Measurement Error Simulation for Lidar

Master's thesis in Computer science and engineering

SENA BAYRAKTAROGLU

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2023

Master's thesis 2023

Measurement Error Simulation for Lidar

SENA BAYRAKTAROGLU

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2023

Measurement Error Simulation for Lidar

SENA BAYRAKTAROGLU

Supervisor: Erik Sintorn, Department of Computer Engineering
Advisor: Sergei Borzov, Volvo Autonomous Solutions
Examiner: Ulf Assarsson, Department of Computer Engineering

Master's Thesis 2023
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2023

Measurement Error Simulation for Lidar

SENA BAYRAKTAROGLU
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Autonomous vehicles require plenty of testing and validation, and simulation environments are a good choice for validation since they enable the testing of multiple scenarios. One of the most common sensors in the field is the Lidar sensor, which allows the retrieval of 3D information from the environment, often used with other sensors and cameras. Measurement errors become important when using a Lidar sensor with other sensors. However, these errors are not correctly included or modeled in most simulation environments. This thesis work aims to explore how measurement errors can be included in the simulation environment. A model was developed to find the error's standard deviation depending on the target material's reflectivity and distance and applied with the Weierstrass fractal function to model the repetitive behavior of the error. Results show that Weierstrass fractal function was appropriate for modeling the repetitive behavior. The model developed for the standard deviation of the error was working well in close distances, while further experimentation was needed for long ranges.

Keywords: Light detection ranging (LIDAR), measurement error, simulation

Acknowledgements

Throughout my thesis journey, I am extremely grateful for the invaluable advice and guidance provided by my supervisor, Erik Sintorn. I would like to express my gratitude to my advisor Sergei Borzov for his guidance, as well as to my colleagues from Volvo Autonomous Solutions. Before I conclude, I would like to express my thanks to my family for their support and to my friends who have made this journey a delightful experience.

Sena Bayraktaroglu, Gothenburg, 2023-06-08

Contents

List of Figures

List of Tables

List of Acronyms

AMCW Amplitude-Modulated Continuous Wave.

BLAS Bottom Level Acceleration Structure.

FMCW Frequency-Modulated Continuous Wave.
FOV Field Of View.

LIDAR Light Detection and Ranging.

NIR Near Infrared.

SNR Signal to Noise Ratio.

TDC Time-To-Digital Converter.
TLAS Top Level Acceleration Structure.
TOF Time of Flight.

1

Introduction

Autonomous vehicle technology is a growing and emerging field requiring plentiful testing and validation to ensure safety. However, real-life testing and validation of autonomous vehicles is a challenging problem as it requires testing numerous scenarios that can occur in real life. Simulation environments play an important role in testing and verification as they allow for testing various scenarios in a safe and controlled environment. Many simulation environments are developed using game engines, enabling virtual worlds with roads, buildings, pedestrians, and vehicles that can be tested in virtual environments.

In order to gather information from the environment, various sensors and cameras are used with the vehicle. The Light Detection and Ranging sensor (LIDAR) is one of the most commonly used sensors to obtain 3D information about the environment. The Lidar sensor creates a 3D representation of the environment by sending laser pulses and calculating the distances by measuring the time when a laser pulse returns. The Lidar sensor plays a vital role in perceiving the environment accurately.

In order to get accurate information from the virtual environment, a Lidar sensor simulator is also used in the simulation environment. The most common method of implementing a Lidar model in the simulation environment is to send rays and find the intersection points with the objects in the virtual simulation environment.

1.1 Problem Overview

Lidar sensor output highly depends on the physical interaction of the sent laser with the scene. When the emitted laser returns to the sensor and a point is registered with a distance value, the distance value can be affected by many factors, creating an error in the measured distance. This error becomes important in the 3D reconstruction of the objects in the environment and the integration with other sensors. Lidar sensors are mostly integrated with other sensors, such as cameras. If the simulation environment does not include any measurement error, then 3D construction or detection algorithms can work well in the simulation environments. However, the performance of those algorithms in real life might be different since it is normal to have some measurement error in the output of the Lidar [1]. In order to make sure the development of algorithms based on the more realistic point cloud, which includes measurement errors, it is beneficial to add those errors into the simulation environment too.

1.2 Research Question

Given that including measurement errors in the simulation environment is essential, a research question is developed around "How to include measurement errors in the Lidar simulation environment?".

In order to answer the research question, data will first be collected from the Lidar sensor to observe the nature of the error. Next, the reasons for the error will be investigated. As a solution, an appropriate measurement error model for the simulation environment will be developed, considering the reflectivity and distance of the target object, and a Weierstrass fractal function will be used to simulate the characteristics of the observed data. Finally, the results of the simulated error and the experimental error will be compared.

1.3 Delimitation

While developing a measurement error model for Lidar simulation, one challenge is obtaining a general method that can be applied to most Lidar models. This challenge occurs because most Lidars have specific algorithms to minimize errors, which can vary from Lidar to Lidar. Because of these difficulties, we have chosen to limit our model to the Ouster OS1-64 Lidar.

Another limitation of this thesis work is the Lidar's experimental setup and changing environmental conditions. Lidar measurements can be influenced by various factors such as temperature, wind, and ambient lighting. In order to have a better measurement error model, reference data will be used rather than developing the model based on conducted experiments.

2

Previous Work

Closing the gap between the simulation environment and real life has become an important research area. This gap consists of a lack of reality in advance weather conditions such as snow [2], rain [3], fog [4], simple intensity values per point [5], lack of ray drop effect [6], and point clouds without including any error [7].

One of the works done in this field is to modify simulated Lidar output with a trained deep neural network model to reflect deviations that can occur in real life to the simulated Lidar output. This model includes predicting correct reflectivity values and predicting which points should be dropped. However, in work, no extensive study has been made on measurement error modeling [6].

Most of the studies consider simulating measurement error in the simulation by an additive noise with a constant standard deviation to measured distance [[8], [9], [10], [7]]. These methods are suitable methods for having a generalized error model that can be applied easily. However, they still don't consider the measurement error increase, which might come from lower reflective material and increased distance [11].

One important work focusing on error analysis is by Martin D. He discusses the performance limits and sources of error for range sensors. The work addresses the problem of having an accurate range estimation with a varying surface albedo. It makes analysis to model systematic and random range errors to have a correct calibration procedure. This work is valuable in terms of understanding the source of errors and modeling however it does not focus on adding the range error in a simulation environment [12].

Another work focuses on analyzing the nature of the error by looking into the proportions of the error and then reflecting analysis into the simulation. This work shows that some portions of the error have a correlated nature and some are not correlated but have a more random nature [1].

This thesis aims to build upon previous research by exploring the effect of adding noise with a specific standard deviation and how it is determined based on the target's distance and reflectivity. Additionally, using a fractal model to simulate the correlated error.

3

Background

In order to create an accurate Lidar simulation model, it is essential to have an understanding of how Lidar works. This section will provide detailed explanations of Lidar's scanner and range finder technology, critical performance metrics, how it interacts with various surfaces, and possible sources of error in a Lidar system. In order to gain a better understanding of the design choices in the methods section, it would be helpful to have knowledge of the rangefinder and scanner technology used. Performance metrics will also be described, which will play a role in the methods part. Furthermore, how Lidar interacts with various surfaces will be explained, which will later be used in the results section for the experimental setup choices. The sources of errors that may affect performance metrics will also be discussed, which will help to interpret the results obtained later.

3.1 Lidar Working Principle

Autonomous vehicles need to get information from the environment continuously. This information can be if any pedestrian, vehicle, or building is in front of them and how much they are away from the vehicle. In order to acquire this information, they need various sensors for the perception and localization of the environment. The Lidar sensor is one of the most common sensors used in autonomous vehicles to acquire an immediate environment. In the simplest case, the Lidar sensor shoots lasers into the surrounding, and distances are calculated from reflected lasers. From calculated distances, a point cloud is formed. Point clouds are 3D representations of the environment.

Lidar technology can be described in two sections; one of them is its rangefinder technology, and the other is the scanning system. First, the range finder technology of Lidar will be explained then the scanning technology will be explained.

3.1.1 Output of Lidars

The point cloud is a set of datasets of points to describe a 3D object or an environment. Point clouds in the context of Lidar sensors represent the output of the Lidar Sensors. Points in a point cloud can be represented as a 4-dimension vector (x,y,z, Intensity). (x,y,z) is the cartesian coordinate representation of the hit point, and Intensity is the return strength of the signal. The following sections will discuss how point cloud is formed for different Lidar types.

3.1.2 Rangefinder Technology

Rangefinder technology for Lidars consists of 4 main components: beam transmitter, optics, photodetector, and signal processing circuit [13].

A beam transmitter transmits a beam with a wavelength near-infrared (NIR). When the beam is reflected from the environment, it is received from a photodetector with the help of the optics inside Lidar. The photodetector converts a reflected beam into an electrical signal [14]. Range estimation for Lidar is given with Equation 3.1, where r is a range, delta t is the time of flight, c is the speed of light, n is the refraction of the propagation medium [13].

There are two main techniques for calculating the Time of Flight (TOF): direct TOF calculation and indirect TOF flight calculation [15]. In the direct time of flight calculation, time is calculated from the round-trip travel of the photon. In order to measure the time spent, photon detectors are used with time to digital converter (TDC). In Figure 3.1, it is possible to see how TDC works. TDC can be thought of as a chronometer, so when the photon sent chronometer starts, and when it is received back, it is being stopped. In the direct TOF flight sensor type, TOF is calculated from arrival time as seen in Figure 3.1. The indirect TOF calculation uses both amplitude-modulated (AMCW) and frequency-modulated continuous waves (FMCW). Time of flight calculation is made through phase shifts between transmitted and received signals. In Equation 3.2, distance calculation for FMCW and AMCW Lidar sensors are given where T is the duration of the signal, $\Delta\phi$ phase shift [15], r is the distance.

$$r = \frac{1}{2n} c \Delta t \tag{3.1}$$

$$r = \frac{1}{2} \cdot T \cdot \left(1 - \frac{\Delta\phi}{2\pi}\right) \tag{3.2}$$

(a) Vertical Field of View

(b) Horizontal Field of View

Figure 3.2: Lidar's vertical field of view and horizontal field of view [13]

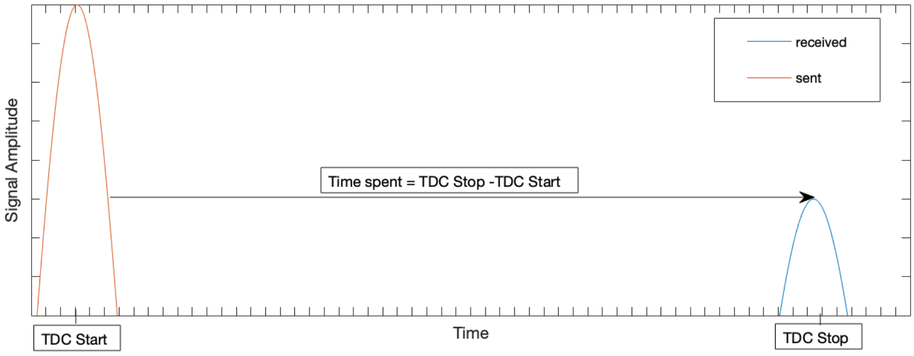


Figure 3.1: Sent and received signal's amplitude

3.1.3 Scanning Technology

Scanning systems for Lidar's used to achieve a good view of the surrounding. Lidar systems have a vertical and horizontal field of view (FOV). There are multiple ways to obtain FOVs vertically and horizontally. In this thesis, only mechanical spinning technology will be used and explained. In the mechanical spinning technology, a vertical field of view is obtained by sending n number of lasers vertically with a field of view with a particular ray distribution. Then, to obtain a horizontal FOV, the base of the Lidar is rotated with a specific frequency. In Figure 3.2, (a) displays vertical rays to obtain vertical FOV while (b) shows the rotation of the base to obtain horizontal FOV.

3.2 Distribution of the Rays

The distribution of rays in the vertical field of view can vary from Lidar to Lidar. Figure 3.5 shows that the angular resolution between rays can either be constant

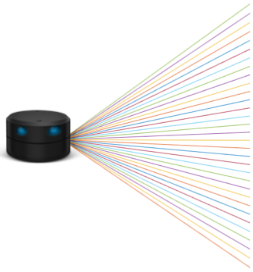


Figure 3.3: Equal angular distribution

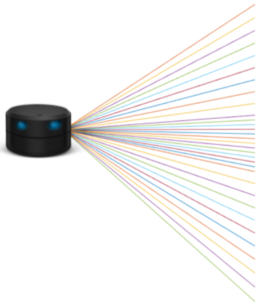


Figure 3.4: Unequal angular distribution

Figure 3.5: Equal and Unequal angular distribution of rays in Lidar system

or varying. Figure 3.3 represents a Lidar ray distribution where the angle between two rays always stays constant, while Figure 3.4 shows a Lidar ray distribution where rays in the middle have a less angular resolution between them and outer rays have more spacing between each other. In an equal ray distribution, the vertical and horizontal angular resolution of the Lidar can be obtained with the following equation;

$$\theta_v = VFOV/n_v \tag{3.3}$$

$$\theta_h = HFOV/n_h \tag{3.4}$$

where θ_v, θ_h are angular resolution and $VFOV$ $HFOV$ are vertical and horizontal field of view respectively and n_v, n_h are the horizontal and vertical number of channels.

3.3 Intensity Measurement

The intensity value for the Lidar is proportional to the received power of the photon. The received power of a Lidar can be modeled with the following formula 3.5:

$$P_r = \frac{P_t c \eta A}{2r^2} \beta T r \tag{3.5}$$

Where P_t is transmitted power, c is the speed of light, A is the area of the receiver, r is the distance of the hit point and sensor, and T_r is the transmission of the medium, β is the Lambertian reflectance of the surface, η is the system efficiency [13]. The received power is used to calculate the intensity of each point in the Lidar point cloud. However, if the received power is smaller than a threshold value, the point is discarded as it is likely to be noise.

3.4 Types of Surfaces

The surface type of the target object in a Lidar system can affect the registered points in the point cloud due to their different light-reflecting properties. In the following sections, the types of surfaces and how they interact with Lidar will be explained.

3.4.1 Specular Surfaces

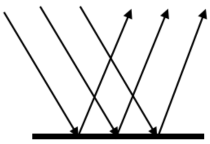


Figure 3.6: Light's reflection on the specular surfaces

Specular surfaces reflect the incoming light at the same angle as seen in Figure 3.6. Examples of specular surfaces are smooth surfaces such as mirrors or standing water. Outgoing radiance for the specular surfaces can be written with Equation 3.6 where $I_{outgoing}$ represents the outgoing radiance, I_{source} is the intensity of the incoming light, ρ is the specular albedo of the material ϕ and θ is the incident direction and ϕ' and θ' is the viewing direction. Therefore, in the case of the specular reflection, if the Lidar's viewing angle is not the same as the reflected laser beam, Lidar will not be able to detect the specular surface [16].

$$I_{outgoing} = I_{source}\rho\delta(\theta - \theta')\delta(\phi - \phi' + \pi) \tag{3.6}$$

3.4.2 Diffuse Surfaces

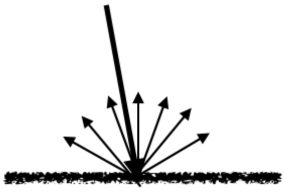


Figure 3.7: Light's reflection on the diffuse surface

When light hits diffuse surfaces, it scatters equally in all directions. To calculate the intensity of the outgoing radiance, we use Lambert's cosine law, which is expressed in Equation 3.7. In this Equation, $I_{outgoing}$ represents the outgoing radiance, I_{source} is the intensity of the incoming light source, ρ is the albedo of the material, and θ_i is the angle of incidence. The Equation shows that the outgoing radiance is proportional to the cosine of the incidence angle, the material's albedo, and the intensity of the incoming light source. As Figure 3.7 shows, incoming light is scattered evenly in all directions. These types of surfaces are easy to identify for Lidar systems since they scatter light in all directions.

$$I_{outgoing} = I_{source} \times \rho \times \frac{\cos\theta}{\pi} \tag{3.7}$$

3.4.3 Retroreflective Surfaces

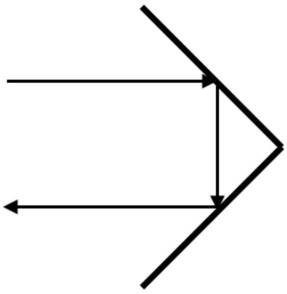


Figure 3.8: Light's reflection on the retroreflective surfaces

Retroreflective surfaces reflect light to their source. These surfaces, such as traffic signs, are not naturally occurring and are made mainly by humans. Retroreflective surfaces can be easily detected by Lidar systems since they reflect the light to their source. How light is reflected on the retroreflective surface can be seen in Figure 3.8.

3.5 Performance Metrics in Lidar

When designing an autonomous vehicle system that will use a Lidar, it is crucial to understand the performance metrics involved. These metrics play an essential role in the system design process. The range, range accuracy, and range precision are the most commonly used metrics for evaluating Lidar system performance. Tests for these metrics are performed on both retroreflective and diffuse surfaces. However, there are no standardized tests for evaluating Lidar performance on specular surfaces due to the high level of uncertainty involved, which varies based on the type of surface used.

The term "range" refers to the maximum distance at which Lidar can detect an object. The determined range of a Lidar system is influenced by the amount of sunlight and the material's reflective properties. For instance, for a low reflective surface range can be lower when there is sunlight in the environment [17].

Range accuracy represents how the measured distance is close to the ground truth value of the measurement.

Precision is defined as the standard deviation of the collected points when multiple measurements are made on the same target under the same conditions [18].

3.6 Signal To Noise Ratio

The signal-to-noise ratio (SNR) affects Lidar systems' performance. In the context of the Lidar sensor, the signal-to-noise ratio (SNR) is defined as the ratio of the received power and received noise, as stated in Equation 3.8. P_n represents the background noise, which is affected by various factors such as atmospheric scattering, photon shot noise, and sunlight [19].

$$SNR = \frac{P_r}{P_n} \tag{3.8}$$

The standard deviation in range measurements is correlated with the signal-to-noise ratio (SNR) of the signal. As the SNR increases, measurement error decreases, while a decrease in SNR results in an increase in measurement error.

3.7 Factors for Measurement Error

Sources of noise that affects SNR in Lidar systems are mainly thermal noise, background noise, and speckle noise, atmospheric effects.

Speckle noise: Speckle noises in the Lidar sensor can come from the scattering of the light on a rough hit surface. When a laser from Lidar hits a diffuse surface, it might backscatter in various directions. This backscattering effect can create speckles in the received data. [20]

Thermal noise: Photons radiate by the temperature. Since the detector's temperature will vary, it will create noise in the Lidar system. Thermal noise in the Lidar system is mostly assessed with the dark current in the detector. Dark current is the current that is produced when there is no light hitting on the detector. [20]

Atmospheric Effects: Another factor that can introduce measurement errors in Lidar data is atmospheric effects, which include atmospheric turbulence and attenuation [21]. Atmospheric turbulence is formed by irregularities in the transmission medium, such as temperature and wind. Furthermore, atmospheric attenuation means the scattering and absorption of the laser beam [22]. Atmospheric attenuation and turbulence can result in the non-linear transmission of the beam and produce deviations in the registered points.

Background Noise: When photons from sources other than the Lidar system hit the detector's surface, they create background noise. These photons do not carry information about the distance of the target hit. The main source of this noise is photons from sunlight. Therefore outdoor measurements can create noisy results in the Lidar system. [20]

Calibration Error: In addition to noises coming from the surrounding, wrongly set parameters or software of the Lidar system can create calibration problems which will cause the misalignment of the points in the point cloud.

3.8 Correlated Noise

As explained previously, some of the noise sources in Lidar can result in correlated noise within the system. This correlated noise can cause dependent errors with a pattern in range measurements because the affected points share the same correlated noise.

4

Methods

Based on the background part about how Lidar works, this section will explain the details of implementing Lidar sensor simulation. Later, the inclusion of errors in the results will also be described.

4.1 Ray Tracing for Lidar

In this thesis work, direct time of flight Lidar will be used, which measures the time spent until a sent signal is received. The type of laser that is sent is a pulsed laser. Since it is a direct TOF lidar, a ray tracing algorithm to find the intersections with the scene is the appropriate technique to simulate the behavior of Lidar.

DirectX Ray Tracing API inside the Unreal Engine will be used as an API for ray tracing. One benefit of using this API is that it enables the parallel production of rays in ray generation shaders and the parallel finding of intersections, thus reducing computation time.

The following sections will describe how the scene is represented in the ray tracing algorithm and how rays are generated for the Lidar sensor.

4.1.1 Acceleration Structure

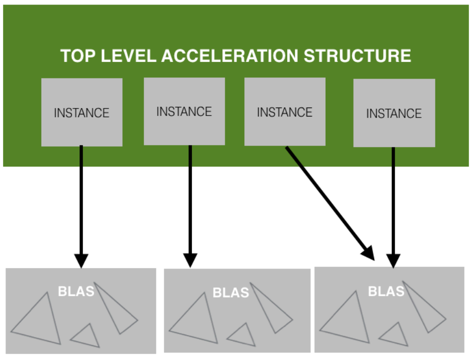


Figure 4.1: Top Level Acceleration Structure

For the efficiency of the ray tracing algorithm, acceleration structures are used to represent objects in the scene. As seen from Figure 4.1, acceleration structure can be divided into two, one is bottom-level acceleration structure (BLAS), and the other is top-level acceleration structure (TLAS). The bottom-level acceleration structure represents the triangles that form one object of TLAS, where TLAS contains multiple objects. In this thesis work, the BLAS structure consists of triangle meshes, and TLAS contains the BLAS. Acceleration structure is built to represent the scene in the GPU.

4.1.2 Ray Generation Shader

In the ray generation part, it is important to consider the properties of the type of Lidar system being used. The rays' direction and start point are important to simulate Lidar correctly. This section will describe how to generate rays for the given sensor model.

A ray in the context of ray tracing can be expressed with Equation 4.1. As the background section discusses, in Lidar, rays must be sent according to a vertical and horizontal resolution to achieve a vertical and a horizontal FOV.

$$\vec{P}(t) = \vec{O} + t\vec{D} \tag{4.1}$$

In this thesis project, the characteristics of the Ouster OS1-64 Lidar will be used to simulate the features of the Lidar. The Lidar has a vertical FOV of 45 degrees, ranging from -22.5 to 22.5 degrees. The number of channels vertically corresponds to the number of rays covering the 45-degree range. Also, horizontal FOV with 360-degree with 1024 channels will be used. However, those values can be modified according to the specific Lidar that will be used.

A spherical coordinate system will generate ray equations for the Lidar system. This system is a three-dimensional space, representing a point by its radial distance from the origin, azimuthal angle, and polar angle. Within the context of a Lidar system, it is possible to represent a single laser using its azimuthal angle, which corresponds to its vertical angle, and its polar angle, which corresponds to its horizontal angle. After determining the appropriate spherical coordinate values, they will be translated into Cartesian coordinates for use with the previous Ray Equation 4.1 where $\vec{O}$ is the Lidar position and $\vec{D}$ is the direction of the Lidar ray that will be sent. [23]

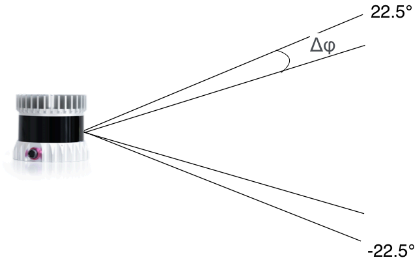


Figure 4.2: Lidars Angular Resolution

To determine the spherical coordinates of each laser emitted from the Lidar, the angular resolution of the Lidar must be known. The angular resolution is the angular separation between two laser rays in the context of the Lidar. In Figure 4.2, φ represents the angular resolution between two vertical laser rays. The Lidar, which will be implemented in this thesis will have an equal distribution of rays. Therefore, the vertical angular resolution $\Delta\varphi$ is given by Equation 4.2, where n_v is the number of vertical channels and θ is the vertical field of view in degrees. The same equation applies to the horizontal resolution, where α is the horizontal field of view and n_h is the number of horizontal lasers emitted.

$$\Delta\varphi = \frac{\theta}{n_v}, \Delta\gamma = \frac{\alpha}{n_h} \tag{4.2}$$

To calculate the spherical coordinates of a single laser with the ith vertical and jth horizontal channel, Equations 4.3 and 4.4 can be used. Equation 4.3 is used to find the azimuth angle at the ith vertical channel where θ_{start} is the starting angle of the vertical field of view, i is the id of the vertically dispatched ray. Similarly, the same process is applied with the horizontal angular resolution as seen from Equation 4.4 to find the azimuthal angle.

$$\theta_i = \theta_{start} + i\Delta\varphi \tag{4.3}$$

$$\alpha_j = \alpha_{start} + j\Delta\gamma \tag{4.4}$$

After the spherical coordinates of each laser are found, their conversion to cartesian coordinates is done with Equation 4.5.

$$x = r\sin\theta\cos\alpha \ , y = r\sin\theta\sin\alpha \ , z = r\cos\theta \tag{4.5}$$

where r is the radial distance, θ is the polar angle and α is the azimuthal angle. The ray description is set inside the ray generation shader of DirectX based on the obtained cartesian coordinates. Then with the TraceRay command closest hit shader is called to modify the payload, which stores the intersection points.

4.1.3 Intensity Modelling

Intensity modeling is crucial in Lidar simulation environments as it is used for segmentation, classification, and visualization purposes. The received energy of the Lidar sensor is determined by Equation 4.6 [24], where R is the distance, θ is the incidence angle, ρ is the reflectance, α is the attenuation coefficient, E_t is transmitted energy and C is the constant related to the receiver and transmitter of the sensor.

$$Er = E_t C e^{-2\alpha R} cos(\theta)\rho \tag{4.6}$$

A simpler model is derived from Equation 4.6 for the designed simulator. The simplified equation, shown in Equation 4.7, replaces $E_t * C$ with a reference reflectivity of I_0.

$$I = I_0 * e^{-2\alpha R} cos(\theta)\rho \tag{4.7}$$

4.1.4 Range Modelling

One of the important metrics in a Lidar sensor is its range. The range of a Lidar depends on the distance of the hit target, the reflectance of the material, and the amount of background noise in the environment. The proposed range method uses a simple quadratic equation model given with Equation 4.8 where $R\%$ is reflectance, and r is the range. Also, this model varies depending on the Lidar specifications that have been used [25]. Range and reflectivity relation for Ouster OS1-64 lidar with a sunlight 100klux can be with Equation 4.9. Figure 4.3 shows the relation between range and reflectivity.

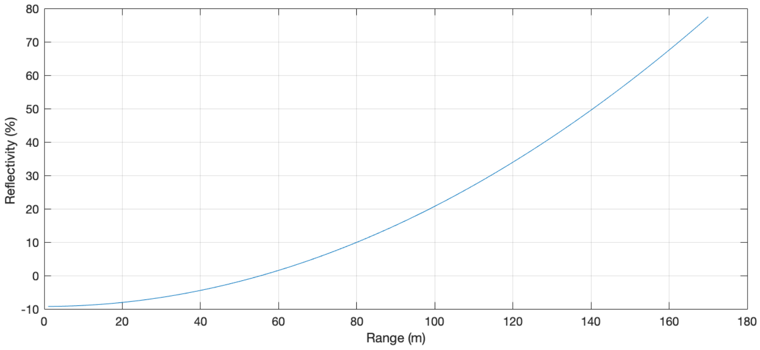


Figure 4.3: Reflectivity and Range [25]

$$R\% = a + b * r^2 \tag{4.8}$$

$$R\% = -9.2500 + 0.003 * r^2 \tag{4.9}$$

4.2 Measurement Error Modelling

In Lidar range measurements, deviations from the real range value can occur, which are called measurement errors. To develop a measurement error model, a test was conducted to observe the nature of these errors. A standard A4 paper was placed in front of the Lidar at a distance of approximately 78 cm, and the results were recorded. The result of one horizontal channel was saved and displayed in Figure 4.4.

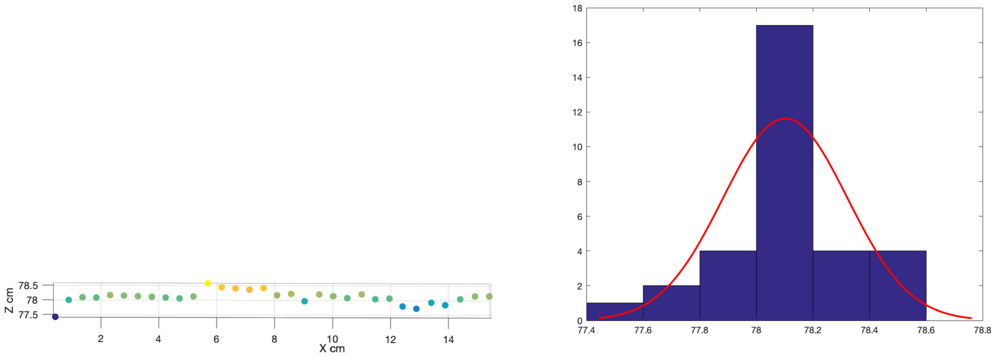


Figure 4.4: Depth values of the Lidar from one vertical Channel

It is essential to analyze the errors to understand the nature and distribution of the precision errors and determine an appropriate precision error model. A histogram is used to visualize the depth values and fitted to a Gaussian distribution, as seen on the left side of Figure 4.4. The results from the experiment show that the range values deviate from the mean with a certain standard deviation, which can be modeled with a Gaussian distribution $N(\mu, \sigma^2)$ [26].

In order to determine appropriate values for σ and μ, it is important to consider the factors that affect them. As mentioned in the background section, a high signal-to-noise ratio provides better precision. The signal-to-noise ratio depends on the target reflectivity, the amount of sunlight in the background, and the distance of the target object. To develop a model that considers the distance and reflectivity of the target, reference data from the datasheet of Ouster OS1-64 will be used.

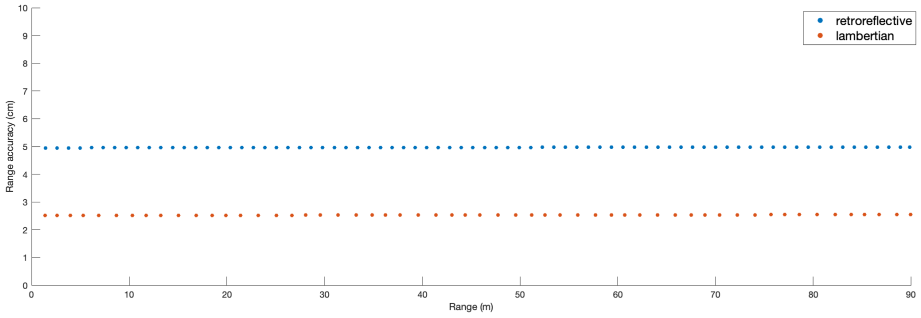


Figure 4.5: Ouster OS1-64 Lidar Range Accuracy Data depending on datasheet [27]

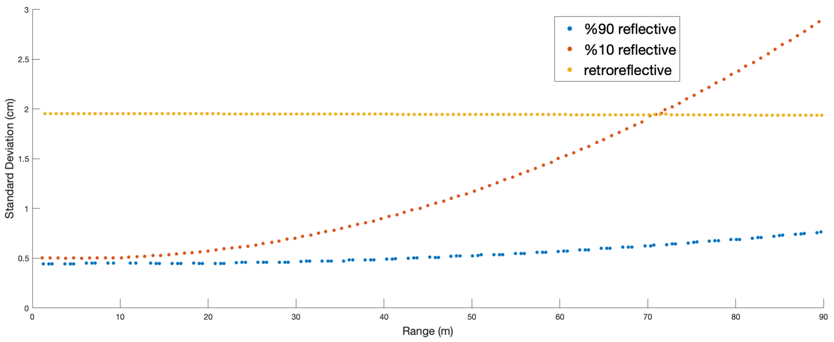


Figure 4.6: Ouster OS1-64 Lidar Range Precision Data depending on datasheet [27]

The reference data used to develop the precision error model of the Lidar is presented in Figure 4.5. As shown in Figures 4.6 and 4.5, it can be observed that as the range increases and reflectivity decreases, the standard deviation of the precision error rises, while the mean of the error remains constant. This data from the datasheet provides the standard deviation values for range error for Lambertian reflective targets at %10 and %90 reflectivity levels. However, the reflectance values between %10 - %90 and ranges greater than 90 meters are not provided. The exponential growth of the data will be used to obtain values for other reflectance levels.

The standard deviation of the precision error is assumed to grow exponentially with reflectivity and target distance. Raw data from the datasheet is fitted with polynomials which are written with Equation 4.11 $P_{10}(TD)$ and Equation 4.10 $P_{90}(TD)$ where TD is the target distance. Polynomials are also plotted in Figure 4.7.

$$P_{\%90}(TD) = 0.00004x^2 - 0.001x + 0.508 \tag{4.10}$$

$$P_{\%10}(TD) = 0.0003x^2 - 0.0031x + 0.5558 \tag{4.11}$$

$$P_{\%R}(TD) = P_{\%90}(TD)b(TD)^{90-\%R} \tag{4.12}$$

Then in order to describe the standard deviation of the error in terms of the logarithmic equation. It will be good to first look at the structure of the exponential equation. The exponential equation is 4.13 where b is the growth rate, a is the initial value, x is the independent variable, and y is the dependent variable. For this thesis, the standard deviation of the range with respect to reflectivity can be expressed as an exponential equation given with Equation 4.14 where TD is the target distance, b(TD) is the exponential growth rate at the target distance and %R is reflectivity and $P_{90}(TD)$ is the Equation 4.10. In Figure 4.8, it is possible to see Equation 4.14 plotted when $TD = 90$. Then for different distances and reflectivities, $\sigma_{x\%R}(TD)$ is plotted in Figure 4.9.

$$y = ae^{bx} \tag{4.13}$$

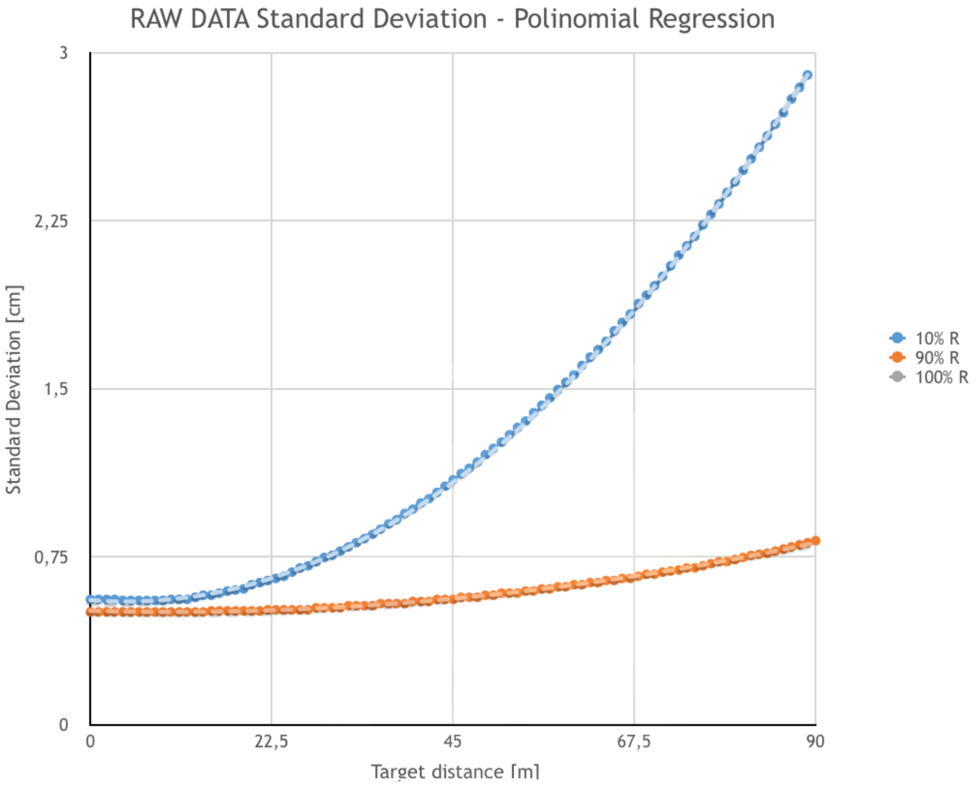


Figure 4.7: Polynomials plotted for standard deviation for %10 and %90 reflectivity

$$\sigma_{x\%R}(TD) = P_{90}(TD) * e^{b(TD)*(90-\%R)} \tag{4.14}$$

$$b(TD) = \frac{ln(\frac{P_{10}(TD)}{P_{90}(TD)})}{(90-10)} \tag{4.15}$$

After developing the model for the standard deviation and mean of the error in the Lidar system, the appropriate model for visualizing the error should be selected. As shown in Figure 4.4, each depth measurement point is correlated with the others, resulting in a wavy pattern. This type of error is commonly known as correlated error and can be observed in Lidar systems when the error has a common noise source that is also correlated. Because the error is correlated and has a repeating nature, it can be represented using a fractal function model.

4.3 Weierstrass Fractal Function

Fractals are a good choice to represent the self-similarity nature of the patterns [28]. Due to this, a fractal function can be used to simulate the repeating and self-similar pattern of the measurement error. In order to achieve this, Weierstrass fractal function and the motivation to choose it will be discussed.

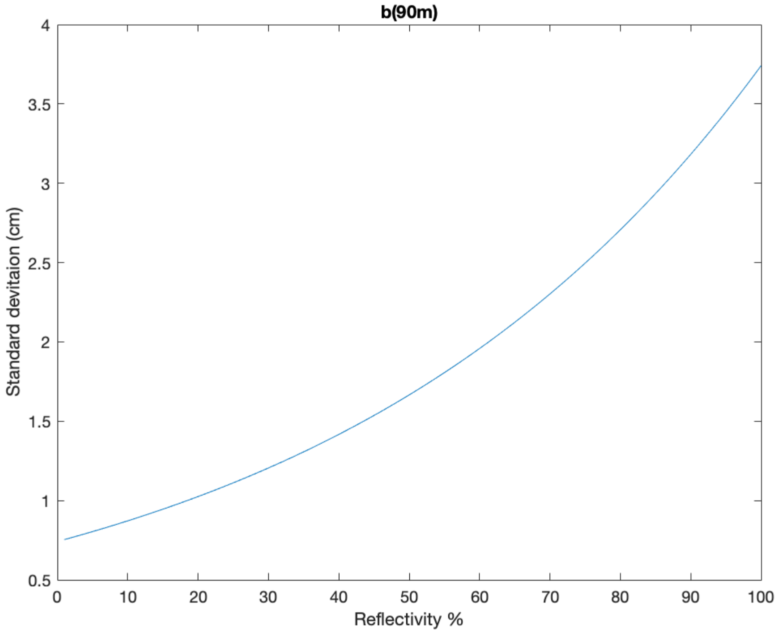


Figure 4.8: Standard deviation of the measurement error at 90 meters with respect to reflectivity

The Weierstrass fractal function is commonly used for modeling rough surfaces because its fractal dimension is a suitable surface roughness parameter [29]. It was also previously used for sea surface modeling. This fractal function is a good choice for sea surface modeling because sea waves can be considered a linear combination of sinusoidal waves, making it an appropriate model [30]. This work done for sea surface modeling was also the inspiration for the reason for choosing the Weierstrass fractal function to model the wavy behavior of the measured distance since it can also be thought of as a linear combination of multiple sinusoidal waves. Weierstrass function can be expressed with Equation 4.16 where $\gamma > 1$, $0 < H < 1$. This fractal function is continuous but nowhere differentiable, and its fractal dimension equals D = 2 - H [31].

Different parameters for the γ values for the fractal function are displayed in Figure 4.10. When γ equals 1, it is like a cosine wave, while as it increases, it is possible to see its fractal characteristics. As a design choice in this thesis, H is chosen as 0.99 while γ is chosen as 5.

$$W_\alpha(x) = \sum_{n=0}^{\infty} \gamma^{-nH} \cos(2\pi\gamma^n x) \tag{4.16}$$

Then in order to combine this model with the previous model developed for precision and accuracy of the range error, Equation 4.17 is used, which enables to shift and scale of the data so that it will have a specific standard deviation and mean where s is the standard deviation of the data, s_n is the desired standard deviation, m is the mean of the data, m_n is the desired mean.

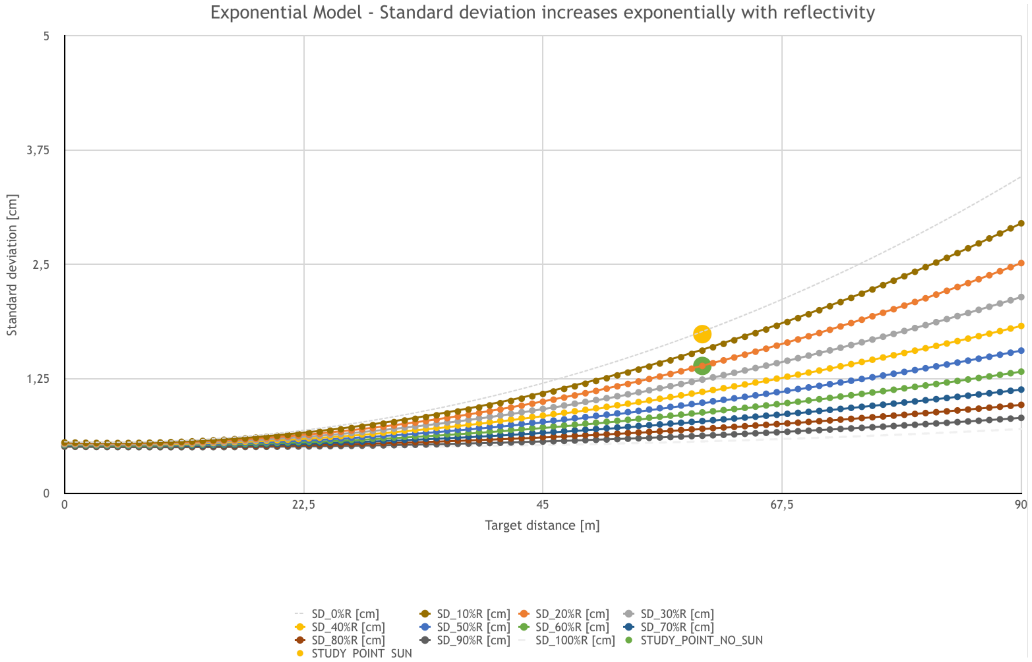
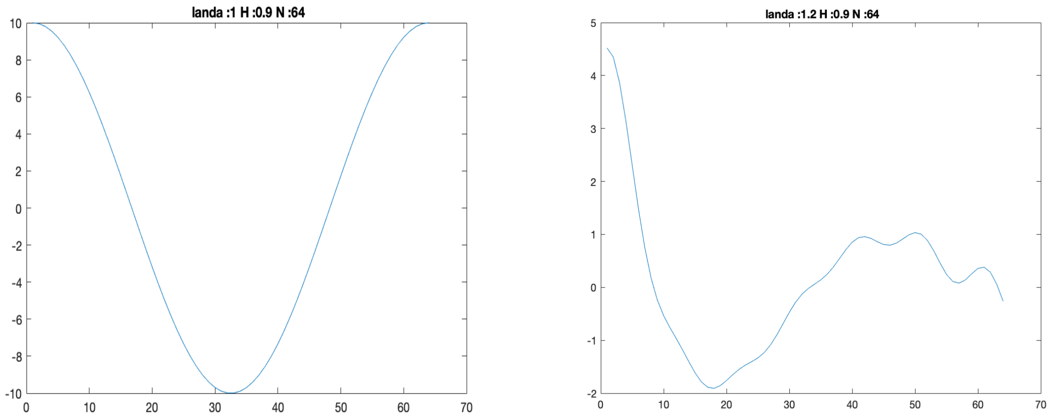


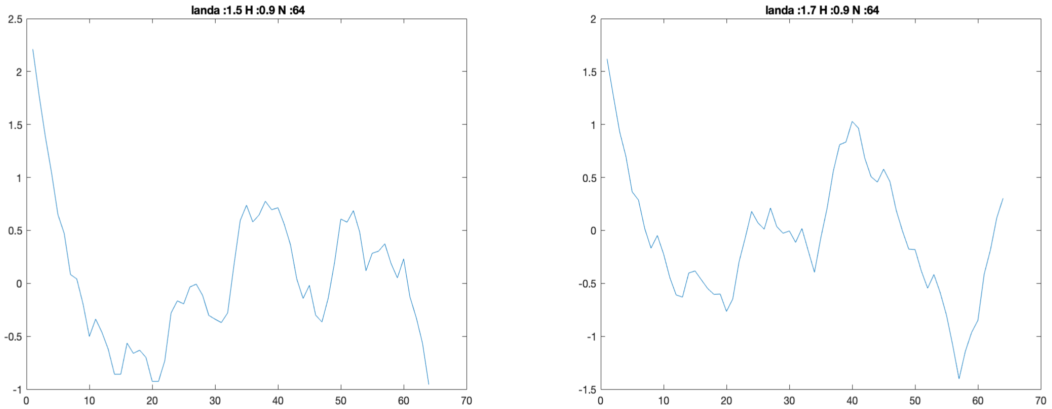
Figure 4.9: Standard deviation of the measurement error based on the different distances and reflectivities

$$w_n = m_n + (w_n - m) \times \frac{s_n}{s} \tag{4.17}$$



(a) H = 0.90, $\gamma = 1$

(b) H = 0.90, $\gamma = 1.2$



(c) H = 0.90, $\gamma = 1.5$

(d) H = 0.90, $\gamma = 1.7$

Figure 4.10: Weierstrass function with different H, γ value

5

Results

The results section provides information on the experimental setup, experiments, and simulation results. The experimental setup description includes details of the specific Lidar used, such as its technical specifications like wavelength, scanning mechanisms, and performance metrics. Furthermore, the output values obtained from the Lidar will be discussed, such as distance, reflectivity, points, ambient lighting, and signal strength. Moreover, how those attributes will be used in this thesis will be discussed. Also, the appropriate target material selection and reasons behind selecting that material as test material will be addressed. Simulation results will be produced alongside experimental results. Then the simulated output will be compared to understand how the simulator works.

5.1 Experimental Set up

We used the Ouster OS1-64 Lidar model as a sensor to test our model. This model has a 45-degree vertical field of view with 32 vertical channels, a 360-degree horizontal field of view with 1024 channels, 865 nanometers wavelength, which is in the NIR spectrum and can rotate at a frequency of 10Hz.

In the selection of target material, it would be helpful to remember the information provided in Section 3.4, which is about types of surfaces. Surfaces can be categorized as diffuse, specular, and retroreflective. Most real-life materials can have both specular and diffuse surface properties. However, using a material with both properties for testing purposes may not be ideal as it would require a detailed analysis of the object's properties and light reflection direction. Moreover, Lidar sensors have no standardized testing method for specular surfaces.

Therefore, standard A4 office paper has been chosen as a target material since it has diffuse surface properties. In terms of reflectivity values for a highly reflective surface, white paper has been selected, and black A4 paper has been selected for low reflectance. A4 papers are placed in a frame. Next, those frames are positioned at varying distances in front of the Lidar with a 90-degree angle, which equals the incidence angle zero. If we remember the Equation 3.7 reflectivity of the material equals the albedo of the material multiplied by the cosine of the incidence angle. For the reflectivity values, we will only refer to reflectance values where the incidence angle equals zero for the papers. The white paper's reflectance value in NIR equals

%75.9 [32]. If we assume this value is true, the black paper's albedo value will be derived through experiments.

The captured result is a collection of data points in a point cloud. Each data point includes information given in Table 5.1. The Signal Photons field indicates the number of photons reflected from the target object, while Ambient Photons come from other sources such as sunlight. Reflectivity value is a calibrated measure based on the sensitivity and distance of the material at that range, and the timestamp is the time when data is registered [33]. Then range is the distance of the target material, and x,y,z is the position in the coordinate system. Also, the channel represents the vertical channel id of the registered data point. Since deviations in the x and y coordinates are in the mm range regarding measurement error, z coordinates will be considered. Also, in this thesis, since the main interest is the error in the distance, reflectance values obtained from Lidar will be only used to compare reflectance between white and black paper.

Table 5.1: Fields of the registered data point

Signal Photons	Reflectivity	Channel	Ambiant Photons	Range	Raw Timestamp	Point ID	x	y	z

5.2 Results from Experiments and Simulation

First, for the 80 cm, to derive the reflectance of the black paper as a percentage, we will compare reflectivity values obtained for white and black paper in Lidar with an 80 cm distance. The mean of the reflectivity values obtained from the Lidar sensor for the white paper is 121 and 28 for the black paper. As mentioned in the experimental setup 5.1, the white paper has %75.9 reflectivity in NIR. Assuming that the %75.9 reflective material has a reflectivity value of 121, we can estimate that black paper has %17 reflectivity in NIR. For this simulation, we will round reflectivity values to %80 for white paper and %20 for black paper.

To measure the standard deviation in Lidar's measurements, white and black paper were placed at distances of 80 centimeters, 4, 8, and 13 meters. To avoid measurement errors caused by the slope of the paper, only one horizontal line of points was selected. In Table 5.2, a comparison of the standard deviation for the z coordinate of the horizontal line of points from the real-life experiments and the Model that is described in Section 4.2 can be seen.

Table 5.2: Experiment Results with Different A4 Paper Colors at Different Distances

Paper Color	Distance	Standart Deviation Experiment	Reference Data
White	4m	0.5447	0.5090
Black	4m	0.5492	0.5442
White	8m	0.7122	0.5100
Black	8m	0.7036	0.5457
White	13m	6.0653	0.5114
Black	13m	7.3356	0.5623

The following figures illustrate the captured points and their fluctuating nature based on experimental results. When the Lidar is close to the target material, the number of registered points is higher, resulting in more visible fluctuations. Figure 5.3 presents the target object's experimental and simulated results. The fluctuations in Figure 5.2 were created using the Weierstrass fractal function. In Figure 5.1, the wavy pattern appears smoother than the simulated output, which is sharper and has a slightly higher frequency.

Figure 5.6 displays the experimental and simulated results of 80 cm away from the white paper from a side view. This perspective allows for easier interpretation of deviations in the z coordinate.

In Figure 5.9, the fluctuations in the data points for both simulated and experimental results are visible. When the paper is 4 meters away, the simulated results look closer to the experimental data than when it is only 80 centimeters away.

Finally, Figure 5.12 shows the recorded points for the paper. However, it is noticeable that the distance between the points increases as the target material moves away from the Lidar. This occurs because the number of points recorded for the paper decreases as it moves away from the sensor. Therefore, no testing has been done beyond eight meters.

Since no experiment can be done further than 8 m, a 400 cm x 400 cm object was created in the simulation environment with %20 reflectances and put into 40-meter and 20-meter distances. In the simulation environment for both distances, Gaussian noise is added with 0.5-centimeter standard deviation, noise is added with the developed model, and no noise is added. Results can be seen in Figure 5.14 and in Figure 5.15. When showing the results, the x, and y-axes are in meters while the z-axis is in centimeters. The graph is centered at zero to make fluctuations more visible.

Lastly, the Signal to Noise ratio is the ratio of signal photons to ambient photons. Signal to Noise ratio and z values were normalized and added an offset to understand their correlation and plotted in Figure 5.13.

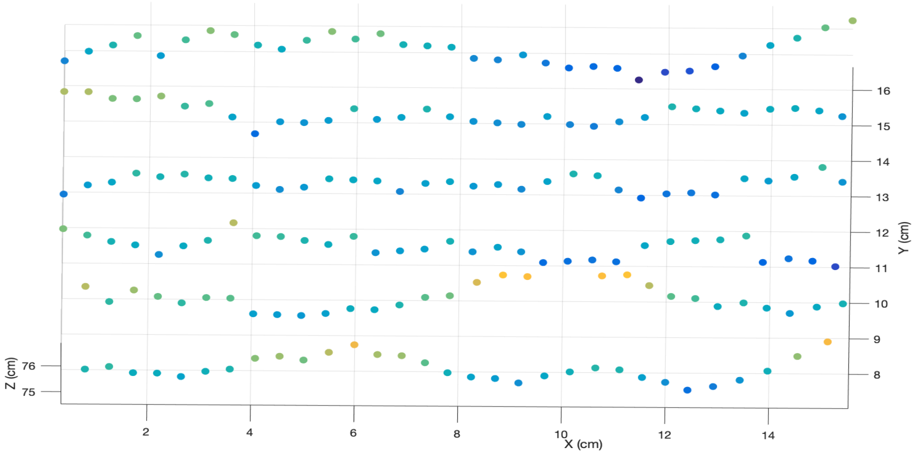


Figure 5.1: Experimental results

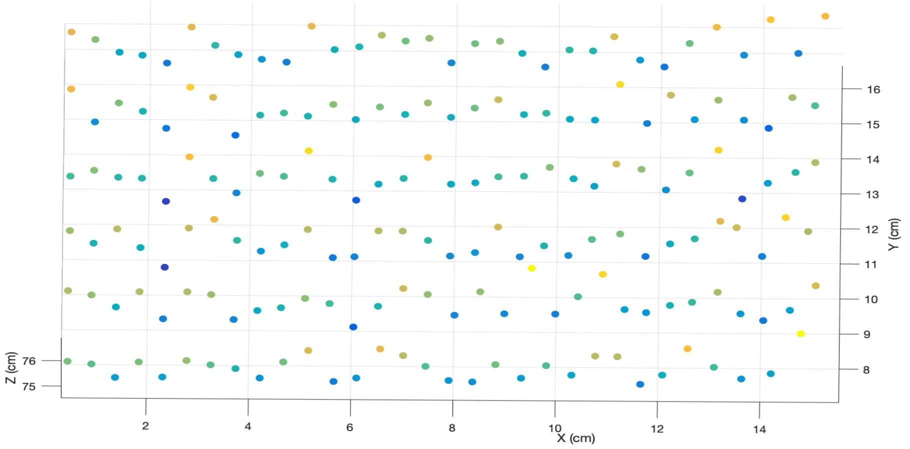


Figure 5.2: Simulated results

Figure 5.3: Results of white paper at a distance of 80cm from the Lidar

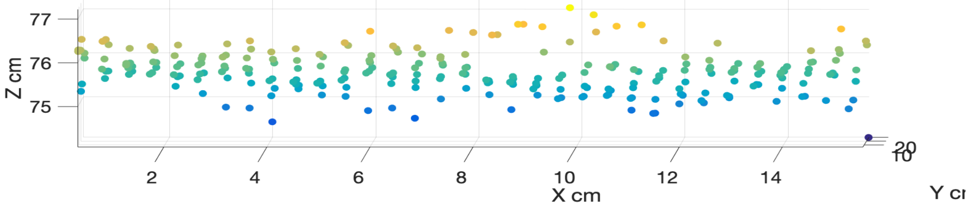


Figure 5.4: Experimental Results

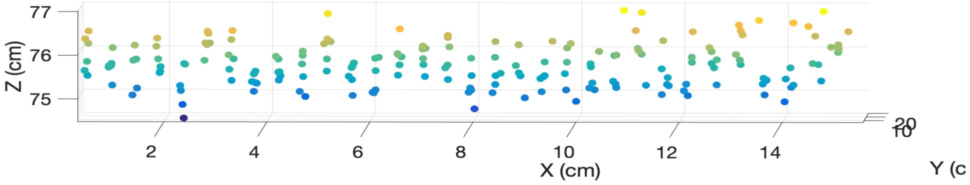


Figure 5.5: Simulated Results

Figure 5.6: Side view of results of white paper at a distance of 80 cm from the Lidar

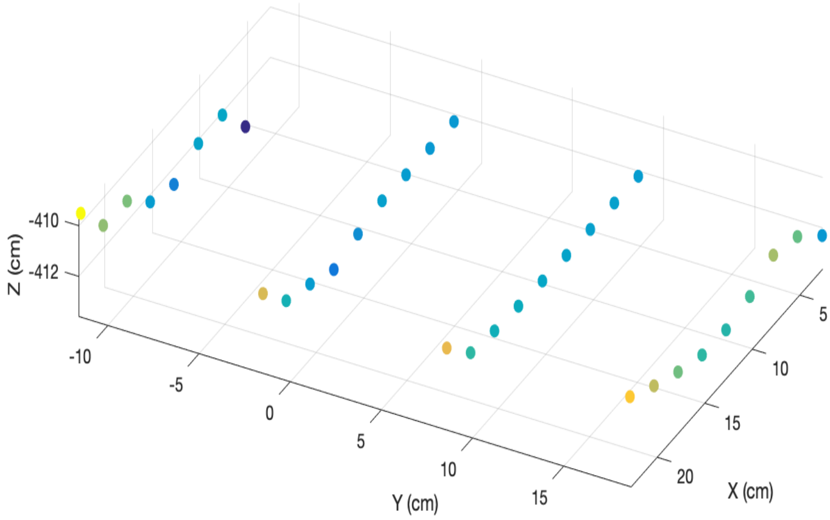


Figure 5.7: Experimental Results

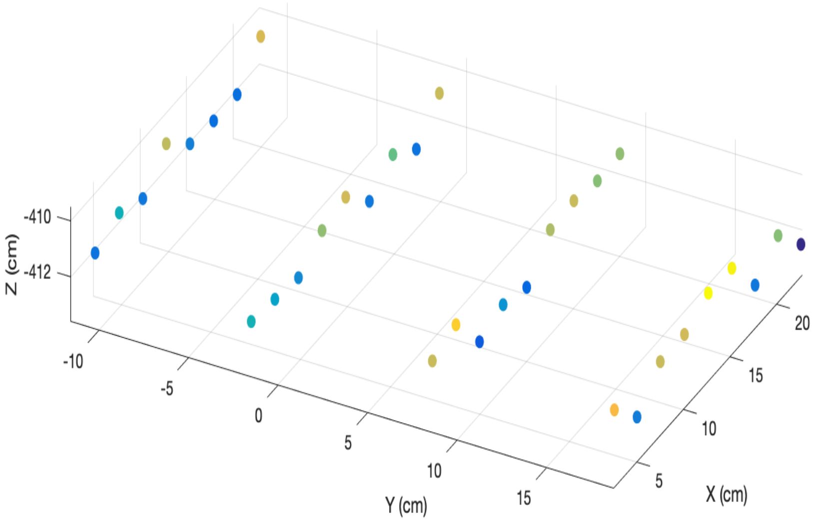


Figure 5.8: Simulated Results

Figure 5.9: Results of white paper at a distance of 4 m from the Lidar

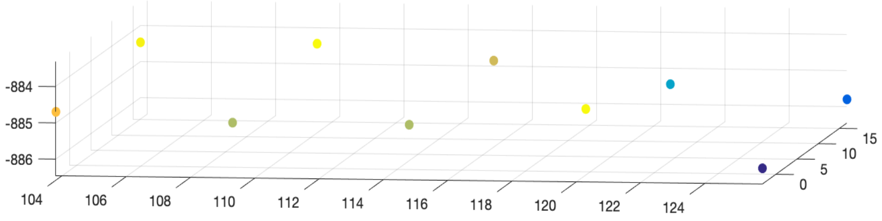


Figure 5.10: Simulated Results

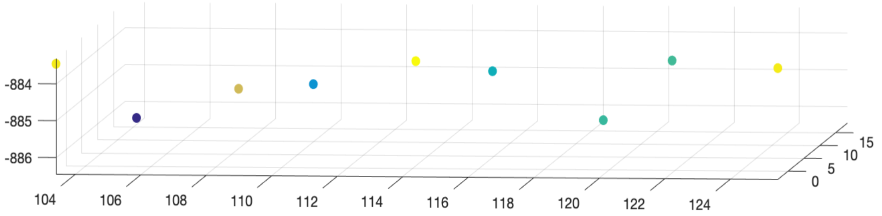


Figure 5.11: Experimental Results

Figure 5.12: Results of white paper at a distance of 8 m from the Lidar

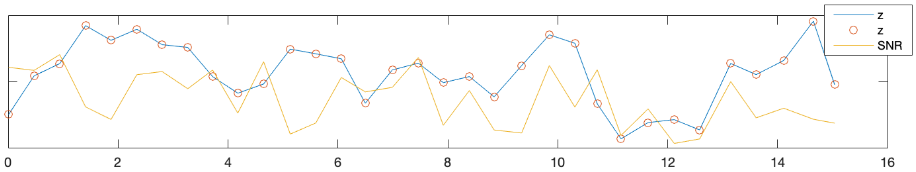
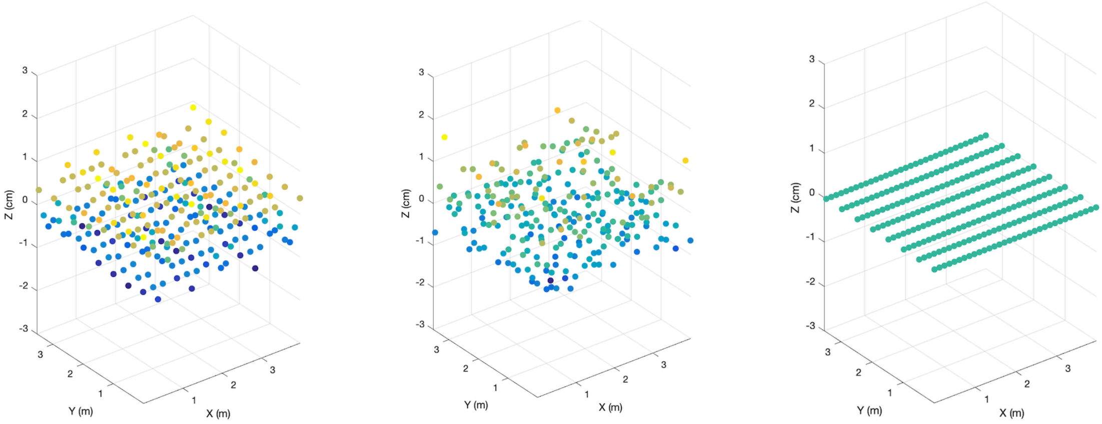


Figure 5.13: SNR and z values comparison

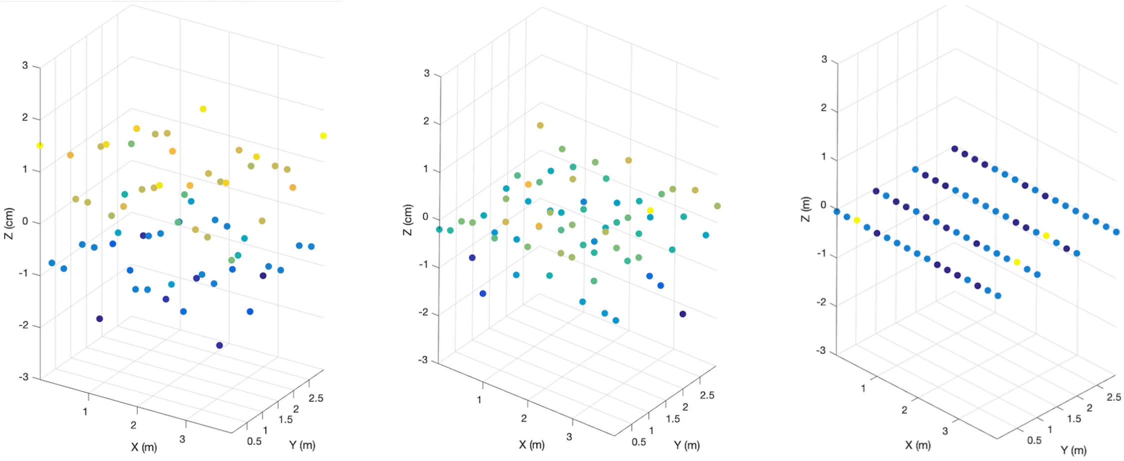


(a) Simulation with developed model

(b) Gaussian random noise with 0.5 cm standard deviation

(c) Without any error model

Figure 5.14: Simulation of %20 reflectance object at 20 m distance



(a) Simulation with developed model

(b) Gaussian random noise with 0.5 cm standard deviation

(c) Without any error model

Figure 5.15: Simulation of %20 reflectance object at 40 m distance

6

Discussion

This chapter will discuss how the experimental setup can be improved, the reasons for differences between simulation and experimental results, difficulties in the implementation, and valuable future work that can be done.

6.1 Comments about Experimental Setup and Results

The Weierstrass fractal function worked well in simulating the repetitive pattern of the experimental output. However, it is still possible to see that experimental output has smoother characteristics than the fractal function. In the comparison of the simulated and real results, it was mainly possible to compare them with their visual properties and their standard deviation in the data.

Furthermore, to compare with previous works, one common method used in the measurement error simulation is adding Gaussian noise with a 0.5-centimeter standard deviation. However, this constant number addition does not consider the material's reflectivity and distance. In Figure 5.14, the result with constant Gaussian noise, the result with the newly developed method and without any error model, is given at 20 meters, and in Figure 5.15, the simulation results are given at 40 meters. As seen in the Figures, results with constant Gaussian noise cannot include increasing error as the distance of the object increases. Also, the wavy pattern of the fractal seems less obvious as distance and standard deviation increase.

Table 5.2, shows that the experiment's standard deviation values are closer to obtained values from the model when the target material is closer to the sensor. However, as the target material moves further away, the standard deviation in the measurement increases significantly more than the standard deviation of the simulation data. This could be due to various factors, such as an unintended slope of the paper, changes in sunlight or temperature, or issues with sensor calibration. This difference can be because the model was obtained from reference data, and we do not exactly know the condition for the obtained data.

6.2 Investigation for the reasons of the Error

The background section 3.7 discusses various factors that may affect the precision of measurements. To determine if the ambient lightning is the cause of the error, Figure 5.13 compares the ambient photons divided by signal photons value with fluctuations in error. However, the error pattern does not precisely match that of Figure 5.13. While ambient lighting may contribute to these fluctuations, it cannot be concluded that it is the primary reason for the precision error. Other factors presented in the background section may also contribute to measurement errors. However, identifying the main cause of these errors remains challenging as each reason must be analyzed separately.

6.3 Working with DirectX in Unreal Engine

The DirectX API within the Unreal Engine is commonly used for image rendering technologies, such as path tracing. It is best suited to use within the engine source code since the ray tracing scene is created within it. However, the Lidar implementation was not implemented within the source code, causing difficulty in accessing the ray tracing scene of the Unreal Engine. As a result, the project was limited because of including only a few objects in the scene.

A good solution could be integrating the sensor model into the source code where most ray tracing operations occur. This would ensure that the sensor is always reaching the ray-tracing scene. However, this solution can still be challenging because the Lidar sensor requires constant communication with the other operations in the simulation, such as communicating between other sensors. It can be beneficial to look into different kinds of engines where ray tracing development is more manageable from the user's perspective.

6.4 Future Work

The reflectivity values and range of the implemented Lidar sensor model have not been tested through experiments. As part of future work, the reflectivity values obtained from experiments can be compared to the proposed method described in the methods section for intensity estimation.

Another future work involves determining the range performance of the sensor, which refers to the maximum distance it can detect an object. This metric is influenced by the ambient lighting in the environment and the material's reflectivity. Additional experiments can be carried out to compare the proposed method with the experimental results.

Furthermore, to improve the experimental setup, it is good to use target materials with precisely known Lambertian reflectances and a more professional setup. Also, the influence of outdoor sunlight has not been tested, and in the case of a better testing environment, it would be good to see how the results are changing.

6.5 Ethical Consideration

In this thesis, since it does not involve participants in the data collection, we believe that the rights of participants are not violated. In autonomous vehicle development, simulation environments are essential in validation and verification. The development of simulation environments should be done in parallel with real-life comparisons to ensure the safety of the developed technologies. This thesis also attempted to ensure the parallel development of real-life and simulation Lidar models. In the design and software of autonomous vehicles, it is beneficial to consider all kinds of scenarios that can happen in real life and try to be prepared for those.

7

Conclusion

To find an answer to the research question of "How can measurement errors of the Lidar sensor be added into the simulation environment?" we first examine the nature of these errors. Then we investigated how the reflectivity of the target material and distance affect these errors. A model is created to determine the standard deviation of the error, taking into account the distance and target reflectivity. This model is combined with the Weierstrass fractal function to represent the repetitive and wavy nature of the measured distance. The simulated results are compared with real-world testing, and it is observed that the Weierstrass function effectively models the wavy pattern. The model developed for the standard deviation of the error based on distance and target reflectivity starts to differentiate from experimental results as the distance between the target and sensor increases. In summary, the Weierstrass fractal function is a suitable option for simulating the wavy pattern of errors. The model developed for the standard deviation of the error based on the target's reflectivity and distance works better at closer distances; however, for further distances, more experimentation is required as a future work.

Bibliography

[1] M. Segata, R. L. Cigno, R. K. Bhadani, M. Bunting, and J. Sprinkle, “A lidar error model for cooperative driving simulations,” in *2018 IEEE Vehicular Networking Conference (VNC)*, 2018, pp. 1–8. DOI: `10.1109/VNC.2018.8628408`.

[2] M. Hahner, C. Sakaridis, M. Bijelic, *et al.*, *Lidar snowfall simulation for robust 3d object detection*, 2022. arXiv: `2203.15118 [cs.CV]`.

[3] S. Teufel, G. Volk, A. Von Bernuth, and O. Bringmann, “Simulating realistic rain, snow, and fog variations for comprehensive performance characterization of lidar perception,” in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–7. DOI: `10.1109/VTC2022-Spring54318.2022.9860868`.

[4] M. Hahner, C. Sakaridis, D. Dai, and L. V. Gool, *Fog simulation on real lidar point clouds for 3d object detection in adverse weather*, 2021. arXiv: `2108.05249 [cs.CV]`.

[5] P. Vacek, O. Jaek, K. Zimmermann, and T. Svoboda, “Learning to predict lidar intensities,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3556–3564, 2022. DOI: `10.1109/TITS.2020.3037980`.

[6] S. Manivasagam, S. Wang, K. Wong, *et al.*, *Lidarsim: Realistic lidar simulation by leveraging the real world*, 2020. arXiv: `2006.09348 [cs.CV]`.

[7] . Sobczak, K. Filus, A. Domaski, and J. Domaska, “Lidar point cloud generation for slam algorithm evaluation,” *Sensors*, vol. 21, no. 10, p. 3313, May 2021. DOI: `10.3390/s21103313`.

[8] J. Fang, D. Zhou, F. Yan, *et al.*, “Augmented lidar simulator for autonomous driving,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1931–1938, 2020. DOI: `10.1109/LRA.2020.2969927`.

[9] W. Jansen, N. Huebel, and J. Steckel, “Physical lidar simulation in real-time engine,” in *2022 IEEE Sensors*, 2022, pp. 1–4. DOI: `10.1109/SENSORS52175.2022.9967197`.

[10] B. Browning, J.-E. Deschaud, D. Prasser, and P. Rander, “3d mapping for high-fidelity unmanned ground vehicle lidar simulation,” *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1349–1376, 2012. DOI: `10.1177/0278364912460288`.

[11] B. Schmitz, C. Holst, T. Medic, D. D. Lichti, and H. Kuhlmann, “How to efficiently determine the range precision of 3d terrestrial laser scanners,” *Sensors*, vol. 19, no. 6, 2019, ISSN: 1424-8220. DOI: `10.3390/s19061466`. [Online]. Available: `https://www.mdpi.com/1424-8220/19/6/1466`.

[12] M. Adams, "Lidar design, use, and calibration concepts for correct environmental detection," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 753–761, 2000. DOI: `10.1109/70.897786`.

[13] 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*, vol. 37, no. 4, pp. 50–61, 2020. DOI: `10.1109/MSP.2020.2973615`.

[14] Y. Yang, S. Ding, B. Plovie, W. Li, and C. Shang, "Soft and stretchable electronics design," in *Encyclopedia of Sensors and Biosensors (First Edition)*, R. Narayan, Ed., First Edition, Oxford: Elsevier, 2023, pp. 258–286, ISBN: 978-0-12-822549-3. DOI: `https://doi.org/10.1016/B978-0-12-822548-6.00087-X`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/B978012822548600087X`.

[15] F. Villa, F. Severini, F. Madonini, and F. Zappa, "Spads and sipms arrays for long-range high-speed light detection and ranging (lidar)," *Sensors*, vol. 21, no. 11, 2021, ISSN: 1424-8220. DOI: `10.3390/s21113839`. [Online]. Available: `https://www.mdpi.com/1424-8220/21/11/3839`.

[16] Y. Wu, *Eecs 432-advanced computer vision notes series 3: Radiometry, brdf and photometric stereo*, Notes, Evanston, IL 60208, 2021.

[17] W. Sun, Y. Hu, D. G. MacDonnell, C. Weimer, and R. R. Baize, "Technique to separate lidar signal and sunlight," *Opt. Express*, vol. 24, no. 12, pp. 12 949–12 954, Jun. 2016. DOI: `10.1364/OE.24.012949`. [Online]. Available: `https://opg.optica.org/oe/abstract.cfm?URI=oe-24-12-12949`.

[18] B. Behroozpour, P. A. M. Sandborn, M. C. Wu, and B. E. Boser, "Lidar system architectures and circuits," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 135–142, 2017. DOI: `10.1109/MCOM.2017.1700030`.

[19] Y. Jiang, J. Zhu, C. Jiang, T. Xie, R. Liu, and Y. Wang, "Adaptive suppression method of lidar background noise based on threshold detection," *Applied Sciences*, vol. 13, no. 6, 2023, ISSN: 2076-3417. DOI: `10.3390/app13063772`. [Online]. Available: `https://www.mdpi.com/2076-3417/13/6/3772`.

[20] R. D. Richmond and S. C. Cain, "Direct-detection ladar systems," *Defense and Security Symposium*, vol. 2010, pp. 1–18, 2010. DOI: `10.1117/3.836466`. [Online]. Available: `https://doi.org/10.1117/3.836466`.

[21] F. Xu, J. Wang, D. Zhu, and Q. Tu, "Speckle noise reduction technique for lidar echo signal based on self-adaptive pulse-matching independent component analysis," *Optics and Lasers in Engineering*, vol. 103, pp. 92–99, 2018, ISSN: 0143-8166. DOI: `https://doi.org/10.1016/j.optlaseng.2017.12.002`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0143816617306565`.

[22] B. Hoffschmidt, S. Alexopoulos, J. Göttsche, M. Sauerborn, and O. Kaufhold, "3.06 - high concentration solar collectors," in *Comprehensive Renewable Energy*, A. Sayigh, Ed., Oxford: Elsevier, 2012, pp. 165–209, ISBN: 978-0-08-087873-7. DOI: `https://doi.org/10.1016/B978-0-08-087872-0.00306-1`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/B9780080878720003061`.

[23] G. F. Gusmão, C. R. Barbosa, and A. B. Raposo, “Development and validation of lidar sensor simulators based on parallel raycasting,” *Sensors*, vol. 20, no. 24, p. 7186, Dec. 2020. DOI: `10.3390/s20247186`. [Online]. Available: `https://doi.org/10.3390/s20247186`.

[24] B. Jutzi and H. Gross, “Normalization of lidar intensity data based on range and surface incidence angle,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 38, pp. 213–218, 2009.

[25] S. Muckenhuber, H. Holzer, and Z. Bockaj, “Automotive lidar modelling approach based on material properties and lidar capabilities,” *Sensors*, vol. 20, no. 11, p. 3309, 2020. DOI: `10.3390/s20113309`.

[26] K. Tanaka and G. Ishigami, “Modeling of lidar measurement uncertainty for rover path planning,” *Department of Mechanical Engineering, Keio University, 3-14-1Hiyoshi, Yokohama, 223-8522, Japan*, 2014.

[27] *Ouster OS1 Datasheet, Firmware Version 3.0, Hardware Version Rev7*, `https://data.ouster.io/downloads/datasheets/datasheet-rev7-v3p0-os1.pdf`, Accessed: May 7, 2023.

[28] M. F. Barnsley, R. L. Devaney, B. B. Mandelbrot, H.-O. Peitgen, D. Saupe, and R. F. Voss, *The Science of Fractal Images* (Springer Book Archive), 1st ed., H.-O. Peitgen and D. Saupe, Eds. Springer New York, NY, 1988, ISBN: 978-1-4612-8349-2. DOI: `10.1007/978-1-4612-3784-6`. [Online]. Available: `https://doi.org/10.1007/978-1-4612-3784-6`.

[29] L. Guo and Z. Wu, “Fractal characteristics investigation on light scattering from two dimensional rough surface,” *International Journal of Infrared and Millimeter Waves*, vol. 21, no. 12, pp. 2099–2106, 2000. DOI: `10.1023/A:1006748923881`. [Online]. Available: `https://doi.org/10.1023/A:1006748923881`.

[30] F. Berizzi and E. Mese, “Scattering coefficient evaluation from a two-dimensional sea fractal surface,” *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 4, pp. 426–434, 2002. DOI: `10.1109/TAP.2002.1003377`.

[31] P. Maragos and F.-K. Sun, “Measuring the fractal dimension of signals: Morphological covers and iterative optimization,” *IEEE Transactions on Signal Processing*, vol. 41, no. 1, pp. 108–, 1993. DOI: `10.1109/TSP.1993.193131`.

[32] K. Dornelles and M. Roriz, “A method to identify the solar absorptance of opaque surfaces with a low-cost spectrometer,” *PLEA 2006 - 23rd International Conference on Passive and Low Energy Architecture, Conference Proceedings*, Jan. 2006.

[33] Ouster, *Software user manual: Firmware v2.1.x for all ouster sensors*, Ouster, Jun. 2021. [Online]. Available: `https://data.ouster.io/downloads/software-user-manual/software-user-manual-v2.1.x.pdf`.