

Real-time reconstruction of indoor ground surfaces in occluded environments filled with smoke based on point clouds obtained using LiDAR

Christiaan W.M. Wiers

Delft University of Technology

Real-time reconstruction of indoor ground surfaces in occluded environments filled with smoke based on point clouds obtained using LiDAR

by

Christiaan W.M. Wiers

Student Name	Student Number
Christiaan Wiers	4715349



Veiligheidsregio Rotterdam-Rijnmond

TU Delft supervisor: Cosimo Della Santina
Company supervisor: Robbert Heinecke
Faculty: Mechanical Engineering (ME)
Department: Cognitive Robotics (CoR)
Company: MIC Veiligheidsregio Rotterdam-Rijnmond

Submission date: 22/02/2024

Defence date: 11/03/2024

Cover: Chunshang Li. "Deep Learning Lidar Odometry". 2018. URL: <https://www.chunshangli.com/2018/04/20/e2e-lidar-odom.html> (accessed Oct. 20, 2023).



Abstract

With the current transition towards renewable and high-tech solutions, the world is becoming increasingly complex. Consequently, the challenges faced by firefighters also intensify. For that reason, firefighting robots are rising in popularity despite being far from perfect. An important area of improvement is the perception capabilities of those robots, given the fact that firefighting robots suffer from occluded camera views in environments filled with smoke. To overcome this challenge a LiDAR sensor may be used but experiments in this work show that even those point clouds are adversely affected by smoke. Consequently, this work presents a method for real-time reconstruction of ground surfaces in occluded environments filled with smoke. The developed method functions in ROS Noetic and merges segmented ground points, when available, with ground surfaces which are reconstructed based on information from segmented wall points. In this way, the method works even without the presence of ground points. To achieve this, a combination of established techniques from scientific literature, along with newly developed techniques were implemented. Doing so gives the robot's operator an improved representation of the ground surface within environments filled with smoke. Ultimately the developed method may allow for autonomous navigation based on LiDAR data within environments filled with smoke. This research shows that a method consisting of techniques which tackle the independent sub-challenges arising from the use of LiDAR in indoor environments filled with smoke can effectively reconstruct the ground surfaces within those environments. Furthermore, the developed method has the potential to do so in a real-time manner.

Contents

1	Introduction	1
1.1	The thesis goal and research question	2
1.2	Scope definition and illustration	4
1.3	Main contributions	5
1.4	Report structure	5
2	State of the art and selection of techniques	6
2.1	The effects of smoke on point clouds	6
2.2	Point cloud denoising	8
2.2.1	State of the art	9
2.2.2	Selection of the technique	10
2.3	Ground segmentation	11
2.3.1	State of the art	12
2.3.2	Selection of the technique	13
2.4	Ground surface reconstruction	15
2.4.1	State of the art	16
2.4.2	Selection of the technique	17
3	Method	18
3.1	Overview	19
3.2	Point cloud denoising	20
3.3	Ground and wall segmentation	21
3.4	Ground surface reconstruction and merging results	23
4	Experiments	26
4.1	Equipment used	26
4.2	Experiments in the basic training room	27
4.3	Experiments in the complex training room	28
5	Results	29
5.1	Qualitative results	29
5.2	Quantitative results	32
5.2.1	Occupancy grid evaluation	32
5.2.2	Reconstructed height evaluation	33
5.2.3	Computational speed	34
6	Discussion	35
6.1	The selection of techniques	35
6.2	The developed method	36
6.3	The conducted experiments	37
6.4	Research implications	37
7	Conclusion	38
8	Future work	40

References	41
A Appendix	43
A.1 Additional pseudo-code	43
A.2 Parameters used by the developed method to obtain results	45
A.3 Enlarged qualitative results	46
A.3.1 Basic training room without smoke	46
A.3.2 Basic training room filled with heavy smoke	47
A.3.3 Complex training room without smoke	49
A.3.4 Complex training filled with heavy smoke	50
A.4 LiDAR selection considerations	52
A.4.1 Selection procedure for the LiDAR purchase	52
A.4.2 Overview table of commercial LiDAR models	54

1

Introduction

In today's increasingly complex and electrified world, firefighting is becoming increasingly complex as well. To enhance firefighters' safety and expand the array of dangerous environments to operate in, the fire brigade of region Rotterdam-Rijnmond in The Netherlands, called "MIC Veiligheidsregio Rotterdam-Rijnmond" has started to use a remotely controlled firefighting vehicle called "Brutus" (Figure 1.1). This robot is used in situations which would be too dangerous for the firefighters to work in and is equipped with CCTV cameras and a water canon for fire extinguishing purposes. The cameras allow the operator to see and navigate Brutus inside dangerous environments, hence obtaining valuable information.

The problem in these situations is that the smoke encountered during operation can occlude the view of the cameras. For that reason, MIC Veiligheidsregio Rotterdam-Rijnmond is interested in the implementation of a LiDAR sensor on Brutus. However, as will be shown in this thesis, environments filled with heavy smoke result in degraded point clouds, which limits the amount of useful information available to the operator. For that reason, the goal of the fire department is to leverage the limited information from those degraded point clouds to aid the operator - and consequently firefighters - with additional spatial information about smoke-filled environments. This can help to improve the operator's control of Brutus and also be an important step toward the fire department's longer-term ambition of enabling autonomous navigation on Brutus, even in these smoke-filled environments.

Following the fire department's goal regarding the use of a LiDAR sensor, this introduction will begin by presenting the goal of this thesis as well as the main research question and associated subquestions. Next, the scope of this research is defined and the main contributions are presented. Finally, the structure of the report will be outlined.



Figure 1.1: Firefighting robot Brutus [1].

1.1. The thesis goal and research question

In line with the goal outlined by the fire department and in consideration of the requirements for an MSc Robotics master's thesis, the goal of this work is to develop a method which can reconstruct ground surfaces using point clouds obtained in indoor environments filled with smoke. The reason to focus on ground surface reconstruction is that this can provide useful, additional information to the operator. Furthermore, the obtained ground surfaces can form a basis for ultimately enabling autonomous navigation on Brutus. The *thesis goal* is visualised in Figure 1.3 and can be summarised as follows:

- *Developing a method which can leverage the limited information from degraded point clouds obtained in environments filled with smoke, to reconstruct indoor ground surfaces in real-time.*

In order to facilitate smooth integration between the developed method in this thesis and future developments on Brutus, such as autonomous navigation algorithms, it was decided to implement this method in ROS Noetic [2]. Furthermore, a Jetson Orin Nano single-board computer [3] running on Ubuntu 20.04 [4] was used given its unique trade-off between computational performance and size. The LiDAR sensor used in this research is the Hesai QT128 mechanical LiDAR [5] which has a 360° horizontal field of view. The LiDAR was chosen after carefully considering multiple LiDAR models based on their field of view, resolution, range, operational temperature and price. This procedure can be found in Appendix A.4. An overview of the developed method in the broader context of Brutus' components is presented in Figure 1.2 and more details about the exact functionality of the developed method can be found in Chapter 3.

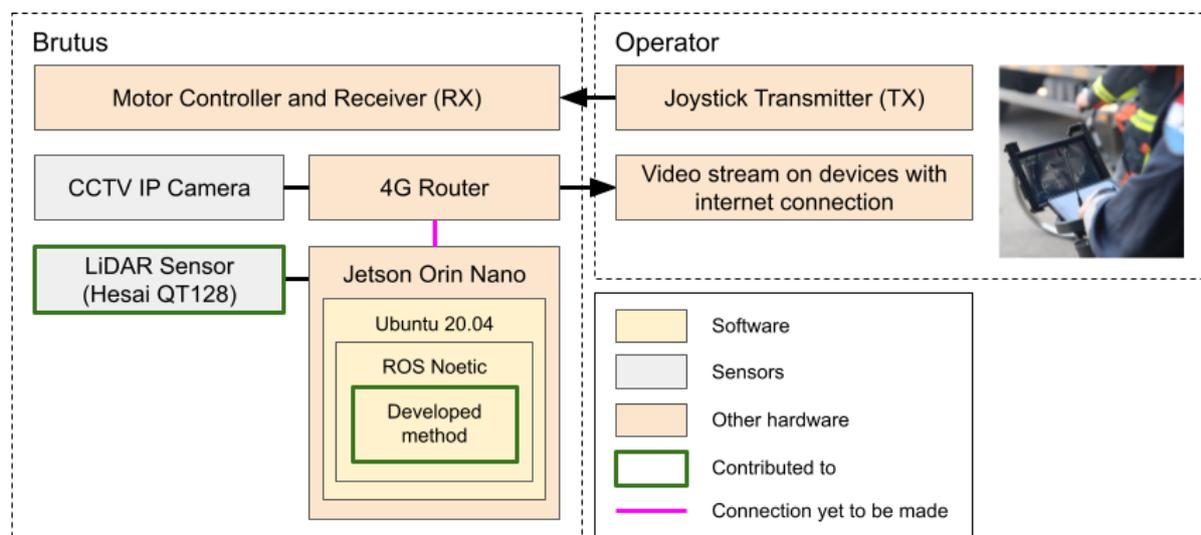


Figure 1.2: Simplified overview of the different components involved when operating Brutus. Image of the operator via RFGFotografie [6].

Following the thesis goal presented above, a main research question and associated subquestions are formulated. These subquestions are established to address the main research question as effectively as possible.

Main research question

- *How to enable real-time reconstruction of indoor ground surfaces in occluded environments filled with smoke based on point clouds obtained using LiDAR?*

Subquestions

- *How does smoke affect the visibility of ground points in a point cloud obtained with LiDAR?*
- *What method can be developed to tackle the main research question?*

To answer the first subquestion, experiments are performed using the LiDAR sensor in a room filled with artificial smoke. This is done to assess how the smoke affects the point clouds obtained by the LiDAR sensor. The results obtained by these experiments can be found in Section 2.1. Furthermore, the second subquestion is answered by analysing the challenges caused by the smoke on point clouds and finding independent solutions to those challenges. This process will be described by Section 2.2 and the remaining sections of Chapter 2.

1.2. Scope definition and illustration

Following the research question presented above, it is important to clearly define the scope of the project - what this research aims to include and, consequently, what it does not include. Within this thesis project, a method has been developed for reconstructing ground surfaces in both clear indoor environments as well as indoor environments filled with smoke. Consequently, it is important to note that this method aims to *reconstruct the full indoor ground surface*, disregarding any objects on top of it (Figure 1.3) and therefore the output shall not be confused with a traversable map. For clarification, a traversable map shows the ground with the exact locations of where the robot can or cannot drive, hence including objects on the ground surface. Furthermore, in this research *reconstructing the full indoor ground surface* means that the ground points are reconstructed according to the characteristics of the LiDAR. This means reconstructing the points according to the LiDAR's horizontal resolution and with the same amount of vertical channels (rings). Finally, this research is only conducted in artificial smoke, so not in actual smoke. The included and excluded scope considerations for this thesis are summarised in Table 1.1.

Scope considerations	Included
Full ground surface reconstruction	Yes
Creating a traversable map	No
Reconstructing the expected ground points	Yes
Operational in clear and smoke-filled environments	Yes
Testing and designing for indoor usage	Yes
Testing and designing for outdoor usage	No
Testing and designing on real smoke	No
Testing and designing on artificial smoke	Yes

Table 1.1: Summary of the included and excluded scope considerations.

The included scope considerations are chosen based on a combination of time constraints and the available resources throughout this project. The rationale behind choosing for the full indoor ground surface reconstruction is that this will directly help the operator's understanding of a smoke-filled environment. Additionally, when eventually combining the obtained results with object detection, it can be an important step towards enabling autonomous navigation in occluded indoor environments using LiDAR.

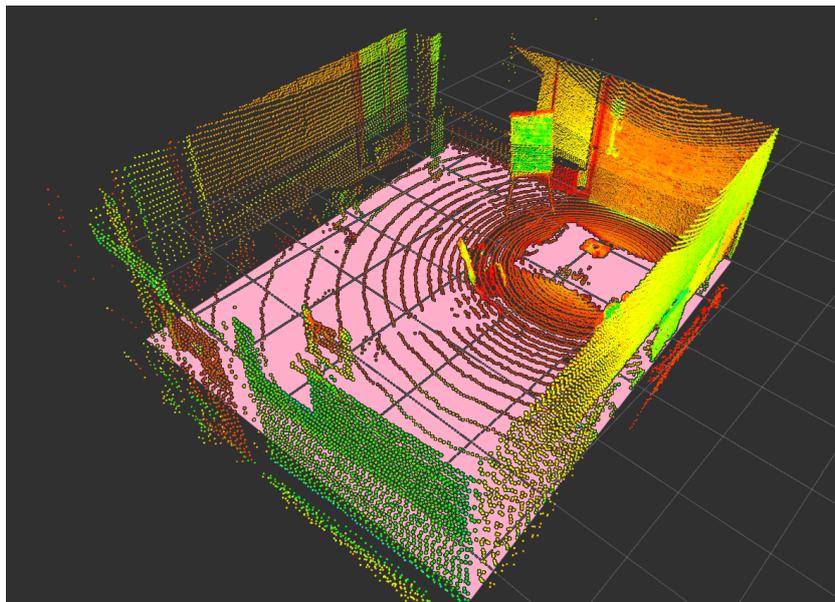


Figure 1.3: The ground points lying on the pink surface shall be reconstructed by the method.

1.3. Main contributions

- Visualisation of the adverse effects caused by artificial smoke on the quality of point clouds obtained using LiDAR.
- A novel method - functioning in ROS Noetic [2] - for real-time reconstruction of ground surfaces in occluded environments filled with smoke, which works even without the presence of ground points.
 - Reproduction of the leading AGDOR denoising technique by L. Minh-Hai et al. [7] including proof of effectiveness in environments filled with smoke.
 - Extension and reproduction of the channel-based ground segmentation work by C. Phuong et al. [8]. Resulting in more precise ground point segmentation for environments filled with smoke.
 - Design and development of a wall-based ground surface reconstruction technique in both Python and C++ to enhance speed and ease future development of this technique.
 - Developed a technique for combining the results of ground point segmentation with those of reconstructed ground surfaces on a per-segment basis. This is done to obtain the best possible representation of the ground in environments filled with smoke.

1.4. Report structure

Next to this introduction chapter, this report consists of several chapters. In the chapter hereafter (Chapter 2), the effects of smoke on point clouds obtained using LiDAR are introduced. Additionally, the state-of-the-art techniques and process behind selecting the techniques used in the developed method are presented. Next, in Chapter 3, the developed method is presented and explained in detail. After that, Chapter 4 introduces the experiments which were conducted and Chapter 5 will present the qualitative and quantitative results from the experiments. Finally, Chapter 6 discusses the results obtained after which Chapter 7 concludes the findings of this research and Chapter 8 presents the future work.

2

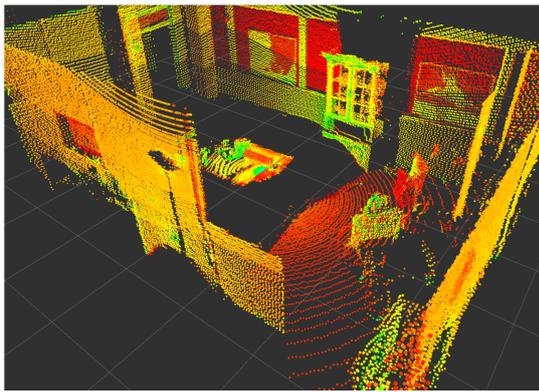
State of the art and selection of techniques

This chapter describes the relevant techniques and the selection thereof regarding the developed method in this thesis. This chapter will first assess and present the challenges caused by smoke on LiDAR sensors. This is useful because there is no directly relevant scientific literature or solution available related to the overarching goal of this thesis. Knowing the independent challenges caused by smoke on LiDAR sensors, the relevant scientific research direction for each of the challenges is identified. These scientific research directions contain potential solutions to the independent challenges caused by smoke on LiDAR sensors and are therefore presented as dedicated sections within this chapter. For each scientific research direction related to an independent challenge, the most relevant (state-of-the-art) techniques/solutions are presented as subsections. Finally, per scientific research direction, a second subsection describes how the most suitable technique was selected, hence clarifying how the design and techniques used in the developed method have come about.

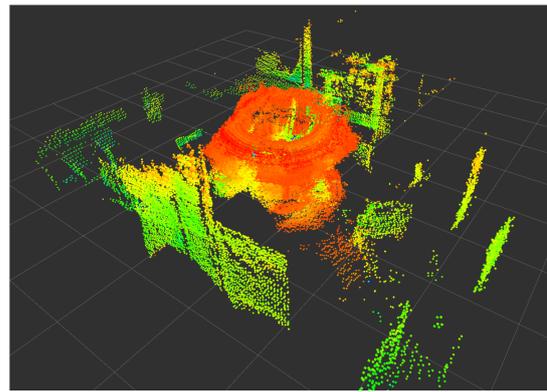
2.1. The effects of smoke on point clouds

Scientific research on the effect of smoke on point clouds is very scarce, for that reason an evaluation of these effects has been performed as part of this thesis. These tests have been performed in a dedicated training room at fire station 'Slotlaan' in Capelle aan den IJssel (The Netherlands). An important side note here is that the tests conducted here were performed using artificial smoke instead of real smoke. This means that the precise effect of smoke on LiDAR remains uncertain while at the same time, this environment is the closest representation of actual smoke given the resources available throughout this research. Looking back at the first subquestion, namely: *How does smoke affect the visible ground surface in a point cloud obtained using LiDAR technology?*. The following may be concluded when looking at results obtained using LiDAR in smoke as shown in Figure 2.1:

- Ground points exhibit significantly lower intensity levels when compared to other points. This can be explained by the typically darker colour of ground surfaces as well as the shallower angle made between the LiDAR sensor and the ground compared to other objects. Both of these can cause less light to be reflected onto the LiDAR's receiver.
 - Vice versa, the vertically oriented wall points exhibit higher intensity levels.
- Depending on the density of smoke, most ground points are absorbed by the smoke, creating false negative points. This can be explained due to the lower intensity reflections caused by the often darker coloured ground surface in combination with the light-absorbing effect of the water-based artificial smoke [9].
- The absorbed (ground) points by the artificial smoke create (orange-coloured), low-intensity false positive points around the LiDAR sensor.



(a) Raw point cloud without any smoke in the room.



(b) Raw point cloud occluded by heavy smoke in the room.



(c) Visibility in the room without smoke.



(d) Visibility in the room occluded by heavy smoke.

Figure 2.1: Visual impression regarding the effect of artificial smoke on point clouds obtained using LiDAR technology.

With an understanding of how ground points disappear in the presence of smoke and create false positives, it is reasonable to investigate whether addressing this challenge is covered in scientific literature. Unfortunately, it is not when looking at this challenge as a whole. Therefore, to effectively deal with the challenge caused by smoke, the main research question is tackled by developing a method which combines the most suitable - often already existing - techniques for solving the independent (sub)challenges caused by smoke. This is done because the challenge for LiDAR sensors in the presence of smoke can also be seen as multiple independent challenges and for those independent challenges, solutions do exist within scientific literature. Using this perspective, the following independent challenges introduced by smoke on point clouds have been identified:

1. *False positives:* As can be seen in Figure 2.1, smoke introduces a dense layer of (orange-coloured) false positive points. This means that these points imply the presence of objects while in reality, there are none.
2. *False negatives:* At the same time, ground points are not present at the places where you would expect them. This takes away the chance to identify any ground points at those locations.
3. *Irregular presence of ground points:* Some parts of the point cloud do contain actual ground points (true positives) and are thus of significant value. Effectively filtering out those ground points can be very challenging.

Knowing the independent challenges caused by smoke on LiDAR sensors, the scientific research directions for each of these challenges are identified and presented below. After presenting these scientific research directions, dedicated sections will introduce, for each research direction, the concept, and the most relevant (state-of-the-art) techniques, followed by an explanation of how a technique was selected for eventual implementation in the developed method.

1. For the challenge of *false positives*, numerous scientific papers are available under the phrase of "*point cloud denoising techniques*".
2. For the challenge of *false negatives*, a multitude of scientific literature is available albeit mostly in an aerial or outdoor context [10], [11]. Techniques which deal with missing points are often referred to as "*surface reconstruction techniques*" but also as "*semantic scene completion*".
3. For the problem regarding *irregular presence of ground points*, numerous scientific papers are available. Techniques which can identify those ground points are often referred to as "*ground segmentation techniques*".

2.2. Point cloud denoising

With the rise of autonomous driving-related research, the LiDAR sensor gained popularity within the scientific community as well [12]. Consequently, LiDAR sensors have been tested in different adverse weather conditions such as rain and snow. By doing so, the challenge of false positives caused by reflections from snow and rain became more widely known and as a result point cloud denoising methods for adverse weather conditions started to be developed. With the first variants being introduced with the launch of the popular C++ based ROS package called Point Cloud Library [13]. In essence, the goal of a point cloud denoising method is identifying and removing the false positive points caused by unwanted reflections of the LiDAR's pulses from for example, snow and rain. Many different point cloud denoising methods exist but roughly six different types can be identified:

- *Radius outlier removal*: These methods filter out points when the number of neighbouring points - within a certain radius of a point - is below a specific threshold. Figure 2.2 illustrates this concept with the orange dot being the points around which a radius outlier removal is performed.
- *Statistical outlier removal*: These methods filter out points based on the statistical characteristics of the surrounding points. For example, it may calculate for each point, the mean distance and standard deviation of the neighbouring points. Points which fall outside of a certain multiple of the standard deviation from the mean of a given point will be filtered out.
- *Intensity-based methods*: Like the name suggests these methods filter out points with an intensity below a certain threshold.
- *Dynamic methods*: These methods are often based on radius outlier removal methods but allow points at large distances to have fewer neighbouring points within a certain search radius. This results in improved performance because the density of a LiDAR's point cloud decreases with increasing distance and therefore one expects to find fewer neighbouring points at larger distances when using the same search radius. This concept is illustrated in Figure 2.2b.
- *Hybrid methods*: The more novel and best-performing methods are often a combination of the denoising method types presented above.
- *Machine learning-based methods*: These types of point cloud denoising methods work primarily by being trained on large datasets of labelled point clouds. Doing so, these models leverage the data in order to create effective point cloud denoising strategies.

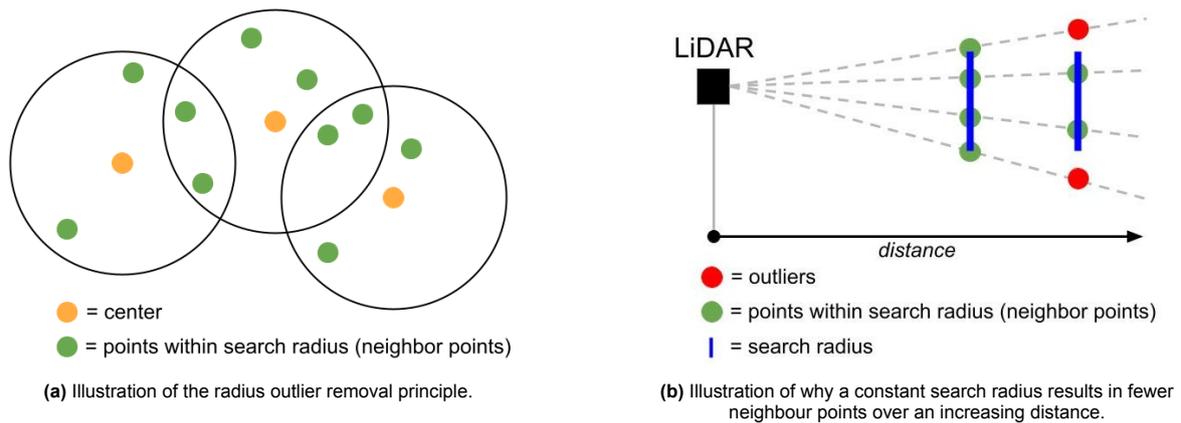


Figure 2.2: Visualisation of the radius outlier removal and dynamic methods.

2.2.1. State of the art

Regarding the state of the art, four different denoising methods are identified: two hybrid methods and two machine learning-based methods.

Hybrid denoising methods:

- **AGDOR:** This method is developed by Mukhopadhyay et al. [7], and is made of two parts, an intensity threshold filter and a dynamic radius outlier removal method. The method starts by using an intensity threshold below which points are filtered out. The authors suggest this threshold to be the mean between the obstacles' mean intensity value and that of the snow's reflections (the false positives) which would require some prior knowledge about the environment. Otherwise, this threshold value shall be chosen empirically. After the points below the intensity threshold are filtered out, the second part of the method will place back some of the points which had just been removed. It does so by leveraging the sparse characteristics of falling snowflakes using a dynamic radius outlier filter. This means that the method only keeps a point if it contains a certain amount of neighbours within the search radius, but this search radius increases linearly with an increasing point's distance to the LiDAR sensor. Unfortunately, this method does not directly compare its performance to that of other state-of-the-art methods which are described in this section but it does show outperformance compared to other well-known denoising methods.
- **AORI:** In this method created by Donny et al. [14], the observation is used that LiDAR channels are often not uniformly distributed meaning that fewer neighbour points have to be expected for the LiDAR's sparser top and bottom channels. To deal with this non-uniform distribution the point cloud is converted into a range image and next, a dynamic radius outlier removal filter is used in the form of a kernel to filter out the false positives. After filtering out the false positives, the range image is converted back into a point cloud. Unfortunately, this method does not compare itself to the machine learning-based methods but it does show a slight out-performance in terms of accuracy and F1 scores compared to AGDOR while lagging in precision. Additionally, it shows a computational speed of 4 FPS compared to 2 FPS for AGDOR using a 2.20 GHz CPU and 32 GB of RAM.

Machine learning-based methods:

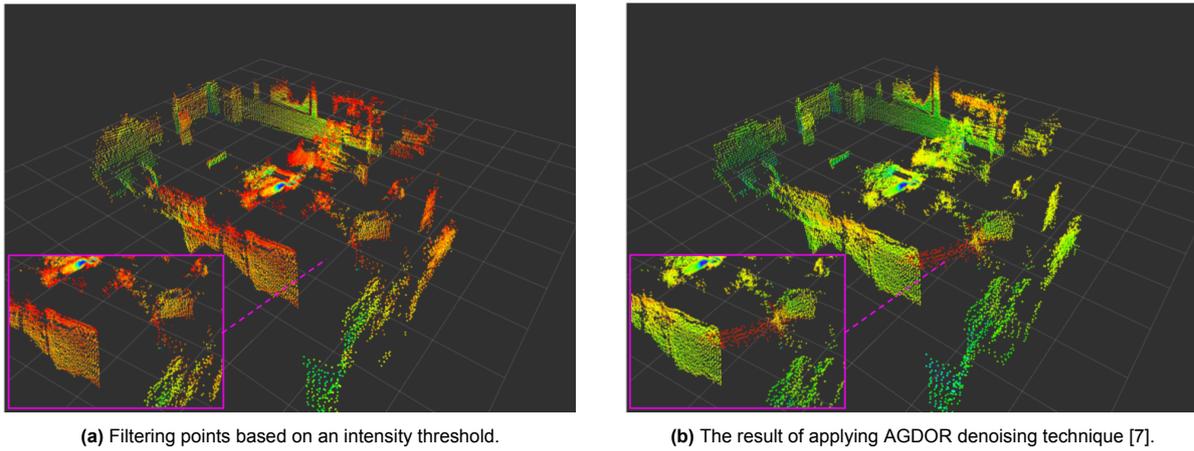
- *WeatherNet*: Heinzler et al. [15] proposed a CNN method which is based on the semantic segmentation network called LiLaNet [16]. The method is adjusted so it only segments 3 different classes namely: clear, rain and fog. The class 'clear' refers to any point of the point cloud which is not classified as rain or fog. Additionally, a dropout layer was added to improve the methods' robustness. As input the method uses a point cloud converted into an intensity image and distance matrix. Unfortunately, this method does not compare itself to any of the state-of-the-art methods mentioned in this subsection but it is reasonable to say that it outperforms hybrid denoising methods given the impressive runtime of only 34 ms per frame.
- *4DenoiseNet*: This deep learning-based denoising method was developed by Seppänen et al. [17] uses time as a fourth dimension - next to a point cloud's 3D characteristics - to improve its denoising performance. More specifically, the method uses a spatial and a temporal branch both of which use a kNN-convolution module. Later, the two branches are fused back together. The work by the authors shows how their method outperforms WeatherNet on their own SnowyKITTI dataset in terms of IoU and computational speed.

With the state-of-the-art methods presented, it is important to emphasize that the methods presented here are often trained or tuned on a limited amount of datasets while also being designed for adverse weather conditions such as snow and rain instead of smoke. Additionally, the hardware specifications and programming languages used with these methods are often unknown.

2.2.2. Selection of the technique

To select the most promising techniques for point cloud denoising within this research's method, different denoising methods have been assessed. The main considerations to make in this selection process are *speed*, *accuracy* and *chances of successful implementation*. For example, AORI seems very promising in terms of precision and recall but has a low chance of successful implementation since very limited pseudo-code is provided and because of added complexity given that the point cloud is converted and processed as a range image. Similarly, the machine learning-based denoising methods have not been implemented because they require a lot of labelled point clouds which are simply not available in the context of indoor environments occluded by smoke and did not seem feasible to obtain during this project. The intensity-based methods, on the other hand, are easy to implement and fast but only provide limited accuracy. All things considered, the AGDOR denoising technique created by L. Minh-Hai et al. [7] seems to provide the best trade-off between speed, accuracy and chances of successful implementation. Additionally, it offers a leading combination of precision, accuracy and recall scores compared to more 'classical' methods such as DROR [18], DSOR [19] and LIOR [20]. Furthermore, an acceptable processing speed is achieved compared to 'classical' point cloud denoising techniques.

To prove the added value of the AGDOR denoising technique, an evaluation was performed in which a simplistic intensity threshold filter was compared to the AGDOR technique. The intensity threshold method was chosen for comparison because it allows for very fast processing as well as the observation that the false positives caused by dense smoke all seem to have a significantly lower intensity compared to other points. As can be seen in Figure 2.3, the AGDOR denoising technique manages to retrieve ground points which were initially filtered out based on the intensity threshold. This intensity threshold filter works by removing all points below a certain intensity value. The optimal value is obtained by empirically testing different intensity threshold values. A detailed explanation of the AGDOR method can be found in the section 3.2.



(a) Filtering points based on an intensity threshold.

(b) The result of applying AGDOR denoising technique [7].

Figure 2.3: Overview of results when applying two different types of filters (intensity threshold versus AGDOR).

2.3. Ground segmentation

The concept of ground segmentation means identifying which points from a point cloud belong to the ground surface. Ground segmentation serves two main purposes [21]. First of all, it determines the area on which a robot or vehicle may be able to move. Secondly, knowing where the ground is helps with identifying and/or tracking moving objects and can lower the computational complexity of doing so. After all, when knowing which points belong to the ground surface, there are a lot fewer points left to access. Within the scientific literature, a lot of research has been done in the context of ground segmentation. This results in a wide variety of different techniques which can be divided into the following four categories [22]:

- *Ground modelling methods:* This refers to all the techniques that try to fit one or multiple planes - to the point cloud - which overlaps with as many ground points as possible. Well-known ground modelling methods include, among others, RANSAC and PCA, as well as methods extending on those techniques. The concept of RANSAC ground segmentation is illustrated in Figure 2.4a, with the green dots representing the segmented ground points since they lie on the grey plane which is fitted to the point cloud using RANSAC.
- *Elevation maps:* These techniques divide the point cloud into a grid of cells when looking at the points cloud's xy -plane. In this grid representation of the point cloud, each cell contains a certain height value assigned to it. This can be e.g. the median height value of all points lying within that cell or even the largest height value within the cell. Next, the ground cell can be identified based on different techniques such as clustering adjacent cells with similar low height values. These techniques provide advantages in terms of robustness to noise but are known to struggle with sparse/occluded point clouds and lag in terms of computational speed. In Figure 2.4b, an elevation map is visualised using a side view. The cells highlighted in green represent the ground whereas the grey cells contain larger height values and are therefore not considered ground surface.
- *Adjacent points:* Techniques in this category, segment ground points by looking at characteristics such as distance or the elevation angle between adjacent points. Most of these techniques evaluate points from consecutive LiDAR channels. This concept is illustrated in Figure 2.4c, where the h represents the difference in height between two consecutive points and α represents the angle formed between the next consecutive pair of points. For points to be considered ground points, both the h and α values have to stay below certain threshold values.
- *Deep learning:* These techniques use convolutional neural networks and a lot of labelled data to learn how to identify ground points. These methods can be fast and accurate but also require a lot of resources, often in the form of NVIDIA GPUs as well as large amounts of labelled point clouds.

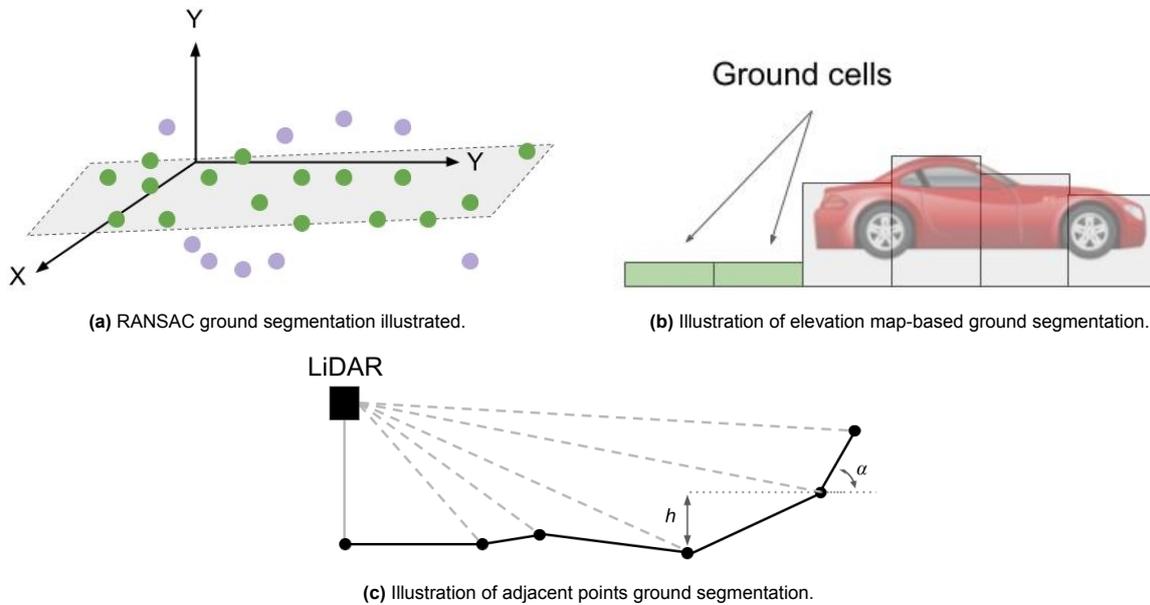


Figure 2.4: Different ground segmentation techniques visualised.

2.3.1. State of the art

Considering the state-of-the-art ground segmentation techniques, four are found to excel. This concerns one ground modelling technique, two deep learning-based techniques and one adjacent point technique.

Ground modeling method:

- *Patchwork++*: This method created by L. Seungjae et al. [23] is state of the art with an F1 score of 96% and recall of 98% on the SemantickITTI dataset [24]. While no direct comparison is made with the other state-of-the-art methods in the section it does show out-performance compared to older methods in terms of F1 score and recall. In general terms, the method works by first filtering out reflected noise and subsequently dividing the point cloud into different polar grid cells. To each of these cells, a wall plane is fitted using the PCA method and the points lying on those planes are considered ground points. The reason for fitting planes to the different polar grid cells is because this allows for more robust ground fitting (e.g. also working for segmenting uneven ground surfaces).

Deep learning-based methods:

- *GnDNet*: This method developed by A. Paigwar [25], uses the well-known PointNet network [26] in combination with a pillar feature encoding network to extract height features from the point cloud. The authors augment the SemantickITTI point cloud dataset [24] upon which they train their model. Doing so, they claim their method achieves state-of-the-art performance compared to older methods which are not included in this state-of-the-art overview.
- *RangeNet++*: Created by A. Milioto [27], their work claims to achieve state-of-the-art performance. The developed method works by first converting the input point cloud into a range image and performing 2D convolution semantic segmentation on those range images. Next, the processed 2D range image is converted back into a point cloud.

Adjacent points method:

- *Fast ground segmentation*: Developed by C. Phuong et al. [8], this method stands out due to its exceptional processing speed of just 4.7 ms per frame, making it the fastest ground segmentation method by a large margin. This state-of-the-art processing speed is the reason for including this technique in this section. Additionally, the design of this method is well documented making it easier to implement. While no other quantitative metrics have been provided about the method, its qualitative results look very promising as well. Albeit reasonable to assume that the state-of-the-art methods mentioned earlier will perform better in terms of the other qualitative metrics.

2.3.2. Selection of the technique

After knowing the most promising ground segmentation methods, the most suitable one - in the context of this research - had to be selected. To get up to speed with ground segmentation the most simple and common ground segmentation technique called RANSAC was initially assessed on some of the point clouds obtained during the experiments of the LiDAR in artificial smoke. To do so, the Scikit-learn Python version of the RANSAC model [28] was tested in ROS Noetic. Python version of the RANSAC model is based on the original RANSAC model created by M. Fischler et al. [29]. Unfortunately, it did not lead to promising results. This is because applying it to the full point cloud leads to a plane which crosses the cloud and therefore does not (at all) represent the surface on which the ground points lie. Using knowledge such as the sensor height, a better estimation was achievable but this method remains prone to unevenness and sudden changes in the surface. After performing the initial tests with the "classic" RANSAC ground modelling method, the state-of-the-art methods presented above were tested on some of the clean point clouds obtained during the experiments.

Evaluating the state-of-the-art techniques started with implementing the deep learning methods. A lot of time was spent trying out these methods but unfortunately, all of these methods required CUDA software as well as an NVIDIA GPU which is beyond the reach of this project. Furthermore, most of those methods are designed and trained for automotive and thus outdoor purposes which increases the chances of lagging performance within the assessed environments in this research. Choosing such a method would also mean that a lot of additional indoor data would have to be collected and manually labelled which is very hard to do.

Next up was the leading ground modelling work model created by L. Seungjae [23] called "Patchwork++". This method was well documented and therefore the implementation of this method in ROS Noetic was successful. Results of the method on indoor Point Clouds obtained with our LiDAR can be seen in Figure 2.5a. The figure shows how the Patchwork++ method seems to struggle with properly segmenting the ground points. While some ground points were correctly identified, many vertical points (false positives) were also included and at the same time, segments of ground points have not been segmented at all (false negatives). One reason for this can be that the method is designed towards more outdoor environments and may therefore be less robust within indoor environments. For example, the input point cloud did not contain all the ground points one may expect to see (many false negatives) because they were located behind certain objects.

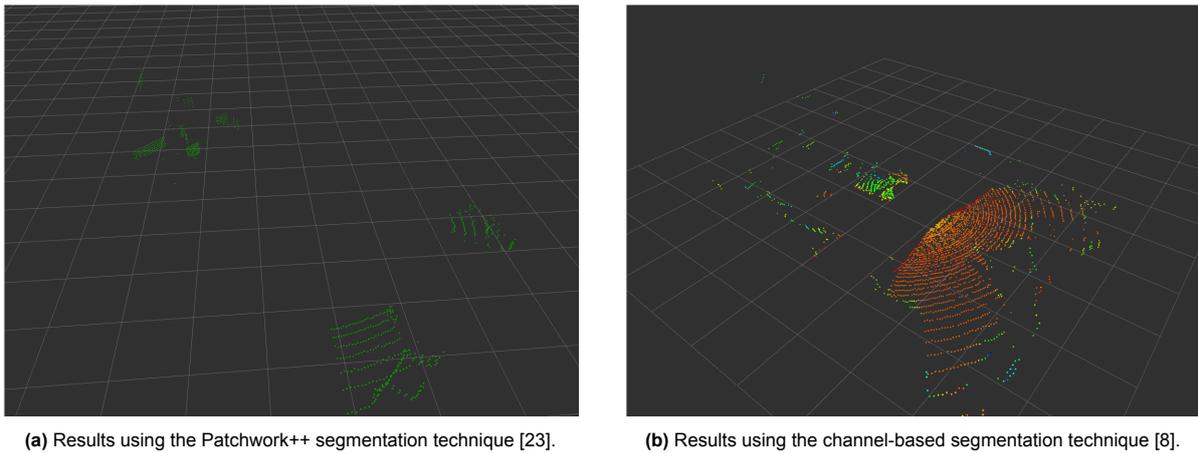


Figure 2.5: Comparison of the Patchwork++ and channel-based segmentation method in a smoke-free environment.

Finally, the channel-based ground segmentation technique created by C. Phuong et al. [8] was recreated and successfully implemented in ROS Noetic. The evaluation of this method led to very promising results both in terms of speed as well as accuracy as can be seen in Figure 2.5b. In contrast, to Patchwork++, the 'channel-based' technique shows significantly more correctly identified ground points (true positives). Unfortunately, it still shows false positives (noise) throughout the point cloud. To mitigate those unwanted false positives as effectively as possible, the problem was evaluated in more detail and a new feature was developed and added to the channel-based segmentation technique was developed. The mechanics behind this new feature as well as the original version are explained in section 3.3. The differences in the obtained results between the original 'channel-based' segmentation technique and the newly developed (updated) version are visualised in Figure 2.6.

As can be seen in Figure 2.6, the version with the new feature shows fewer false positives compared to the original version, hence indicating an increase in the technique's precision. The improved performance of this newly created feature on top of the channel-based segmentation technique resulted in the decision to use this combination of techniques as the ground segmentation approach within the method.

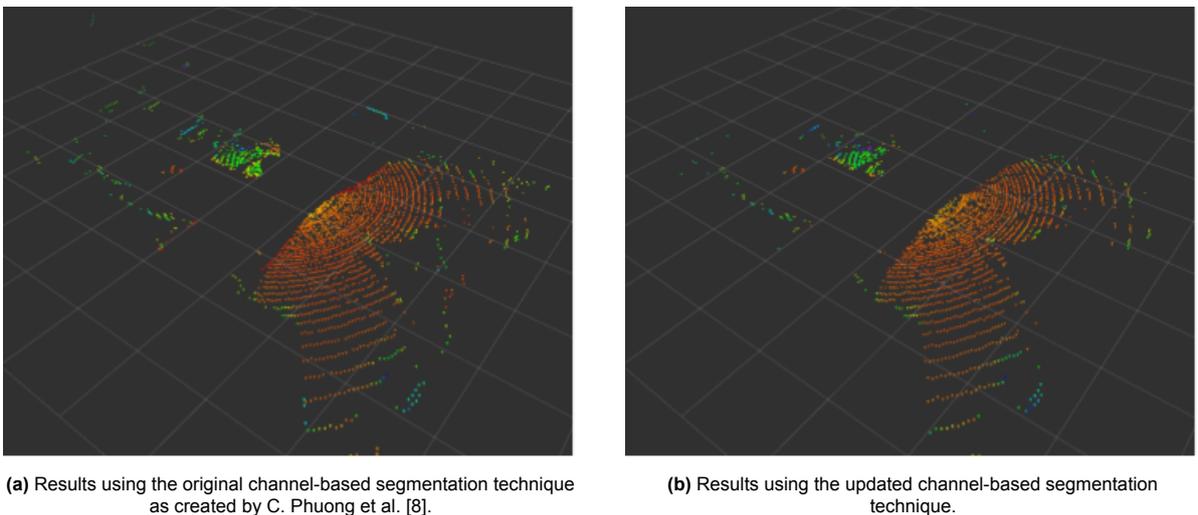


Figure 2.6: Visualisation of the different results using the original and updated channel-based segmentation technique.

Given the obtained results of the different ground segmentation methods in combination with practical implementation limitations of other state-of-the-art methods, the channel-based ground segmentation method was selected as it seemed to be the most suitable method in the context of this research.

2.4. Ground surface reconstruction

Smoke can lead to false negative (ground) points [30], which can be a problem because those ground points provide the information on which the robot (or its operator) eventually determines where it can or can not drive. The reason that ground points are missing is often due to either the presence of dense smoke which absorbs the emitted signal [9] or because the emitted signal is blocked by another object so that the actual ground is never reached. To properly deal with these false negative points, (ground) surface reconstruction techniques exist of which the two main categories can be identified within scientific literature. Additionally, the exact output created by these (ground) surface reconstruction techniques depends on the category to which the technique belongs.

- *Physics-based reconstruction*: These methods are often used for reconstructing the exterior surfaces of houses based on point clouds obtained with LiDAR. These methods, often use incomplete point clouds as input (e.g. from the outside of the house) and obtain an output in the form of some 3D CAD model consisting of several intersecting planes which represent the estimated exterior surfaces (e.g. from the house used as input). These methods estimate those exterior surfaces based on the input point cloud, by leveraging high-level, often physics-based knowledge about the object of interest. For example, when many points are vertically aligned over a large surface, it assumes that those points represent a wall. Using that assumption, it is then known that this wall has to intersect with a ground surface since a wall cannot float. Using high-level logical reasoning like just described, these methods use the limited knowledge from the point clouds to make educated guesses about surfaces to accurately reconstruct them. The main disadvantage of these techniques in the context of this research is the large runtime they require which is often in the tens of seconds.
- *Semantic scene completion*: The second and final category of ground surface reconstruction techniques identified within the scientific literature is that of semantic scene completion. These techniques are used in the context of self-driving cars because they enable an improved understanding of the scenes given that these methods outperform sole segmentation-focused methods [31]. In the context of these methods, ground surface reconstruction refers to reconstructing point clouds with false negatives. In other words, it is about turning the false negative into true positive ground points. These semantic scene completion techniques are types of deep learning networks which use encoders and decoders to reconstruct the point clouds. These techniques can be very accurate and offer real-time performance but do require a lot of labelled point clouds to train upon as well as expensive hardware like NVIDIA GPUs.

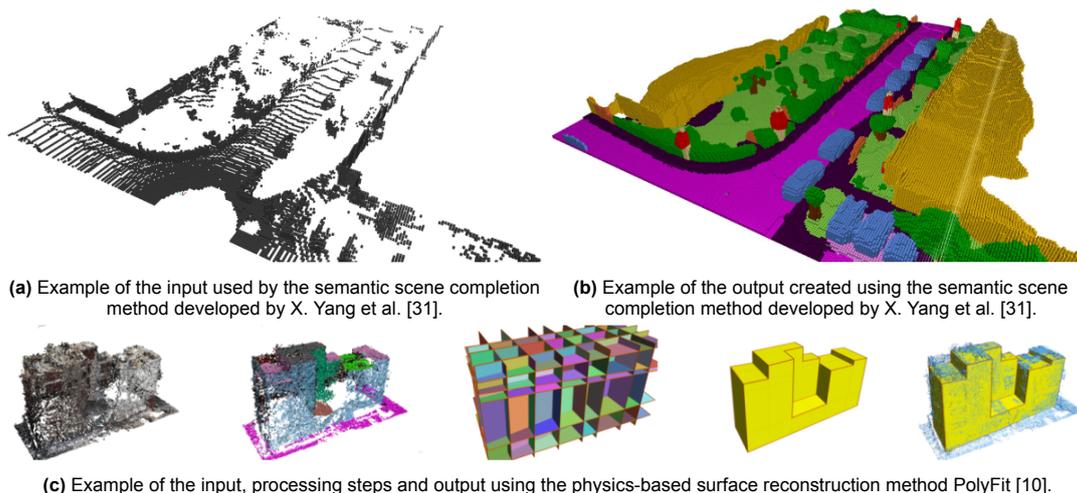


Figure 2.7: Visual impression of two different categories of point cloud-based surface reconstruction methods.

2.4.1. State of the art

When looking at the state-of-the-art methods, two physics-based and two semantic scene completion techniques are identified and discussed in this subsection.

Physics-based reconstruction:

- *Vectorized indoor surface*: As the name suggests, this method developed by J. Han et al. [11] focuses on reconstructing indoor door surfaces. It does so by first segmenting permanent structures from the scene and completing the walls and floors using Integer Linear Programming. In general terms, the wall structures are detected using k-nearest neighbours which determine which points are below a certain threshold in terms of the curvature they form. To the points below the threshold, RANSAC plane fitting is applied to determine which points form a wall surface. Additionally, RANSAC plane fitting is applied to the bottom and top points in the points cloud. The precise threshold values for this are determined based on the height of the wall points obtained earlier. Finally, the resulting planes are converted into a 3D CAD model which is the eventual output of this technique.
- *PolyFit*: This technique developed by L. Nan et al. [10], uses RANSAC plane fitting to fit a multitude of planes to the point cloud. Cleverly enough, it does so using the principle that intersecting planes have to meet watertight constraints which ensures physically valid models. Finally, the optimal number of planes is obtained using an objective function which takes into account, among other things, the model complexity and point coverage of the planes. This technique is visualised in Figure 2.7c.

Semantic scene completion:

- *SSA-SC*: While this method developed by X. Yang et al. [31] is already surpassed in terms of mIoU, it is still included as state-of-the-art because it does offer the best trade-off between mIoU and speed. In the context of this research, the trade-off between performance and speed is very important since the goal is to obtain a method which works in real-time. Their method is a network which consists of a 2D completion branch and a 3D semantic segmentation branch. Probably the most significant contribution is the 2D completion branch which turns the point cloud into a BEV feature map upon which the reconstruction of missing points is performed. The BEV feature map turns the point cloud into a 2D representation which leads to significant computational benefits. Their technique achieved a mIoU of 23.5%, giving it the third rank on the SemanticKITTI semantic scene completion challenge [32]. At the same time, this technique achieves a speed of up to 20 FPS. The input and output of this method are visualised in Figure 2.7a and b.
- *SCPNet*: Without going into too much detail about how the functionality of the method created by Z. Xia et al. [33], it works by having not one but two networks (a student and a teacher), each consisting of a 3D completion method as well as an encoder and decoder for segmentation purposes. By having these two networks, the features identified by both networks are matched which makes the method more robust to moving objects. These moving objects are what other semantic scene segmentation methods struggle with, resulting in unwanted false positives. Doing so, this method has the first rank in the SemanticKITTI semantic scene completion challenge with a mIoU of 36.7% [32]. Furthermore, this technique achieves speeds of up to 5.8 FPS.

2.4.2. Selection of the technique

To choose the ground surface reconstruction technique which is best for the method developed in the research, a couple of practical observations about the state-of-the-art techniques presented in Subsection 2.4.1 were assessed. The first one was that the *physics-based* methods take very long to compute which is understandable as they focus purely on reconstruction accuracy and are not designed for real-time applications. The runtime of these techniques is in the tens of seconds which in essence makes them unsuitable for implementation within the developed method in the research.

Next, there are the *semantic scene completion methods* which are designed for real-time application and also achieve impressive performance in terms of mIoU. Numerous attempts have been made to successfully implement these methods. Unfortunately, this did not lead to any success because these semantic scene completion techniques all require NVIDIA GPUs to run them. Even if these techniques would successfully run there was still the uncertainty of how to obtain enough relevant indoor point cloud for those techniques to train upon.

With both states of the categories showing serious practical limitations, it was decided to develop a novel ground surface reconstruction technique. This technique is designed to work in the context of the research assignment as presented in more detail within Section 3.4.

3

Method

In order to leverage the limited information from degraded point clouds obtained in environments filled with smoke for the reconstruction of indoor ground surfaces in a real-time manner, a method is developed based on the challenges caused by smoke as described in Section 2.1. The identified solutions to these challenges are point cloud denoising to remove the false positives, ground segmentation to identify the irregular presence of ground points and finally, ground surface reconstruction to better visualise where the ground is despite the many false negatives in the point clouds. After selecting the most suitable techniques for dealing with the independent challenges caused by smoke, an actual method which combines these techniques has to be created. This chapter will start by presenting an overview of the developed method. Next, each main component of the developed method will be explained in more detail using dedicated sections.

3.1. Overview

The developed method works using a degraded 3D point cloud obtained from an indoor environment filled with smoke as input. The output is a reconstructed point cloud of the full indoor ground surface reaching up to the surrounding walls. Depending on the amount of segmented ground points despite the smoke, this reconstructed point cloud consists of segmented ground points and/or reconstructed ground surfaces in the form of a point cloud. With the input and output explained, it is relevant to look at the different components of which the method consists in order to better understand how the output is obtained. The developed method consists of three main components namely, the point cloud denoising, ground and wall segmentation, and finally the ground surface reconstruction and merging of results. All these components will be described in more detail in dictated sections within the chapter but first, they will be briefly introduced in this section. Furthermore, a visualisation of all functionalities within the method is presented in Figure 3.1.

- **Point cloud denoising:** The first step of this method is to properly filter out the false positive points created by the smoke. These (noise) points are characterised by their dense distribution close to the LiDAR sensor as well as by their low-intensity values. The denoising of the point cloud is done by reproducing and implementing a leading point cloud denoising filter.
- **Ground and wall segmentation:** As the second step, ground segmentation is applied on the denoised point cloud. This is useful because true positive ground points which are still visible in smoke remain the most accurate representation of the ground surface. The selected ground segmentation technique for this component of the method is known for its leading processing speed. Next, segmentation of wall points is performed using the same technique but with different parameters. The segmentation of wall points is relevant since those points contain valuable information for the reconstruction of ground surfaces and are often among the points with the highest intensities which keep these points visible in even the heavier smoke conditions.
- **Ground surface reconstruction and merging results:** This third and final step uses the segmented wall points and divides them into multiple angular segments - when looking at the xy-plane - using a polar grid. Next, per angular segment: the method checks how many ground points it expects to find and compares that with the actual number of segmented ground points found for that segment. If the number of segmented ground points is above the threshold for that segment, those segmented ground points are shown. Otherwise, the method reconstructs the ground surface for that segment using the information from the wall points in that segment in order to determine to what distance and height, the ground points shall be reconstructed.

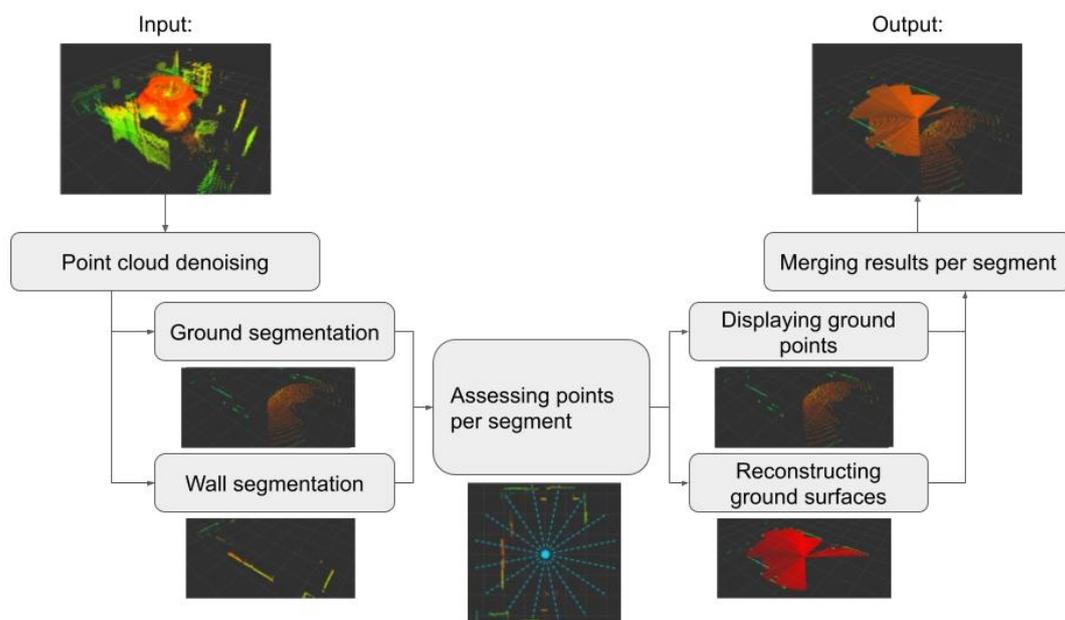


Figure 3.1: Overview of the different steps within the developed method.

3.2. Point cloud denoising

As the first step of this method, the leading point cloud denoising technique called AGDOR [7] is recreated in Python, fine-tuned and implemented to work in ROS. This technique works by first filtering out points based on a user-defined intensity threshold. This is done with the high-level knowledge that noise (false positives) caused by adverse conditions, have mostly lower intensities compared to reflections from objects. Now, the low-intensity points may also include points of objects such as the actual ground. To retrieve low-intensity points from actual objects (true positives), a KDTree neighbour search [13] is performed. This is done with the rationale that adverse conditions (for example rain and snow) often create sparse noise in point clouds. In smoke, it is the other way around namely: false positives are often the denser points since they arise close to the LiDAR sensor and because they concern reflections from very fine particles. Therefore in this context, a KDTree neighbour search is used to filter out the false positive points - which are assumed to be denser compared to reflections from actual objects. Normally, a KDTree search is performed with a fixed search radius. However, as described in sub-section 2.2.1, the search radius in AGDOR scales linearly with the distances of the point from which the KDTree neighbour search is performed. Figure 3.2b, shows the point cloud obtained after denoising. Additionally, the pseudo-code in Algorithm 1 shows the functioning of the implemented denoising method.

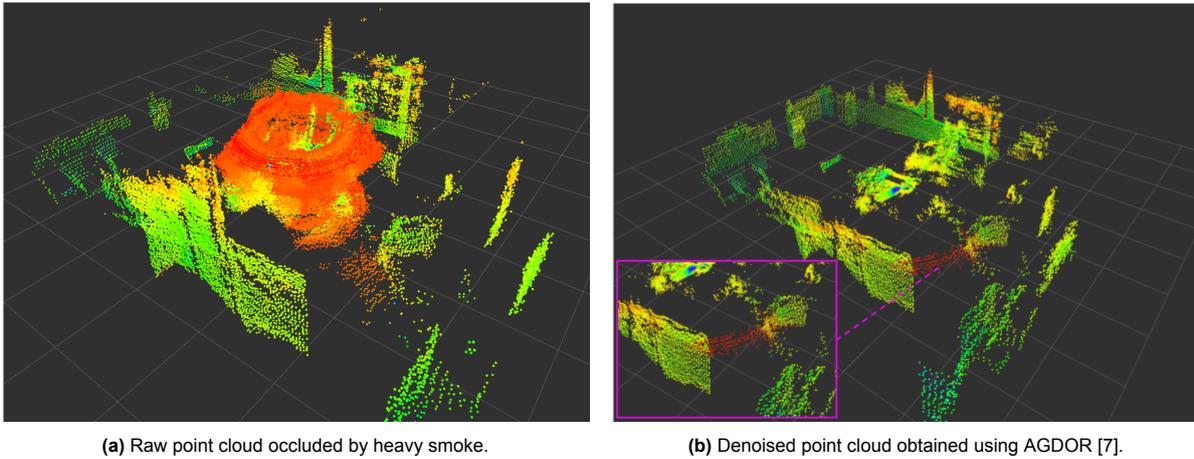


Figure 3.2: Visualisation of output from point cloud denoising.

Algorithm 1 AGDOR cloud denoising method [7].

Input: $P = p_1, p_2, \dots, p_n; p_i = x_i, y_i, z_i, intensity_i, timestamp_i, ring_i$
Output: $P_{noise}, P_{denoised}$

- 1: $ITh \leftarrow$ defined intensity threshold
- 2: $neighborsthreshold \leftarrow$ defined number of neighbors threshold
- 3: **Step 1:**
- 4: **for** $p \in P$ **do**
- 5: **if** $p[‘intensity’] < ITh_i$ **then**
- 6: $P_{low} \leftarrow p$
- 7: **else**
- 8: $P_{denoised} \leftarrow p$
- 9: **end if**
- 10: **end for**
- 11: **Step 2:**
- 12: **for** $p \in P_{low}$ **do**
- 13: $r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$
- 14: $SR = \alpha * \theta * r_i$
- 15: $number\ of\ neighbors \leftarrow KDTreeNeighborSearch(SR)$
- 16: **if** $number\ of\ neighbors > neighbors\ threshold$ **then**
- 17: $P_{denoised} \leftarrow p$
- 18: **else**
- 19: $P_{noise} \leftarrow p$
- 20: **end if**
- 21: **end for**

3.3. Ground and wall segmentation

Ground segmentation: In the second step of the method, ground points are identified using an adjusted version of the channel-based ground segmentation technique created by C. Phuong et al. [8]. This method analyses the denoised point cloud $P_{denoised}$ based on each of the vertical segments. For example, the LiDAR used in this research has a 360° horizontal field of view and a vertical resolution of 0.4° which means that the ground segmentation technique will assess $\frac{360^\circ}{0.4^\circ} = 900$ vertical segments. These vertical segments can be obtained by grouping points based on similar timestamps or by dividing the points using a certain angular resolution. Next, an elevation angle α , height h and *distance* for each of the consecutive point pairs within a vertical segment are determined. In the original work, the elevation angle α , may not exceed a certain α_{max} threshold. Also, if the height between two consecutive points exceeds a certain h_{min} threshold, the second point is considered a non-ground point. Finally, the *distance* between the sensor and point $i + 1$ has to exceed the *distance* between the sensor and point i . This technique is illustrated in Figure 3.3a.

The new feature which is developed and added to the original technique involves adding a different distance requirement. Given a vertical segment with n points, the distance requirement used to be about requiring the distance between the sensor and point $i + 1$ to exceed the distance between the sensor and point i . With the new feature, the distance requirement is updated and says that for two consecutive points, the distance between the sensor and point $i + 1$ must be at least 30 mm larger than the distance between the sensor and point i . This requirement is added because, as illustrated in Figure 3.3b, smoke can create a false positive point (marked in red) which is seen as noise by the original method because the angle it creates exceeds the maximum allowed threshold value. However, the second false positive point (marked in orange) caused by smoke does not exceed the threshold angle and is thus included as a false positive. In smoke the distance between such points is often very small and via empirical testing, a minimum difference in distance between the LiDAR and subsequent points was found to be 30 mm . Adding this new feature significantly reduces the number of false positives (noise) caused by smoke within the segmented ground points. The resulting point cloud containing just the segmented ground points is called: P_{ground} . The technique is also described using pseudo-code by Algorithm 4 within the Appendix.

Wall segmentation: Using the same technique but with different values for the threshold parameters α_{max} and h_{min} , the wall points have been segmented which results in the following point cloud: $P_{wallseg}$.

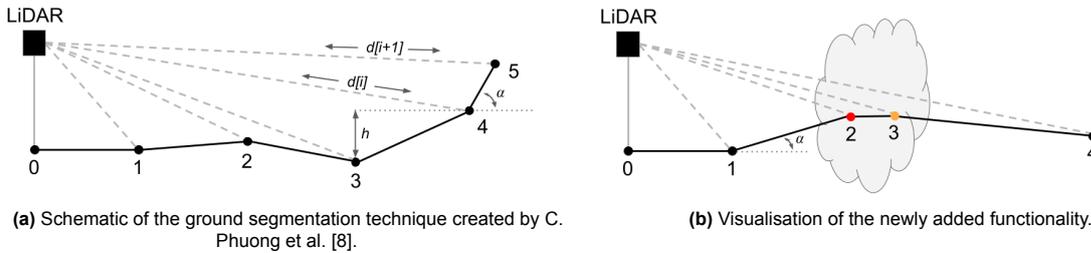


Figure 3.3: Illustrations of the original (left) and updated (right) channel-based ground segmentation techniques.

Clustering: After the wall points $P_{wallseg}$ have been identified, clustering of the different walls has to take place. This way, the different wall segments can be distinguished from each other. Which allows for independent analysis of the points per wall. The clustering method is implemented using the KDTree function from the PCL library [13] which is available for both Python and C++ scripts used in ROS. The KDtree takes the $P_{wallseg}$ as input after which the min, max cluster size and search radius have to be specified. The output of the KDTree clustering function will be referenced as $P_{clusters}$ and like the name suggests this output consists of multiple clusters, each of which can be accessed independently.

Creating the polar grid representation: Now that each wall in the point cloud is represented as a unique cluster of points, the point cloud is divided into a multitude of angular segments also known as a polar grid, this idea is visualised in Figure 3.4a. This is relevant because later on, this allows for effective reconstruction of ground surfaces between the LiDAR and the wall clusters in case ground points are missing. The precise angle of the segments and therefore number of angular segments can be set by the user. The smaller this angle, the more precise the ground surface reconstruction can become but also the higher its computational complexity.

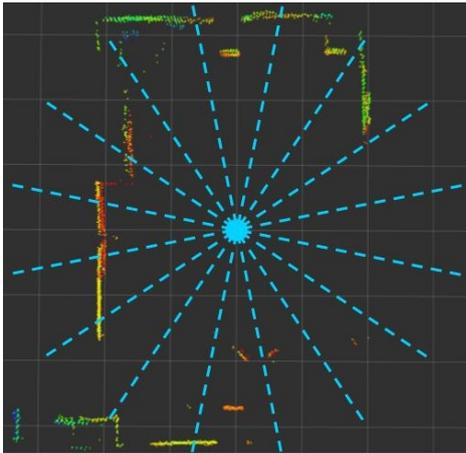
3.4. Ground surface reconstruction and merging results

The final step of this work's method involves reconstructing ground surfaces and merging them with the results from the ground segmentation step (Section 3.3). To do so, four independent steps have been taken which are described in this section. To enable a more in-depth understanding of these steps, a description of steps one and three using pseudo-code (Algorithm 5) can be found in Appendix A.1. Furthermore, step two is illustrated using Equation 3.1 and a description of step four in the form of pseudo-code (Algorithm 3) can be found within this section.

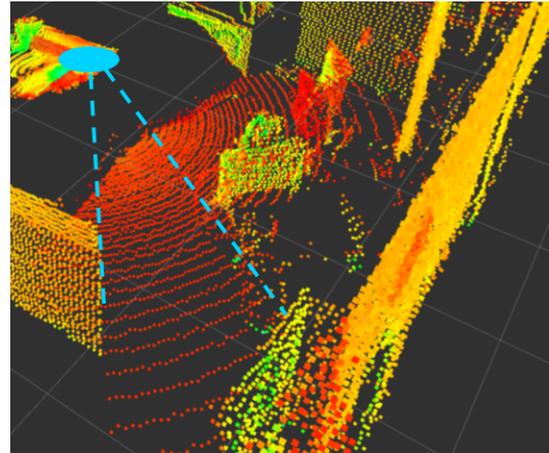
First, the lowest z and ring values are extracted from the clusters of wall points $P_{clusters}$ on a per-segment basis. This means that an array with the lowest z values and subsequently lowest ring values is obtained - and saved for later use - for each of the angular segments using the information of the corresponding wall point clusters.

As a *second* substep, the number of segmented ground points is compared on a per-segment basis with the expected number of ground points for that segment. If the actual number of segmented ground points is above a percentage (*gain*) of the expected number of ground points for that segment, the segmented ground points are stored inside an array of points which will be part of the actual output. When the threshold is not met for a specific segment, it means that very few ground points are present and therefore the ground points are reconstructed for that segment using a wall-based ground surface reconstruction function. The number of expected ground points per segment is calculated as follows:

$$expected\ points = \min(ring\ value) * \frac{segment\ angle^\circ}{horizontal\ resolution^\circ} * gain \quad (3.1)$$



(a) The creation of a polar grid in the xy -plane with user-defined angular segments.



(b) Visualisation with the blue striped lines indicating the segment on which the threshold is being applied.

Figure 3.4: Visualisation of the polar grid creation (left) and segmented ground points threshold (right).

The *third* step is the adjustment of z-values for segments which did not meet the threshold. This works by comparing per segment: the z-values of segment i with those of adjacent segments $i - 1$ and $i + 1$. Additionally, the sensor's floor height (which is known) is used to obtain the elevation angle between the lowest z-value assigned to segment i , this elevation angle shall remain below a certain threshold. This is done to prevent any unrealistic wall-based ground surface reconstruction. In case the z-values assigned to a segment which will be reconstructed differ more than a certain amount from those of adjacent segments, it will adjusted to the lowest z-value of the adjacent segment (but only after all segments have been assessed). If the assigned elevation angle obtained for this segment is (still) more than a 'global' threshold, the z-value assigned to the segment will be set to the sensor's floor height since that is the only actual ground truth available. In other cases, the segment's z-value is left untouched. A simplified illustration of this function is presented in Figure 3.5a, where the continuous black lines with letters represent the clusters of wall points.

The *fourth* step is the actual wall-based ground surface reconstruction using the adjusted z-values obtained in the previous step. By and large, this works as follows: First of all, the lowest ring value per wall segment is known and the assumption is made that the lowest ring value of a wall-cluster, has a high probability of representing the edge of the ground surface which intersects with the wall. Furthermore, the distance to the wall is known as well as the horizontal resolution of the LiDAR. The script which uses this information for the reconstruction of likely ground surface points between the sensor's ground point and wall segments is shown in Algorithm 2.

Algorithm 3 Pseudo-code of basic functioning of the wall-based ground surface reconstruction.

Input: $P_{segments_vec}, segment_angles, adjusted_lowest_z, origin_z, lowest_ring$

Output: $P_{reconstructed}$

```

1:  $P_{reconstructed} \leftarrow [...]$ 
2: for  $i, \in P_{clusters}$  do
3:    $distances \leftarrow r[1, \dots, n]; r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ 
4:    $R \leftarrow$  an array from 0 to  $\min(distances)$ , in  $lowest\_ring[i]$  number of steps
5:    $z\_values \leftarrow$  an array from  $origin_z$  to  $adjusted\_lowest\_z[i]$ , in  $lowest\_ring[i]$  number of steps
6:    $angles \leftarrow segment\_angles[i]$  to  $segment\_angles[i + 1]$ , in increments of  $0.4^\circ$ 
7:   for  $j \in R$  do
8:      $x_j \leftarrow r[j] * \cos(angles)$ 
9:      $y_j \leftarrow r[j] * \sin(angles)$ 
10:     $z_j \leftarrow$  an array with the same length as  $x$  and  $y$ , filled with:  $z\_values[i]$ 
11:   end for
12:    $P_{reconstructed} \leftarrow (x, y, z, intensity, t, r)$ 
13: end for

```

This function is also visualised in Figure 3.5b, where the pink dots represent the reconstructed ground surface points and the striped blue lines resemble the segment in which points are reconstructed. Furthermore, in the visualisation of Figure 3.5b, the lowest ring value found in the wall cluster for that angular segment is four, hence rings 1-3 are being reconstructed. Furthermore, the segment's angle is twice that of the LiDAR's vertical resolution and therefore three points are created on each of the added rings. In short, this surface reconstruction technique uses high-level knowledge about the intersection of walls with the ground surfaces in combination with LiDAR's characteristics such as the horizontal resolution and the available ring/channel information in the points to create realistic ground surface reconstructions.

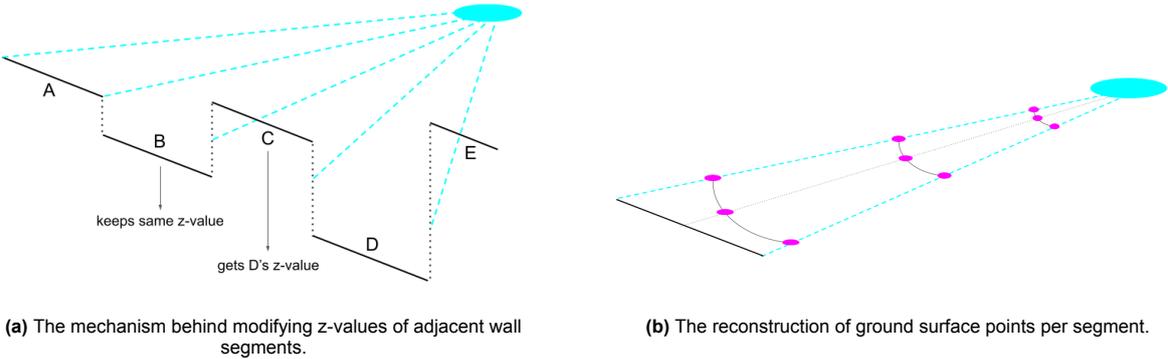


Figure 3.5: Visualisation of the z-value adjustment and ground surface reconstruction techniques.

Finally, the segmented ground points and reconstructed ground surfaces are merged. This is done a on per-segment basis meaning that when the number of segmented ground points within that section is above the expected threshold number, those points are shown within that segment. Otherwise, the wall-based reconstructed ground surface is shown.

4

Experiments

To prove the effectiveness and robustness of the developed method, experiments are conducted in two different dedicated training rooms at fire station 'Slotlaan' in Capelle aan den IJssel (the Netherlands): (1) inside a square-shaped *basic training room* and (2) within a more atypically shaped *complex training room*. These rooms are chosen because they represent realistic environments of operation for the firefighting robot. Furthermore, actual experiments with artificial smoke were allowed within these training rooms. This chapter first presents the equipment used for performing the experiments after which the conducted experiments in the basic training room and complex training room will be explained and visualised using dedicated sections.

4.1. Equipment used

Within these experiments, the following equipment is used:

- *LIDAR sensor*: In this research the Hesai QT128 mechanical LiDAR [5] is used. This Lidar has 128 channels (also known as rings) giving it a 105.2° vertical field of view, while also having a 360° horizontal field of view. Furthermore, this LiDAR has an effective range of 20 meters at 10% reflectivity and can operate in temperatures between $-40^\circ C$ and $85^\circ C$. Additional specifications can be found in Appendix A.4.
- *Smoke machine*: The Antari Z 1000 MKII DMX smoke machine [34] is used, which can output $283 m^3$ of (water-based) artificial smoke per minute. This high-volume output makes the smoke machine very suitable for quickly filling the training rooms with thick smoke.
- *Computer*: The point clouds obtained during the experiments are recorded as 'rosvbag' files via ROS Noetic [2] for later evaluation. The computer used during these experiments is the Lenovo Ideapad 5 laptop [35] with the Ubuntu 20.04 [4] operating system.

4.2. Experiments in the basic training room

The experimental setup used for the experiments performed in the basic training room is shown in Figure 4.1c. As can be seen in this figure, relatively few obstacles are present in the room, with the main ones being two black office chairs and a standing flipover board. Furthermore, the smoke machine is placed on the ground and the LiDAR sensor is placed on top of a white-coloured support. In this training room, six different tests are performed:

- *Without smoke:* First of all, without the presence of smoke, the LiDAR is placed in five different locations throughout the room as shown in Figure 4.1b. Positions 1-4 make up the first four tests and are used to capture the environment from very different locations so that later, these point clouds can be merged to establish the most complete point cloud of the ground surface. Next, a fifth position is used to obtain a reference point cloud from a more centralised position in the room.
- *With smoke:* Finally, in the presence of heavy smoke, the room is captured with the LiDAR located in position five which makes this the sixth test.



(a) The basic training room, showing a black office chair, smoke machine and flipover board.



(b) The five different positions from which point clouds have been recorded in the basic training room.



(c) Visualisation of the experimental setup in position four, showing the LiDAR sensor on its support as well as the smoke machine.

Figure 4.1: Overview of the basic training room from multiple angles, including the experimental setup.

4.3. Experiments in the complex training room

Before performing the experiments in the basic training room, experiments were conducted in the complex training room. This is done to see how well the developed method can deal with environments which contain more and differently shaped objects. The visualisation of this room, including the experimental setup can be seen in Figure 4.2. Within this figure, it can be observed how the LiDAR is positioned on top of a sofa, with the smoke machine being positioned on the ground next to a wall. Additionally, this room contains a wide range of different objects, including, but not limited to: two red chairs, a sofa, a display cabinet, two smaller cupboards, a desk and an office chair. In this experiment, the position of the LiDAR remains the same and therefore, two tests are performed in this training room:

- *Without smoke*: First of all, a test is performed without the presence of smoke. During this test, the LiDAR captures the room from the position on the sofa as shown in Figure 4.2a. This is done to obtain a ground truth of what the (ground) points look like without any degrading effects on the point clouds caused by smoke.
- *With smoke*: Next, the room is captured by the LiDAR in the presence of heavy smoke which makes this the second and final test in the complex training room.



(a) Visualisation of the LiDAR's position within the complex training room.



(b) The LiDAR's position within the complex training room, shown from a different angle.



(c) The experimental setup with the LiDAR on the sofa and smoke machine positioned on the ground.



(d) The complex training room in very light smoke.

Figure 4.2: Overview of the complex training room from multiple angles, including the experimental setup.

5

Results

In this chapter, the results obtained from the experiments as described in Chapter 4 are presented. These results are all obtained by processing the point clouds from the performed experiments using a Jetson Orin Nano single-board computer [3]. This chapter first presents the qualitative results followed by the quantitative results, each of which will be presented in dedicated sections. The qualitative results consist of a visual comparison between the LiDAR's point clouds obtained in different conditions and the output created by the developed method. Finally, the quantitative results consist of a grid-based area and height comparison between the ground points of the point clouds obtained directly using the LiDAR and the output created by the developed method. Additionally, a break-down of the developed method's computational speed is presented.

5.1. Qualitative results

The qualitative type of results have been obtained since they form an effective way of assessing to what extent the method performs as expected. As described in Chapter 4, to obtain insightful qualitative results, the following four scenarios have been assessed:

1. The basic training room without smoke
2. The basic training room filled with heavy smoke
3. The complex training room without smoke
4. The complex training room filled with heavy smoke

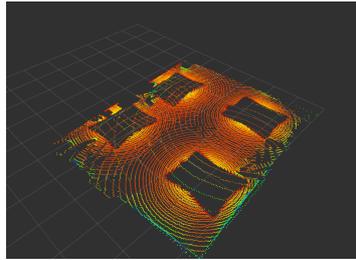
Please note: the parameters used by the method as well as enlarged pictures of the results shown in this section can be found in Appendix A.2 and A.3 respectively. Furthermore, besides the reconstructed ground surface, sub-figures 5.2c, 5.3c and 5.4c, all show some outlines of wall points as well. This is done for visualisation purposes only, since it helps to judge how accurately the surface is reconstructed.

1. Basic training room without smoke

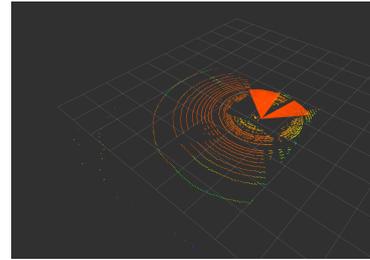
To assess the performance in a relatively straightforward environment, tests were performed in the basic training room without any smoke. In Figure 5.1, the results obtained using a camera (a), merging the ground points of the LiDAR's point clouds from position 1-4 (b) and the output of the ground surface reconstruction method based on the point cloud from position 3 (c) are shown. The merging of the point clouds from four different locations is done to obtain a ground truth point cloud of the ground surface.



(a) Picture of the basic training room from a similar perspective as the point clouds.



(b) The ground truth ground points obtained by merging point clouds from four locations.



(c) The output point cloud obtained using the ground surface reconstruction method.

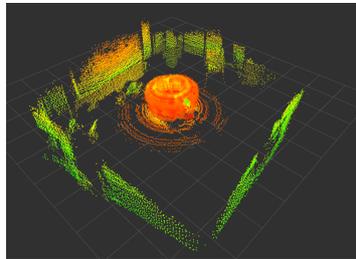
Figure 5.1: Visualisation of the different results obtained in the basic training room without smoke.

2. Basic training room filled with heavy smoke

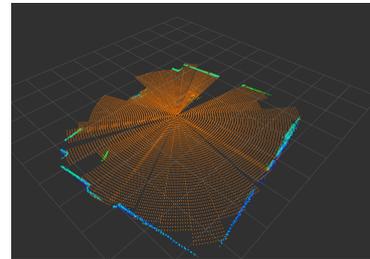
Next, the results obtained in the basic training room filled with heavy smoke are shown. This is done as a way to show the robustness of the method in challenging conditions caused by smoke. Figure 5.2 shows the results obtained using a camera (a), the LiDAR's point cloud from position 5 (b) and the output of the ground surface reconstruction method based on the point cloud from position 5 (c).



(a) Picture of the basic training room filled with smoke.



(b) The point cloud obtained in the basic training room filled with smoke.



(c) The output point cloud obtained using the ground surface reconstruction method.

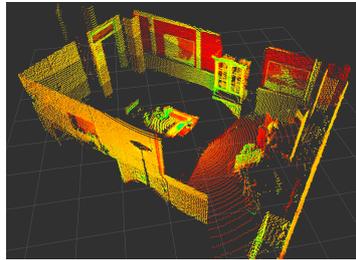
Figure 5.2: Visualisation of the different results obtained in the basic training room filled with heavy smoke.

3. Complex training room without smoke

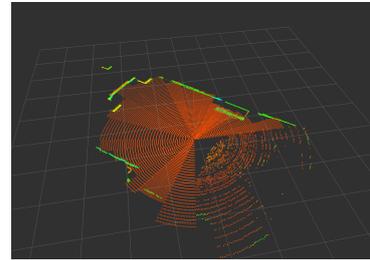
Next to the basic training room, results obtained in the complex training room are shown. These results are relevant since they show how the developed method deals with more complex environments. In Figure 5.3, the results obtained using a camera (a), the LiDAR's point cloud (b) and the ground surface reconstruction method (c) are shown.



(a) Picture of the complex training room from a similar perspective as the point clouds.



(b) The point cloud obtained in the complex training room without heavy smoke.

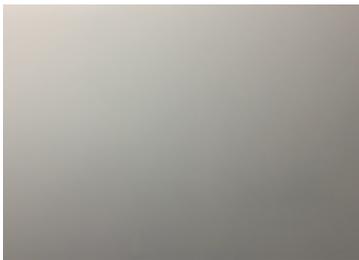


(c) The output point cloud obtained using the ground surface reconstruction method.

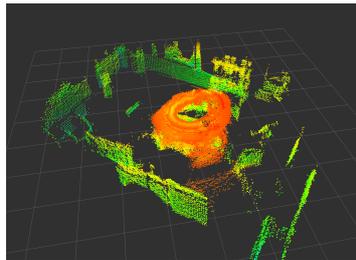
Figure 5.3: Visualisation of the different results obtained in the complex training room without smoke.

4. Complex training room filled with heavy smoke

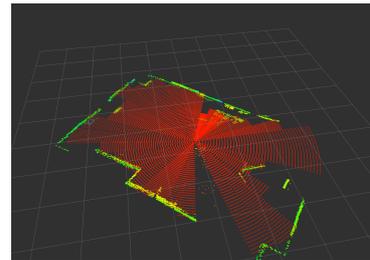
Finally, the results of the complex training room filled with heavy smoke were obtained. Figure 5.4 shows the results obtained using a camera (a), the LiDAR's point cloud (b) and the ground surface reconstruction technique (c).



(a) Picture of the complex training room filled with smoke.



(b) The point cloud obtained in the complex training room filled with heavy smoke.



(c) The output point cloud obtained using the ground surface reconstruction method.

Figure 5.4: Visualisation of the different results obtained in the complex training room filled with heavy smoke.

5.2. Quantitative results

In this section, the quantitative results obtained using point clouds from the "basic training room" are presented. Only this "basic training room" is used for the quantitative evaluation since the point clouds obtained in the "complex training room" contain too many false negative ground points (even without smoke) resulting from large amounts of the LiDAR's emitted laser pulses being blocked by objects.

The many blocked points added significant uncertainty about the actual area of the ground surface, hence making it unrealistic to obtain an accurate and usable ground truth point cloud of the actual ground surface. Consequently, this would give an inaccurate and misleading comparison between the occupancy grids obtained using the method's output and those obtained from the point clouds.

The quantitative results in the section will be presented in the following order. First, the method's ability to reconstruct the ground surface is assessed using occupancy grids obtained from point clouds in different scenarios. Next, the height of the reconstructed surface is compared to that of the ground truth based on the occupancy grids obtained earlier. Finally, a breakdown of the method's computational speed is presented.

5.2.1. Occupancy grid evaluation

To evaluate how well the developed method reconstructs the ground surface, 2D (binary) occupancy grids have been created which are based on different types of point clouds as input. To create these occupancy grids from the LiDAR's point clouds, z-value thresholds were used to extract all the ground points present in the point cloud. Furthermore, the xy-plane is divided into a fixed number of cells and if one or more points lie within the xy-limits of a cell, that cell's area will be coloured black in the grid. In case, no ground points were present within a cell, the area is coloured white. These obtained occupancy grids are presented in Figure 5.5.

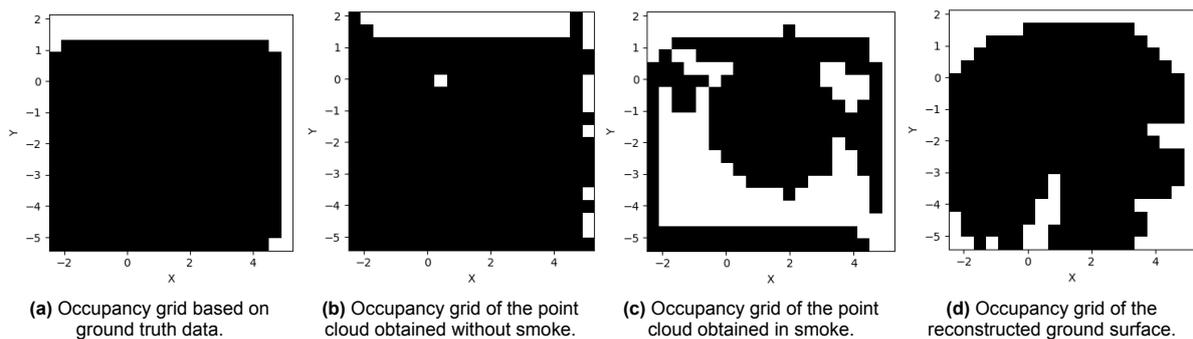


Figure 5.5: Visualisation of the four different binary occupancy grids.

The occupancy grids presented in Figure 5.5 are obtained in the following manner:

- (a) *Ground truth (ideal) occupancy grid*: This occupancy grid uses a point cloud from the LiDAR without the presence of any smoke as a basis. To create an occupancy grid which forms the most ideal representation of the actual ground surface, false positive points and outliers were identified and removed from this point cloud. Furthermore, pictures of the room have been analysed to remove any additional false positives and to fill in possible false negative points manually.
- (b) *Occupancy grid of the LiDAR's point cloud without smoke*: This occupancy grid is obtained directly from the LiDAR in an environment without any smoke. Only the false negatives created by the support on which the LiDAR stands have been filled in to create a fair comparison with the result from the ground surface reconstruction method.
- (c) *Occupancy grid of the LiDAR's point cloud in smoke*: This occupancy grid is obtained directly from the LiDAR in an environment filled with smoke. Only the false negatives created by the support on which the LiDAR stands have been filled in to create a fair comparison with the result from the ground surface reconstruction method.
- (d) *Occupancy grid of reconstructed ground surface in smoke*: This occupancy grid is obtained using the ground surface reconstruction method developed in this thesis. The input used by this method was the same as used for the "occupancy grid in smoke", namely: a point cloud obtained in an environment filled with smoke.

By overlapping the different occupancy grids with the "ground truth (ideal) occupancy grid", it was determined what cells make up the true positives, true negatives, false positives and false negatives for each of the different occupancy grids. The resulting grids will be referred to as "classification grids" (Figure 5.6). Using these grids, performance metrics such as precision, accuracy, recall, F1 and IoU have been obtained. These metrics are presented in Table 5.1 and indicate how well the developed method reconstructs the ground surface from a purely xy-plane perspective.

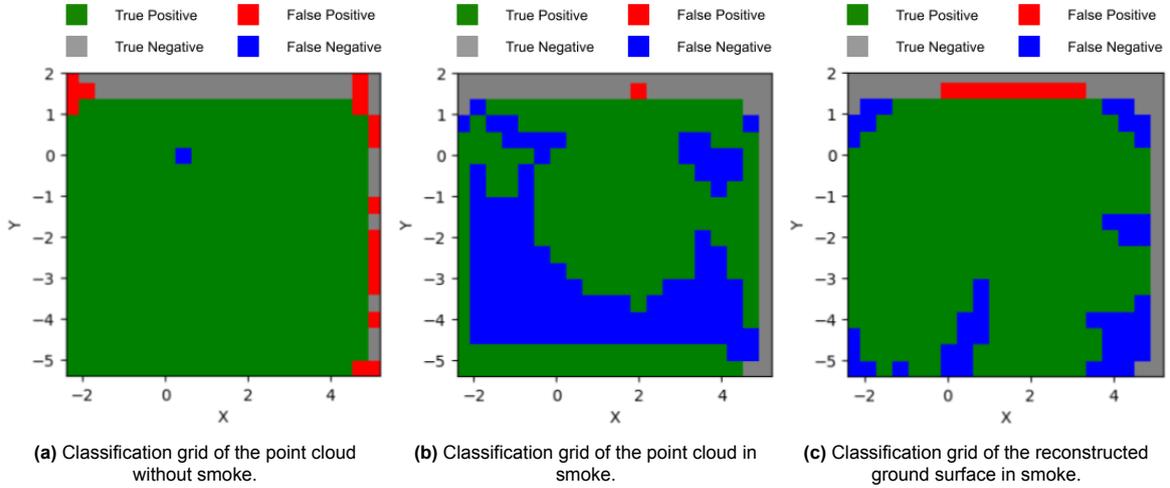


Figure 5.6: Visualisation of the three different classification grids.

Classification grid	Precision	Accuracy	Recall	F1	IoU
Point cloud without smoke	0.95	0.95	0.99	0.97	0.95
Point cloud in smoke	0.99	0.69	0.63	0.77	0.63
Reconstructed ground surface in smoke	0.97	0.86	0.86	0.91	0.84

Table 5.1: Table showing the performance metrics obtained using the three classification grids.

5.2.2. Reconstructed height evaluation

Next to the method's ability to reconstruct the ground surface from a xy-plane perspective, the ability to estimate the correct height of a ground surface has also been assessed on a per-cell basis for all the true positive cells. This height evaluation is done for the classification grid which is obtained by overlapping the "ground truth (ideal) occupancy grid" with the "occupancy grid of the reconstructed ground surface in smoke", since this comparison most accurately resembles the difference in height of the reconstructed ground surface from the actual ground surface. The obtained results are visualised in Figure 5.7 with sub-figure a, showing the combined height differences for all true positive cells in the form of a boxplot and sub-figure b, showing the height difference on a per-cell basis using a heatmap. In this heatmap, cells which were not true positives were given a 0.0 mm height difference. The resulting mean and median height differences for all true positive cells are -27 mm and -23 mm as indicated by the green triangle and orange line respectively in Figure 5.7a.

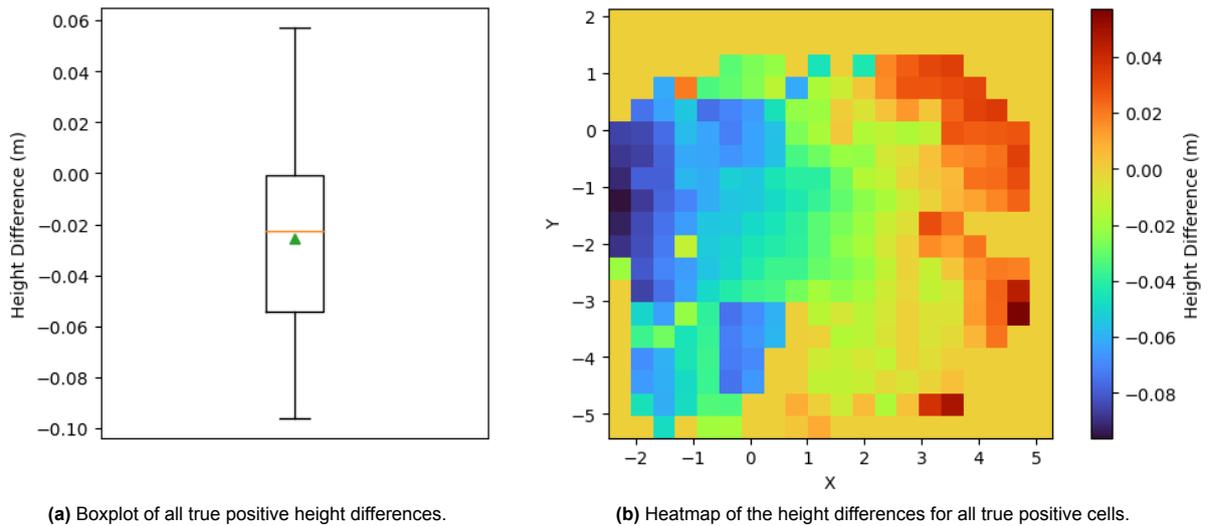


Figure 5.7: Visualisation of height differences between the ground truth and reconstructed surfaces.

5.2.3. Computational speed

In this subsection, the total computational speed of the developed method is presented together with the speed of the components on which the method depends. These results are the 'net' speeds and contributions to the total runtime, meaning that if for example, component B uses component A's output, the runtime of A is subtracted from that of B. This way the 'net' computational speed of component B, excluding the impact of component A on which it does depend, is obtained. An overview of these results is presented in Table 5.2:

Component	Script type	Runtime (s)	Hz	Contibution to total runtime
LiDAR	N/a	0.10	10.0	4.0%
Denoiser	Python	1.48	0.7	59.2%
Ground segmentation	C++	0.01	196.9	0.2%
Wall segmentation	C++	0.01	196.9	0.2%
Ground surface reconstruction	C++	0.91	1.1	36.4%
Full method	C++ & Python	2.50	0.40	100%

Table 5.2: Overview of the computational speed per main component of the developed method.

6

Discussion

This chapter will focus on the main limitations of this work and in doing so, also give an interpretation of the obtained qualitative and quantitative results. To effectively discuss these limitations, they will be discussed per stage of this research. Therefore this chapter covers the limitations concerning *the selection of techniques*, followed by *the developed method* and finally, *the conducted experiments*. In addition to the limitations, this chapter will end with a section dedicated to this work's implications.

6.1. The selection of techniques

When reflecting on the selection of techniques used within this method, the first limitation to be remarked concerns the trade-off between speed and accuracy which often had to be made when selecting one of the considered techniques for implementation in the method. This was the case for all three of the method's main techniques namely: point cloud denoising, ground as well as wall segmentation and ground surface reconstruction. The second limitation concerned the large inconsistency in datasets, required hardware and programming languages used among these techniques. This makes it very hard to create fair comparisons between the different techniques, hence making it hard to select the best one. Especially because many techniques did not directly compare themselves to other state-of-the-art techniques. The third and final limitation in this context concerns the implementation difficulties of the different techniques. This was especially true for the machine learning-based point cloud denoising, ground segmentation and ground reconstruction techniques. The requirements of these methods, in terms of hardware (such as NVIDIA GPUs) and large amounts of labelled point clouds, exceeded the available resources of this project. Therefore, they could not be successfully implemented. This is unfortunate because when trained on relevant data, these techniques may lead to significant results. Furthermore, many of the considered techniques had only very limited documentation available often without actual code.

6.2. The developed method

Next to the process of selecting the techniques, the developed method itself also has several noteworthy limitations. In order to thoroughly discuss these limitations, they will be presented per main component of the developed method.

Point cloud denoising: As shown by the results in Section 3.2, the point cloud denoising technique [7] used in the developed method is an effective technique for reducing the number of false positives in the point cloud while keeping valuable true positive ground points. However, it has two main limitations. First of all, it filters out some true positive points as well. This can be explained because the technique used is designed for sparser false positive point clouds (e.g. those from snow and rain). In this work, the technique's parameters were tuned on the observation that false positives caused by artificial smoke create denser (instead of sparser) false positives. Unfortunately, this has the consequence that some true positive points are filtered out as well. The second and final limitation is the relatively low processing speed of this method with a runtime of 1.48 seconds per point cloud (Table 5.2). This has the additional downside of limiting the speed of the overall method since it depends on the output of this point cloud denoising technique. The total contribution of this technique to the overall method's runtime is 59%, as is presented in Section 5.2.

Ground and wall segmentation: Within the developed method, the technique used for both ground and wall segmentation is based on the work of C. Phuong et al. [8]. This technique is known to be one of the fastest ones available without the need to train on large amounts of data. By assessing the results in Figure 2.6 of Section 2.3, it can be observed how the technique effectively segments the ground, albeit together with some false positive ground points. On the flip side, this segmentation technique also comes with note-worthy limitations. The first one is that this technique requires three parameters to be correctly chosen. Doing so comes down to a trade-off between more false positive points (and more true positive ground points) or fewer false positive points (and fewer true positive ground points). The next main limitation of this technique manifests itself during the segmentation of wall points. More specifically, if the point clouds are obtained in rooms with smoke and/or a lot of windows. That is because in such scenarios the points representing those walls are more noisy and inaccurate compared to point cloud representations of the actual walls. This may create misleading results later on when per angular segment, the wall points closest to the LiDAR as well as the lowest wall point are used to reconstruct the ground surface. This effect is also reflected in the quantitative results of the ground surface reconstruction method where more false positive and false negative cells are visible in Figure 5.6c compared to the ground truth scenario of Figure 5.6a.

Ground surface reconstruction: The custom ground surface reconstruction technique, aims to reconstruct the ground points at their expected locations on the ground surface, disregarding possible interference from smoke or other objects which may reflect or absorb these points. When closely assessing and comparing Figure 5.2b with c, it can be seen how the ground surface of the basic training room is fairly accurately being reconstructed. The same comparison can be made by looking at Figure 5.4b and c. These observations are underlined by the satisfactory results including, but not limited to, an IoU of 84% - compared to 64% of the ground surface covered by the point cloud in smoke - and a mean height difference of -27 mm compared to the ground truth. At the same time, the technique also introduces several limitations given the assumptions made within the technique, which can differ from reality.

Firstly, per segment: the technique assumes the number of channels it has to reproduce based on the lowest ring value found within those wall points for that segment. In practice, the identified wall points often do not contain the wall points which intersect with the ground surface. This is because, especially in smoke, not all wall points are correctly reflected back towards the LiDAR. This has two consequences: first of all, too many channels and consequently points can get reconstructed which results in very dense reconstructions of the ground surfaces (Figure 5.1c). Secondly, this overestimated the number of expected ground points which influences the merging of results (more on this in the *Merging the results* bullet). Thirdly, the method assumes that per segment: the length of the reconstructed surface equals that of the smallest distance from the wall points to the sensor. While this is a reasonable estimate, reality often requires the length of the reconstructed surface per segment to be adaptive to the detected walls. The false negative cells and false positive cells which are visible

in Figure 5.6c can largely be explained by this effect. Finally, the method introduces a limitation by reconstructing the ground surface linearly in an ascending or descending way from the sensor's floor height towards the lowest identified wall point for a given angular segment. This effect can also be observed when looking at the quantitative results of Figure 5.7b. In reality, the height of the ground surface may for example be curved, which weakens the robustness of this reconstruction approach.

Merging the results: The developed method chooses on a per-segment basis, whether to display the segmented ground points or to reconstruct and display a ground surface. This choice is made based on whether a given segment contains a certain minimum fraction of the amount of expected ground points. This comes with two limitations: First of all, this decision-making approach can lead to a sub-optimal output performance in cases where only a part of the ground points are available but still those are shown as output instead of a reconstructed ground surface. The rationale for doing so is that the segmented ground points (if available) are often still the most accurate representation of the actual ground surface. Secondly, the number of expected ground points is often overestimated since the segmented wall points within thick smoke may not accurately resemble the wall points which are close to intersecting with the ground surface. In those cases, the identified lowest ring value for the reconstruction of a certain segment is often too high which leads to an overestimation of the expected number of ground points based on Equation 3.1.

6.3. The conducted experiments

The experiments performed during this research also introduce limitations which shall be taken into consideration. First, these experiments have been conducted using artificial smoke, which consists of water-based particles instead of the soot particles found in real smoke. This creates uncertainty about how a LiDAR and the developed method will perform in actual smoke given the potentially differing effects that such particles may have on the reflection and absorption of emitted LiDAR pulses. Finally, the method has been tested in just two different rooms, both consisting of static environments. This limits the proof of the developed method's robustness given the uncertainty about the different, possibly dynamic environments.

6.4. Research implications

To be best of my knowledge there is an absence of directly relevant scientific literature which deals with the main research question of this work. For that reason, this work can be seen as a valuable contribution to the scientific literature related to surface reconstruction based on point clouds. More specifically, the method developed in this work forms a (modular) foundation for fast and accurate ground surface reconstruction without relying on machine learning techniques which may require large amounts of labelled point clouds.

The developed method in this work will provide the operator and firefighters with additional spatial information about smoke-filled environments. This additional spatial information can be helpful information for the operator to accurately and safely control the navigation of Brutus. Additionally, this information can complement the fire brigade's information-gathering procedure concerning environments filled with smoke. Next to firefighting robots - with some minor modifications - the developed method in this research may also be used for indoor drones equipped with a LiDAR sensor since these types of drones are already being used by the fire department.

7

Conclusion

This work aims to develop a method which can leverage the limited information from degraded point clouds obtained in environments filled with smoke, to reconstruct indoor ground surfaces in real-time. This goal comes from the desire of the fire department to use the limited information from degraded point clouds to aid the operator - and consequently firefighters - with additional spatial information about smoke-filled environments. This can be an important step towards the fire department's longer-term ambition of enabling autonomous navigation on Brutus, even in environments filled with smoke.

Following the desire presented above, the following main research question was obtained: *How to enable real-time reconstruction of indoor ground surfaces in occluded environments filled with smoke based on point clouds obtained using LiDAR?* The overall answer is: by constructing a method composed of novel and existing techniques which tackles the independent sub-challenges introduced by smoke on the point clouds obtained with LiDAR. These techniques include point cloud denoising, wall and ground segmentation, and ground surface reconstruction. The effectiveness of this approach is underlined by the satisfactory results obtained in the research which include, but are not limited to, an IoU of 84% and a mean height difference of -27 mm compared to the ground truth. This main research question will be further answered in the context of the obtained sub-questions. Additionally, it has to be remarked that the conclusions presented here are based on artificial smoke instead of actual smoke.

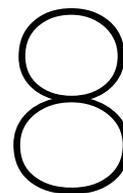
Continuing with the first sub-question: *How does smoke affect the visibility of ground points in a point cloud obtained with LiDAR?* This question is answered by Section 2.1 of this report, which found two main effects caused by smoke on point clouds obtained with LiDAR: Firstly, ground points are among the first points to be absorbed by smoke given their lower intensity values compared to other points. Secondly, smoke creates many low-intensity, false positive points around the LiDAR sensor. Both effects degrade the quality of the point cloud and consequently the perception within environments filled with smoke. Especially the ground points suffer from this effect given their low intensity values, which underlines the importance of the second sub-question.

The second and final sub-question is as follows: *What method can be developed to tackle the main research question?* As stated at the beginning of this chapter, the overall answer to this question is a method which effectively tackles the independent sub-challenges introduced to the point clouds obtained in environments filled with smoke. Following this very broad answer, the key takeaways from the results obtained using the developed method give a more nuanced answer to this second sub-question.

The most important takeaways per component of the developed can be summarised as follows:

1. *Point cloud denoising*: As shown by the results in Section 3.2, the point cloud denoising technique used is an effective technique for reducing the number of false positives in the point cloud while keeping valuable true positive ground points. Therefore, this is an effective first step towards enabling ground surface reconstruction. The downside of this technique is the relatively poor runtime, which contributes to 59% of the method's total runtime, as is presented in Section 5.2.
2. *Ground segmentation*: Based on the results presented in Figure 2.6 of Section 2.3, it can be observed and concluded that the implemented techniques effectively segment the ground with only a limited number of false positive ground points.
3. *Wall segmentation*: The wall points are reasonably well segmented when looking at the qualitative results (comparing Figure 5.2b and c). At the same time, there are some inaccuracies, of which false positives have the most negative impact on the ground surface reconstruction performance.
4. *Ground surface reconstruction*: This technique significantly reduces the number of false negatives and increases the number of true positives compared to the point cloud obtained in smoke. This can be observed by comparing Figure 5.6b and Figure 5.6c and the obtained IoU of 84% versus 63% for the ground truth surface covered by the point cloud in smoke. This underlines the added value of the developed technique. Despite these promising results, it remains the component where the most significant advancements can be made in the context of accurate and real-time ground surface reconstruction. As stated in the discussion, this mainly concerns the equation which determines whether to display segmented ground points or reconstructed ground points, the processing speed and finally, having adaptive surface lengths and heights.

In short, when looking at the results obtained using the method developed in this work, the answer to the research question can be summarised as follows: The reconstruction of indoor ground surfaces in occluded environments filled with smoke based on point clouds obtained using LiDAR can be realised using a method consisting of different techniques which tackle the independent sub-challenges for such environments. However, doing this in real-time will require additional processing power or an accelerated way of point cloud denoising and ground surface reconstruction.



Future work

Concerning the performance of the developed method, the most important drawbacks to be tackled concern four categories: *implementation of techniques*, *parameters*, *ground surface reconstruction* and *speed*.

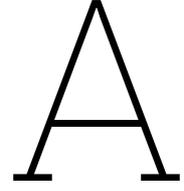
- *Implementation of techniques*: As described within the 'Selection of techniques' sections of Chapter 2, several state-of-the-art techniques have not been implemented due to practical problems encountered while trying to install, recreate, run or implement these techniques. This was especially true for the machine learning-based techniques. In case additional efforts do succeed in implementing improved solutions for independent challenges introduced by the smoke, this can and is likely to improve the output of the developed method.
- *Parameters*: The point cloud denoising, ground and wall point segmentation and ground surface reconstruction techniques all require multiple parameters to be correctly set. The performance and robustness of the developed method can be significantly improved by enhancing the techniques so these parameters no longer have to be manually chosen.
- *Ground surface reconstruction*: The technique developed to enable this type of reconstruction can still be improved. The primary areas of improvement are how per segment, the lowest z-value and ring-value are extracted as well as finding an alternative for linearly reconstructing points between such extracted points and the sensor's floor height. Finding additional high-level knowledge which can complement or even replace the current high-level reasoning may result in notable performance gains and a more robust method.
- *Speed*: The drawback concerning speed comes from the point cloud denoising technique and ground surface reconstruction technique used. Besides simply upgrading the available computational resources, the point cloud denoising technique as well as the ground surface reconstruction technique can be developed in a more efficient manner which may significantly increase the computational speed. Additionally, the point cloud denoising method may be converted into C++ to improve its processing speed.

Next to these drawbacks, another area of future work would be to combine object detection with this developed method in order to obtain traversable maps of the environment (even in the presence of smoke). These traversable maps can eventually be used by Brutus to know precisely where it can and cannot drive, hence being an important step towards enabling autonomous navigation for Brutus.

References

- [1] Fernando Bot. Mar. 2023. URL: <https://fernandofotografie.nl/2023/03/26/experience-day-in-rotterdam-bij-de-kuip/>.
- [2] *Robotic Operating System*. May 2018. URL: <https://www.ros.org> (visited on 01/19/2024).
- [3] *NVIDIA Jetson Orin Nano*. URL: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/> (visited on 02/15/2023).
- [4] *Ubuntu*. URL: <https://releases.ubuntu.com/focal/> (visited on 02/15/2024).
- [5] *QT128 Ultra-Wide View Short-Range Lidar | HESAI Technology*. zh-CN. URL: <https://www.hesaitech.com/product/qt128/> (visited on 01/19/2024).
- [6] Remco F. Gerritsen. *RFGFotografie*. Dutch. Social media. May 2021. URL: <https://twitter.com/RFGFotografie/status/1398932385081745408> (visited on 02/16/2024).
- [7] Minh-Hai Le et al. "An Adaptive Group of Density Outlier Removal Filter: Snow Particle Removal from LiDAR Data". en. In: *Electronics* 11.19 (Jan. 2022). Number: 19 Publisher: Multidisciplinary Digital Publishing Institute, p. 2993. ISSN: 2079-9292. DOI: 10.3390/electronics11192993. URL: <https://www.mdpi.com/2079-9292/11/19/2993> (visited on 01/19/2024).
- [8] Phuong Chu et al. "A Fast Ground Segmentation Method for 3D Point Cloud." In: *J. Inf. Process. Syst.* 13.3 (2017), pp. 491–499.
- [9] Valentin Vierhub-Lorenz et al. "Development of a LiDAR system for low visibility conditions". In: *Optical Measurement Systems for Industrial Inspection XIII*. Vol. 12618. SPIE, 2023, pp. 314–322.
- [10] Liangliang Nan and Peter Wonka. "PolyFit: Polygonal Surface Reconstruction from Point Clouds". In: *ICCV* (2017).
- [11] Jiali Han et al. "Vectorized indoor surface reconstruction from 3D point cloud with multistep 2D optimization". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 177 (2021), pp. 57–74. ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2021.04.019>. URL: <https://www.sciencedirect.com/science/article/pii/S0924271621001222>.
- [12] You Li and Javier Ibanez-Guzman. "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems". In: *IEEE Signal Processing Magazine* 37.4 (2020). Publisher: IEEE, pp. 50–61.
- [13] Radu Bogdan Rusu and Steve Cousins. "3D is here: Point Cloud Library (PCL)". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, May 2011.
- [14] Minh-Hai Le, Ching-Hwa Cheng, and Don-Gey Liu. "An Efficient Adaptive Noise Removal Filter on Range Images for LiDAR Point Clouds". In: *Electronics* 12.9 (2023). Publisher: MDPI, p. 2150.
- [15] Robin Heinzler et al. "CNN-Based Lidar Point Cloud De-Noising in Adverse Weather". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2514–2521. DOI: 10.1109/LRA.2020.2972865.
- [16] Florian Piewak et al. "Boosting LiDAR-based Semantic Labeling by Cross-Modal Training Data Generation". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. Sept. 2018.
- [17] Alvari Seppänen, Risto Ojala, and Kari Tammi. "4DenoiseNet: Adverse Weather Denoising From Adjacent Point Clouds". In: *IEEE Robotics and Automation Letters* 8.1 (2023), pp. 456–463. DOI: 10.1109/LRA.2022.3227863.
- [18] Nicholas Charron, Stephen Phillips, and Steven L. Waslander. "De-noising of Lidar Point Clouds Corrupted by Snowfall". In: *2018 15th Conference on Computer and Robot Vision (CRV)*. May 2018, pp. 254–261. DOI: 10.1109/CRV.2018.00043. URL: <https://ieeexplore.ieee.org/document/8575761> (visited on 01/19/2024).

- [19] Akhil Kurup and Jeremy Bos. *DSOR: A Scalable Statistical Filter for Removing Falling Snow from LiDAR Point Clouds in Severe Winter Weather*. arXiv:2109.07078 [cs]. Oct. 2021. DOI: 10.48550/arXiv.2109.07078. URL: <http://arxiv.org/abs/2109.07078> (visited on 01/19/2024).
- [20] Ji-II Park, Jihyuk Park, and Kyung-Soo Kim. “Fast and Accurate Desnowing Algorithm for LiDAR Point Clouds”. In: *IEEE Access* 8 (2020). Conference Name: IEEE Access, pp. 160202–160212. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3020266. URL: <https://ieeexplore.ieee.org/document/9180326> (visited on 01/19/2024).
- [21] Hyungtae Lim, Minho Oh, and Hyun Myung. “Patchwork: Concentric Zone-Based Region-Wise Ground Segmentation With Ground Likelihood Estimation Using a 3D LiDAR Sensor”. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6458–6465. DOI: 10.1109/LRA.2021.3093009.
- [22] Tiago Gomes et al. “A Survey on Ground Segmentation Methods for Automotive LiDAR Sensors”. en. In: *Sensors* 23.2 (Jan. 2023). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 601. ISSN: 1424-8220. DOI: 10.3390/s23020601. URL: <https://www.mdpi.com/1424-8220/23/2/601> (visited on 01/19/2024).
- [23] Seungjae Lee, Hyungtae Lim, and Hyun Myung. *Patchwork++: Fast and Robust Ground Segmentation Solving Partial Under-Segmentation Using 3D Point Cloud*. arXiv:2207.11919 [cs]. Sept. 2022. DOI: 10.48550/arXiv.2207.11919. URL: <http://arxiv.org/abs/2207.11919> (visited on 01/19/2024).
- [24] J. Behley et al. “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences”. In: *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*. 2019.
- [25] Anshul Paigwar et al. “GndNet: Fast ground plane estimation and point cloud segmentation for autonomous vehicles”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2150–2156.
- [26] Charles R. Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [27] Andres Milioto et al. “Rangenet++: Fast and accurate lidar semantic segmentation”. In: *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019, pp. 4213–4220.
- [28] *sklearn.linear_model.RANSACRegressor*. en. URL: https://scikit-learn/stable/modules/generated/sklearn.linear_model.RANSACRegressor.html (visited on 01/19/2024).
- [29] Martin A. Fischler and Robert C. Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: <https://dl.acm.org/doi/10.1145/358669.358692> (visited on 01/19/2024).
- [30] Joseph W Starr and BY Lattimer. “Evaluation of navigation sensors in fire smoke environments”. In: *Fire Technology* 50 (2014). Publisher: Springer, pp. 1459–1481.
- [31] Xuemeng Yang et al. “Semantic segmentation-assisted scene completion for lidar point clouds”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3555–3562.
- [32] *Leaderboard - 3D Semantic Scene Completion on SemanticKITTI*. 2023. URL: <https://paperswithcode.com/sota/3d-semantic-scene-completion-on-semantickitti>.
- [33] Zhaoyang Xia et al. “SCPNet: Semantic Scene Completion on Point Cloud”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17642–17651.
- [34] *Antari Z 1000 MKII DMX smoke machine*. URL: <https://www.bax-shop.nl/rookmachines/antari-z-1000-mkii-dmx-rookmachine-controller>.
- [35] *Lenovo Ideapad 5-15ARE05 Laptop*. URL: <https://pcsupport.lenovo.com/nl/nl/products/laptops-and-netbooks/5-series/ideapad-5-15are05> (visited on 02/19/2024).



Appendix

A.1. Additional pseudo-code

Since the method created in this thesis involves a combination of many different techniques, pseudo-code is provided as well to help better understand the precise working of each of those techniques. While some of the pseudo-code is presented within the actual chapters, not all were suitable for direct placement within the text and therefore the remaining pseudo-code is presented here.

Algorithm 4 Channel-based ground segmentation method.

Input: $P = p_1, p_2, \dots, p_n; p_i = x_i, y_i, z_i, intensity_i, timestamp_i, ring_i$

Output: P_{ground}

```
1:  $R \leftarrow r[1, \dots, n]; r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ 
2:  $H \leftarrow h[1, \dots, n - 1]; h_i = z_{i+1} - z_i$ 
3:  $D \leftarrow d[1, \dots, n - 1]; d_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}$ 
4:  $\alpha \leftarrow \arcsin(H/D)$ 
5:  $\alpha_{max}, h_{min}, d_{min} \leftarrow 3^\circ, 0.05m, 0.03m$ 

6: for  $i \in n - 1$  do
7:   if  $-h_{max} \leq h_i \leq h_{max}$ 
8:     and
9:      $-\alpha_{max} \leq \alpha_i \leq \alpha_{max}$ 
10:    and
11:     $r_{i+1} - r_i \geq d_{min}$  then
12:       $P_{ground} \leftarrow p_i$ 
13:    end if
14: end for
```

Algorithm 5 Combining wall-based ground surface reconstruction with segmented ground points.**Input:** $P_{clusters}, P_{ground}$ **Output:** P_{final}

- 1: **Step 1:** Create segment angles for the polar grid and assign the $P_{clusters}$ points to the segments in which they belong. Additionally, extract the minimum z and ring values per segment.
- 2: $segment_angles \leftarrow$ an array from 0° to 360° , in steps of 10°
- 3: $P_{segment} \leftarrow [...]$
- 4: $P_{segments_vec} \leftarrow [...]$
- 5: $lowest_z \leftarrow [...]$
- 6: $lowest_ring \leftarrow [...]$

- 7: **for** $i \in segment_angles$ **do**
- 8: **for** $j \in P_{clusters}$ **do**
- 9: **if** $segment_angles[i] \leq \arctan(\frac{y_j}{x_j}) \leq segment_angles[i + 1]$ **then**
- 10: $P_{segment} \leftarrow p_j$
- 11: **end if**
- 12: **end for**
- 13: $P_{segments_vec} \leftarrow P_{segment}$
- 14: **end for**

- 15: **for** $P_{segment} \in P_{segments_vec}$ **do**
- 16: $lowest_z \leftarrow \min(P_{segment}.z)$
- 17: $lowest_ring \leftarrow \min(P_{segment}.ring)$
- 18: **end for**

- 19: **Step 2:** Per segment, check if the number of segmented ground points exceeds the expected amount as described by Equation 3.1 in Section 3.2. The segmented ground points which exceed the threshold will eventually be part of the method's output and are referred to as $gp_segments_included$.
- 20: **Step 3:** Segments which did not contain the expected amount of segmented ground points, will have their lowest z-values compared to, and if needed, adjusted to the lowest z-values of adjacent segments.
- 21: $origin_z \leftarrow -0.70m$
- 22: **for** $i \in P_{segments_vec}$ **do**
- 23: $min_distance \leftarrow$ the minimum distance value of the points in $P_{segments_vec}[i]$
- 24: $elevation_angle \leftarrow \arctan(\frac{lowest_z[i] - origin_z}{min_distance})$
- 25: $adjusted_lowest_z \leftarrow [...]$
- 26: **if** $lowest_z[i - 1] < lowest_z[i]$ **or** $lowest_z[i + 1] < lowest_z[i]$
- 27: **and**
- 28: $-5^\circ < elevation_angle < 5^\circ$ **then**
- 29: $adjusted_lowest_z \leftarrow \min(lowest_z[i - 1], lowest_z[i + 1])$
- 30: **else**
- 31: $adjusted_lowest_z \leftarrow origin_z$
- 32: **end if**
- 33: **end for**
- 34: $P_{segments_vec} \leftarrow adjusted_lowest_z$

- 35: **Step 4:** Reconstruct ground surface between the origin and lowest z values of the wall clusters as described in more detail by Algorithm 3 in Section 3.4.
- 36: $P_{reconstructed} \leftarrow reconstruction_algorithm(P_{segments_vec})$

- 37: **Step 5:** Merge the segments with ground points $gp_segments_included$ obtained in step 2, with the segments containing the reconstructed ground surfaces $P_{reconstructed}$.
- 38: $P_{final} \leftarrow P_{reconstructed} + gp_segments_included$

A.2. Parameters used by the developed method to obtain results

In Table A.1, the different parameters used to obtain the results are shown per component of the developed method.

Component	Parameters	Values
Point cloud denoiser	Search radius	dynamic
	Min. / max. cluster size	100 / 5000
	Intensity threshold	25
	Alpha	0,5
Ground and wall point segmentation and clustering	Search radius	dynamic
	Min. / max. cluster size	50 / 5000
	Min. / max. ground angle	-3 / 3 deg.
	Min. / max. wall angle	60 / 120 deg.
	Relative distance margin	15%
Ground surface reconstruction	Height difference threshold	0.05
	Azimuth segment angle	10 deg.
	Reconstruction threshold	15%
	Min. wall points per segment	10
	Max. global elevation angle	5%

Table A.1: Overview of the parameters used per component of the developed method.

A.3. Enlarged qualitative results

In this section, the results obtained from the experiments are shown using larger image sizes.

A.3.1. Basic training room without smoke

This subsection shows in order: the results from the basic training room without smoke using a picture, merged ground truth points and the reconstructed output.



Figure A.1: Picture of the basic training room, shown from a similar perspective as the point clouds.

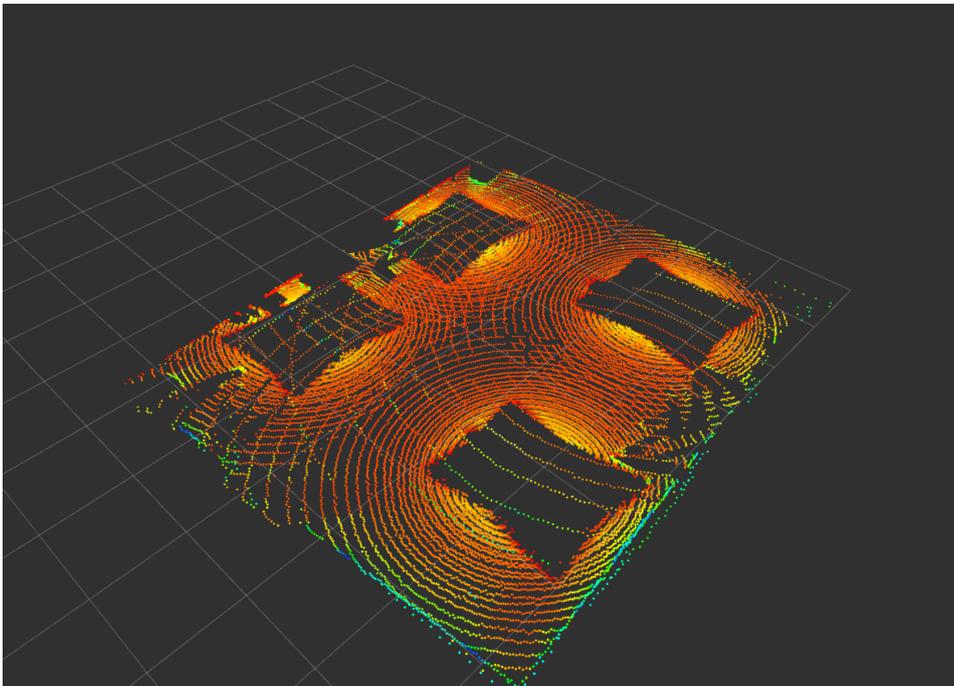


Figure A.2: The ground truth ground points obtained by merging point clouds from four locations.

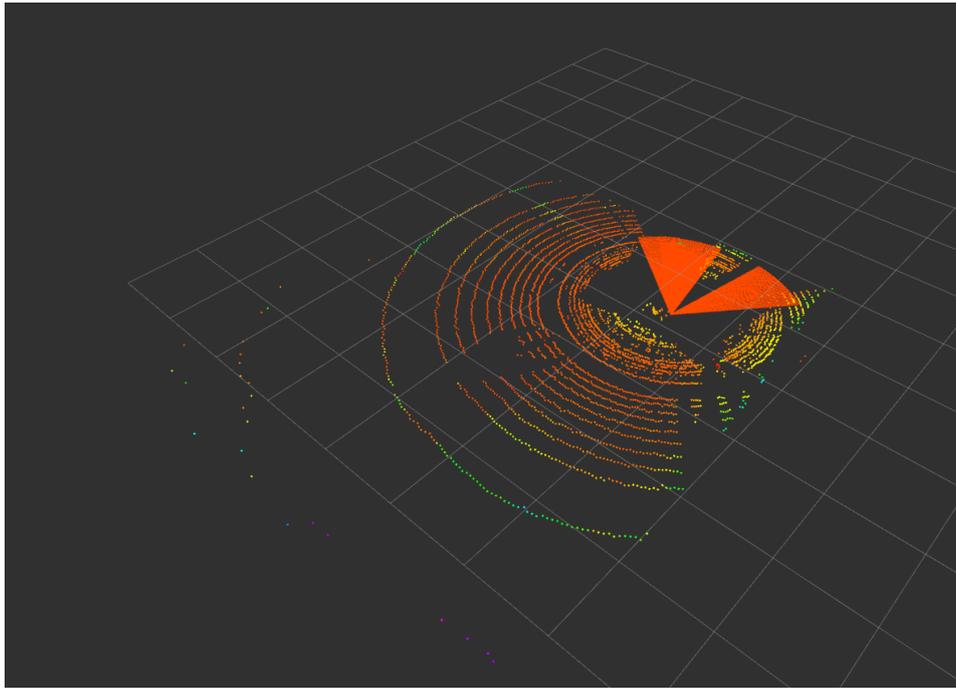


Figure A.3: The output point cloud obtained using the ground surface reconstruction method.

A.3.2. Basic training room filled with heavy smoke

This subsection shows in order: the results from the basic training room filled with smoke using a picture, the LiDAR's point cloud and the reconstructed output.



Figure A.4: Picture of the basic training room filled with smoke.

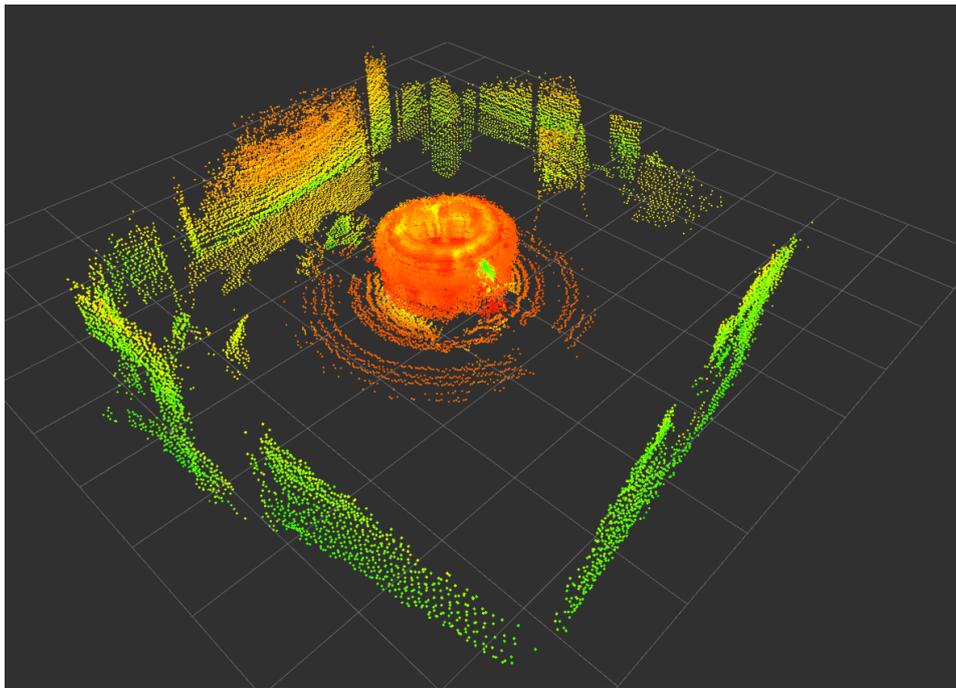


Figure A.5: The point cloud obtained in the basic training room filled with smoke.

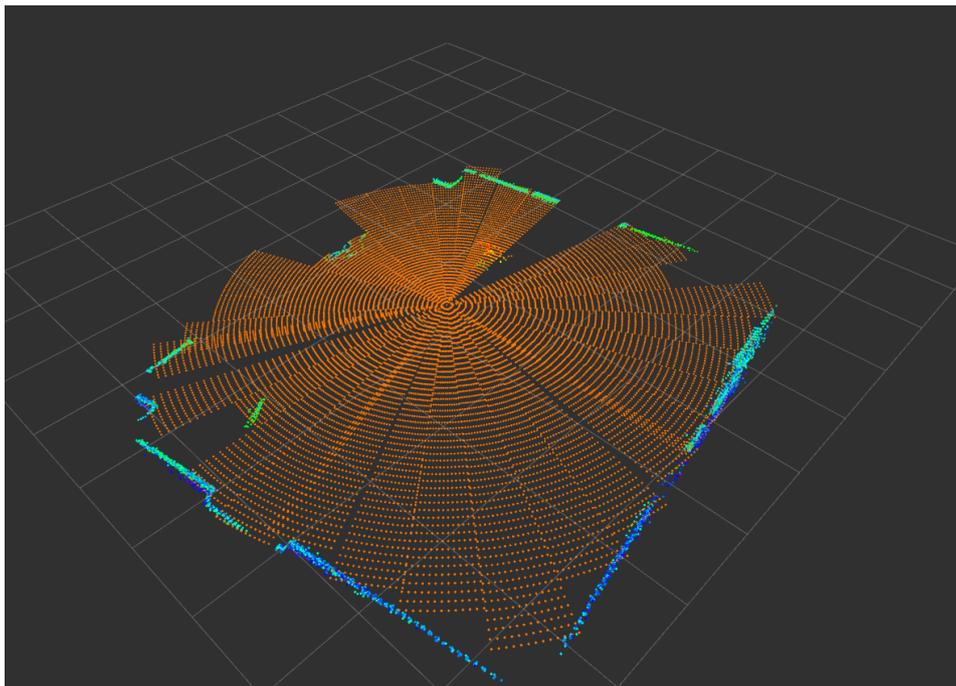


Figure A.6: The output point cloud obtained using the ground surface reconstruction method.

A.3.3. Complex training room without smoke

This subsection shows in order: the results from the complex training room without smoke using a picture, the LiDAR's point cloud and the reconstructed output.



Figure A.7: Picture of the complex training room, shown in a similar angle as the point clouds.

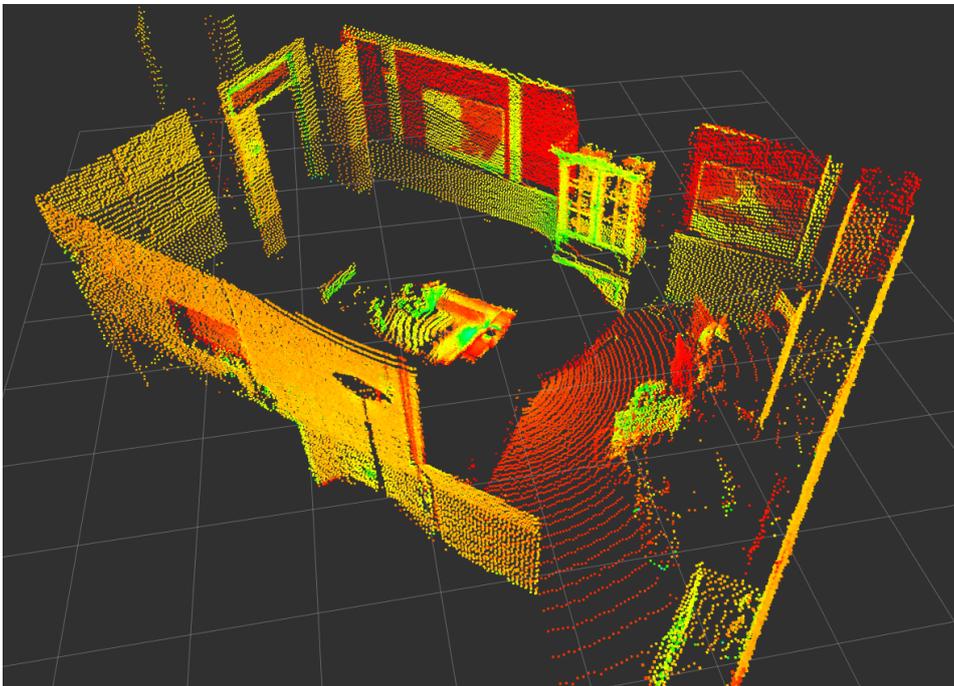


Figure A.8: Point cloud obtained in environments without smoke.

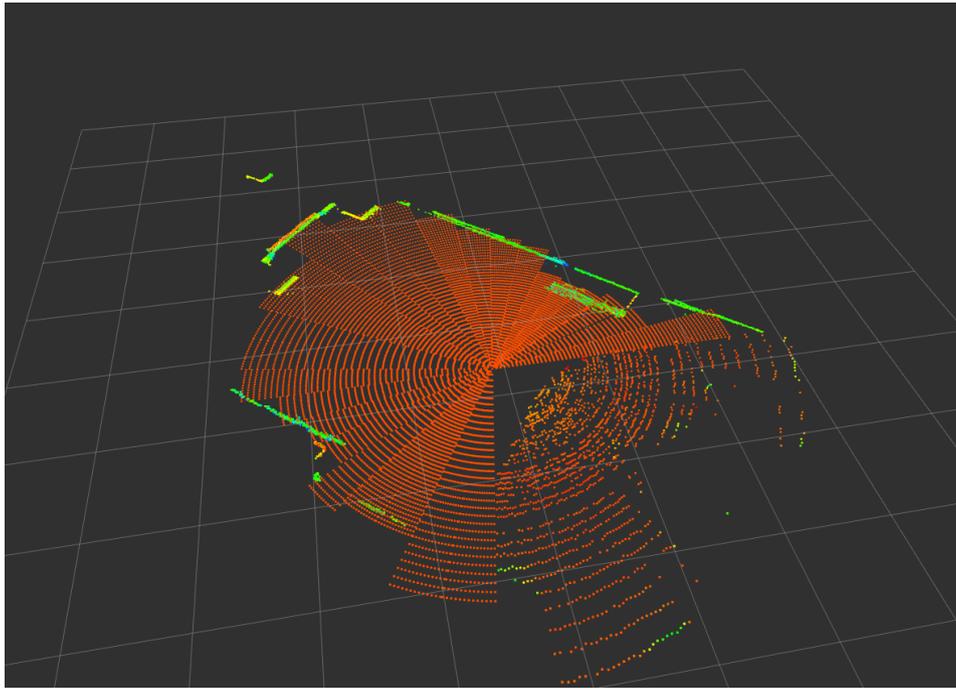


Figure A.9: The output point cloud obtained using the ground surface reconstruction method.

A.3.4. Complex training filled with heavy smoke

This subsection shows in order: the results from the complex training room filled with smoke using a picture, the LiDAR's point cloud and the reconstructed output.



Figure A.10: Picture of the complex training room filled with smoke.

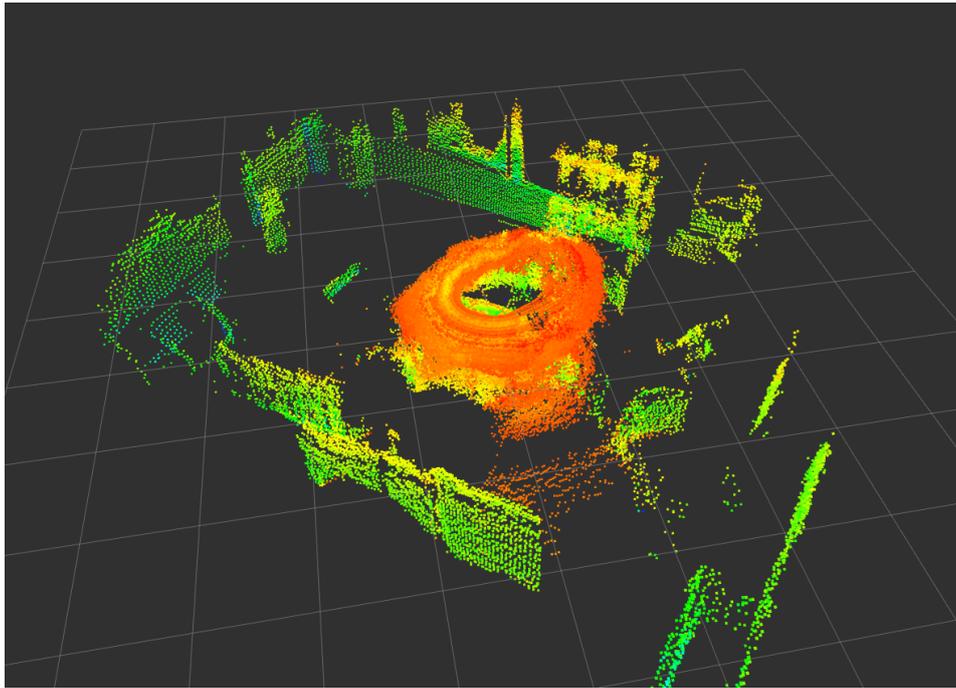


Figure A.11: The point cloud obtained in the complex training room filled with heavy smoke.

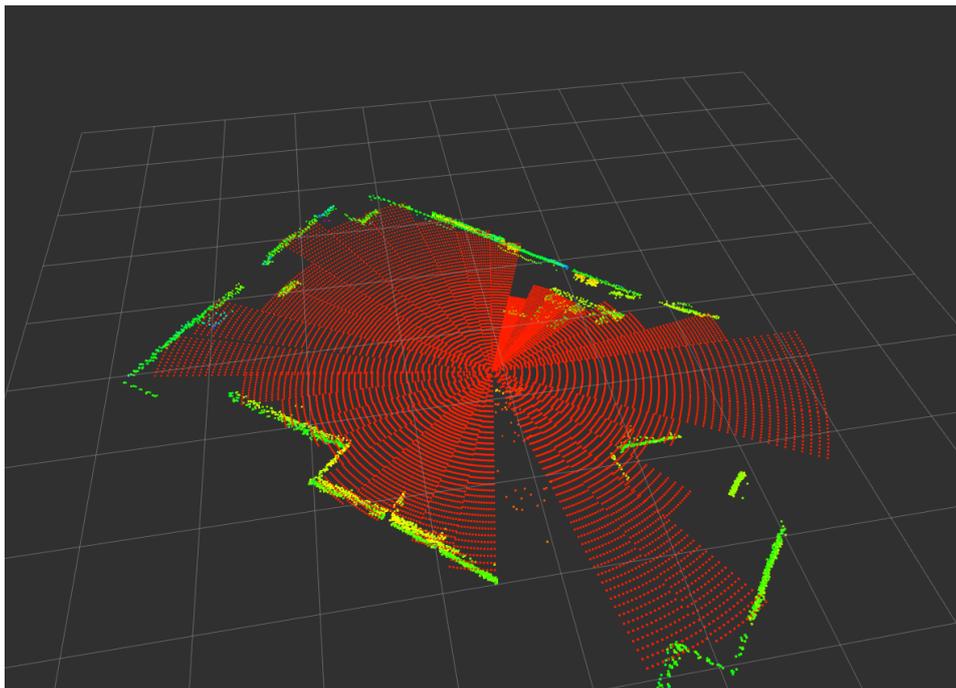


Figure A.12: Reconstructed ground surface (merged result) in heavy smoke.

A.4. LiDAR selection considerations

Since my thesis includes the implementation of a LiDAR sensor on a firefighting robot, this appendix provides information on how the best possible LiDAR for this thesis was selected. Furthermore, this section presents an overview of different well-known commercial LiDAR models including the technical specifications.

A.4.1. Selection procedure for the LiDAR purchase

In order to make a well-considered choice when it comes to selecting the best LiDAR model for our firefighting robot, a sequential procedure has been followed. This procedure consists of the following steps:

1. Creating a long list of relevant LiDAR companies by:
 - Writing down all LiDAR companies found in the relevant scientific literature.
 - Adding additional relevant LiDAR companies via Google, video calls with LiDAR suppliers and meetings with TU Delft staff from the intelligent vehicles group on 3mE.
2. List all of the commercially available LiDAR models, both mechanical (only 128 layer models) and solid-state models from the companies in the long list.
3. Collect the following (technical) selection criteria for all of the LiDAR models found thus far:
 - Max. operating temperature (since we want to robot to be able to operate at high temperatures).
 - FoV horizontal (since we want to get a complete view of the operating environment).
 - FoV vertical (since we want to get a complete view of the operating environment).
 - Horizontal resolution (since we to get a detailed overview of the operating environment).
 - Vertical resolution (since we to get a detailed overview of the operating environment).
 - Range @ 10% reflectivity (since see at larger distances).
 - Price (since there is a limited budget assigned to this project).
4. Narrow down the list of LiDAR models by only keeping the models for which all of the technical selection criteria could be found (see Figure A.14)
5. Rank the LiDAR models using a weighted scoring method (see Figure A.13).

Eventually, the HESAI QT128 mechanical LiDAR received the highest score and has therefore been purchased.

1 = low importance, 2 = medium importance, 3 = high importance

Weights

Max. operating temp.	3	
FoV horizontal	3	
FoV vertical	3	
Horizontal res.	3	*uses a "reversed" normalisation formula
Vertical res.	3	*uses a "reversed" normalisation formula
Range @ 10% reflectivity	1	
Points per m2 at 10m per scan	0	*not using this one at the moment since vert. and horiz. resolution are used instead
Price	2	*uses a "reversed" normalisation formula

Min Max normalised data (0-1)	OS1 - Mid Range	Alpha Prime	QT128	Pandar128	AT128	M1	Cube 1 Outdoor	Falcon Kinetic	MOVIA (near range)	XenoLidar-X
Max. operating temp.	0,2	0,4	1,0	1,0	1,0	1,0	0,4	1,0	0,4	0,0
FoV horizontal	1,0	1,0	1,0	1,0	0,2	0,2	0,0	0,2	0,2	0,0
FoV vertical	0,4	0,3	1,0	0,3	0,2	0,2	0,2	0,2	0,7	0,1
Horizontal res.	0,8	0,9	0,6	0,9	0,9	0,8	1,0	0,8	0,5	0,7
Vertical res.	0,7	0,9	0,2	0,7	0,8	0,8	0,7	0,8	0,2	0,7
Range @ 10% reflectivity	0,3	1,0	0,0	0,6	0,6	0,5	0,0	0,8	0,0	0,1
Points per m2 at 10m per scan	0,3	0,6	0,0	0,4	1,0	0,5	0,1	0,3	0,0	0,1
Price	0,7	0,0	0,9	0,2	0,8	0,7	0,9	0,9	0,9	0,9

Weighted scores	OS1 - Mid Range	Alpha Prime	QT128	Pandar128	AT128	M1	Cube 1 Outdoor	Falcon Kinetic	MOVIA (near range)	XenoLidar-X
Max. operating temp.	0,7	1,3	3,0	3,0	3,0	3,0	1,3	3,0	1,3	0,0
FoV horizontal	3,0	3,0	3,0	3,0	0,6	0,6	0,1	0,6	0,6	0,0
FoV vertical	1,1	0,9	3,0	0,9	0,5	0,5	0,6	0,5	2,1	0,3
Horizontal res.	2,5	2,8	1,8	2,8	2,8	2,5	3,0	2,5	1,6	2,1
Vertical res.	2,0	2,8	0,5	2,1	2,5	2,5	2,0	2,4	0,6	2,2
Range @ 10% reflectivity	0,3	1,0	0,0	0,6	0,6	0,5	0,0	0,8	0,0	0,1
Points per m2 at 10m per scan	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Price	1,5	0,0	1,7	0,4	1,7	1,4	1,9	1,7	1,7	1,9
Total (max score = 18)	11,0	11,9	13,0	12,9	11,6	10,9	9,0	11,5	7,9	6,6

Figure A.13: The weighted scoring method used for selecting the best LiDAR model for my thesis.

A.4.2. Overview table of commercial LiDAR models



Name LiDAR	OS1 - Mid Range	Alpha Prime	QT128	Pandar128	AT128	M1	Cube 1 Outdoor	Falcon Kinetic	MOVIA (near range)	XenoLidar-X
Company	Ouster	Velodyne	Hesai	Hesai	Hesai	RoboSense	Blickfeld	Innovusion	MicroVision/Ibeo	XenomatiX
LiDAR type	Rotating	Rotating	Rotating	Rotating	MEMS	MEMS	MEMS	MEMS	Flash	Flash
Number of echos	2	2	2	2	2	2	3	2	3	1
Wavelength (nm)	865	903	940	905	905	905	905	1550	885	940
Layers/channels	128	128	128	128	128	-	-	-	-	-
Horizontal res. (°)	0,2	0,1	0,4	0,1	0,1	0,2	0,0	0,2	0,5	0,3
Vertical res. (°)	0,4	0,1	0,8	0,3	0,2	0,2	0,4	0,2	0,8	0,3
Frequency (Hz)	10	5	10	10	10	10	2,5	10	25	20
FoV vertical (°)	45	40	105	40	25	25	30	25	75	20
FoV horizontal (°)	360	360	360	360	120	120	70	120	120	60
Range @ 10% reflectivity (m)	90	300	20	200	180	150	30	250	18	50
Range @ 90% reflectivity (m)	170	-	-	-	-	-	-	-	-	-
Min. operating temp. (°C)	-20	-20	-40	-40	-40	-40	-30	-40	-20	-10
Max. operating temp. (°C)	50	60	85	85	85	85	60	85	60	40
Ingress	IP67	IP67	IP6KX, IPX7, IPX9K	IP6K7, IP6K9K	IP6KX, IPX7, IPX9K	IP67, IP6K9K	IP65	IP67, IP69K	-	IP67
Points per second (all echos)	5.242.880	4.800.000	1.728.000	6.912.000	3.072.000	1.500.000	90.000	900.000	960.000	300.000
Price (EUR)	14.000	50.000	8.000	40.000	9.500	16.000	4.500	7.500	7.800	4.000

Figure A.14: Overview of well-known commercial LiDAR models including the relevant technical specifications.