havok [50]. Some alternative physically based renderers are: Arnold [51], RenderMan [52] and LuxCoreRender [53].

2) Particles System: Particles are defined as a collection of objects which are often represented using a simple shape [54]. Particles are characterized by having a source called an emitter. Each particle also possesses a lifespan, which starts from the moment the particle is created. The particles disappear when their lifespan is reached or they collide with certain objects [54]. Besides having a shape, particles also possess a material property that defines how they are rendered. Physically, particles are treated as point masses with a trajectory depending on the forces acting upon them [55].

3) Particle Systems Toolbox: The Particles System Toolbox (PST) [21] was used to validate the trajectories of the particles generated by the physics solver in Blender. PST uses a conventional fixed-step fourth-order Runge-Kutta solver in MATLAB to solve the nonlinear differential equations that govern the movement of particles under the effect of multiple forces.

4) Principled BSDF: The principled shader was used because it provides a physically accurate microfacet model. This model can be used to simulate the scattering properties of many different materials by changing its parameters without having to use different shaders [56], [57]. By fitting the BSDF (Bidirectional Scattering Distribution Function) from a measured material at the corresponding wavelength with the parameters of the principled BSDF, a material database can be created, which can be shared between different tools [58].

5) Using a Path Tracer to Calculate Distances: In order to calculate not only intensities per pixel but also the distance traversed by the rays, some modifications are required over the normal computations done with a path tracer. This is illustrated in Fig. 5 .

The resulting intensity, considering one color and one pixel, is the product of the falloff, material reflection terms and intensity of the lamp [59], [60].

By adjusting in the shader the base color of two of the primary colors in the following way [59]:

- For red, setting: $\boldsymbol{R}_3, \boldsymbol{R}_2,\ \boldsymbol{R}_1 =\ 1$

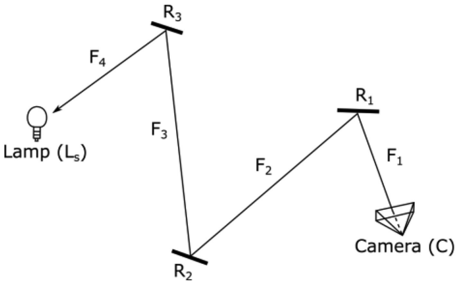


Fig. 5. Ray tracing principle. The ray(s) is/are sent from the camera (C) and is/are affected by falloff terms (F) and the reflection of the different materials on its way (R). Eventually, if the ray reaches a lamp (L_s), the intensity of the pixel can be calculated [59], [60].

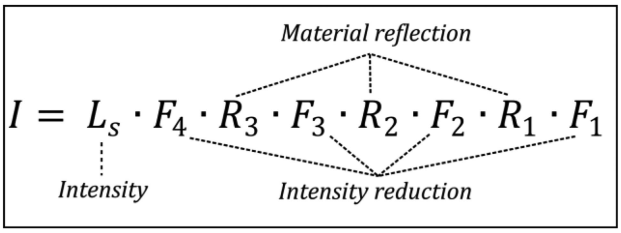

Material reflection
$I = L_s \cdot F_4 \cdot R_3 \cdot F_3 \cdot R_2 \cdot F_2 \cdot R_1 \cdot F_1$
Intensity
Intensity reduction

Fig. 6. Calculation of the intensity per pixel. The configuration of the shader for each material and the light source determine the result. The shader determines the material reflection and the light source determines the intensity and the intensity reduction parameters.

- For blue, setting: $\boldsymbol{R}_3 = \boldsymbol{a}^{\boldsymbol{x}_3}$, $\boldsymbol{R}_2 = \boldsymbol{a}^{\boldsymbol{x}_2}$, $\boldsymbol{R}_1 = \boldsymbol{a}^{\boldsymbol{x}_1}$

Where *a* is a number chosen close to one in order to avoid an overflow in the numerical calculation for very short or long distances and x_1, x_2, x_3 correspond to the length of the ray segment. The following intensities are generated:

$$\boldsymbol{I_{red}} = \ \boldsymbol{L_s} \cdot \boldsymbol{F}_4 \cdot \boldsymbol{F}_3 \cdot \boldsymbol{F}_2 \cdot \boldsymbol{F}_1 \tag{4}$$

$$\boldsymbol{I_{blue}} = \boldsymbol{L_s} \ \cdot \boldsymbol{F}_4 \cdot \boldsymbol{a}^{\boldsymbol{x}_3} \cdot \boldsymbol{F}_3 \cdot \boldsymbol{a}^{\boldsymbol{x}_2} \cdot \boldsymbol{F}_2 \cdot \boldsymbol{a}^{\boldsymbol{x}_1} \cdot \boldsymbol{F}_1 \tag{5}$$

Which means we can calculate [**59]:**

$$\boldsymbol{log_a} \left(\frac{\boldsymbol{I_{blue}}}{\boldsymbol{I_{red}}} \right) = \boldsymbol{x}_3 + \boldsymbol{x}_2 + \boldsymbol{x}_1 \tag{6}$$

which corresponds to the total path length of the ray. If we assign to the green channel the real reflection of the materials at the sensor's wavelength, it is possible to get, after the render finishes, a matrix containing the distance and intensity per ray.

For the method to work correctly, anti-aliasing needs to be deactivated. Additionally, the number of samples in the render needs to be adjusted to try to reach a compromise between precision and rendering time.

III. Methodology

The methodology is divided into two parts. Part A explains how the sensor is simulated. Part B describes the physical setup and the characterization of the water film measurement device (WFMD).

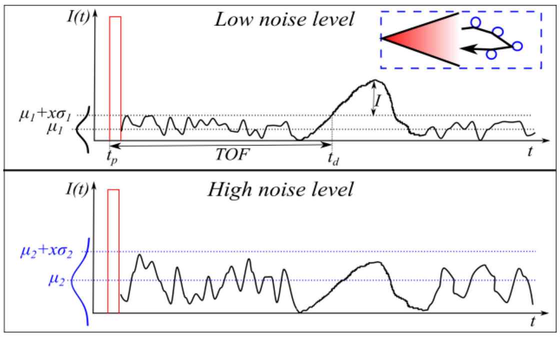


Fig. 7. Electrical signals generated by the receiver in a LIDAR sensor under low and high noise levels. Multiple parameters combine to create a signal, which may or may not be over the minimum required level over noise (here defined as a certain number x of standard deviations σ from the mean noise value µ) [30]. Although in reality, the scattering is a very complex process (dotted blue box) similar results can be generated in simulation using particles if their number is adjusted based on the real number of detections.

A. Sensor Simulation

As mentioned in Fig. 4 and supplemented in Fig. 7 the process through which a certain concentration of particles generates a detection in a time of flight LIDAR depends on multiple factors [40], [61]:

- The particles distribution, sizes, shapes, and compositions.
- The intensity and shape of the light source pulse.
- The wavelength of the LIDAR.
- The divergence of the light source beam.
- The receiver parameters.
- The scanning pattern.

When the detection process is considered for the whole field of view of the sensor, over multiple frames, a certain probability distribution for the number of detections per frame is generated (Fig. 12, 14).

A purely physically based simulation including all these factors together with the factors mentioned in the background section would be too complex to be done with a high frame rate. For that reason as mentioned in the introduction section, a hybrid model is proposed. In this model, the distribution of the number of spray detections per frame is learnt from the real data. This information is used to adjust the number and lifetime of the particles used for the simulation. These particles do not correspond in size and number to real spray drops. Their size is optimized based on the divergence of the sensor light source to generate a similar number of detections per frame.

With this approach, the sender and receiver can be simplified. Their purpose is to illuminate the scene and provide a certain intensity per pixel, which can be compared to a fixed noise level. If the intensity is higher than the noise level, a detection is registered.

This approach has the following advantages:

- The computing power required to calculate the particles' trajectories is reduced to a minimum making high frame rates possible.
- Although the used particles do not correspond to real drops, a detection still depends on their relative position, size, material and distance from the light source.

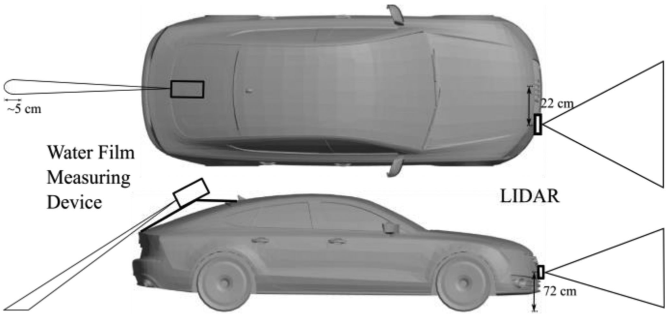


Fig. 8. Water film measuring device placed on the back of a car having also a LIDAR sensor placed in the grill region. The information provided by both devices is internally synchronized.

- The formation of complex patterns, as the one shown in Fig. 1 are still possible as long as the particles remain smaller than the angular and distance resolution of the used LIDAR.
- Because the mesh size is independent from the mass of the particles, the calculation of the trajectories is not affected by this approximation.
- As particles can collide and are effected by forces, the versatility of the model is high.

The following disadvantages are however present:

- The model is valid only for the LIDAR technology used to collect the data.
- The acquired distribution will depend on the background radiation and rain intensity present during the measurements [30]. For that reason our measurements were done in intervals where those conditions remained more or less constant.
- The spray detections per frame distribution does not provide information about the spatial distribution of the particles. The model parameters need to be adjusted to achieve an acceptable spatial distribution.

To complete the sensor model, the modified path tracer mentioned on Section II-D can be used by placing the lamp relative to the camera as in the real sensor and adjusting the parameters of both: camera and lamp, to approximate those of the real sensor.

Because in each frame the lamp illuminates the whole field of view of the sensor, the scanning pattern is ignored for all objects except for the particles, for which it implicitly affects their number. This may be a drawback for some use cases but is an acceptable approximation in our case.

B. Experimental Setup

The setup consisted of a device to measure the water film thickness (Water Film Measuring Device) placed at a distance around 2 m diagonally from the asphalt and at an angle of around 45°. The device was mounted on the back as shown in Fig. 8 to be able to satisfy the recommended installation angle (35° to 65° from the street) without blocking the field of view of the LIDAR sensor or the driver. It was placed in the middle to reduce the impact from the spray generated by the wheels. This device

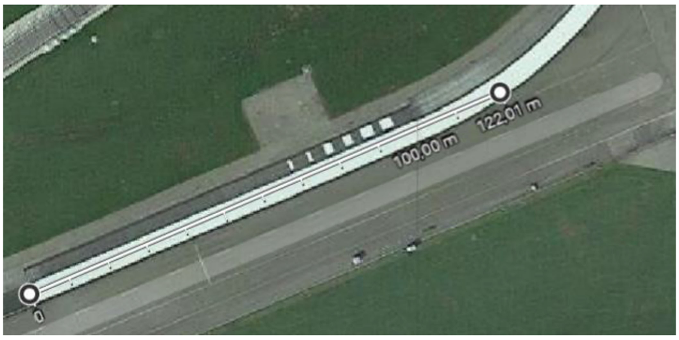


Fig. 9. Measurement in the proving ground. Track length of approximately 122 m. Surface covered by white tiles and filled with water [64].

uses an optical spectroscopic measuring principle to measure the amounts of water, ice, and snow on the street [62].

The same car had an automotive LIDAR sensor [63] in front from which the point cloud and object list were collected. The two devices were synchronized using their UTP timestamps.

1) Measurement of Water Level at Walking Pace: In order to get an impression about the characteristics of the WFMD, a measurement of water level was done in a porous asphalt region with a depression having the dimensions: 32 × 47 cm filled with water with a maximum depth of 11 mm. The spot of the sensor was driven over the water region at walking pace multiple times going forwards and backwards. Additionally, the film thickness of the same region was manually measured with a caliper. The relation between the manual measurement of the layer thickness $(\boldsymbol{h_m})$ and the value provided by the WFMD $(\boldsymbol{h_s})$ is given by the equation: $\boldsymbol{h_s} = 0.36 \cdot \boldsymbol{h_m}$. The maximum value registered by the WFMD was from 4 mm. It must be mentioned that the WFMD is constructed to measure values up to 2 mm, but such small layer thickness are very difficult to measure on granular asphalt. Additionally, it seems that when the spot covers an inhomogeneous region, the reported value tends to be the smallest water thickness of the region. In order to compensate for some of these issues, in subsequent results the presented water levels are averaged over a high number of frames. Additionally, when used for classification (Fig. 14), only broad ranges are utilized.

2) Measurement of Water Level in the Proving Ground: A second type of water level measurement was done using a special track on a test area. The track is covered with white tiles. The used region had a length of approximately 122 m and is shown in Fig. 9. The whole track was filled with water, using nozzles at ground level. The water flows from the nozzles to a drainage creating an uneven distribution.

The experiment was divided into two parts: In the first part, the car with the WFMD and LIDAR sensor was driven at different speeds and positions on the water filled track in order to get an idea about the effect of increasing speeds upon the water film measurement, further described in this section. In the second part, two cars where used: One car traveled in the spray generated by the water filled track, captured by the car with the LIDAR sensor driving behind. This measurement is described in Section IV.

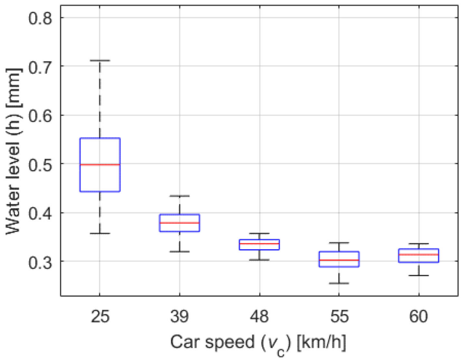


Fig. 10. Box plot showing the dependencies between measured water level and mean speed. With higher speeds, the mean water level decreases.

Due to the uneven water distribution, three water level measurements were done per speed: left, middle and right of the track. The speeds used were (25 ± 2) km/h, (39 ± 2) km/h, (48 ± 4) km/h, (55 ± 3) km/h and (60 ± 3) km/h. The average results are shown in Fig. 10.

A notable effect is the reduction of the measured water level with higher speeds. This effect could have been caused by the wind pushing the water under the car away and hence reducing the water film thickness in the place where the spot measures it. Although this effect was observed in the used track, which is covered by tiles, it could also take place in normal streets covered by an ice layer. Considering (1), when the street is covered by porous asphalt, the pores should obstruct the water being pushed by the wind, reducing the dependency between water level and vehicle speed.

IV. Results

A. Measurements in the Proving Ground

As mentioned before, a second experiment was done using two cars. The car with the LIDAR sensor (ego) and the target car were driven in a way to try to keep a distance of around 17m in x-direction and 4m in y-direction between them. The speeds used were the same as in the previous experiment: (39 ± 2) km/h, (48 ± 4) km/h, (55 ± 3) km/h and (60 ± 3) km/h, excluding the (25 ± 2) km/h as, in this case, the amount of spray detected by the sensor was very low. The track was driven six times for each speed. The accumulated point clouds for all six measurements and speeds are shown in Fig. 11. The data is filtered so that the variation in the vertical and horizontal distances between the two cars remains low. The data shown corresponds to 6858 scans. The position of the target car was extracted by tracking the middle of the line formed by the detections hitting the rear part or the car.

Given its trajectory, most of the spray seems to come from the tread pick up, only at 60 km/h some spray detections appear to come from the side wave. For speeds up to 48 km/h, the dispersion remains relative low but with higher and higher speeds, turbulence increases dispersion, and the contribution from each wheel cannot be clearly separated. Additionally, the

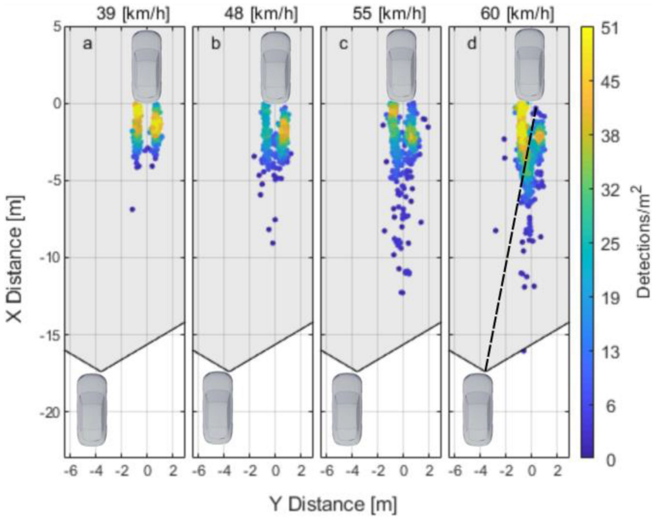


Fig. 11. Accumulated point cloud for measurements in the test area at different speeds. The field of view of the LIDAR is shown. The distance from the ego to the target car has a value of (17.4 ± 0.3) m in x and of (3.6 ± 0.3) m in y. The water level is (0.5 ± 0.08) mm. The spray cloud spreads mostly in the x direction for higher speeds due to the higher rotation speed of the wheels and the effect of the wind. The contribution of each wheel, which is initially clearly separated, is combined for the higher speeds. The effect of occlusion is shown in d. The spray from the left wheel absorbs or scatters the rays before they can reach the right wheel.

spray from the left wheel reduces the detections on the right wheel by occluding them from the perspective of the sensor.

The size of the spray cloud increases mostly longitudinally with speed. Besides the increased wind force and initial velocity of the drops, the reduction of the drop size (3) also probably causes the particles to fly farther away from the wheel given their reduced mass.

As predicted by (2), the number of spray detections and hence the amount of spray increases with speed as can be seen in Fig. 12. Here, N_{mean} corresponds to the average number of spray detections per scan. The relation between LIDAR spray detections and speed, although initially close to lineal, seems to reach saturation for higher speeds due to occlusion.

Even though the previous results help to understand these phenomena, it is unlikely to find a street area with such a constant water distribution. Most real streets are built so water can flow into a drainage as fast as possible. Besides, the effect of rain is not included. For these reasons and also in order to study the effect of higher speeds, a series of measurements were made on the regional A9 autobahn close to Ingolstadt during rainy days. Section V shows the results.

B. Measurements in the Highway

The measurements were done on three different rainy days:

In the first measurement, only one car was used in order to sample spray from different leading vehicles of similar size to include more variation in the data. In the other two measurements, two cars were used in order to coordinate the maneuver and hence be able to realize longer measurements. The ego car tried to remain on the low speed lane collecting the spray generated by the target car in the middle lane while trying to maintain a similar speed. Fig 13 shows the accumulated point cloud from the three measurements. The vertical distance was filtered into five different distances, going from 15.5 m to 32.6 m with an average standard deviation in each case from 0.3 m. The data was filtered for a speed of (89 ±4) km/h, a water level of (0.7± 0.4) mm and a horizontal distance of (3.6 ± 0.2) m. The data shown corresponds to 13162 scans. In this case, the information provided by the sensor internal object tracker was used as ground truth for the leading vehicle position.

It was raining during the whole measurement and for that reason a very high number of detections are seen in the region from 0 to 5 m. These detections, although partially caused

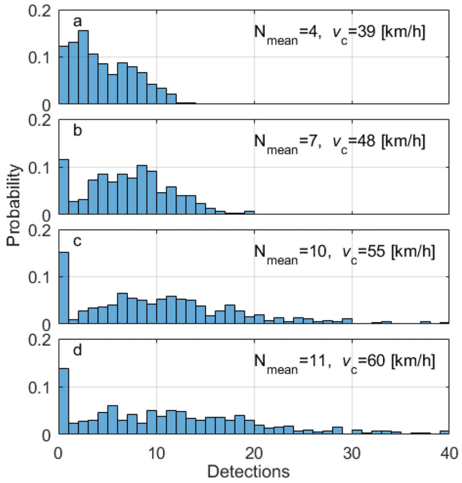


Fig. 12. Histogram showing the probability distribution of the number of spray detections under the same conditions as in Fig. 11. Bin size 1 detection. The probability of a higher number of spray detections increases with the car speed, as also the average number of spray detections per scan N_{mean}. The increment is caused by the higher wheel rotation (2).

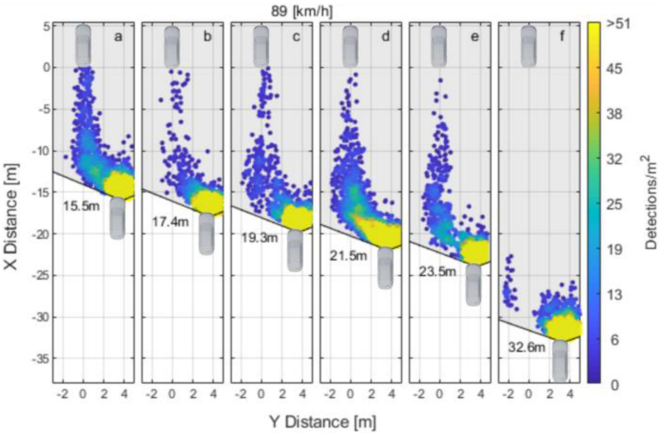


Fig. 13. Accumulated point cloud for highway measurements with an average speed of (89 ±4) km/h. The field of view of the LIDAR is shown. The x distance of the ego to target car is shown in the figure. The y distance from ego to target has a value of (3.6± 0.2) m. The water level is (0.7 ± 0.4) mm. With increasing distance between the ego and the target car, fewer detections happen on the spray cloud due to the exponential decrease in intensity of the LIDAR with distance. After a certain distance, mostly only the detections corresponding to raindrops remain.

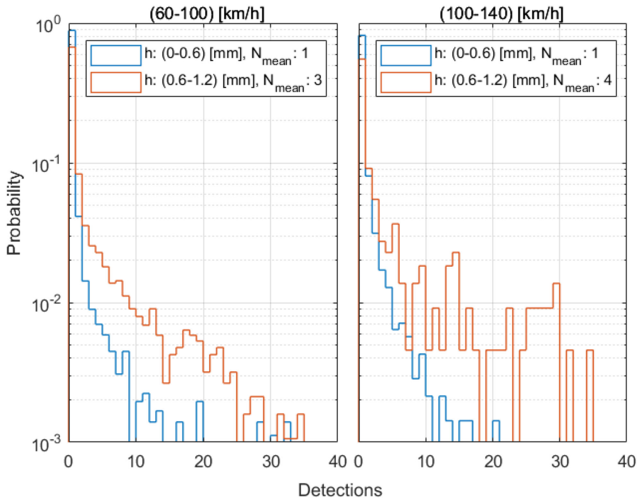


Fig. 14. Histogram showing the probability distribution of the number of spray detections for a speed of 60 to 100 km/h on the left side and 100 to 140 km/h on the right side for measurements in the highway. Detections with a distance to the ego car smaller than 5 m are subtracted from the sample. The distance from ego to target car has a value of (17.5 ± 0.3) m in x and (3.7 ± 0.1) m in y. Bin size 1 detection. As also seen in the test area (Fig. 7), the probability of a higher number of spray detections increases slightly with speed but most notably with water level. Due to the increased noise level caused by rain the average number of detections is smaller than it would be expected without rain (Fig. 12).

by spray water, are mostly due to raindrops. This agrees with previous results [30] in which the extra detections caused by rain, fog and snow took place mostly in this region.

In contrast to the results in the test area, the spray detections are not concentrated close to the wheels anymore and it is not possible to differentiate the contribution of each wheel. Due to occlusion and turbulences, a big and unified cloud is seen. It is also interesting to notice that after a certain distance from the ego, the number of spray detections radically decreases. This is due to the divergence of the beam, the reduction in mean drop size due to higher speeds (3) as in the test area and increased noise level due to rain. The drops acquire a dynamic corresponding to a small Stokes number tending to follow more the wind flow instead of their own inertial trajectory. They also tend to remain suspended longer in the air, thereby affecting the LIDAR detections even when the target car is far away. The net effect is an increased number of drops but a more drastic reduction in the number of spray detections with distance.

As similarly done for the test area, a histogram of the spray detections for different speeds and in this case, also for water levels, is shown in Fig. 14. The region from 0 to 5 m is filtered out. The effect of increasing the speed is relatively small for both low and high water levels. This was already the tendency seen in Fig. 12 and is attributed to occlusion. When the wheel speed increases, the water pick up per second also increases but at the same time, smaller drops are generated, the occlusion generated by this many small drops close to the sensor results in an overall reduction in the SNR, therefore only a small increment in the number of detections is seen. The change in water film thickness, on the other hand, noticeably increases the average number of spray detections as it increases the number of drops of all sizes.

In comparison with the spray measurements in the test area, the probability of zero to one detection is much higher for the highway. This is due to the highly uneven distribution of the water on the street surface and the fast drainage. Nevertheless, the maximum number of spray detections coincides with what was registered for speeds higher than 55km/h in the test area (Fig. 12). Those relatively big point clouds are the main cause of false object detections as seen in Fig. 1.

After having identified the main factors that influence the creation, dispersion and detection of spray in theory and in practice using measurements with different water levels and speeds, we proceed to introduce the simulation framework in Section IV-C. Section IV-D compares the simulation results with the measurement results presented in this section.

C. Simulation

As already mentioned in Section IV-A, most of the registered spray detections are caused by tread pick up. The spray generation section will hence focus on simulating this part. The second part related to spray dispersion mentions how the effect of the wind and drag forces is added to the simulation. These two parts complete the physical simulation in regards to the particle trajectory. Nevertheless, in order to simulate the sensor, the scene needs to be rendered and this requires defining a material and mesh size for the particles. This is done in the spray detection section.

1) Spray Generation: The physics engine simulates the particles assuming they are point masses. With this assumption and considering the frame rate of the simulation to be high enough, a simple integrator can be used to calculate the trajectories in real time [55].

Besides the solver it is important to know how the initial position and velocities of the particles are calculated and how the forces are defined.

Regarding the initial position and velocity of the particles, it was found that the vector fields coincide when using the following equations:

$$\boldsymbol{v_{vertex_x}} = -\frac{\boldsymbol{v_c \cdot i}}{\boldsymbol{N}} \cdot \sin\left(\frac{2\pi \boldsymbol{i}}{\boldsymbol{N}} - \frac{\pi}{2} + \frac{\boldsymbol{v_c \cdot \Delta t \cdot i}}{\boldsymbol{Nr}}\right) \tag{7}$$

$$\boldsymbol{v_{vertex_z}} = \frac{\boldsymbol{v_c \cdot i}}{\boldsymbol{N}} \cdot \cos\left(\frac{2\pi \boldsymbol{i}}{\boldsymbol{N}} - \frac{\pi}{2} + \frac{\boldsymbol{v_c \cdot \Delta t \cdot i}}{\boldsymbol{Nr}}\right) \tag{8}$$

$$\boldsymbol{p_{vertex_x}} = \boldsymbol{r} \cdot \cos\left(\frac{2\pi \boldsymbol{i}}{\boldsymbol{N}} - \frac{\pi}{2} + \frac{\boldsymbol{v_c \cdot \Delta t \cdot i}}{\boldsymbol{Nr}}\right) \tag{9}$$

$$\boldsymbol{p_{vertex_z}} = \boldsymbol{r} \cdot \sin\left(\frac{2\pi \boldsymbol{i}}{\boldsymbol{N}} - \frac{\pi}{2} + \frac{\boldsymbol{v_c \cdot \Delta t \cdot i}}{\boldsymbol{Nr}}\right) + \boldsymbol{r} \tag{10}$$

where v_{vertex} and p_{vertex} correspond to the initial velocity and position of the particles emitted by a vertex, N to the number of vertices, i to an index going from 0 to N, v_c to the wheel velocity, r the radius of the circle and Δt to the time difference between two frames.

Fig. 15 shows an example of a quiver plot for a circle with 36 vertices as obtained by using (11-14). The radius of the circle corresponds to a wheel of type 255/40R19, which is the wheel size of the target car used in the test area and should be a common

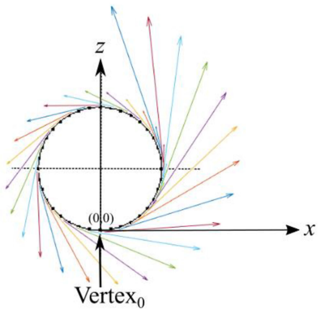


Fig. 15. Quiver plot showing the initial velocities and positions of the particles emitted from each of the different vertices of a circle. This circle will be used to construct the wheel. The number of vertices N is 36. The vertex marked as $Vertex_0$ emits particles with a velocity of zero. The velocity increases counter clock wise for each vertex, as stated in (7, 8).

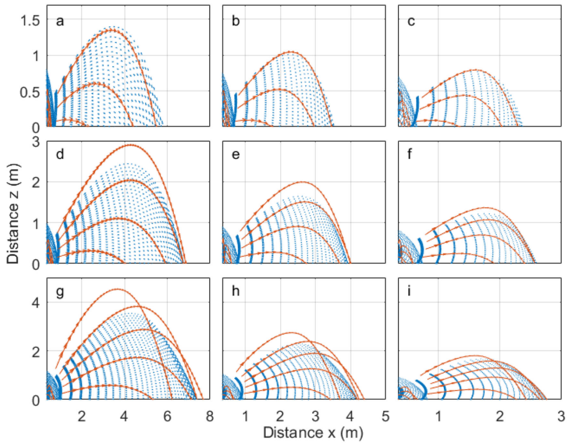


Fig. 16. Trajectories generated by a rotating circle using different rotation velocities and drag values. The trajectories in blue are from Blender and those in red are calculated using the PST using (7-10) to set the initial velocity and position of the emitted particles. a) $v_c = 40$ km/h, $K_D = 0.2$, b) $v_c = 40$ km/h, $K_D = 0.5$, c) $v_c = 40$ km/h, $K_D = 1$, d) $v_c = 50$ km/h, $K_D = 0.2$, e) $v_c = 50$ km/h, $K_D = 0.5$, f) $v_c = 50$ km/h, $K_D = 1$, g) $v_c = 60$ km/h, $K_D = 0.2$, h) $v_c = 60$ km/h, $K_D = 0.5$, i) $v_c = 60$ km/h, $K_D = 1$. For particles with a high injection angle Blender tends to underpredict the trajectories. When adding the rest of the vehicle, however, most of these particles will hit the vehicle mesh and be removed.

wheel size. This kind of rotating circle is used later as basis for constructing the whole wheel. The vertex corresponding to $i = 0$ is marked as $Vertex_0$.

Regarding how the forces are defined, a similar method was used comparing in this case the trajectories of the particles for different combinations of particles masses and sizes, wheel velocities, drag values and wind forces.

Fig. 16 shows an example of the trajectories obtained in Blender (blue) and PST (red) for different values of drag and wheel speeds for a particle with a mass of 1 and size of 1. Except for the wheel speed, these values do not relate with real physical values. For the same reason, the different constants as implemented in Blender are shown using variables in capital letters to differentiate them from the real physical constants. The forces are implemented as follows:

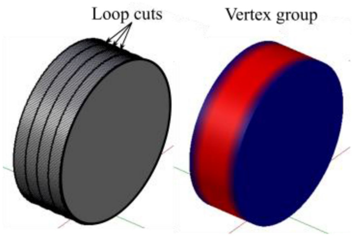


Fig. 17. Construction of a wheel model using three loop cuts on a cylinder with dimensions based on a real wheel of type 255/40R19. The left side shows the loop cuts and the right side a vertex group used to control where particles are created (red region).

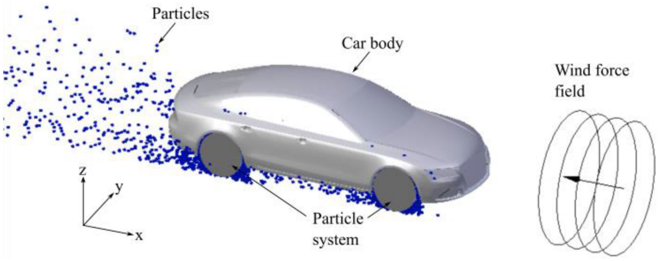


Fig. 18. 3D scene showing the different elements used for the simulation, excluding the sensor. The wheels are placed in a low poly car body, which acts as collision object. Particles (blue) flow away from the wheel due to the initial velocity acquired from the wheel rotation and the wind force. The wind force is constant and acts only in the x direction.

$$\boldsymbol{F_D} = \boldsymbol{K_D} \cdot \boldsymbol{L}^2 \cdot \boldsymbol{v}^2 \tag{11}$$

$$\boldsymbol{F_W} = \boldsymbol{K_W} \tag{12}$$

$$\boldsymbol{F_G} = \boldsymbol{M} \cdot \boldsymbol{G} \tag{13}$$

Here, L corresponds to the particle diameter, K_D to the drag value, K_W to the wind force value, M to the particle mass, G to the gravity acceleration value, and $\boldsymbol{F_D}$, $\boldsymbol{F_W}$ and $\boldsymbol{F_G}$ to the drag, wind and gravity forces, respectively.

From the previous results, it is concluded that the particle simulation in Blender for the parameter region tested is accurate and stable enough to be used for the simulation of spray in real time. With this knowledge, the rest of the wheel is completed using a cylinder with three loop cuts, as shown on the left side of Fig. 17. As mentioned before, the width of the wheel corresponds to the 255/40R19 wheel type. The loop cuts are positioned trying to imitate the profile of a real wheel with three treads. Each of these loops is equivalent to the circles previously used in the validation.

A vertex group, as shown on the right side of Fig. 17, is used in order to limit the emission of particles to only the three loop cuts (red region) without including the edges at each side of the cylinder (blue region).

After having completed the wheels, each of them is positioned in a 3D car body. The model is simplified to include only the external hull. Any gap in the mesh of the car is closed. When a particle hits the car body, its lifetime is set to zero to simulate the breaking of the drop against the real car hull. An openGL rendering of the scene is shown in Fig. 18. The particles are

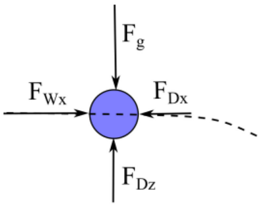


Fig. 19. Force diagram illustrating the forces that act upon one particle. The dotted line marks the trajectory of the particle as generated by the different forces and the initial velocity and position with which the particle was created.

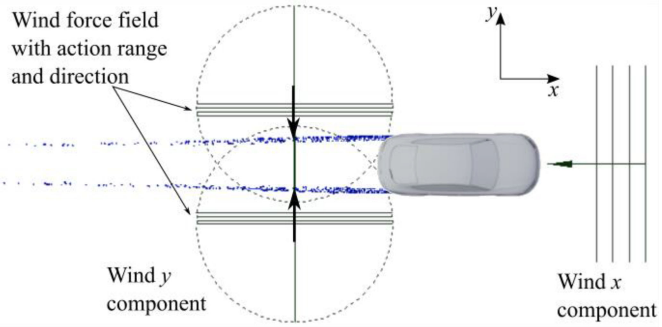


Fig. 20. Top view of a scene showing the addition of two extra wind force fields that bring the spray tracks from each wheel close to each other depending on the car speed. The range of these fields and their strength is chosen to make the simulations coincide with the spray measurements. Particles are shown in blue.

colored in blue for easier visualization. For the real sensor simulation, different materials are used for the wheels, car body and particles approximating the real reflection properties of each material. The wind force field is also shown. In this case, only a constant force in the x direction (15) is used. The effect of the y component of the wind force is shown in Fig 20.

2) Spray Dispersion: The spray dispersion is simulated using the same forces, which were characterized in the previous section (11–13) by adjusting the values of the coefficients used by the physics engine in a way that the trajectories of the particles coincide with the theoretical trajectories. The real values for the different coefficients cannot be directly used as their value is too small and conflict with the solver. For example, we can calculate the mean particle size by using (3). Using the values provided in [35] to get an approximate value of the constant k and considering a car speed of 40 km/h and tread height $b = 11$ mm we obtain an average particle diameter of $D_{mean} = 330\mu m$, after considering a spherical water drop shape, this corresponds to a mass of only $m = 1.95x10^{-8}kg$.

The proposed model is expanded upon by adding a component of the wind force in the y direction and finally turbulences using the curl of a Perlin noise texture.

To start with, the forces acting upon one particle, which is the analogous of one or more drops, are shown in Fig. 19. As the wind force field F_{W_x} is aligned with the *x-axis*, this force affects only the x component of the particle acceleration vector. The wind force is opposed by the drag force F_{D_x}. In the z direction, the gravity force F_g is opposed by the drag force F_{D_z}.

The wind and drag forces are defined as [67]:

$$\boldsymbol{F_D} = -\frac{1}{2}\boldsymbol{\rho_{air}} \boldsymbol{v_p}^2 \boldsymbol{C_D A_p} \tag{14}$$

$$\boldsymbol{F_W} = \frac{1}{2}\,\boldsymbol{\rho_{air}} \boldsymbol{v_w}^2 \boldsymbol{C_D A_p} \tag{15}$$

where ρ_{air} is the air density, $\boldsymbol{v_p}$ the velocity of the particle, $\boldsymbol{v_w}$ the wind velocity, A_p the cross sectional area of the particle and C_D the drag coefficient, which is a dimensionless number dependent on the shape of the object. It has a value of around 0.5 for spherical objects [67].

The values of the constants $K_{\boldsymbol{D}}, L,\ M,\ G$ and K_W can then be calculated by solving (16-17) for the z-direction and (20-21) for the x-direction. Notice that although the magnitude of each force will be different between simulation and theory, the resulting acceleration will be the same. If the initial position and velocity also coincide, then the resulting trajectory should coincide.

$$-\boldsymbol{K_D L^2 v_{pz}}^2 + \boldsymbol{MG} = \boldsymbol{Ma_z} \tag{16}$$

$$-\frac{1}{2}\boldsymbol{\rho_{air} v_{pz}}^2 \boldsymbol{C_D A_p} + \boldsymbol{mg} = \boldsymbol{ma_z} \tag{17}$$

If we assign $M = 1$ and $G = g$ and replace (16) in (17) we obtain:

$$-\frac{1}{2}\boldsymbol{\rho_{air} v_{pz}}^2 \boldsymbol{C_D A_p} + \boldsymbol{mg} = \boldsymbol{m}\left(-\boldsymbol{K_D L^2 v_{pz}}^2 + \boldsymbol{g}\right) \tag{18}$$

After simplifying and solving for $\mathrm{K_D}$ we get:

$$\boldsymbol{K_D} = \frac{1}{2}\,\frac{\boldsymbol{\rho_{air} C_D A_p}}{\boldsymbol{mL}^2} \tag{19}$$

The same can be done in the x direction:

$$\boldsymbol{K_{W_x}} - \boldsymbol{K_D L}^2\ \boldsymbol{v_{px}}^2 = \boldsymbol{ma_x} \tag{20}$$

$$\frac{1}{2}\boldsymbol{\rho_{air} v_w^2 C_D A_p} - \frac{1}{2}\boldsymbol{\rho_{air} v_{px}}^2 \boldsymbol{C_D}\ \boldsymbol{A_p} = \boldsymbol{ma_x} \tag{21}$$

Replacing (20) in (21) we get:

$$\frac{1}{2}\boldsymbol{\rho_{air} v_w^2 C_D A_p} - \frac{1}{2}\boldsymbol{\rho_{air} v_{px}}^2 \boldsymbol{C_D}\ \boldsymbol{A_p} = \boldsymbol{mK_{W_x}} - \boldsymbol{mK_D L^2 v_{px}}^2 \tag{22}$$

Using the result from (23) we can solve for $\mathrm{K_{W_x}}$:

$$\boldsymbol{K_{W_x}} = \frac{1}{2}\,\frac{\boldsymbol{\rho_{air} v_w^2 C_D A_p}}{\boldsymbol{m}} \tag{23}$$

As can be observed in Fig. 11, with increasing speed the wind stream starts to push the spray generated by each wheel in the y-direction towards each other. This effect increases with speed up to the point that a single spray source is seen for highway speeds (Fig. 13). In order to simulate this effect, it is possible to use two extra wind force fields placed as illustrated in Fig. 20. These fields have an effect area from 0 to 6 m behind the car and are used to add an extra force in the required direction. The size of the effect area was chosen based on the test area spray measurements. The left field affects only the particles from the left wheel. The same happens for the right wheel. As the relation between K_{W_x} and K_{W_y} controls how much the two spray tracks come together, it is possible to vary K_{W_y} to make the real and simulated point clouds coincide.

The effect of turbulence is also modeled phenomenologically using a similar strategy as for the y component of the wind force. In this case, the particles from the left wheel generate a turbulence force that affects the particles on the right wheel and vice versa. This effect increases the more the particles from each wheel come closer to each other.

The strength of the turbulence is adjusted to make the real and synthetic distributions coincide. In this matching, it is important to consider that the tracking of the target car in both the test area and highway is not perfect and hence part of the effect that seems to be caused by turbulence may have been caused by tracking errors. Additionally, as wind direction was not measured some of the perturbations may have been caused by side wind.

3) Spray Detection: If we consider the average size of the particles as previously calculated, it is of a fraction of a millimeter. The drops generated by the bow and side waves (splash) have a larger average size between 1 and 2 mm [31] but travel only a short distance from the wheel before hitting the floor or breaking. Therefore, it would be required to simulate thousands of very small particles from which only few are detected. As mentioned in the Section III-A a better alternative is to use a bigger mesh size for the particles. These bigger particles would then represent a group of drops that in a certain moment create a concentration high enough to cause a detection. The number of particles required and their lifetime can be extracted from histograms like the ones shown in Fig. 12 and Fig. 14. The exact shape of the histogram cannot be reconstructed in simulation due to the way particles are buffered in version 2.79 of Blender. Blender uses a fixed-size monolithic buffer to store the particles data. The size of the buffer is defined at the beginning of the simulation and the user can only control the total amount of particles and the start and end frame of their emission resulting in a fixed emission rate [68].

The exact size of the mesh can be chosen based on the parameters shown in Fig. 21. Based on the characteristic of the sensor there will be a maximum distance after which spray is rarely detected (see also Fig. 13). Besides, there will also be a minimum valid distance for the model, in our case caused by the non-separable presence of rain drops in the region from 0 to 5 m. The mesh size can hence be chosen equal to D_{min} or between D_{min} and D_{max}. This analysis is similar to the LOD (Level Of Detail) considerations in computer graphics.

The mesh size of the particles and their quantity also influence the amount of occlusion (Fig. 11). This effect is nevertheless difficult to accurately reconstruct in simulation because a very high amount of particles is required. An alternative is to manually reduce the number of particles of the occluded wheel or increase the hardware resources.

Regarding the materials, the principled BSDF was used as mentioned in the background section. Four different materials were defined, one for the car body, one for the wheels, one for the license plate and one for the particles. In order to keep the rendering time as low as possible, only the albedo and the index of refraction were adjusted. All other parameters were set to zero.

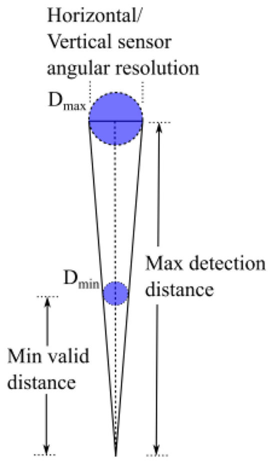


Fig. 21. Based on the sensor resolution and the region in which the spray model is valid, is possible to choose a value for the mesh size of each particle. This value is only relevant for rendering and does not influence the particles trajectory.

The albedo is obtained by setting the subsurface scattering to $1 - SSA$ and using black as scattering color, which is equivalent to absorbing that portion of the light. For the particles, the SSA (Single Scattering Albedo) is known from the literature to approximate 0.53 [30]. For the car body and wheels, the reflection was measured manually for the required wavelength and averaged. For the license plate, it was extracted from the literature [69].

D. Validation

In order to have a better overview of how the different parameters of the simulation are selected, Fig. 22 presents a summary of the information provided in parts A, B and C (Measurements in the proving ground, Measurements in the Highway and Simulation). All parameters were previously explained except for the detection level. This variable defines the minimum intensity that a pixel needs to have in order to be considered a detection (Fig. 7c). The rest of the section presents a comparison between the real measurements and the obtained synthetic point cloud for the test area (Fig. 23) and highway (Fig. 24), respectively.

In order to evaluate the quality of the simulation 2D histograms are used. The Pearson correlation coefficient between the real distributions and the synthetic ones is calculated.[1] This follows previous studies related with the validation of virtual point clouds [70].

A grid size from 0.5 m by 0.5 m is used is in the same order of magnitude as the standard deviation of the distance measurements. Besides measurement inaccuracies, the grid size and correlation coefficient required from the simulation would depend on the algorithm that uses the synthetic data (e.g., the algorithm creating the object hypothesis in Fig. 1). If object segmentation is done fitting lines using Ransac, the grid size can

[1] Other metrics are presented in Table VI and VII.

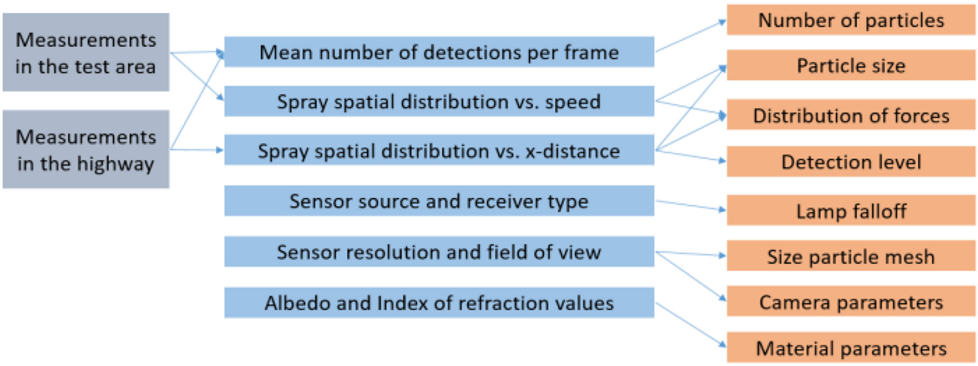

Measurements in the test area
Measurements in the highway
Mean number of detections per frame
Spray spatial distribution vs. speed
Spray spatial distribution vs. x-distance
Sensor source and receiver type
Sensor resolution and field of view
Albedo and Index of refraction values
Number of particles
Particle size
Distribution of forces
Detection level
Lamp falloff
Size particle mesh
Camera parameters
Material parameters

Fig. 22. Relation between the main simulation parameters and the source(s) used for their extraction. A combination of real measurements with physically based values is used.

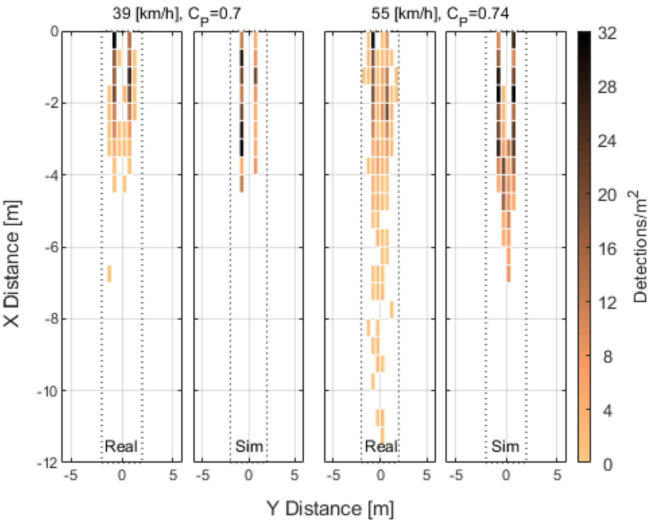


Fig. 23. Comparison of the real measurements as show in Fig. 11 and the respective simulation results using a 2D histogram with a bin size of 0.5×0.5 m. For each pair, the average speed and Pearson correlation coefficient (C_p) are shown. The whole region inside the dotted box is used to calculate the correlation. Although it is possible to increase the similarity between simulation and measurements to obtain better correlation values, over fitting should be avoided. Additionally, the simulation must be based on known physical effects as much as possible.

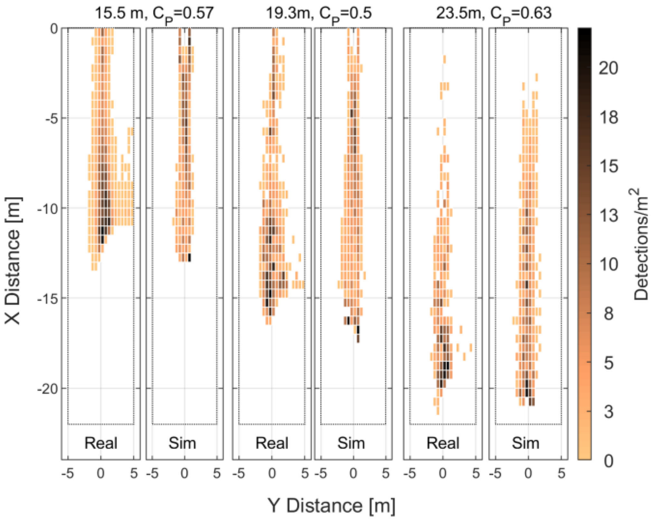


Fig. 24. Comparison of the real measurements as shown in Fig. 13 and the respective simulation results using a 2D histogram with a bin size of 0.5 × 0.5 [m]. For each pair, the corresponding distance between ego and target car and the Pearson correlation coefficients (C_p) are shown. The whole region inside the dotted box is used to calculate the correlation. The detections in the region from 0 to 5 m from the sensor are removed from the measurements as they are not considered in the simulation.

be defined based on the inlier distance threshold. If the algorithm is a neural network, it can be trained to identify detections caused by spray using real data. The same network can then be used to classify synthetic spray detections; if the accuracy is similar, it can be assumed that for the network both real and synthetic are not distinguishable. Otherwise, the simulation would have to be adjusted to reach a higher correlation coefficient for an equal/smaller grid size until the network is not able to differentiate between the two cases anymore. Notice that this is similar to what GANs (Generative Adversarial Networks) do with the difference that in our case, the generation uses a physically based simulation instead of a neural network.

Unknowns regarding the sensor internal algorithms can also cause a low correlation value. This issue can be solved if the sensor manufacturer provides these algorithms in the form of compiled code. The data provided by the simulation can then be used as input for this module.

Finally, it is important to balance the desired correlation value and the complexity of the simulation. For example, by increasing the number of particles or the number of forces is possible to increase the similarity between the real and synthetic spray clouds. However, this would require better hardware for the same number of frames per second.

Based on the previously mentioned aspects, the results shown in Fig. 23 for the test area and Fig. 24 for the highway have the main goal of demonstrating that with the proposed method, many of the characteristics of the spray generation, dispersion and detection can be reproduced. Additionally, it shows that the same model can be used to cover a wide range of speeds: (39 to 89 km/h) by only adjusting the simulation parameters.

The particle diameter used for the simulation was 4 mm (Table II), which is the maximum value measured in spray

TABLE I
HIGHWAY MEASUREMENTS

Date	Duration	Rainfall [65] [66]
13.04.2018	1 hour 17 min	2 mm
16.05.2018	3 hours, 26 min	3.6 mm
17.05.2018	2 hours, 20 min	3.1 mm

TABLE II
MATERIAL PARAMETERS (PRINCIPLED BSDF)

	IOR (Index of refraction)	Absorption
Particles	1.33 (Water)	0.47
Car body	1.45 (Average different materials)	0.57
Wheels	1.5 (Rubber)	0.93
License plate	1.48 (Acrylic-retroreflective)	0.20

TABLE III
SIMULATION PARAMETERS (PROVING GROUND MEASUREMENTS)

Particle diameter 4 mm Mesh diameter 1.8 cm, # Render samples 10		
Speed [km/h]	39	55
K_D	0.025	0.025
K_{W_x}	4.21	8.37
K_{W_y}	0.05	3.7
L	3	3
T_s	0.3	0.3
$\boldsymbol{v_w}$	$0.4 \cdot \boldsymbol{v_c}$	$0.4 \cdot \boldsymbol{v_c}$
Particles/frame	8	20
Render time/frame	40 ms	60 ms

from tread throw experiments [31]. Additionally, the wind speed $(\boldsymbol{v_w})$ was set as 40% of the vehicle speed $(\boldsymbol{v_c})$. If a smaller particle, size is used or in a higher wind speed, the spray detections concentrate much farther from the target car as in the measurements. In the future, CFD simulations can be used to get a better approximation of the wind speed value behind the car close to the wheels. This value can be used to calculate K_{W_x} for each required speed. This parameter will nevertheless remain a challenge as each particle represents one or multiple drops that come together to cause a detection.

Additionally, the turbulence in the real measurements appears to be higher. This could be easily corrected but care is required not to attribute to turbulence the effects caused by the movement of the reference point (tracking errors). Therefore, more data capture using two cars with a differential GPS system would be needed before adjusting this value.

The values for each of the different coefficients are shown in Table II.

For the highway (Fig. 24), the region from 0 to 5 m in the real data was removed because it contains mostly detections caused by rain drops (compare with Fig. 13).

In this case, the wind speed was set equal to the car speed. From the results it appears that with distance, the reduction in the number of spray detections is stronger in the measurements. Because during the highway measurements it was raining (rainfall rate <3.6 mm), the noise level of the real sensor increased causing a reduction in the number of spray detections for the whole region (>5 m) [30]. This effect is included in the simulation by adjusting the number of particles per frame (Table III). The use of a constant emission rate and particles size reduce however the quality of the approximation. For instance, if the rate were exactly as shown in Fig. 14 there will be many frames in which few particles are generated. With few particles being generated close to the wheels most of the detections would have been caused by particles still floating on the air and closer to the sensor. As there is a unique particle size however, the behavior of the particles is to follow inertial trajectories and hence relatively quickly fall to the ground.

For higher rainfall rates (>3.6 mm), further modifications on the simulation model may be needed. In this case is likely that most of the detections concentrate directly in front of the sensor, as the noise level is very high occluding spray detections.

In the region close to the wheels in the synthetic data, the degree of turbulence is small and increases only after a certain distance when the particles come closer to each other. In reality, the contribution from each wheel is indistinguishable. As already mentioned, in the simulation the effect of occlusion is difficult to generate without a very high number of particles. When the distance of the ego increases (see Fig. 24 at 23.5 m), the region close to the wheels is not visible anymore and the correlation improves. A second problem is the concentration of particles. In the simulation, they tend to concentrate on the middle of the two streams coming from each wheel. In reality, though, their concentration increases the closer they are to the sensor. As explained in Section IV-B (Measurements in the highway), this is due to the accumulation of small particles in the air. To include this effect a second type of particles would have to be added to the simulation with a behavior corresponding to low Stokes number. The number of particles of each type (low and high Stokes number) would have to be optimized depending on the real number of spray detections per frame and their distribution. Whether the improved correlation value justify higher complexity and rendering times is an open question.

Because occlusion and the change from high to low Stokes numbers are most notable at higher speeds the correlation values for the highway are lower than those for the proving ground (Fig. 21).

Table III shows the parameters used for the highway region. In this case, there is only one set as only one speed is used (which corresponds to the average speed of the measurements). The mean particle size is kept the same as for the proving ground as the average behavior is still mostly inertial (Table II). The mesh size remained the same as it depends on the sensor resolution.

The wind speed was set equal to the car speed in this case. As mentioned before, these two values are difficult to adjust because one detection can correspond to one or more real spray drops.

The constant L was arbitrarily chosen in all cases. It acts as a scale factor for the drag constant to keep its value within an acceptable numerical range.

V. CONCLUSION

This paper shows that it is possible to reconstruct the spatial distribution of the detections caused by spray on a LIDAR sensor by using a path tracer, a real time physics engine and particle systems. The simulated and measured spatial distributions correlate with a C_p going from 0.5 in the worst case to 0.74 in the

TABLE IV
SIMULATION PARAMETERS (HIGHWAY MEASUREMENTS)

Particle diameter 4 mm, Mesh diameter 1.8 cm, Speed 89 [km/h], # Render samples 10	
K_D	0.025
K_{W_x}	140
K_{W_y}	30.6
L	3
T_s	1.5
v_w	v_c
Particles/frame	50
Render time/frame	260ms

best case. This kind of synthetic data can be used to improve the performance of algorithms generating object hypothesis in SD and ADAS functions increasing their robustness against phantom breaking.

A simulation setup was defined as simply as possible in order to keep the simulation time low while still capturing the main forces that influence the movement of spray water. The parameters of the simulation were partially deduced from real measurements and partially based on known physical values. In the simulation, each particle represented a spray detection which corresponds to one or more real drops with a concentration high enough to generate a reflection that can be registered by the LIDAR sensor.

It was also shown that if the water level is constant, the mean and maximum number of spray detections per frame increases with higher speeds. After a certain speed, the amount of spray from one wheel is high enough to occlude the spray from the other. The effect of occlusion is difficult to recreate in the simulation and is one of the causes for lower correlation values.

With higher vehicle speeds, the source of spray changes from being clearly generated by each wheel to a cloud coming somewhere from the back of the target car. If the distance between the target and ego vehicles increases, the spray cloud concentrates mostly in front of the ego car. This is caused by the reduction of the spray drop sizes with increasing wheel rotation speed (smaller drops remain longer in the air). The presence of these drops in the near region of the sensor increases the noise level, reducing the number of detections on other regions. The net effect on the sensor perception is similar to that of rain [30].

The inertial trajectory of the particles in Blender was validated using a conventional solver (PST). The effect of the turbulence, which increases with vehicle speed, was simulated as a random perturbance upon the inertial trajectory.

VI. OUTLOOK

The integration of particle systems into a real time physics based simulation of the LIDAR sensor provides a framework in which other phenomena can be easily included. For example: dirt on the LIDAR cover, exhaust gases, rain, fog and snow.

Even though these phenomena can be simulated using volumetric scattering or stochastic simulations [25], the use of particles allows for the easy inclusion of external forces making the simulation more realistic.

TABLE V
PARAMETERS OF THE USED LIDAR SENSOR (IBEO SCALA [63])

Parameter	Value
Wavelength	905 nm
Horizontal Field-of-View	145°
Vertical Field-of-View	3.2°
Layers	4 (0.8° each)
Data refresh time	40/80 ms
Distance resolution	< 0.1 m
Typical range	150 m

TABLE VI
PARAMETERS OF THE USED WFMD (VAISALA DSC111 [62]*)

Parameter	Value
Water film measurement range	0 to 2 mm
Resolution	0.01 mm

*USED VERSION WAS OPTIMIZED FOR ROAD MEASUREMENTS.

TABLE VII
ADDITIONAL PERFORMANCE METRICS FIG 23

Parameter	Speed [kmh]	Value [%]
Jaccard's distance binary image*	39	52
	55	54
Hamming distance**	39	6
	55	16

*The Jaccard's distance is the percentage of non-zero samples that differ. If the number of spray detections is binarized (setting a bin to one if it has one or more detections) it corresponds to comparing the shapes of the two point clouds. As it ignores zero samples it is a very strict metric.
**The Hamming distance is the percentage of samples that differ (including zero samples).

TABLE VIII
ADDITIONAL PERFORMANCE METRICS FIG 24

Parameter	Distance [m]	Value [%]
Jaccard's distance binary image	15.5	46
	19.3	34
	23.5	47
Hamming distance	15.5	10
	19.3	7
	23.5	9

The validation can be done in each case based on the correlation between the real and synthetic point clouds using a 2D histogram that captures the relevant properties.

The use of a variable injection rate for the particles as well as variable particles sizes should improve in the future the correlation between the simulation model and the real measurements.

A precise measurement of the water film thickness using a single device remains a challenge. In the future, the use of V2X technologies should improve these kinds of measurements.

Table IV and V present extra information related with the used sensors.

Table VI and VII present additional performance metrics to evaluate the correlation between the simulation results and the real point clouds.

References

[1] Y. Kang, H. Yin, and C. Berger, “Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments,” *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 171–185, Jun. 2019.

[2] H.-P. Schöner, “Simulation in development and testing of autonomous vehicles,” in *Proc. 18. Internationales Stuttgarter Symp.*, 2018, pp. 1083–1095.

[3] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner *et al.*, “Sim-to-Real: Learning Agile Locomotion For Quadruped Robots,” in *Proc. Robotics: Sci. Syst.*, 2018.

[4] K. Bousmalis *et al.*, “Using simulation and domain adaptation to improve efficiency of deep robotic grasping,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4243–4250.

[5] V. S. R. Veeravasarapu, R. N. Hotay, C. Rothkopf and R. Visvanathan, “Simulations for Validation of Vision Systems,” 2015, *arXiv:1512.01030*.

[6] nvidia, “Nvidia drive constellation: VR-Simulator für autonome Fahrzeuge.” [Online]. Available: https://www.nvidia.com/de-de/self-driving-cars/drive-constellation/

[7] M. O’Kelly, A. Sinha, H. Namkoong, J. Duchi and R. Tedrake, “Scalable end-to-end autonomous vehicle testing via rare-event simulation,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9849–9860.

[8] J. Thieling, and J. Roßmann, “Highly-scalable and generalized sensor structures for efficient physically-based simulation of multi-modal sensor networks,” in *Proc. 12th Int. Conf. Sens. Technol.*, 2018, pp. 202–207.

[9] S. O. Wald, and F. Weinmann, “Ray tracing for range-doppler simulation of 77 GHz automotive scenarios,” in *Proc. 13th Eur. Conf. Antennas Propag.*, 2019, pp. 1–4.

[10] P. Degerman, J. Pohl, and M. Sethson, “Ultrasonic sensor modeling for automatic parallel parking systems in passenger cars,” SAE Tech. Paper, 2007. [Online]. Available: https://doi.org/10.4271/2007-01-1103

[11] K. Majek, and J. Bedkowski, “Range sensors simulation using GPU ray tracing,” in *Proc. 9th Int. Conf. Comput. Recognit. Syst. CORES 2015*, 2016, pp. 831–840.

[12] R. Bridson, J. Houriham, and M. Nordenstam, “Curl-noise for procedural fluid flow,” *ACM Trans. Graph.*, vol. 26, no. 3, p. 46, 2007.

[13] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim, “Unified particle physics for real-time applications,” *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–12, 2014.

[14] F. Wotawa, B. Peischl, F. Klück, and M. Nica, “Quality assurance methodologies for automated driving,” *Elektrotech. Inftech.*, vol. 135, no. 4-5, pp. 322–327, 2018.

[15] E. Egea-Lopez, F. Losilla, J. Pascual-Garcia, and J. M. Molina-Garcia-Pardo, “Vehicular networks simulation with realistic physics,” *IEEE Access*, vol. 7, pp. 44021–44036, Apr. 2019.

[16] E. Coumans, and Y. Bai, Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016 [Online]. Available: http://pybullet.org/

[17] NVIDIA, PhysX library. 2008. [Online]. Available: https://developer.nvidia.com/gameworks-physx-overview

[18] Q. Chao *et al.*, “A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving,” *Comput. Graph. Forum*, vol. 2019, no. 1, pp. 287–308. 2020, doi: 10.1111/cgf.13803.

[19] Blender Foundation, *Cycles: Open source production rendering.* 2011. [Online]. Available: https://www.cycles-renderer.org/

[20] Blender Foundation, Blender. 2020. [Online]. Available: https://www.blender.org/about/

[21] J. Buchholz, Particle System Toolbox. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/14314-particle-system-toolbox

[22] E. D. R. Shearman, E. G. Hoare, and A. Hutton, “Trials of automotive radar and lidar performance in road spray,” IEE Colloquium on Automotive Radar and Navigation Techniques (Ref. No. 1998/230), pp. 10/1–10/7, 1998.

[23] J. Sauder, and B. Sievers, “Context prediction for unsupervised deep learning on point clouds,” 2019.*arXiv:1901.08396*.

[24] S. Hasirlioglu, and A. Riener, “A model-based approach to simulate rain effects on automotive surround sensor data,” in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2609–2615.

[25] M. Berk *et al.*, “A stochastic physical simulation framework to quantify the effect of rainfall on automotive lidar,” *SAE Int. J. Adv. Curr. Practices Mobility*, vol. 1, no. 2, pp. 531–538, 2019, [online] Available: https://saemobilus.sae.org/content/2019-01-0134/.

[26] M. Elgharbawy, A. Schwarzhaupt, G. Scheike, M. Frey, and F. Gauterin, “A generic architecture of ADAS sensor fault injection for virtual tests,” in *Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl.*, 2016, pp. 1–7.

[27] J. R. V. Rivero *et al.*, “Characterization and simulation of the effect of road dirt on the performance of a laser scanner,” in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–6.

[28] M. S. Awan, L. Csurgai-Horváth, S. S. Muhammad, E. Leitgeb, F. Nadeem, and M. S. Khan, “Characterization of fog and snow attenuations for free-space optical propagation,” *JCM*, vol. 4, no. 8, pp. 533–545, 2009.

[29] S. Hasirlioglu, A. Riener, W. Huber, and P. Wintersberger, “Effects of exhaust gases on laser scanner data quality at low ambient temperatures,” in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1708–1713.

[30] J. R. Vargas Rivero, T. Gerbich, V. Teiluf, B. Buschardt, and J. Chen, “Weather classification using an automotive LIDAR sensor based on detections on asphalt and atmosphere,” *Sensors*, vol. 20, no. 15, 2020, Art. no. 4306.

[31] D. H. Weir, J. F. Strange, and R. K. Heffley, and others, “Reduction of adverse aerodynamic effects of large trucks, volume i. Tech. Rep.,” U.S. Federal Highway Administration, 1978.

[32] G. W. Flintsch *et al.*, “Splash and spray assessment tool development program: Final report,” Federal Highway Administration, Rep. FHWA-HRT DTFH61-08-C00030, 2014.

[33] X. Hu *et al.*, “A numerical simulation of wheel spray for simplified vehicle model based on discrete phase method,” *Adv. Mech. Eng.*, vol. 7, no. 7, 2015, Art. no. 1687814015597190.

[34] A. P. Gaylard, K. Kirwan, and D. A. Lockerby, “Surface contamination of cars: A review,” *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 231, no. 9, pp. 1160–1176, 2017.

[35] S. A. Lottes and C. Bojanowski, “Computer Modeling and Analysis of Truck Generated Salt-Spray Transport Near Bridges,” No. ANL/ESD/13-1. Argonne National Lab, Argonne, IL, USA, 2013.

[36] C. Radovich, and D. Plocher, “Experiments on spray from a rolling tire,” in *The Aerodynamics of Heavy Vehicles II: Trucks, Buses, and Trains*, pp. 403–417.

[37] A. Kabanovs, A. Garmory, M. Passmore, and A. Gaylard, “Investigation into the dynamics of wheel spray released from a rotating tyre of a simplified vehicle model,” *J. Wind Eng. Ind. Aerodyn.*, vol. 184, pp. 228–246, 2019.

[38] F. A. Gifford Jr, “Use of routine meteorological observations for estimating atmospheric dispersion,” *Nucl. Saf.*, vol. 2, no. 4, pp. 47–51, 1961.

[39] F. Pasquill, “The estimation of the dispersion of windborne material,” *Met. Mag.*, vol. 90, no. 1,063, pp. 33–49, 1961.

[40] R. H. Rasshofer, M. Spies, and H. Spies, “Influences of weather phenomena on automotive laser radar systems,” *Adv. Radio Sci.: ARS*, vol. 9, B.2, pp. 49–60, 2011.

[41] J. S. Paschkewitz, “Simulation of spray dispersion in a simplified heavy vehicle wake,” No. UCRL-TR-218207. Lawrence Livermore National Lab, Livermore, CA, USA, 2006.

[42] G. Echeverria, N. Lassabe, A. Degroote, and S. Lemaignan, “Modular open robots simulation engine: MORSE,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 46–51.

[43] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, “BlenSor: Blender sensor simulation toolbox,” in *Int. Symp. Vis. Comput.*, 2011, pp. 199–208.

[44] J. Allworth, L. Windrim, J. Wardman, D. Kucharski, J. Bennett, and M. Bryson, “Development of a high fidelity simulator for generalised photometric based space object classification using machine learning,” 2020.*arXiv:2004.12270*.

[45] A. D. Cornetto, and J. Suway, “Validation of the cycles engine for creation of physically correct lighting models,” No. 2019-01-1004. SAE Tech. Paper, 2019.

[46] J. Hu *et al.*, “Sensor-realistic synthetic data engine for multi-frame high dynamic range photography,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 516–517.

[47] P. Martinez-Gonzalez, S. Oprea, A. Garcia-Garcia, A. Jover-Alvarez, S. Orts-Escolano, and J. Garcia-Rodriguez, “Unrealrox: An extremely photo-realistic virtual reality environment for robotics simulations and synthetic data generation,” 2018.*arXiv:1810.06936*.

[48] F. Wang, Y. Zhuang, H. Gu, and H. Hu, “Automatic generation of synthetic LiDAR point clouds for 3-D data analysis,” *IEEE Trans. Instrum. Meas.*, vol. 68, no. 7, pp. 2671–2673, Jul. 2019.

[49] R. L. Smith, Open dynamics engine. 2005. [Online]. Available: http://www.ode.org/

[50] HAVOK, “Havok Physics Engine,” 2000.

[51] I. Georgiev *et al.*, “Arnold: A brute-force production path tracer,” *ACM Trans. Graph. (TOG)*, vol. 37, no. 3, pp. 1–12, 2018.

[52] C. Per *et al.*, “Renderman: An advanced path-tracing architecture for movie rendering.” *ACM Trans. Graph. (TOG)*, vol. 37, no. 3, pp. 1–21, 2018.

[53] LuxCoreRender project, LuxCoreRender. 2018. [Online]. Available: https://luxcorerender.org/

[54] W. T. Reeves, “Particle systems - a technique for modeling a class of fuzzy objects”, *ACM Trans. Graph.*, vol. 2, no. 2, pp. 91–108, 1983.

[55] I. Millington, *Game Physics Engine Development: How to Build a Robust Commercial-Grade Physics Engine for Your Game*:. Boca Raton, FL, USA: CRC Press, 2010.

[56] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, “Microfacet models for refraction through rough surfaces,” in *Proc. 18th Eurographics Conf. Rendering Techn.*, 2007, pp. 195–206.

[57] B. Burley, “Physically-Based Shading at Disney,” *ACM SIGGRAPH*, vol. 2012, pp. 1–7, 2012.

[58] B. Karis, “Real shading in unreal engine 4,” in *Proc. SIGGRAPH*, 2013.

[59] R. Sedman, How to calculate for every ray the total distance it has traveled from camera to emitter. Accessed: Mar. 5, 2019. [Online]. Available: https://blender.stackexchange.com/questions/81485/how-to-calculate-for-every-ray-the-total-distance-it-has-traveled-from-camera-to

[60] M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*. Burlington, MA, USA: Morgan Kaufmann, 2016.

[61] Y. Li, P. Duthon, M. Colomb, and J. Ibanez-Guzman, “What happens for a ToF LiDAR in fog?,” 2020.*arXiv:2003.06660*.

[62] Vaisala, Condition Patrol DSP310. [Online]. Available: https://www.vaisala.com/sites/default/files/documents/DSP310-Datasheet-B211197EN-B.pdf

[63] Ibeo ScaLa B3.3. IBEO, 2019. [Online]. Available: https://hexagondownloads.blob.core.windows.net/public/AutonomouStuff/wp-content/uploads/2019/05/ibeo_Scala_datasheet_whitelabel.pdf

[64] Google (n.d.). Accessed: May 15, 2019. [Online]. Available: https://www.google.de/maps/place/Audi+Pr%C3%BCfgel%C3%A4nde/@48.7709667,11.7275916,294m/data=!3m1!1e3!4m8!1m2!2m1!1spr%C3%BCfgel%C3%A4nde+neustadt+maps!3m4!1s0x479fb2ad9762ff5b:0xfb8d954c46558a9b!8m2!3d48.7713316!4d11.7169924

[65] Bayerisches Landesamt für Umwelt, GewässerkundlicherDienst Bayern. Accessed: Jun. 14 2019. [Online]. Available: https://www.gkd.bayern.de/en/meteo/precipitation/kelheim/hepberg-200106

[66] Bayerisches Landesamt für Umwelt, GewässerkundlicherDienst Bayern. Accessed: Jun. 14, 2019. [Online]. Available: https://www.gkd.bayern.de/en/meteo/precipitation/kelheim/baumannshof-200036

[67] Y. Nakayama, “*Introduction to fluid Mechanics*,” Cambridge, MA, USA, pp. 25–132, 2018.

[68] L. Stockner, Paged Buffers for Particles. [Online]. Available: https://code.blender.org/2011/03/paged-buffers-for-particles/

[69] R. Gurney, M. Rhead, W. E. Martin, S. Ramalingam, and N. Cohen, “The effect of retro-reflectivity and reflectance of U.K. number plates on ANPR performance,” in *Proc. 5th Int. Conf. Imag. for Crime Detection Prevention*, 2013, pp. 1–13.

[70] T. Hanke, A. Schaermann, M. Geiger, K. Weiler, N. Hirsenkorn, A. Rauch *et al.*, “Generation and Validation of Virtual Point Cloud Data for Automated Driving Systems,” in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2017.

Jose Roberto Vargas Rivero was born in Guantánamo, Cuba, in 1986. He received the B.Sc. degree in electronics from the Universidad Distrital Francisco José de Caldas of Bogotá, in 2009, and the M.Sc. degree in photonics from the Abbe School of Optics and Photonics, in Jena, in 2016. Since 2016, he has been working as a Ph.D student in the simulation of Lidar sensors in a virtual environment considering weather related effects. He is currently working in the development of simulation tools in Audi AG.

Thiemo Gerbich was born in Lollar, Germany, in 1985. He received the Dipl. Phys. degree in 2011 and the Dr.rer.nat. degree in 2015, both from the University of Würzburg, Germany. From 2011 to 2016, he was Research Assistant with the Institute of Physical and Theoretical Chemistry of the University of Würzburg. He is currently working as a Sensor Developer with Audi AG and responsible for testing of automotive Lidar systems. His Ph.D. thesis has focused on the dynamics of radiationless deactivation of electronic excited states of organic molecules.

Boris Buschardt was born in Berlin, Germany, in 1969. He received the Dipl.-Ing. degree in 1997 and the Dr.-Ing. degree in 2002, both in automotive engineering from the Technical University of Berlin. His Ph.D. study about Synthetic Steering Torque was honoured with the Hans-Peter Willumeit Award. From 1998 to 2000, he was a Research Assistant with the Man Machine Interaction Group of the Technical University of Berlin. Since 2000, he is working in different R&D positions with Audi AG. He is currently responsible for the Radar and Lidar development. He is the author of more than 40 papers, books and patents. His research interests include man machine interaction, sensor perception and safety of automated driving vehicles.

Jia Chen (Member, IEEE) received the Diploma degree from the University of Karlsruhe, Germany, in 2006 and the Ph.D. degree from TUM, in 2011. From 2011 to 2015, she was a Postdoctoral Fellow with the Environmental Science and Engineering Department, Harvard University. In 2015, she was appointed as a Professor with TUM, and currently heads the Environmental Sensing and Modeling Group with the Department of Electrical and Computer Engineering. She is also Associate with the Department of Earth and Planetary Sciences, Harvard University. She has authored and coauthored 52 peer-reviewed journals and 124 conference publications.

The Effect of Spray Water on an Automotive LIDAR Sensor: A Real-Time Simulation Study

Requesting permission to reuse content from an IEEE publication

Author: Jose Roberto Vargas Rivero

Publication: IEEE Transactions on Intelligent Vehicles

Publisher: IEEE

Date: March 2022

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

sensors

MDPI

Article

Data Augmentation of Automotive LIDAR Point Clouds under Adverse Weather Situations

Jose Roberto Vargas Rivero [1,2], Thiemo Gerbich [1], Boris Buschardt [1] and Jia Chen [2,3,*]

[1] Audi AG, Auto-Union-Str., D-85057 Ingolstadt, Germany; jose-roberto.vargas-rivero@audi.de (J.R.V.R.); thiemo.gerbich@audi.de (T.G.); boris.buschardt@audi.de (B.B.)

[2] Electrical and Computer Engineering, Technical University of Munich, Theresienstr. 90, D-80333 München, Germany

[3] Institute for Advanced Study, Technical University of Munich, Lichtenbergstraße 2 a, D-85748 Garching, Germany

* Correspondence: jia.chen@tum.de

Abstract: In contrast to previous works on data augmentation using LIDAR (Light Detection and Ranging), which mostly consider point clouds under good weather conditions, this paper uses point clouds which are affected by spray. Spray water can be a cause of phantom braking and understanding how to handle the extra detections caused by it is an important step in the development of ADAS (Advanced Driver Assistance Systems)/AV (Autonomous Vehicles) functions. The extra detections caused by spray cannot be safely removed without considering cases in which real solid objects may be present in the same region in which the detections caused by spray take place. As collecting real examples would be extremely difficult, the use of synthetic data is proposed. Real scenes are reconstructed virtually with an added extra object in the spray region, in a way that the detections caused by this obstacle match the characteristics a real object in the same position would have regarding intensity, echo number and occlusion. The detections generated by the obstacle are then used to augment the real data, obtaining, after occlusion effects are added, a good approximation of the desired training data. This data is used to train a classifier achieving an average F-Score of 92. The performance of the classifier is analyzed in detail based on the characteristics of the synthetic object: size, position, reflection, duration. The proposed method can be easily expanded to different kinds of obstacles and classifier types.

Keywords: LIDAR; point cloud; spray water; ADAS; AV; data augmentation; classification; synthetic data; sensor model

Citation: Vargas Rivero, J.R.; Gerbich, T.; Buschardt, B.; Chen, J. Data Augmentation of Automotive LIDAR Point Clouds under Adverse Weather Situations. *Sensors* **2021**, *21*, 4503. https://doi.org/10.3390/s21134503

Academic Editor: Mengdao Xing

Received: 19 May 2021
Accepted: 25 June 2021
Published: 30 June 2021

1. Introduction

The high degree of reliability required by newer ADAS and AV functions makes the combination of different sensors necessary [1–3]. Among those sensors, radar, cameras, ultrasound and GPS (global positioning system) are common. Recently, LIDAR sensors have also been used mostly due to their high resolution, ability to work under low light conditions and direct distance measurement [4,5]. However, the high resolution obtained by using infrared light comes at the cost of a reduced performance when dirt [6], rain, fog [7], snow [8], exhaust gases [9] and spray water [10] are present. Besides reducing range, these phenomena cause extra detections which can have negative effects like phantom braking. Mechanisms to filter out those extra detections are difficult to implement, as there is a risk of removing detections caused by real objects.

In this context, this paper focuses on correctly classifying solid objects in point clouds affected by spray water. Spray water is especially problematic because it can generate point clouds with characteristics that highly resemble those of solid objects. Figure 1 shows a real example in which the spray cloud generates detections with a shape close to the 'L' shape of a vehicle [10]. True positives are needed to train a classifier to remove this kind of false positive. This corresponds to cases in which real solid objects are present simultaneously with spray, as caused, for example, by objects falling off of a vehicle. As the amount of spray shown in Figure 1 is usually generated at highway speeds, collecting this kind of data is extremely dangerous even if done at a proving ground. Even using a high number of data collecting vehicles, the number of examples obtained may be too low for training.

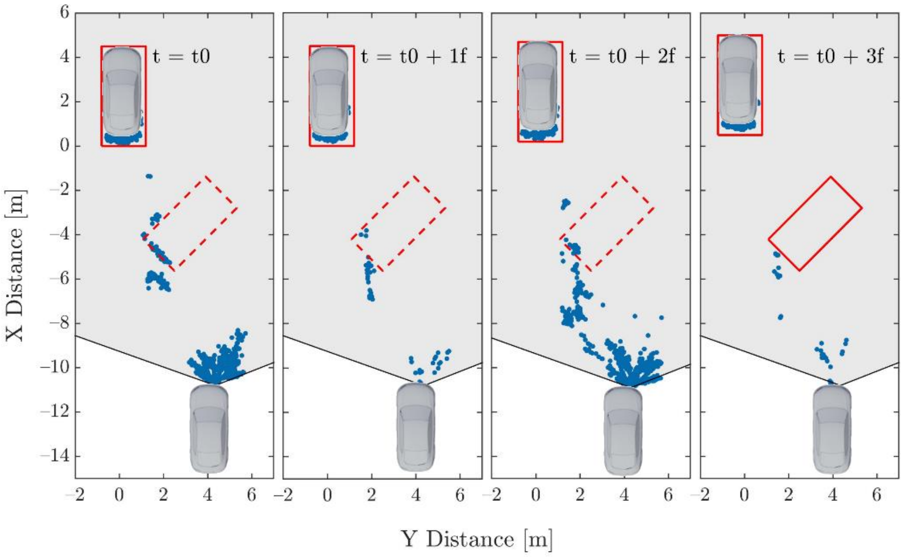


Figure 1. Bird's eye view of a sequence in which a ghost object is generated due to spray water. A wrong object hypothesis created at the beginning (dashed line) is not rejected in subsequent frames; as a consequence, an extra object is generated (solid line). The field of view of the sensor is shown in light grey (based on [10]).

State of the Art

Regarding data augmentation, there have been multiple approaches going from transformations of a real training set [11–16], to combinations of real and synthetic data [17–20], to purely synthetic data [21–23] and domain adaptation techniques [24–26]. In [12], the point clouds of previously labeled objects are added by concatenation at different positions into the training data in order to improve the training of the network. Possible collisions between added objects are automatically removed. In [15], an adversarial strategy is used in order to simultaneously train an 'augmentor,' which is a neural network specialized in creating augmented data which causes a reduction in the performance of a classifier. When trained simultaneously, the classifier learns features that are independent of the possible modifications caused by the augmentor. As there is a broad range of operations that can be applied for augmentation, in [16] a search method is proposed in order to find the optimal augmentation policy, understood as a series of augmentation operations with their respective parameters.

Similar to our method in [17], real scans are used for the background which corresponds to portions of streets in which the movable objects and pedestrians are removed. Synthetic movable objects and pedestrians are then placed with a number, position and pose based on probabilities extracted from labeled real datasets. A sensor model is then used to generate the final point clouds. This permits the generation of high amounts of annotated 3D data for training. A similar approach is used in [18]. In this case, a synthetic object is added on top of the roof of vehicles in real point clouds. This synthetic object is modeled in order to attack the point cloud-based classification algorithm, reducing its accuracy and hence identifying possible vulnerabilities.

Unique to our approach is the use of real point clouds affected by spray water. The collection of this data and the setup used are explained in Section 2. The ROI (region of interest) in which the synthetic object is generated is based on previous results regarding the spatial distribution of spray and the possible actions that can be taken to protect the passengers.

As the synthetic object needs to match the characteristics of a real object placed in the ROI, spray itself has to be simulated. As done in previous works, the scene is rendered. In our case, a physically based render is used to simulate the LIDAR sensor [27] with material properties based on real reflection values. Spray and rain are simulated using particle systems which generate detections with a spatial distribution similar to those caused by real spray [10]. After rendering, the detections caused by the obstacle are extracted and concatenated into the real point clouds. Finally, the effect of occlusion and noise are added.

With the obtained point clouds, a two-layer feedforward neural network is trained. The results are presented in Section 3. The classifier is trained to identify detections caused by the added obstacle. The method uses a bird's eye view of the 3D point cloud in which detections are assigned to bins in a 2D histogram similar to what is done in [28,29] for the extraction of vehicle bounding boxes. The obtained histogram(s) is then convoluted with objects having the dimensions and orientation corresponding to the features that need to be extracted. The use of the bird's eye view with fixed histogram sizes and a simple network for classification allows to keep the required time budget. Classification based on detections instead of a binary classification of the whole scene has as advantage that the decision to activate any safety mechanism can be made based on the extracted obstacle: dimensions, position, rotation and/or reflection and the characteristics of the ego vehicle, such as its ground clearance, for example. The results are discussed in Section 4.

2. Experiment

2.1. Region of Insterest

The region at which spray detections are usually seen, based on our measurements on the highway and considering that the spray is generated by one leading vehicle (LV), corresponds approximately to 6 m to each side of the ego vehicle and up to 20 m in front of it [10]. These measurements were taken with an average ego speed of 89 km/h. Under these circumstances, if a solid object is present in the spray cloud, it is not possible to correct the vehicle trajectory in time. An emergency stop would also not be enough to stop the vehicle before hitting the possible object and would be quite dangerous. There are, however, corrective measures that can be taken. The actuator of the seatbelt pretensioner can be activated as fast as 100 ms to 300 ms [30]. There are also numerical studies regarding the rotation of the seats, in which rotations up to 90° under 200 ms are considered possible and useful in order to protect the passengers [31]. Additionally, a slight brake can be applied. Taking 200 ms as the average for the actuator and adding 300 ms for the detection and classification of the object yields a total of 500 ms. Based on the previously mentioned average speed of our spray measurements, this corresponds to a distance of 12.36 m. Additionally, in order to take curved roads into account and the possibility that a falling object moves from an adjacent lane to the ego lane, we used a cone-shaped ROI. The angle is based on the value of the vehicle exit angle for European roads, which corresponds to the angle between the road edge line and the vehicle straight trajectory [32]. An angle of 30°, which should cover most of the left and right curves found in European highways, was chosen. The final ROI is shown in Figure 2. During the measurements the ego vehicle occupied either the second or third lane while the LV occupied the first or second lane, respectively. For this reason, to the right there could be either a road boundary or another lane. This makes the classification more challenging, as can be seen in Figure A1 in Appendix A.

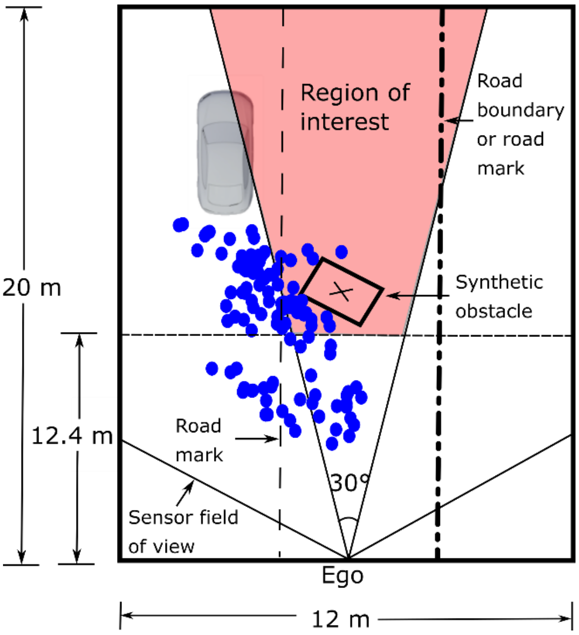


Figure 2. Region of interest. The size is based on the region in which spray is usually detected [10] for the LIDAR and speeds used. Before 12.4 m, even if a solid object is detected no meaningful action can take place. The 30° are based on the curve radii found in European roads. The 'x' marks the center of the added synthetic object.

2.2. Measurement

An automotive LIDAR sensor on the position indicated in Figure 3 was used for data collection. The measurement was taken on a highway during the dates indicated in Table 1. Those days were rainy days. The rainfall during the measurement as reported by nearby weather stations is also indicated [10].

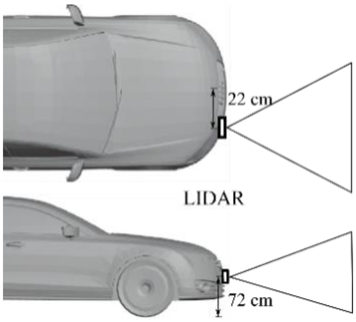


Figure 3. Placement of the LIDAR sensor in the ego vehicle (based on [10]).

Table 1. Highway measurements (taken from [10]).

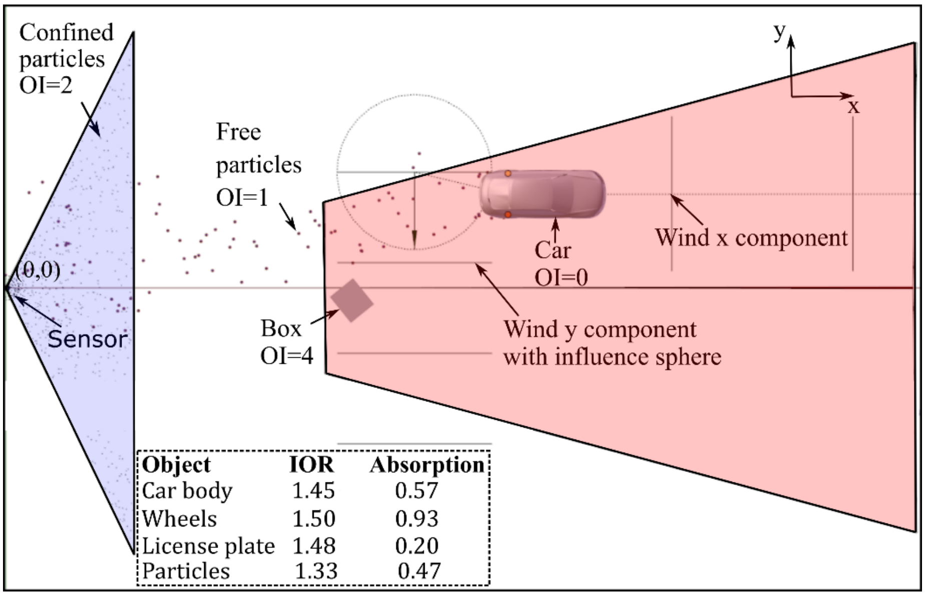
Date	Duration	Rainfall
13 April 2018	1 h 17 min	2 mm
16 May 2018	3 h 26 min	3.6 mm
17 May 2018	2 h 20 min	3.1 mm

A total of 13.162 frames were collected. In each measurement there is a LV, which generates spray, located in the lane to the left of the ego vehicle. The position of the LV was tracked using the internal tracker of the sensor and is reported in Figure A3. The ego speed had a value of (89 ± 4) km/h.

2.3. Simulation

The simulation setup used is presented in Figure 4. This setup is based on the model introduced in [10]. The original simulation setup was based on the following assumptions:

- Simulating real water drops as well as the exact forces that act upon them cannot be done in real time. However, the number of detections that those drops cause is usually much smaller as a high concentration of drops is needed to cause a reflection strong enough to cause a detection. Therefore, a more efficient approach is to use particles to directly simulate detections instead of single drops.
- The solver used by Blender [33] to calculate the trajectories of the particles is stable in the required parameter range.
- The LIDAR sensor itself can be simulated in the same way as a camera image is rendered by adapting the calculation done inside the material shader. The calculation is changed from the default multiplication of intensities to an addition of ray distances. One color channel is left unchanged; as a result, each camera pixel contains the information about the distance that the ray traveled from the light source and its corresponding intensity. The light source is placed next to the camera and both camera and light source are configured based on the resolution and field of view of the sensor.
- If physically based values are used for the materials, as shown inside the dashed line region of Figure 4, the calculated intensities per pixel should be proportional to the ones obtained using the real sensor.



Object	IOR	Absorption
Car body	1.45	0.57
Wheels	1.50	0.93
License plate	1.48	0.20
Particles	1.33	0.47

Figure 4. Simulation setup. The object index (OI) is a parameter used by the render to create a separate mask for the object to add composition effects [34] and makes it is possible to identify which object caused which detections. Notice the correspondence with Figure 2. The ROI is marked in red. The region in violet corresponds to a volume in which confined particles are uniformly distributed to simulate the detections caused by rain. The index of refraction (IOR) and absorption used for the material of each object is shown in the dashed line region [10].

These assumptions were validated by comparing the trajectories of the particles generated by Blender with those generated by a fourth order Runge-Kutta solver for the required parameter range. Additionally, the correlation of the spatial distribution of the real and simulated point clouds was calculated using the Pearson correlation coefficient obtaining values going from 0.5 to 0.74. The rendering time per frame using an Intel CORE i7-8850H CPU varies from 40 ms (8 particles per wheel per frame) up to 260 ms (50 particles per wheel per frame) [10].

In this setup, the camera was configured on the basis of the resolution of the sensor but, instead of a one-to-one relation, a three-to-one relation was used. The extra pixels were used to simulate multiple echoes: secondary and tertiary reflections coming from the same direction. Additionally, the confined particles inside the violet volume were unique to this simulation. These detections were needed in order to make the echo values in simulation and in reality coincide. The number of free and confined particles used is presented in Table 2.

Table 2. Number of particles used.

Particles	Number of Particles	Distribution	Mesh Diameter
Free	~50 per wheel per frame	Defined by acting forces	1.8 cm
Confined	200.000	Uniformly	6 mm

A box was used as an obstacle. Any other obstacle shape could be used, but we decided to use a box for simplicity in the description of its shape.

The position of the LV in the simulation was set equal to the real position for the corresponding frame. This means every real frame has an equivalent synthetic frame. The number of free particles in the simulation was based on the number of detections caused by spray in reality [10]. These particles are emitted from the wheels, and their size and the forces that act upon them (i.e., wind x and y components) are calibrated on the basis of physical values and the spatial distribution of spray in real measurements [10]. For this specific simulation, the effect of the wind in y, which originally is simulated using two forces, was reduced to only one and the value of this force increased to direct the spray towards the ROI.

The confined particles were distributed uniformly over the violet volume in Figure 4 but were emitted randomly with a lifetime of one frame. This generates a distribution of detections similar to the one observed in real rain measurements using the same sensor [8,10].

In order to calibrate the intensity of the light source, the accumulated histogram of the real and synthetic echo pulse width (EPW, proportional to the intensity of the reflection [8]) values were compared. This was done before adding the box to the simulation. The intensity of the lamp in the simulation was adjusted to generate a similar intensity range as in the measurements.

After calibration of the light source was completed, the box was added to the simulation. The position of the center of the box is distributed uniformly over the whole ROI. It was verified that there was no collision between the box and the LV. Possible collisions with the point clouds generated by other vehicles or by the road boundary in the real data were dealt with afterwards. The width, length and height of the box were sampled from a uniform distribution going from 20 cm to 2 m. The z position of the box was adjusted to be half its height to ensure that it remained on the street surface. The rotation was uniformly distributed between 0° and 90° as is the reflectivity, which goes from 0 to 1. Its duration, which corresponds to the number of frames in which it remains in the same position, was varied between 1, 2 and 3. An equal number of examples for each duration was used. Figure 5 presents the results obtained by accumulating the boxes generated over multiple frames keeping the LV position constant. The transparency in the boxes in the figure is proportional to their reflection value.

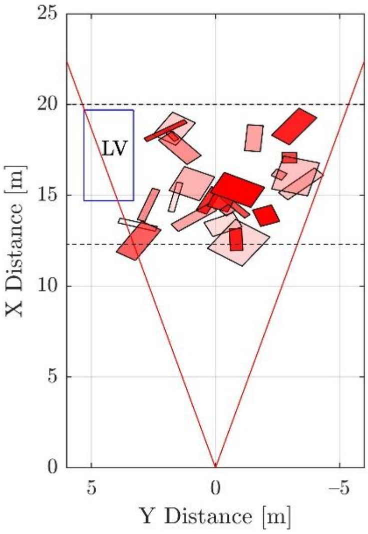


Figure 5. Examples of the generated synthetic obstacles accumulated over multiple frames. The red and dotted lines mark the ROI. The transparency of the box is proportional to its reflection value. The blue box marked 'LV' corresponds to the position of the leading vehicle and collisions with it are automatically removed.

Figure 6 shows the results obtained after rendering. The left side of the figure uses colors for the detections based on their EPW, while the right side uses different colors to identify the corresponding echo number. Three different frames are shown. In Figure 6a,b the box is not added. A direct comparison can be made with Figure 7a,b which corresponds to the real equivalent (same position of LV). As can be noticed, there are similarities in the distributions of the EPWs (before training the classifier the histogram of the EPW values of the detections caused by the synthetic object is adjusted to resemble the one obtained by the real reflections on the rear of the LV (see Figure A4). This is done in order to avoid biasing the classifier towards this difference. The histograms are also trimmed to have the same maximum value.) and echo values. The region close to the sensor (<5 m) contains detections caused mostly by rain, with very low intensity and an echo number of one or two. The detections caused by spray tend to have a very low to low EPW. Mostly first and second echoes are seen with very few third echoes. Something similar happens with the vehicle itself, which is a reason why removing detections caused by spray is difficult. The intensities and echoes are in similar ranges, and only the shape allows for a clear differentiation.

figfig:sensors-1246607-f006c,d shows a frame in which the box is added, causing occlusion to a portion of the vehicle (the box height goes from 20 cm to 2 m which allows for the occlusion of multiple layers). Notice how the distribution of the intensities in the box makes physical sense with the size perpendicular to the sensor having a higher intensity. The echo number also coincides to what would be expected. Finally, in Figure 6e,f an example is given in which the box does not occlude the vehicle. This occlusion, which happens naturally in the simulation, has to be artificially added when the real data is augmented.

After the synthetic data is generated, the object with OI 4 is extracted and concatenated to the corresponding frame of the measured point clouds. Additionally, Gaussian noise is added to the detections caused by the box. This noise has a standard deviation of 5 cm and mean of zero. The value of the standard deviation is extracted from real reflections caused by the rear of the LV. The effect of adding the noise is shown in Figure 7e,f.

As previously explained, the cases in which the box intercepts with the LV are automatically removed. It is, however, possible that the box intersects with the road boundary or with vehicles in the adjacent lane as shown in Figure 8c. Additionally, the presence of the box would occlude objects further in the light path as shown in Figure 8a. In order to tackle these two issues, an algorithm based on the radial histogram of the detections per layer is used (Figure 8 right, radial histograms).

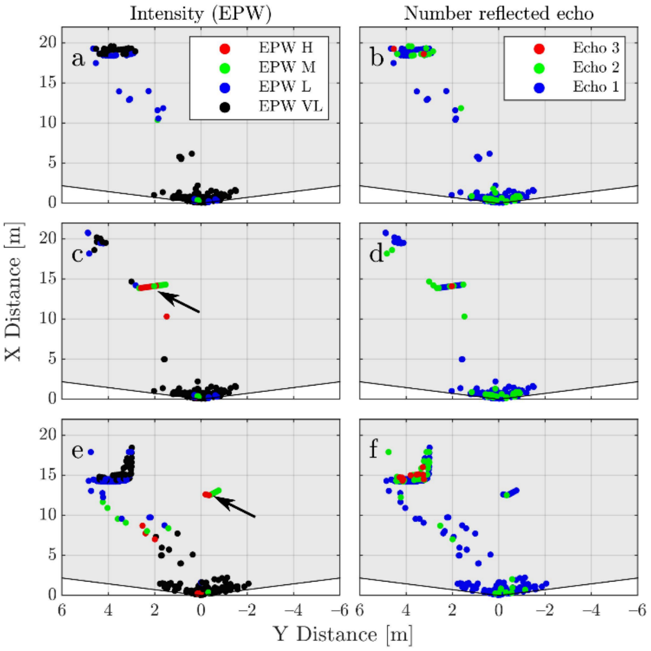


Figure 6. Examples of the generated synthetic frames. On the left side the detections are colored based on the EPW value and on the right are based on the echo number. In (**a**,**b**) there is no added obstacle. In (**c**,**d**) the effect of occlusion caused by the box is shown. In (**e**,**f**) a box that does not cause occlusion is added. The arrows in (**c**,**e**) show the position of the obstacle.

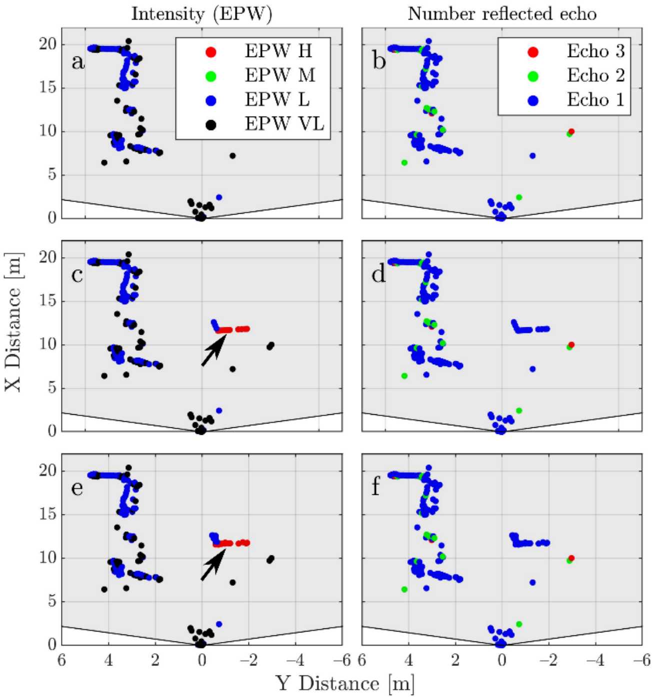


Figure 7. Examples of a real frame before (**a**,**b**), after augmentation (**c**,**d**) and after adding noise (**e**,**f**). On the left side the detections are colored based on the EPW value and on the right based on the echo number. The frame in (**a**,**b**) is the corresponding real frame to the synthetic one shown in Figure 6a,b. The arrows mark the position of the box after augmentation. Notice how the intensity distribution on the box makes physical sense due to the way in which the synthetic data was generated. The echo number is also plausible. The effect of adding noise extracted from real measurements (**e**,**f**) also increases the similarity.

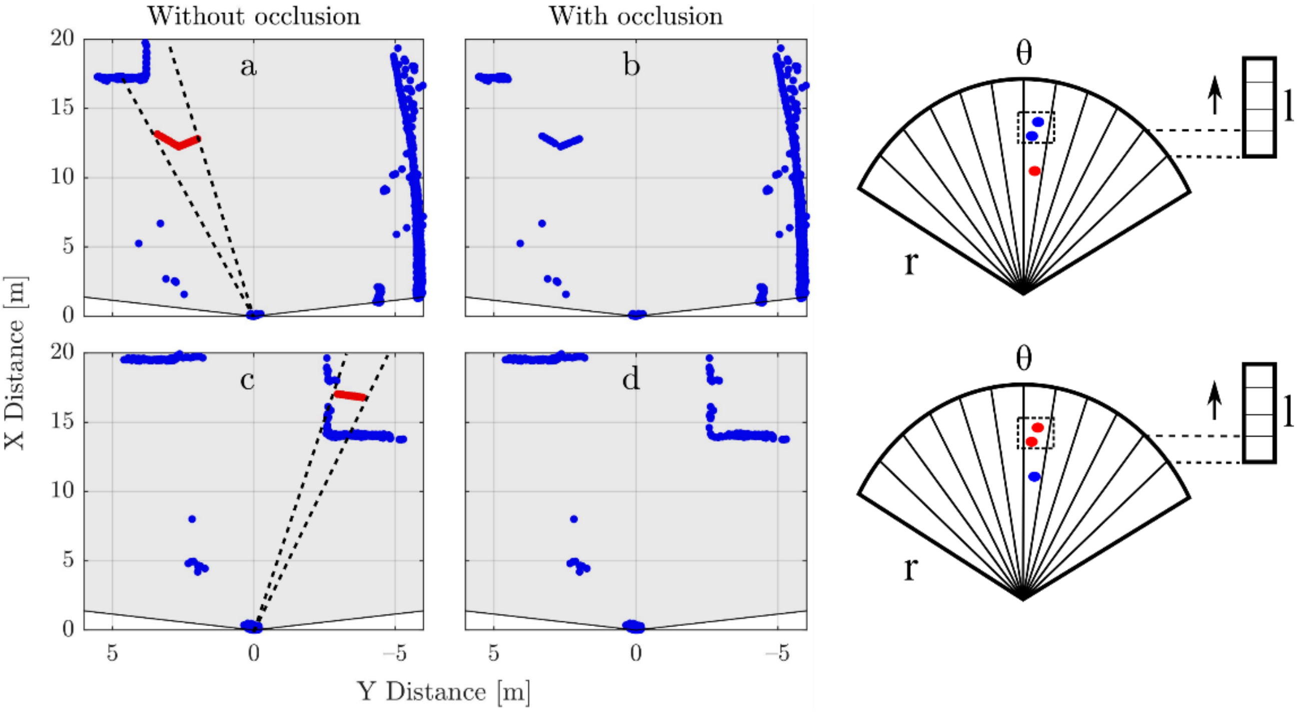


Figure 8. Examples of cases in which occlusion has to be calculated after adding the synthetic object. In (**a**) the detections caused by the LV in the same angular region and layer as the obstacle (dotted line) should be removed. Equivalently in (**c**), the obstacle itself should be removed as its random position happens to be inside another vehicle. The (**b**,**d**), respectively. The angular bin size needs to be selected to avoid strong undersampling of the point cloud if the value is too high. However, if it is too small, the desired occlusion effect does not take place. In our case, a value slightly higher than the resolution of the sensor was used. The occlusion is calculated before the Gaussian noise is added to the synthetic detections.

The algorithm works as follow:

1. For each frame, the radial histogram per layer is calculated.
2. The bins that contain detections caused by the box are extracted.
3. The detections inside the bins are organized in ascending order based on their radial distance from the sensor.
4. The first detection is used and the rest are removed. The detections in the region from 0 to 5 m are not taken into consideration as their effect was already included when generating the synthetic data.

Notice that the examples shown in Figure 8a,c and in Figure 6c,d correspond to cases in which all layers are occluded. This is done to clearly identify the effect but would only happen in rare cases in which the synthetic object's height approaches the upper limit of its range (2 m).

3. Results

As a classifier, a feedforward neural network was used with two layers, one hidden layer with 10 neurons and one output layer. As a training function, scaled conjugate gradient backpropagation was used. Cross entropy was used to evaluate performance.

As output classes: No Box (NB) and Box (B) were used. The NB class included the detections caused by spray, other vehicles, the road boundary and raindrops. The B class corresponded to the detections caused by the synthetic obstacle. In order to compensate for the class imbalance, an error weighting of 20% was used for the NB class and of 100% for the B class.

For training, the region from −6 m to 6 m in 'y' and 0 to 20 m in 'x' was divided into histograms with different bin sizes. The following bin sizes were used on the basis of the features they should help to identify:

- Small features: 10 cm in 'x' by 10 cm in 'y'.
- Vehicle rear: 20 cm in 'x' by 2 m in 'y'.
- Vehicle side: 4 m in 'x' by 10 cm in 'y'.
- Middle size features: 50 cm in 'x' by 50 cm in 'y'.
- Radial: 1°.

The following classification parameters were used (per detection). The number of counts correspond to those of the bin containing the detection for the corresponding histogram:

1. X position.
2. Y position.
3. Echo number.
4. Layer number.
5. EPW value.
6. Number of counts (radial).
7. Number of counts (small features).
8. Number of counts (vehicle rear).
9. Number of counts (vehicle side).
10. Number of counts (middle size feature).
11. Absolute value of the subtraction of parameter six of the current frame from parameter six of previous frame.
12. Convolution with horizontal matrix (Appendix B) (small features).
13. Convolution with corner matrix[1] (small features).
14. Convolution with corner matrix[1] rotated 90° (small features).
15. Convolution with impulse matrix[1] (small features).

Table 3 presents the classification results using the F-Score as metric. The results are divided between 'All durations,' which corresponds to the classification results for durations of one, two and three frames, and 'Duration 3f' which contains only the examples in which the object remains in the same position for three frames. The results for durations of one and two frames were very similar and hence are not shown. An F-Score for the B class of 92.13 and 92.26 was obtained, respectively. This indicates, as expected, that the classification improves the longer the box remains in the same position and that adding more parameters like parameter 11 could further improve the results.

Table 3. Classification results.

		Predicted Class			
		All Durations		**Duration 3f**	
		NB	**B**	**NB**	**B**
Actual class	NB	12,454,236	20,036	4,150,856	6686
	B	3848	141,545	1178	47,055
	F-Score	99.90	92.22	99.91	92.29

In order to find problematic regions in the parameter space, Figure 9 shows the results obtained by using the previously trained classifier on smaller clusters of the data. Here the focus is on the effect of changes in the obstacle size and the position of the LV. Further dependencies can be seen in the Figure A2.

Regarding execution times, a single threaded, not optimized version of the proposed algorithm is required to extract the feature vector and complete the inference of 182 ± 51 ms per frame, using C++ in a computer with an Intel CORE i7-8850H CPU. The variation was caused by the changing number of detections. However, current platforms used for AV like RENESAS R-Car and NVIDIA Drive integrate accelerators for image processing tasks [35,36], which, given our use of fixed histogram sizes, can be employed to highly parallelize the computation, reducing the execution time. If for example three consecutive sensor frames are used, this corresponds to ~120 ms [37], and segmentation of the ROI for each of those frames should take few milliseconds, leaving around 150 ms for the calculation of the feature vector and classification which should be attainable. The time required to generate the synthetic data and train the neural network is not included as it is assumed that it can be done offline. The pretrained network can then be used in a vehicle and updated if needed over the air.

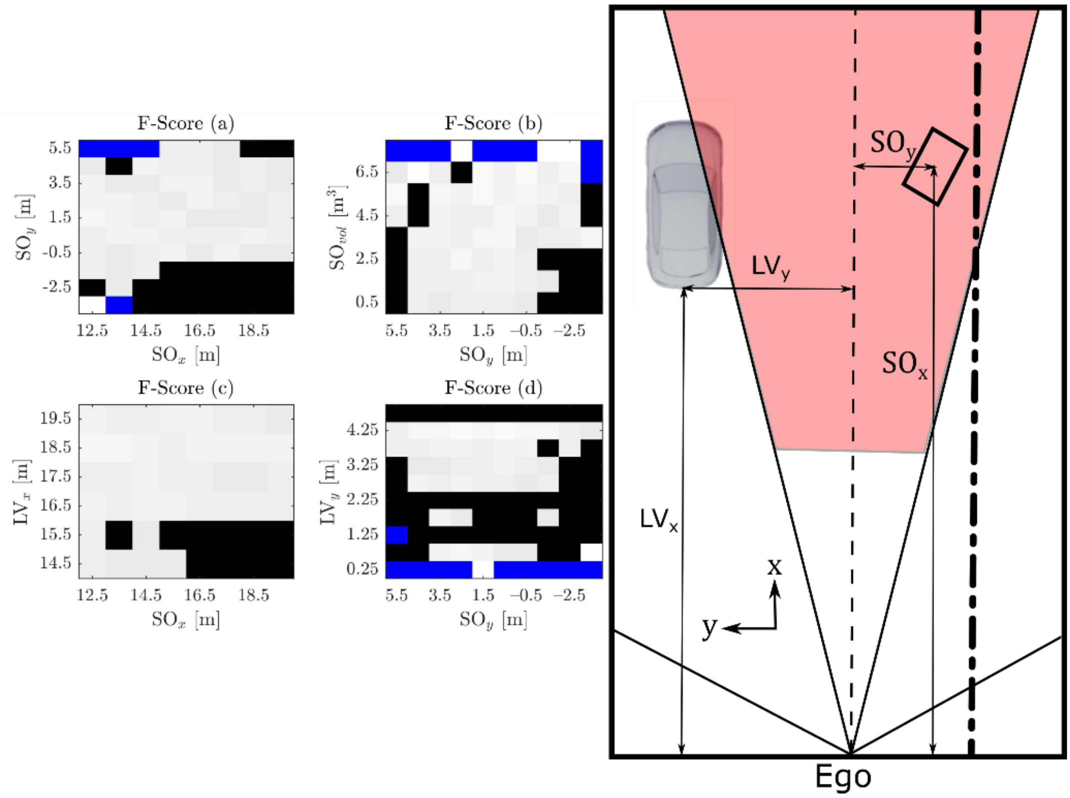


Figure 9. Classification results are analyzed in detail by dividing the available data into small portions. Regions where the F-Score is lower than 92% are shown in black. Blue regions are empty (the combination of parameters currently does not contain any example). On the right, the meaning of each of the parameters is shown. SO_{vol} corresponds to the volume of the synthetic object, SO_x and SO_y to its coordinates. LV_x and LV_y correspond to the coordinates of the leading vehicle. In both cases, the position is calculated with respect to the position of the LIDAR sensor on the Ego vehicle. The difficulty of the classification task increases when the synthetic object is located close to the road boundary (**a**,**b**) or when is close to the LV and has a small size (**b**), additionally when the synthetic object is farther away in x-direction as the LV (**c**) or when the LV moves to the Ego lane (**d**).

4. Discussion

Figure 9 shows that the classification becomes difficult when the center of the box moves closer to the road boundary $SO_y < -2.5$ m Figure 9a,b. This is due to the multitude of shapes that the road boundary can generate (see Figure A1), which can be easily confused with a rectangular-shaped object. Additionally, when the LV moves into the ego lane while overtaking $LV_y < 2.25$ m Figure 9d, the accuracy of the classifier is reduced as it probably learns that when there is an object in the ego lane, it is most likely an obstacle. Something similar happens when the synthetic object is far away from $SO_x > 16.5$ m Figure 9c. In this case especially, if the box is not big enough and there is lots of spray, it is likely to result in a misclassification. Finally, when the LV is quite far to the left $LV_y > 4.25$ m Figure 9d or close to the beginning of the ROI $LV_x < 15.5$ m Figure 9c, only a portion of the vertical sideline of its shape can be seen. This is easier to misclassify than the usually expected 'L' shape. Although these regions are challenging to classify, accuracy in general is high. This would not be the case if the probability of finding objects in the driving lane for the used ROI were not as low. For instance, at lower speeds or in urban traffic the task would be more challenging, as the position of the synthetic object would not be as important a classification parameter as it is on a highway at high speeds. Additionally, although care was taken to make the synthetic data as similar as possible to the real data, it can be that some of the parameters are still different enough to make the classification task easier. For instance, a unique reflection value was used while real obstacles would probably be composed of different materials with different reflection values.

The presented model constitutes a basis upon which more complex scenarios can be simulated with relatively small changes. For example, the type of synthetic object can be easily changed and its position can be animated or simulated as desired. Their number can also be increased and they do not have to be static. For instance, they can simulate other vehicles. Real and synthetic data can also be combined in different ways. The spray could be generated completely synthetically, keeping just the real detections caused by the LV, or a synthetic LV can be used leaving just the real spray detections.

The proposed methodology is based on open-source tools and can be easily applied to other LIDAR sensors. It can also be used to simulate other weather-related effects. For instance, the simulation of the detections caused by exhaust gases.

A certain amount of data collection is needed as many of the simulation parameters are extracted from the characteristics of the real point clouds. It is, however, a much easier collection task as no real obstacles need to be present on the spray region.

5. Conclusions

Regarding future research directions, even though a shallow neural network was considered enough for the current use case, the use of a more sophisticated network type and architecture can increase classification accuracy. This is especially true if other obstacle types are introduced, for instance bicycles, tires, vehicle parts, etc., in which case manually defining the classifier features is difficult. Additionally, even though only one reflection value for the whole obstacle object was used, a texture can be used to give different parts different reflection values. More complex physics can also be added for the obstacle, such as bouncing or breaking apart.

Author Contributions: Conceptualization, J.R.V.R. and T.G.; methodology, J.R.V.R. and T.G.; software, J.R.V.R.; validation, J.R.V.R. and T.G.; investigation, J.R.V.R.; resources, J.R.V.R. and B.B.; data curation, J.R.V.R.; writing—original draft preparation, J.R.V.R.; writing—review and editing, J.R.V.R., T.G., J.C. and B.B.; visualization, J.R.V.R.; supervision, T.G., J.C. and B.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ADAS	Advanced driver assistance systems.
AV	Autonomous vehicles.
EPW	Echo pulse width.
GPS	Global positioning system.
IOR	Index of refraction.
LIDAR	Light detection and ranging.
LV	Leading vehicle.
OI	Object index.
ROI	Region of interest.

Appendix A

Some of the road boundary shapes are shown in Figure A1a–c. Figure A1d shows another vehicle. As can be noticed, the road boundary can generate shapes that can be easily confused with other objects and, due to curves, it is not limited to a specific region. Other vehicles like trucks can also generate complex point clouds.

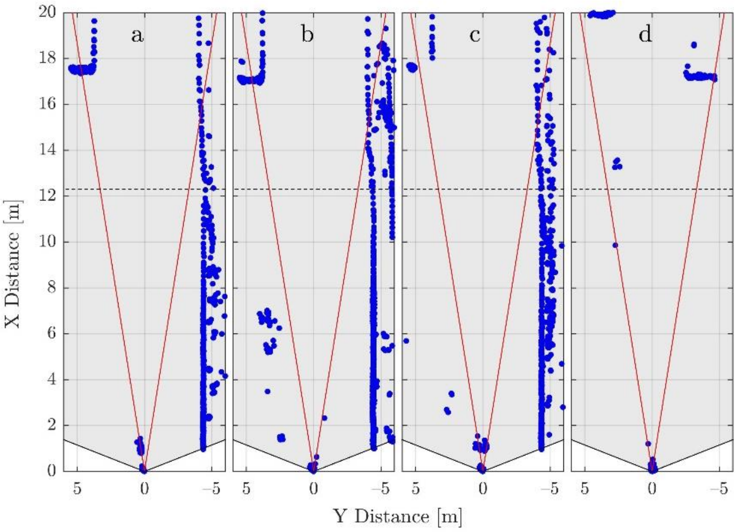


Figure A1. Examples of the point clouds generated by different road boundaries (**a**–**c**) and vehicles in other lanes (**d**). The ROI is limited in 'y' by the read lines and in 'x' by the dotted line.

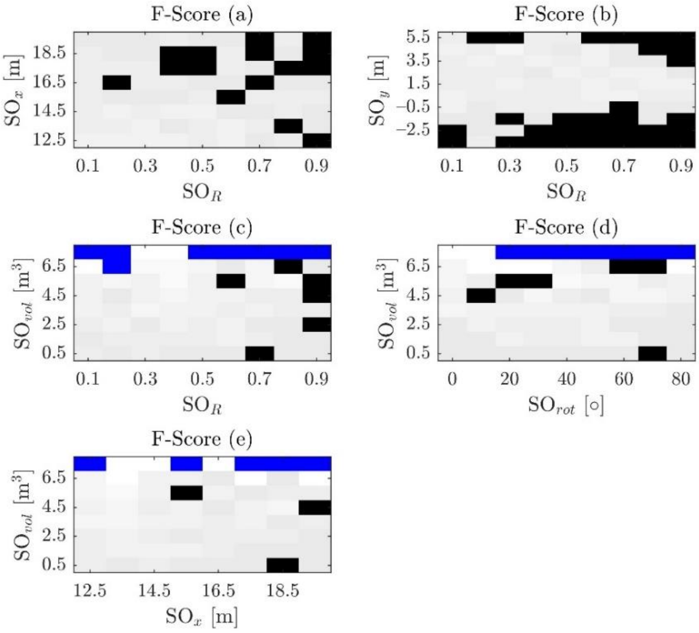


Figure A2. Continuation of Figure 9, including the dependencies for other parameters. In black regions with a F-Score lower than 92% are shown, and blue regions that are empty are shown (the combination of parameters currently does not contain any example). Besides the already explained parameters, SO_R corresponds to the reflection of the synthetic object and SO_{rot} to its rotation in degrees. In (**a**–**c**) high reflection values for the synthetic object seem to reduce the F-Score. A high reflection generates a high EPW value, high EPW values are more likely on the LV (Figure A4) making the classification harder. The rotation (**d**) and volume (**c**–**e**) do not seem to have a strong influence by themselves.

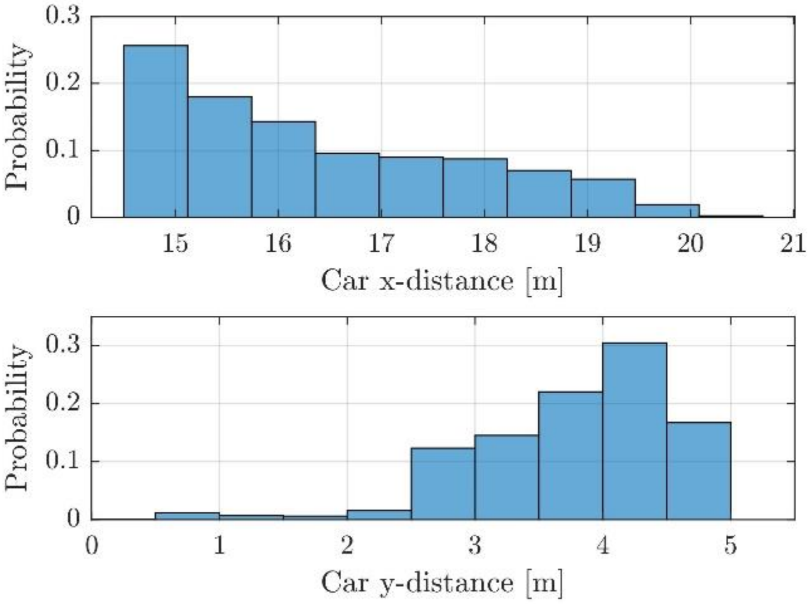


Figure A3. Probability distribution for the position of the LV with respect to the ego vehicle. The reference point in the LV is the middle of the lower side of its bounding box (considering bird's eye view).

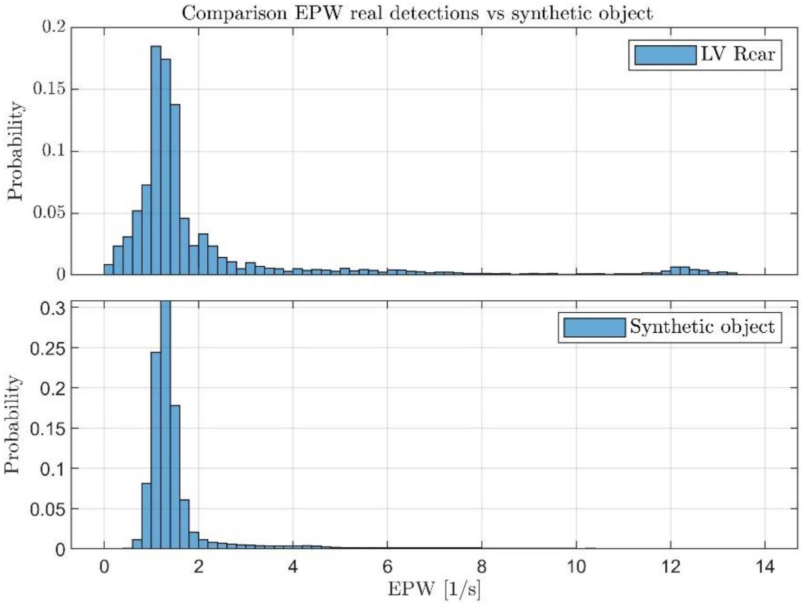


Figure A4. Histograms showing the probability distributions of the EPW values for the rear part of the real LV (up) and the synthetic object (down). The mean and standard deviation as obtained after rendering the scene for the synthetic object are adjusted to resemble those of the LV.

Appendix B

Matrices used in the convolutions: impulse (A1), horizontal (A2) and corner (A3). These matrices are used to identify features such as isolated detections as present in the spray cloud or parts of the synthetic object or leading vehicle.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 4 & -1 \\ -1 & -1 & -1 \end{pmatrix} \tag{A1}$$

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{pmatrix} \tag{A2}$$

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & -1 & 2 & 2 & -1 \\ -1 & -1 & -1 & 2 & -1 \\ -1 & -1 & -1 & 2 & -1 \end{pmatrix} \tag{A3}$$

References

1. Kocić, J.; Jovičić, N.; Drndarević, V. Sensors and sensor fusion in autonomous vehicles. In Proceedings of the 2018 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 20–21 November 2018; pp. 420–425.
2. Kim, J.; Han, D.S.; Senouci, B. Radar and vision sensor fusion for object detection in autonomous vehicle surroundings. In Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech, 3–6 July 2018; pp. 76–78.
3. Wang, Z.; Wu, Y.; Niu, Q. Multi-sensor fusion in automated driving: A survey. *IEEE Access* **2019**, *8*, 2847–2868. [CrossRef]
4. Göhring, D.; Wang, M.; Schnürmacher, M.; Ganjineh, T. Radar/lidar sensor fusion for car-following on highways. In Proceedings of the 5th International Conference on Automation, Robotics and Applications, Wellington, New Zealand, 6–8 December 2011; pp. 407–412.
5. Verucchi, M.; Bartoli, L.; Bagni, F.; Gatti, F.; Burgio, P.; Bertogna, M. Real-Time clustering and LiDAR-camera fusion on embedded platforms for self-driving cars. In Proceedings of the 2020 Fourth IEEE International Conference on Robotic Computing (IRC), Taichung, Taiwan, 9–11 November 2020; pp. 398–405.
6. Rivero, J.R.V.; Tahiraj, I.; Schubert, O.; Glassl, C.; Buschardt, B.; Berk, M.; Chen, J. Characterization and simulation of the effect of road dirt on the performance of a laser scanner. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
7. Heinzler, R.; Schindler, P.; Seekircher, J.; Ritter, W.; Stork, W. Weather influence and classification with automotive lidar sensors. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1527–1534.
8. Vargas Rivero, J.R.; Gerbich, T.; Teiluf, V.; Buschardt, B.; Chen, J. Weather Classification Using an Automotive LIDAR Sensor Based on Detections on Asphalt and Atmosphere. *Sensors* **2020**, *20*, 4306. [CrossRef]
9. Hasirlioglu, S.; Riener, A.; Huber, W.; Wintersberger, P. Effects of exhaust gases on laser scanner data quality at low ambient temperatures. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1708–1713.
10. Vargas Rivero, J.R.; Gerbich, T.; Buschardt, B.; Chen, J. The Effect of Spray Water on an Automotive LIDAR Sensor: A Real-Time Simulation Study. *IEEE Trans. Intell. Veh.* **2021**. [CrossRef]
11. Yang, B.; Luo, W.; Urtasun, R. Pixor: Real-time 3D object detection from point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7652–7660.
12. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
13. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3D object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
14. Hahner, M.; Dai, D.; Liniger, A.; van Gool, L. Quantifying Data Augmentation for LiDAR based 3D Object Detection. *arXiv* **2020**, arXiv:2004.01643.
15. Li, R.; Li, X.; Heng, P.-A.; Fu, C.-W. PointAugment: An Auto-Augmentation Framework for Point Cloud Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6378–6387.
16. Cheng, S.; Leng, Z.; Cubuk, E.D.; Zoph, B.; Bai, C.; Ngiam, J.; Song, Y.; Caine, B.; Vasudevan, V.; Li, C.; et al. *Improving 3D Object Detection through Progressive Population Based Augmentation*; Springer: Cham, Germany, 2020.
17. Fang, J.; Zhou, D.; Yan, F.; Zhao, T.; Zhang, F.; Ma, Y.; Wang, L.; Yang, R. Augmented LiDAR Simulator for Autonomous Driving. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1930–1937. [CrossRef]
18. Tu, J.; Ren, M.; Manivasagam, S.; Liang, M.; Yang, B.; Du, R.; Cheng, F.; Urtasun, R. Physically Realizable Adversarial Examples for LiDAR Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
19. Yue, X.; Wu, B.; Seshia, S.A.; Keutzer, K.; Sangiovanni-Vincentelli, A.L. A lidar point cloud generator: From a virtual world to autonomous driving. In Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, Yokohama, Japan, 11–14 June 2018; pp. 458–464.
20. Feng, Y.; Liu, H.X. Augmented reality for robocars. *IEEE Spectr.* **2019**, *56*, 22–27. [CrossRef]
21. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. *arXiv* **2017**, arXiv:1711.03938.

22. Johnson-Roberson, M.; Barto, C.; Mehta, R.; Sridhar, S.N.; Rosaen, K.; Vasudevan, R. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv* **2016**, arXiv:1610.01983.
23. Griffiths, D.; Boehm, J. SynthCity: A large scale synthetic point cloud. *arXiv* **2019**, arXiv:1907.04758.
24. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1887–1893.
25. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4376–4382.
26. Zhao, S.; Wang, Y.; Li, B.; Wu, B.; Gao, Y.; Xu, P.; Darrell, T.; Keutzer, K. ePointDA: An End-to-End Simulation-to-Real Domain Adaptation Framework for LiDAR Point Cloud Segmentation. *arXiv* **2020**, arXiv:2009.03456.
27. Blender Project. Cycles: Open Source Production Rendering. Available online: https://www.cycles-renderer.org/ (accessed on 10 January 2021).
28. Yu, S.-L.; Westfechtel, T.; Hamada, R.; Ohno, K.; Tadokoro, S. Vehicle detection and localization on bird's eye view elevation images using convolutional neural network. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 102–109.
29. Mohapatra, S.; Yogamani, S.; Gotzig, H.; Milz, S.; Mader, P. BEVDetNet: Bird's Eye View LiDAR Point Cloud based Real-time 3D Object Detection for Autonomous Driving. *arXiv* **2021**, arXiv:2104.10780.
30. Skutek, M. *Ein PreCrash-System auf Basis Multisensorieller Umgebungserfassung*; Shaker: Düren, Germany, 2006.
31. Wu, H.; Hou, H.; Shen, M.; Yang, K.H.; Jin, X. Occupant kinematics and biomechanics during frontal collision in autonomous vehicles—can rotatable seat provides additional protection? *Comput. Methods Biomech. Biomed. Eng.* **2020**, *23*, 191–200. [CrossRef]
32. RISER Consortium. Roadside Infrastructure for Safer European Roads. Available online: https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/pdf/projects_sources/riser_guidelines_for_roadside_infrastructure_on_new_and_existing_roads.pdf (accessed on 29 June 2021).
33. Blender Online Community. Blender—A 3D Modelling and Rendering Package. Available online: https://www.blender.org/ (accessed on 29 June 2021).
34. Blender 2.91 Manual. Rendering/Layers and Passes/Passes. Available online: https://docs.blender.org/manual/en/latest/render/layers/passes.html (accessed on 9 January 2021).
35. Liu, S.; Tang, J.; Zhang, Z.; Gaudiot, J.-L. Computer Architectures for Autonomous Driving. *Computer* **2017**, *50*, 18–25. [CrossRef]
36. Alcaide, S.; Kosmidis, L.; Hernandez, C.; Abella, J. Software-only Diverse Redundancy on GPUs for Autonomous Driving Platforms. In Proceedings of the 2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS), Rhodes, Greece, 1–3 July 2019; pp. 90–96.
37. Zeisler, J.; Maas, H.-G. Analysis of the performance of a laser scanner for predictive automotive applications. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *2*, 49–56. [CrossRef]