FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Multimodal Terrain Traversability Analysis for Unmanned Ground Vehicles in Forestry Environments

Miguel Vasques

Coimbra, September 2023

Multimodal Terrain Traversability Analysis for Unmanned Ground Vehicles in Forestry Environments

Supervisor:

Prof. Paulo Peixoto

Jury:

Prof. Lino José Forte Marques

Prof. Paulo José Monteiro Peixoto

Prof. Rui Paulo Pinto da Rocha

Dissertation submitted in partial fulfillment for the degree of Master of Science in Electrical and Computer Engineering - Robotics, Control and Artificial Intelligence Branch

Coimbra, September 2023

Acknowledgements

Agradeço a Deus que me foi capacitando para chegar ao início e ao fim desta jornada que foi o ensino superior. Sempre que consegui esforçar-me fui muito abençoado.

Ao Professor Paulo Peixoto, por ter aceite e orientado o trabalho,pelo ensino e pelas constantes palavras de ânimo.

Aos meus pais, irmãos e avó que sempre me ajudaram.

Aos meus companheiros e bons amigos do SVA de 2017. Bruno tens *mea culpa* na minha vinda para Coimbra

Aos amigos que estiveram desde o ínicio a partilhar dificuldades e alegrias semelhantes e que estão para ficar: Marcos, João, Tiago, Ju, Guilherme, Bianca, Lucas e Tiago, OBRIGADO.

Ao Jopa e à Lila que me fizeram sempre sentir em casa, muito muito obrigado!

Ao Fernando Esteves que foi sempre um avô desde o primeiro dia!

A todos no clube de Desbravadores de Coimbra por me acolherem aqui, vão ter sempre um lugar no meu coração onde estiver!

Ao Fernando, um amigo a qualquer hora.

Aos meus parceiros das classes avançadas com quem passei uma parte significativa de todas as férias de verão, depois de ir a demasiados recursos, a todos os dirigentes que muito ensinaram e se tornaram bons amigos, Rita, Júlio e Caixeiro, Obrigado por todos os bons momentos!

Rebeca e Pablo obrigado por toda a amizade e pela família na Figueira.

Aos meus colegas e AMIGOS que aqui encontrei e com quem partilhei de perto vitórias e derrotas aqui. Miguel, Cunha, Tiago, Neves, Proença, Roger, Sénica, Sara, Roberto. Foi excelente meus amigos!

Thanks Habibu for the exchange of advice on work and about life.

A mis amigos de México, Dorancy somos Maestros tia!

Resumo

Com o aumento da intensidade de fenómenos climáticos extremos, como a seca, o risco de incêndios florestais tornou-se ainda maior. Uma ferramenta fundamental para mitigar os riscos apresentados por essas condições cada vez mais comuns é a gestão do perímetro florestal, que envolve a remoção de materiais altamente inflamáveis, como ramos e arbustos mortos - uma tarefa frequentemente negligenciada devido à falta de recursos e ao perigo envolvido. Os avanços na navegação autónoma criam a oportunidade de automatizar esse processo de limpeza com virtualmente nenhum recurso humano, equipando o veículo com sensores capazes de fornecer uma compreensão do ambiente para navegação e implantação segura de atuadores que permitem a identificação e remoção de material combustível. A precisão da tarefa de perceção é de extrema importância e muitas vezes impossível de ser alcançada com apenas um único sensor, sendo assim importante usar sensores auxiliares, como IMU, GPS, etc. Para realizar a navegação autónoma em ambientes florestais, dada a variedade de estruturas observáveis nessas áreas, que muitas vezes não possuem características fixas como em ambientes urbanos (por exemplo, passeios, passagens para peões) e superfícies construídas para fins de navegação, é necessária uma análise única para determinar se a navegação pode ou não ser realizada ao longo do caminho enfrentado. Esta análise visa obter uma medida de *traversability* e é realizada levando em consideração vários elementos, como a rugosidade, inclinação, elevação e as próprias limitações do veículo.

Nesta dissertação, é apresentado um método de análise de traversibilidade que utiliza uma nuvem de pontos 3D obtida de um sensor LIDAR e dados de localização de GPS e IMU para distinguir zonas onde a passagem é possível/impossível, usando parâmetros como inclinação e elevação entre superfícies. O método foi submetido a vários testes em ambientes florestais para os quais não havia testes documentados, e o seu desempenho foi avaliado em diferentes ambientes florestais e com diferentes sensores. Os resultados são apresentados e analisados no documento.

Palavras-Chave: análise de travessia; LIDAR; deteção multimodal; robótica florestal

Abstract

With the increasing intensity of extreme climate phenomena such as drought, the risk of forest fires has become even greater. A key tool for mitigating the risks posed by these increasingly common conditions is forest perimeter management, the removal of highly flammable material such as branches and dead shrubs – a task often neglected due to the lack of resources and the danger involved. Advances in autonomous navigation create the opportunity to automate this cleanup process with virtually no human resources by equipping the vehicle with sensors capable of providing an understanding of the environment for navigation and safe deployment of actuators that enable identification and removal of combustible material. The accuracy of the perception task is of paramount importance and often impossible to achieve with just a single sensor, so it is important to use auxiliary sensors such as IMU, GPS, etc. To perform autonomous navigation in forest environments, given the variety of observable structures in these areas, which often do not have fixed features as in urban environments (e.g., curbs, crosswalks) and surfaces constructed for navigation purposes, a unique analysis is required to determine whether or not navigation can be performed along the faced path. This analysis aims to obtain a measure of traversability and is performed taking into account various elements such as roughness, slope, elevation and the vehicle's own limitations.

In this dissertation, a method of traversability analysis is presented that uses a 3D point cloud obtained from a LIDAR sensor and localization data from GPS and IMU to distinguish zones where traversability is possible/impossible, using parameters such as slope and elevation between surfaces. The method was subjected to various tests in forest environments for which there were no documented tests, and its performance was evaluated in different forest environments and with different sensors. The results are presented and analyzed in the document.

Keywords: Terrain Traversibility Analysis; LIDAR; Multimodal Sensing; Forestry Robotics

"A vida é uma grande universidade, mas pouco ensina a quem não sabe ser um aluno"

— Augusto Cury

Contents

List of Acronyms

Full Name	Acronym
Bayesian Generalized Kernel	BGK
Global Positioning System	GPS
Inertial Measurement Unit	IMU
Inertial Navigation System	INS
Light Detection and Ranging	LIDAR
LIDAR Odometry and Mapping	LOAM
Point Cloud	PCD
Robot Operating System	ROS
Real Time Kinematics	RTK
Simultaneous Localization And Mapping	SLAM
Terrain Traversability Analysis	TTA

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Context

In recent years, the alarming increase in the frequency and intensity of forest fires has cast a shadow over ecosystems, communities, and landscapes around the world, highlighting the urgent need to address this pressing environmental challenge. Statistics from the European Forest Fire Information System (EFFIS) show that of all the fires in Europe in 2022, 17,000 fires resulted in a burned area of more than 30 ha, and that these incidents represent only a small fraction of the total [6].

The data show that 85% of these fires occur in the Mediterranean region. Five of EU-MED countries - Portugal, Spain, France, Italy, and Greece - have data available in EFFIS for a longer period of time. Of these countries, Portugal is particularly noteworthy, as 35% of the fires and 39% of the area burned in the last 30 years belonged to Portuguese forests [7].

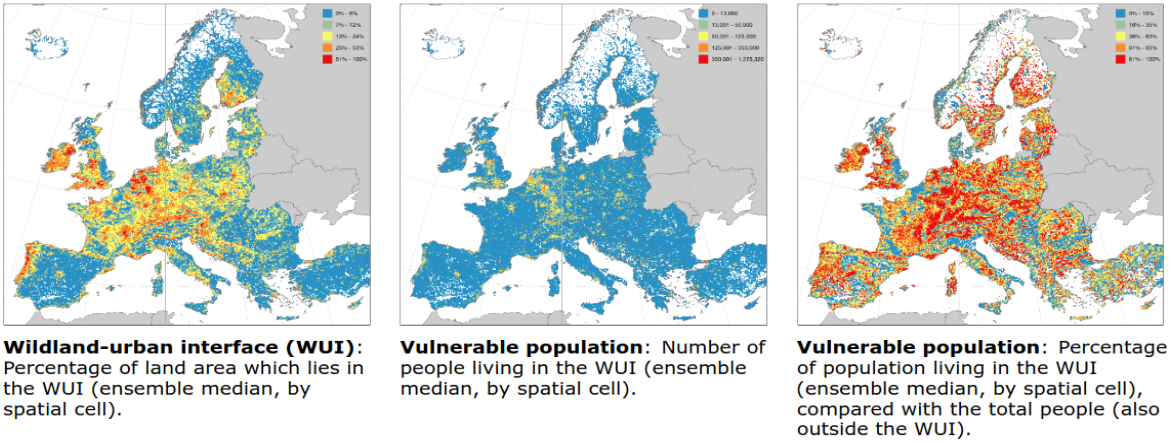
The impact of forest fires reaches different sectors:

- Economy - Destruction of forested areas leads to economic losses in sectors ranging from logging to forest tourism to agriculture, leading to an increase in unemployment that promotes migration from rural areas, which in turn promotes accumulation, which in turn promotes new fires.

- Environment - Forest fires are the cause of species extinction and the emission of gasses into the atmosphere, leading to an increase in global warming.

Wildfire risk can be described as the sum of the magnitude of wildfire risk in a given area and the potential impact on citizens. As a major component of risk, wildfire risk is associated with several factors, including weather and climate conditions that can affect the likelihood of ignition or subsequent fire behavior. Although they do not always appear in fire risk definition systems, 90% of wildfires are caused by human activities [1] and occur in

the WUI - Wildland Urban Interface where human activities are more present (see Figure 1.1, 1.2). These fires fall into the category of most dangerous to the public, but the larger burn areas are in forests in less densely populated areas where there is no forest monitoring [8].



Wildland-urban interface (WUI): Percentage of land area which lies in the WUI (ensemble median, by spatial cell).

Vulnerable population: Number of people living in the WUI (ensemble median, by spatial cell).

Vulnerable population: Percentage of population living in the WUI (ensemble median, by spatial cell), compared with the total people (also outside the WUI).

Figure 1.1: Vulnerable population distribution by proximity to wildland fuels.[1]

According to reports from Portuguese authorities, the causes are mostly criminal in nature, but also include negligence in activities such as uncontrolled campfires or flying sparks from heavy machinery [9] [10].

Of the possible measures to prevent forest fires and their effects, such as awareness campaigns, improving vegetation resilience, and others, one of the most effective is to promote regular "cleaning" of forests to reduce the accumulation of fuel through pruning, mowing, raking, and removal [11]. This task is not easy because, despite the introduction of measures such as education or laws on forest management, most of the land is owned by non-industrial private companies and it is difficult to make the process as professional and efficient as possible, since the owners may not agree with the measures planned for their land [12].

Research has shown that forest debris such as brush, scrub, and pine needles are highly combustible and promote the spread of fires and that it is necessary to maintain forest areas by reducing the accumulation of fuel. The big problem is that this work requires high investment, focusing on attracting labor for this purpose, which is a difficult task due to the demanding and dangerous work [9].

Work in the forestry sector involves a high risk of injury, which accounts for the majority of occupational accidents [13]. This is due to the high loads, exposure to the environment and the use of power tools and machines, which make the work more efficient, but also more risky, causing problems in the muscular and auditory system, back injuries, as well as many

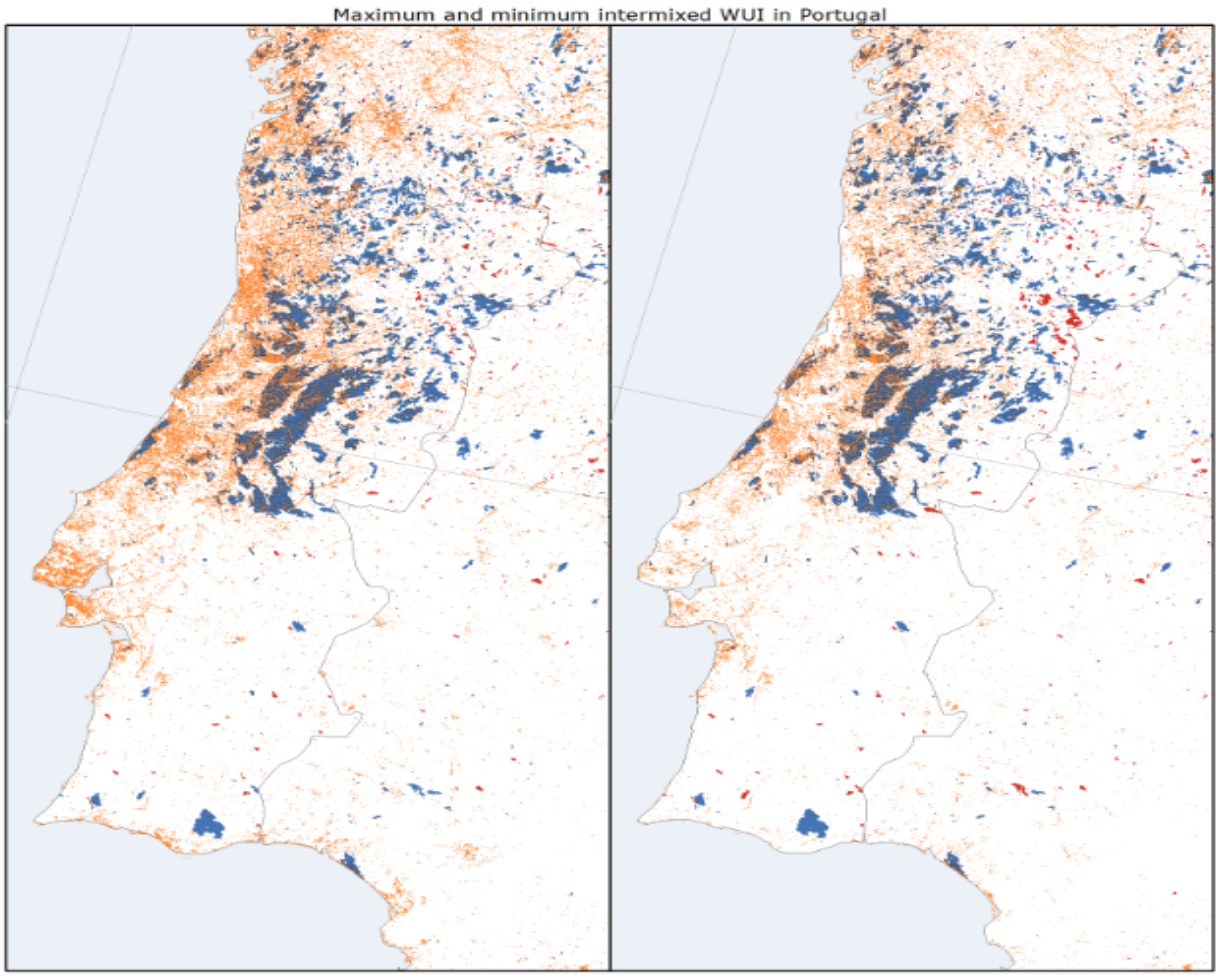


Figure 1.2: WUI - Orange, Fires intersecting WUI are marked in blue, Fires not intersecting marked in red [1]

other occupational accidents resulting from the use of said tools [7][14].

Among the various tools available for this type of work, mulchers are increasingly recognized as one of the most efficient methods for maintaining forest areas. They allow the shredding of various types of fuel material, which is then spread on the ground as mulch. This conversion leads to a potential reduction in fire spread by reducing the available oxygen supply in this dense fuel. Research also shows that mulch can be helpful in restoring a burned area.

Given these reasons and the increasing development and availability of better solutions, it is imperative to complete this important task using the most efficient and safest processes, with one of the options being to apply the technology using robotic forestry [15]. To take full advantage of this category of tools, said robots must be able to navigate autonomously. For this, a perception module that can determine the regions to be cleaned and navigated is essential [16]. In this work, we address the navigation problem, i.e., the robot's ability to

respond to the perceived environment and move from the initial state to a desired target state. A core component for solving the navigation problem is the analysis of the traversability of the terrain, which provides information for path planning and obstacle avoidance, where the information about obstacles is determined by their characteristics and the constraints of the robot.

1.2 Objectives

The main objectives of this thesis are:

- Select and implement a baseline terrain traversibility analysis (TTA) method.
- Examine a number of different graded forest datasets (off-road, park, forest) and evaluate and document performance of the selected method in each scenario.

1.3 Main Contributions

Contributions provided by this work include:

- Performance evaluation of the selected TTA method - Most of the TTA methods proposed in the literature deal with autonomous navigation in urban environments. In this work, we evaluated several of them and selected the one that seemed most suitable for a forestry environment. We tested it on three different datasets to prove their suitability for different application scenarios. We also conclude that different LiDARs with different point cloud densities and acquisition patterns can be used without significantly degrading performance.
- Multi-dataset testing - Tests were performed on different datasets recorded from different locations and with different sensory conditions. The specifics of the various landscape features encountered are documented.
- Identification of the method limitations - We highlight the challenges of the selected method in creating traversability maps that take into account the unique characteristics of forested areas.
- Suggestions for improvement - Propose some solutions to address the identified problems and improve the performance of the method in forest environments.

1.4 Outline

The Outline of this document is as follows :

- Chapter 1 provides the motivation and context of this work and explains the purpose of this type of work in fire prevention.
- Chapter 2 presents related work in traversability analysis using various strategies and sensors, and introduces the most commonly used methods to accomplish this task.
- Chapter 3 analyzes the chosen method in detail and shows what happens at each step until the final result is obtained.
- Chapter 4 presents and describes the datasets used for the evaluation of the proposed method, their specifics and the results obtained, and analyzes the peculiarities found in the behavior of the method in the scenarios studied.
- Chapter 5 focus on the conclusions drawn after the development of the work and provides suggestions for methods that can be used to improve the performance of the algorithm on some of the challenges found during the tests.

2 State of Art and Background

2.1 State of Art

The present chapter is focused on the study of the state of the art regarding terrain traversability analysis, with special focus in off-road and forest environments.

2.1.1 Multimodal Sensing

Since precise positioning is one of the vital components for autonomous execution of tasks by mobile robots, and these are becoming more sophisticated as robot technology develops, it is no longer possible to fulfill every application with a single sensor.

There was a time when 2D LIDAR was the main choice for less demanding navigation and positioning - in structured interior environments - as it offered high accuracy with minimal data volume.

With the increasing availability of computing power, 3D LIDAR gained popularity as it offered point clouds with better matching methods and better stability for matching between frames. In addition to these improvements, developments brought the ability to integrate information from other popular sensors such as odometers and image sensors, which are used in many fields such as unmanned driving, autonomous robot navigation, and others.

On the other hand, SLAM systems require accurate position and orientation data that cannot be obtained from a single sensor:

- Dependence of vision on initialization and sensitivity to illumination drift cause instability.
- Sparse data from LIDAR makes positioning in more complex, unstructured scenes(e.g. forest environments) easy to get out of hand.
- Odometry information becomes invalid due to rapid movement and long-term error accumulation.

In forestry or off-road applications, uneven movement often occurs, leading to distortion of localization data. To mitigate this problem, several auxiliary sensors, such as inertial measurement units (IMU), global positioning system (GPS) and ultra-wideband (UWB) sensors, can be added to the localization system [17].

2.1.2 Terrain Traversability Analysis

Terrain sensing is one of the most challenging problems in field robotics. A robot is often confronted with scenes of varying complexity when navigating the terrain, so the ability to classify the surrounding terrain is a key property of intelligent unmanned ground vehicles (UGVSs) [18]. Terrain traversability analysis adresses two different problems:

- Safety - identifying traversable areas in the robot's upcoming potential path, avoiding collisions and unrecoverable states [19][18].
- Efficiency - operating in an optimal state, determining appropriate and efficient paths while considering simultaneously defined priorities such as risk, fuel consumption, and time [20] and considering vehicle kinematics and non-holonomic constraints [18].

Although traversability analysis does not yet have a firm formal definition in the robotics community [19], we conclude from the various sources that it is correct to say that TTA measures the ability of the vehicle to traverse a terrain region, taking into account the characteristics of the terrain and the capabilities of the robot. This analysis is then translated into a cost map of said terrain and can then be applied in various path planning approaches such as Roadmaps or Potential Fields [20].

The analysis can include data from multiple sensors and sensor types, either separately or combined [19].

- Exteroceptive Sensors - The sensors most commonly used for traversability analysis, consisting primarily of visual cameras and LIDAR capture data about the upcoming terrain to assess the traversability value when interpreted [18].
- Propriocetive sensors - Self-sensing sensors [21] simultaneously characterize the terrain as the robot traverses it or learn the robot's behavior by analyzing terrain cells previously observed remotely through exteroceptive sensors. This analysis enhances the robot's understanding of the terrain's shape, terramechanical properties, and their impact on the robot's state, including speed and orientation. Examples of such sensors include IMUs, accelerometers, RTK , GPS, and others.

In structured environments such as the normal urban environment where autonomous vehicles are increasingly common, there are many distinguishable geometric elements that help classify an area as clear or occupied by an obstacle and infer whether a particular path is a viable option for reaching a determined goal and thus stereo vision is a cost-effective and reliable method for information retrieval.

On the other hand, in unstructured environments, using RGB data to perform traversability analysis could be a good option, as there is already some work on performing semantic segmentation in camera images not only of urban but also of off-road or forest environments [22] [23].It is imperative to interpret beyond the geometric classification, e.g., classify sparse vegetation as non-traversable as it is often traversable or even denser vegetation that the deployed vehicle can actually pass and provide more options for efficient paths[24].

Nevertheless, problems such as color shifts and weather-related illumination changes pose a major challenge to the use of color-based classification approaches.

Given these limitations, this task usually relies on data from LIDAR point clouds, which, despite their limitations (sparse points and being an active sensors that require more energy and generate more measurement noise [25]), are resilient when confronted with adverse conditions such as varying surface textures and light shifts, and provide a good range measurement and reliable data over a long distance. The increasing availability of these sensors and the resulting drop in price make them an option to count on.

Other approaches can even use data from other sensors such as RADAR or combine some of the mentioned with others.

2.2 Terrain Semantic Segmentation

Valada et al. [23] proposed an approach using a system that worked with RGB, near infrared (NIR), and depth data and evaluated semantic segmentation with their UpNet architecture based on deep convolutional neural networks (DCNNs) with two main components: contraction and expansion. In contrast to UNet [26], UpNet uses the 13-layer VGG [27] architecture as the foundation on the contraction side. The expansion side consists of five up-convolutional upfold refinement segments that refine the coarse segmentation masks generated by the contraction segment. Each up-convolutional refinement consists of one upsampling layer followed by a convolution layer. This network was trained on various inputs in the dataset, identified the optimal inputs, and then fused them. This fusion improved the results of both unimodal approaches, even in the most extreme environments. The addition

of NIR data provides better detection of chlorophyll in vegetation compared to monocular cameras and provides better discrimination of obstacles in the environment [28].

2.3 Bayesian Inference

Shan et al. [29] presents a solution to the sparse data problem that often occurs when data is acquired as a point cloud. The navigation environment is treated as an elevation map, and each cell of the discretized grid map that lies within a user-defined distance threshold is assigned an elevation value using Bayesian kernel elevation regression. The training data, consisting of LIDAR scans and the estimated elevation map, allows the inference of a traversability value (using a weak forced prior for the Bayesian method) for cells that lie within the distance threshold to be calculated. This threshold is defined based on the map resolution so that inference is performed only for the neighboring cells of the cell, allowing the remaining cells to be computed by BGK-derived traversability inference and reducing the required computational power. This method had good results in urban environments, but was not properly evaluated in the forest. Therefore, this method was chosen and is discussed in more detail in Section 4 of this document.

2.4 Camera and LIDAR traversability Map Fusion

Sock et al. [2] fused LIDAR and camera data to overcome the usual problems of classifying unstructured elements using only the LIDAR point cloud, and computed two independent traversability maps. The camera traversability map divided the image into blocks that were classified as traversable/non-traversable. This served as the basis for an support vector machine [30] classifier, which was fed with the appropriate features and converted into a probability value for the remaining images. The LIDAR map uses the acquired data to create a 2.5D elevation grid map, compares the elevation difference (slope) between cells, and then converts it to a traversability value using an exponential function with experimentally determined parameters. The resulting map is created by a Bayesian fusion of the two maps generated from data from both sensors after a spatial and temporal alignment (see Figure 2.1).

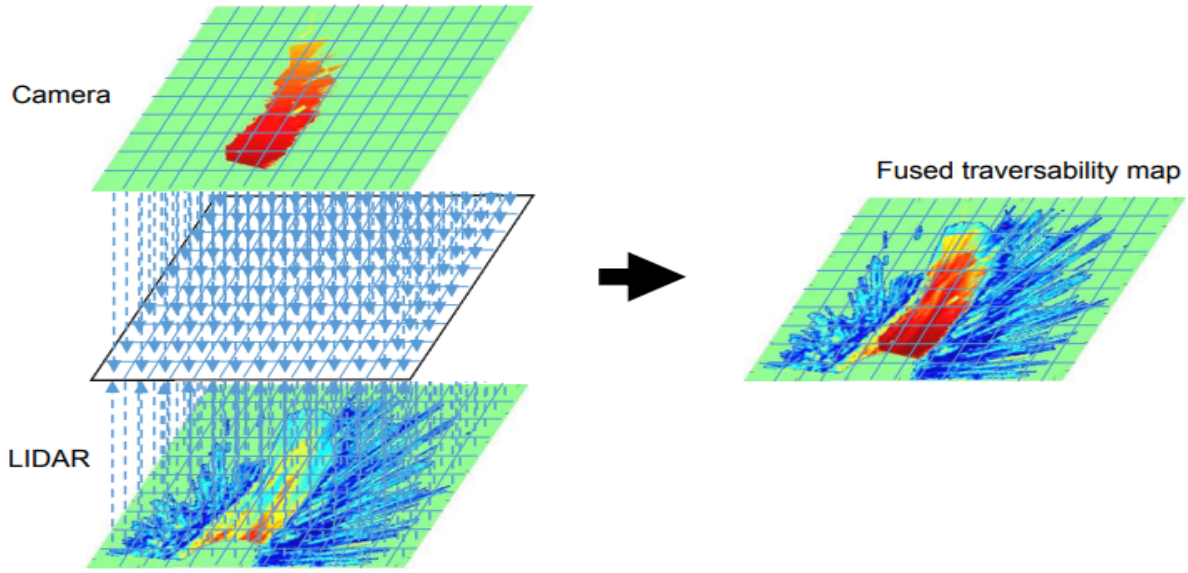


Figure 2.1: Fusion of LIDAR (Geometric Information) and RGB Camera (Texture and Colour Information) Traversability map [2].

2.5 Bayesian Integration in Fused Map

Zhou et al. [31] improved the results of the above two methods. In their approach, LiDAR data and camera images are combined to generate a comprehensive terrain map. LiDAR data provides depth information, while the camera images offer visual details. By merging these two data sources, the authors aim to create more accurate and informative terrain maps, enhancing a robot's ability to assess the terrain it encounters. LIDAR creates the elevation map by directly extracting the ground returns and using them in the elevation map using a Kalman filter [32] to process the overlapping data of successive scans. Starting from the elevation map already obtained, the traversability analysis begins with the calculation of the traversability cost and the assignment of a value for each grid cell with three geometric features commonly used for this task: Slope, Elevation, and Roughness. The final traversability value is determined using Bayes' rule to combine the traversability by LIDAR and the RGB sensor. This method showed a clearer traversability map with a similar runtime, but requires annotated data, which we are assuming is not available in our case. A summarized look at the reffered approaches is presented in Table 2.1.

Table 2.1: Overall view of methods with similar objectives to ours.

Author	**Focus**	**Method**	**Sensors**	**Output**	**Open--Source**	**Strengths/ Weaknesses**
Valada et. al [23]	Forest	UpNet and DCNN	RGB, NIR, Depth	Feature Map/ Segmented Image	x	Needs Anotated Data
Shan et. al[29]	Urban/Off-road	BGK, 2.5D Elevation Grid Map	LIDAR	2D Grid Traversability Map	*	Only Needs LIDAR data
Sock et. al[2]	Off-road	2.5D Elevation Grid Map, SVM	LIDAR, CCD	2D Grid Traversability Map	x	Combines Sensors Needs Annotated data
Zhou et. al[31]	Urban/Off-road	GANav	LIDAR, RGB	2D Grid Traversability Map	x	Needs Annotated Data

2.6 Background

This section focuses on some work developed in the past that is the cornerstone of some state of the art methods, with emphasis on the chosen method.

2.6.1 LIDAR Odometry and Mapping (LOAM)

Unlike applications where only the LIDAR moves (rotates) to capture the environment and measurement errors are constant with distance, accurate mapping in applications that require movement of the sensor itself requires knowledge of the sensor position throughout the scan [3].

LOAM - LIDAR Odometry and Mapping splits the SLAM problem into high frequency odometry to estimate LIDAR velocity and a slower process of mapping to register and match the point cloud (see Figure 2.2), allowing real-time mapping and updating of sensor position with higher precision after performing point cloud features matching.

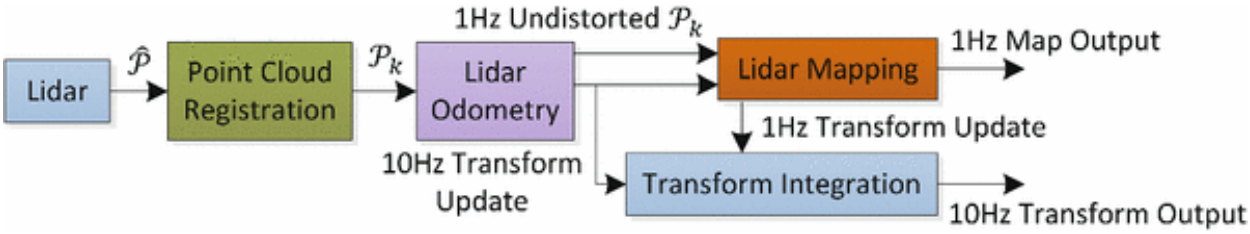


Figure 2.2: LOAM Flow Diagram [3]

LOAM functioning has been improved over the time in many ways. SLOAM [33] uses semantic features instead of geometric ones, making it a good option in unstructured environments but requiring annotated datasets. LEGO-LOAM [34] brings in label matching to further improve object matching between scans, while F-LOAM [35] outperforms the other methods in this category in terms of runtime.

For LIDAR with a smaller FOV that provide fewer observations for each feature, such as most solid-state LIDAR, problems arise, such as motion blur due to the constantly rotating LIDAR with non-repetitive sampling, which invalidates continuous feature matching between frames LOAM-LIVOX [36] addresses this problem with good results by implementing point selection and iterative pose optimization.

2.6.2 Range Image

The projection of a 3D LIDAR point cloud into a 2D image greatly simplifies the process of analyzing neighboring points. The lines of the range image can be obtained in a variety of ways. LIDAR such as Velodyne's use different laser channels, with each beam exiting in a "fan" pattern and diverging by a certain angle (vertical resolution) that provides the vertical field of view of the sensor (see Figure 2.3). To achieve the 360^{o} horizontal FOV, the sensor rotates around the vertical axis of the scanner [37]. Each channel is associated with a value ID stored in the ring field of each point in the point cloud, and this value addresses the image row [37]. In other capturing architectures, row attribution can be done based on the said elevation angle (see Figure 2.4) and sensor parameters given by the manufacturer [38].

$$VerticalAngle = \text{atan2}\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \tag{2.1}$$

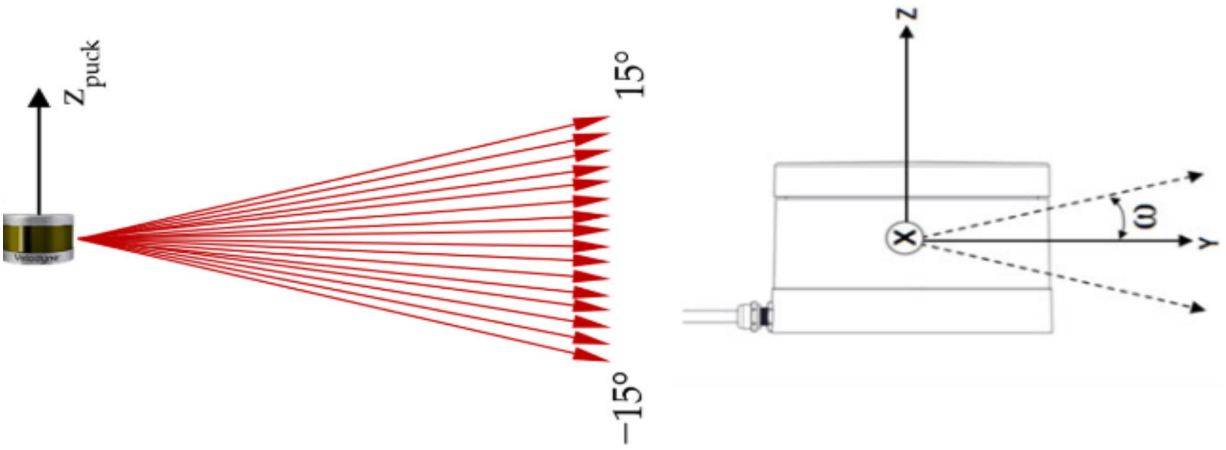


Figure 2.3: The "Fan" capture pattern. The divergence between beams multiplied by the number of channels gives the vertical FOV while the rotation around the sensors's vertical axis allows a 360^{o} horizontal FOV [4] .

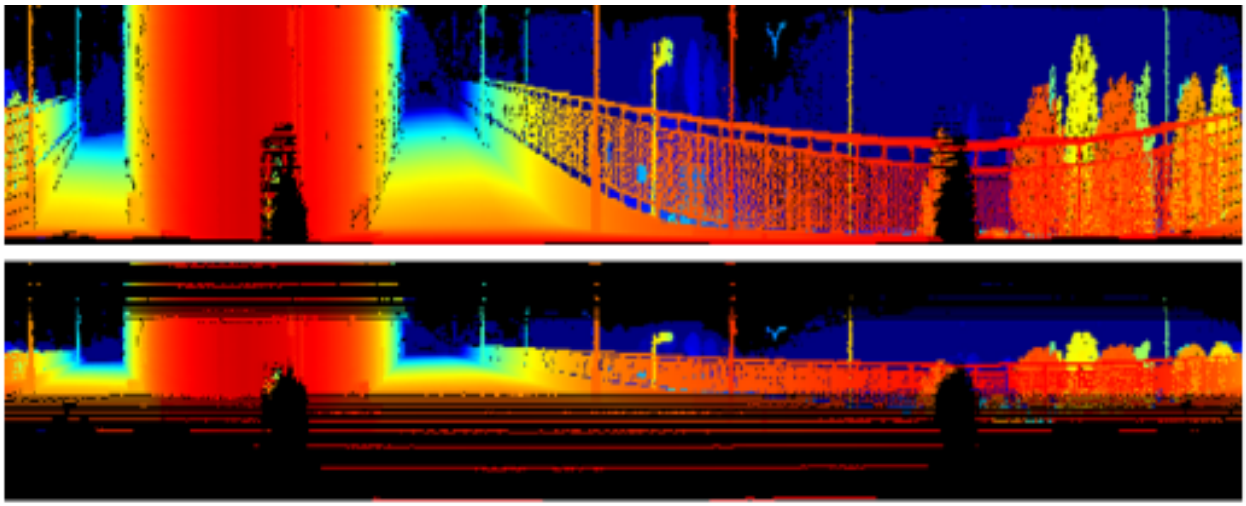


Figure 2.4: Difference between Row Attribution by channel ID (top) and elevation angle (bottom) [5].

2.6.3 Occupancy Grid Mapping

An occupancy grid map consists of a 2D grid composed of equal cells, where each cell stores quantitative information that distinguishes free or occupied areas of the robot operating environment. This is achieved by each cell storing a probability value of being occupied

[39] after evaluating readings from its sensors. Such maps play a crucial role in mobile robot applications, as they are of great help in tasks such as navigation, path planning, localization, and collision avoidance [40]. When a ground truth occupancy grid is available, each cell can be analyzed to determine the performance of the created grid [41]. In this case, the process is simplified by binarizing the value of the cells into occupied or free and evaluating the 4 logical states: Occ - Occ, Occ - Free. Free- Occ, Free - Free [42].

	Predicted Class	
Actual Class	**Occupied**	**Free**
Occupied	True Positives (TP)	False Positives (FP)
Free	False Negatives (FN)	True Negatives (TN)

Table 2.2: Confusion Matrix of the possible readings of the grid when compared to ground truth

2.6.4 Sparse data in 3D point clouds

The problem of sparse data often occurs in 3D LIDAR because they produce noisy [43] and sparser images compared to dense RGB images [44]. The terms "dense" and "sparse" are easily defined if we consider the world as a discretized grid.

In sparse data, many of these cells are empty, while dense point clouds contain data points in most cells [44] (see Figure 2.5). The occurrence of sparse data is not only inconvenient for similarity computation, but also for other applications such as object tracking or obstacle detection [45] (see Figure 2.6).

2.6.5 Bayesian Kernel Inference

Gaussian processes are probabilistic models that have proven useful in machine learning, as they allow for modeling functions and making predictions by defining a distribution over an arbitrary function, assigning each input point a random variable that follows a multivariate Gaussian distribution and providing a way to model uncertainty over functions along with the output value.

A Gaussian process is fully described by its mean function m(x) and its covariance function $k(x, x')$ [46] with this second component often referred to as the kernel function. The goal of the kernel function is to define the existing correlation between the function values

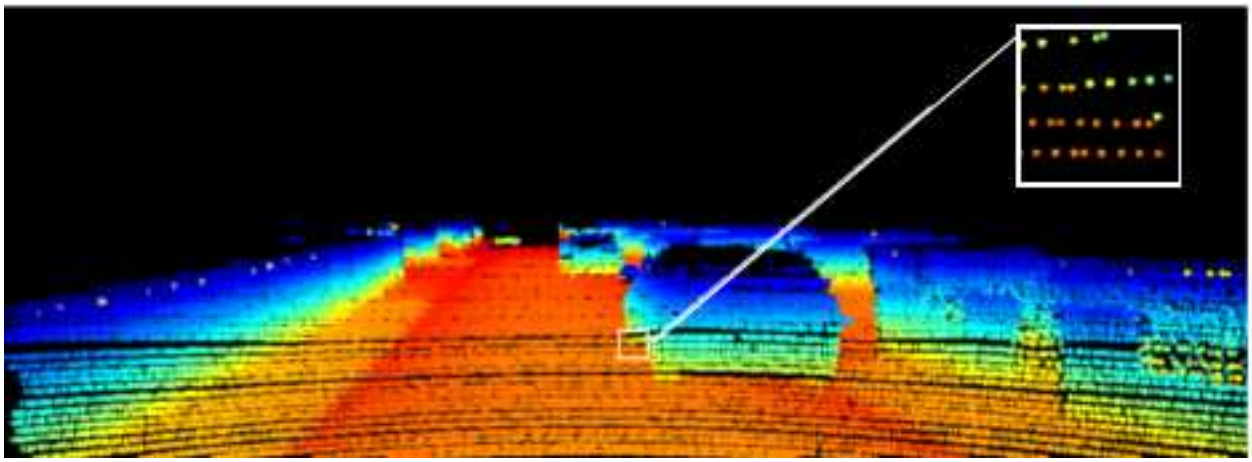


Figure 2.5: Height image extracted from the point cloud from a Velodyne HDL-64E. LIDAR data sparsity - void space between points, discontinuity in the scan, highlighted on the top right corner - happens even in high-end LIDAR.

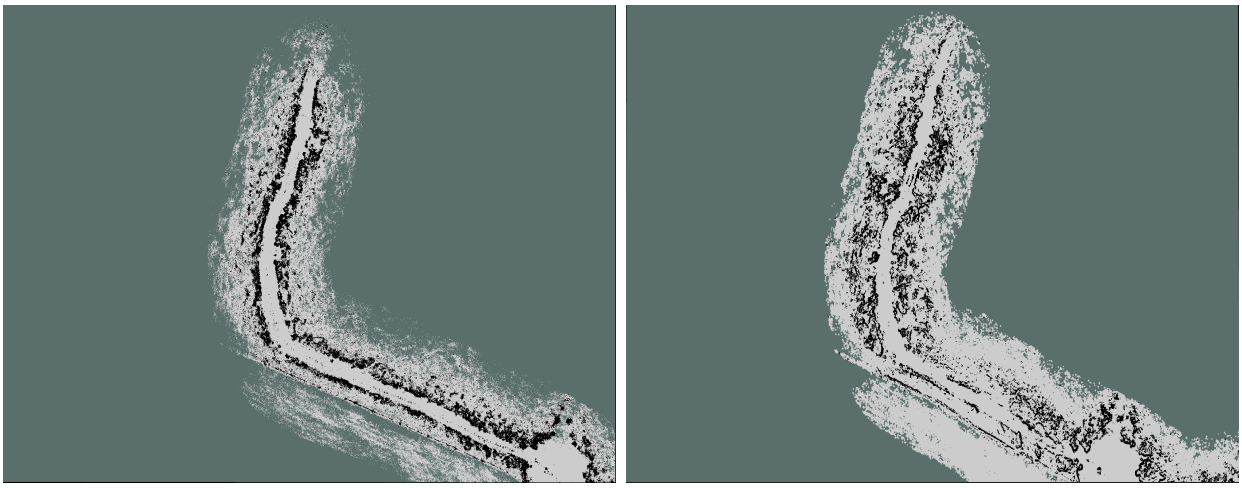


Figure 2.6: Occupancy grid from the sparse LIDAR point cloud (left image) and after applying inference (right image). The raw point cloud leaves many cells with unknown data that can't be used for path planning as the occupied/free state of that spaces is unknown.

at different input points and thus quantify the output changes of the function relative to the input changes.

Different kernel functions can have different relationships, ranging from periodic behavior to abrupt changes, and it is important to find the best kernel for a given dataset. To determine the most appropriate kernel, a Bayesian method is used in which prior distributions are assigned to the kernel's hyperparameters (random variables), which are then updated using the posterior distributions, with uncertainty reflecting the kernel's ability to describe the data.

2.6.6 Path Planning

Advances in autonomous navigation have enabled operations that require a lot from humans, such as agriculture [47] or others that pose a high risk of danger, such as mining [48] or underwater exploration [49].

Spontaneously, the components of this problem are the robot and its dynamics, the environment, and the initial and goal states. Many distinguishable solutions have been developed using different methods (see Table 2.3).

Artificial potential field methods such as VFF [50], VFH [51] and others let the robot endure the effect of the combined repulsive forces of obstacles, gravity and the target. This category offers a lower computational cost while being efficient in collision avoidance, but is subject to the risk of finding a local solution and being "trapped" in it.

Graph search algorithms such as Visibility Graphs [52] graphs that connect lines starting and ending at the vertices of visible obstacles to create a road map connecting the start and end positions, and then search for the shortest path, or Voronoi diagrams [53] that plot equidistant lines between the nearest obstacles to show the possible paths and add lines to connect this graph to the start and end positions.

Sampling-based algorithms simplify the search space by discretizing the continuous environment and connecting the sampled points that lie in a free space. Since there is no prior description of the obstacle space, the entire space is searched by a collision detector to check whether a particular robot position is in a free or occupied space.

Rapidly-exploring random tree (RRT) [54], randomly searches points around the robot position, keeping the points that are in free space and creating nodes in those points and attemping to connect to the initial node, creating one first edge. The process is repeated, incrementing the nodes and edges that differ from the previous ones, and ends when the node in the target region is created and connected, allowing extraction of a start-goal path.

Probabilistic roadmap (PRM) [55] starts by taking the sampled points, creating the nodes on all of them that belong to the free space, and then creating the connections that form the roadmap between not all of them, but only those that are within a user-defined radius of each other. The algorithms can be improved, e.g., by setting a limit on the number of connections attempted or by using k-nearest neighbors to select candidate nodes.

Table 2.3: Overall view of path planning methods.

Cathegory	Name	Method
Artificial Potential Fields	Virtual Force Field (VFF) [50]	Repulsing Forces From Obstacles and Target
Artificial Potential Fields	Vector Field Histogram (VFH) [51]	Repulsing Forces From Obstacles and Target
Graph Search	Visibility Graphs [52]	RoadMap from obstacle vertices
Graph Search	Voronoi Diagrams [53]	RoadMap Between Obstacles
Sampling Based	Rapidly-Exploring Random Trees (RRT) [54]	Incrementing Edges To Nodes In The Free Space
Sampling Based	Probabilistic Roadmap (PRM) [55]	Incrementing Edges To Nodes In The Free Space

3 Traversability Map Computation

In this chapter, the chosen method will be examined in more detail and its pipeline, as shown in Figure 3.1, will be presented. The algorithm chosen to perform the proposed traversability analysis was the one by *Shan et.al* [29]. This method was public available and also met our objectives. The lack of annotated forest datasets was not a problem for this algorithm, as it uses a LIDAR point cloud as input and does not rely on semantic segmentation.

This method takes as input a point cloud and odometry/localization information and produces an elevation map augmented with inferred cells to resolve the LIDAR sparsity. Elevation and occupancy values are then assigned to the map cells, defining the cells as traversable or non-traversable. By applying filters, which will be discussed, and from the data of this map, it becomes possible to create the occupancy grid that generates the cost map that allows the robot to perform autonomous navigation (see Figure 3.1).

The authors of the original paper presented a qualitative evaluation of their method. However, the scenarios chosen were mainly urban and, as such, not as challenging as the forested areas, so further evaluation is needed.

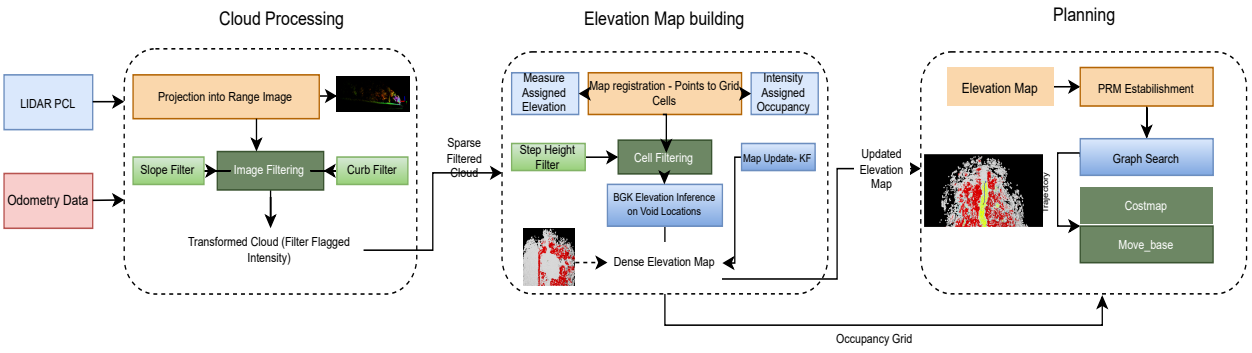


Figure 3.1: Flow chart of the presented TTA method.

3.1 LEGO-LOAM - Point Projection

The first step in any TTA method is always to register the successive LiDAR scans/RGB-D information to a common 3D representation. Range images are a good solution because they allow feature extraction. Our main goal, traversability analysis, also requires the detection of certain features to distinguish traversable / non-traversable terrain. LEGO-LOAM [34] takes the LIDAR point cloud and the position of the robot and computes the range using the well-known 3D distance formula with the coordinates of the sensor $[x_s, y_s, z_s]$ and the point $[x_p, y_p, z_p]$. Each point stores X, Y, and Z coordinates for its position, as well as an intensity value that measures the reflectivity of the surface providing a descriptor for different materials. The calculated distance then replaces the actual intensity value of the point (see Figure 3.2). In this way, a distance image similar to the one shown in Figure 2.4 is created, simplifying the process of calculating distances between points since the intensity measurement is used for the entire calculation.

$$Distance = \sqrt{(x_s - x_p)^2 + (y_s - y_p)^2 + (z_s + z_p)^2} \tag{3.1}$$

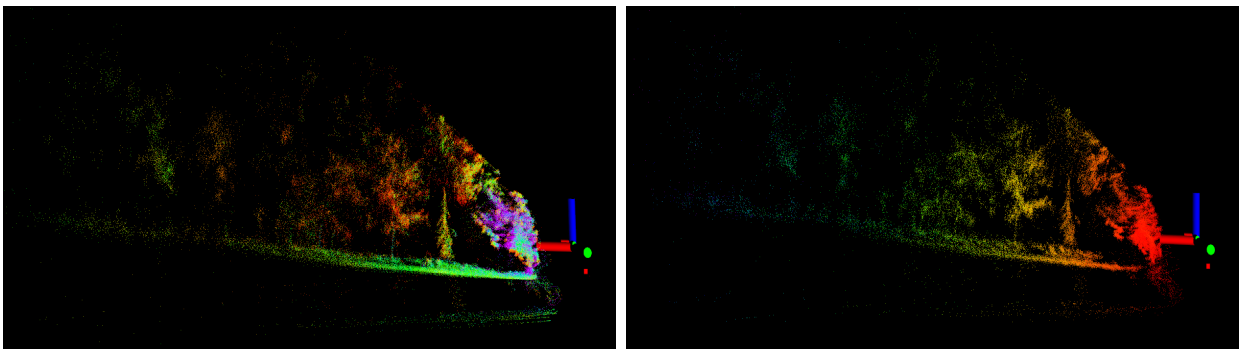


Figure 3.2: Point Cloud with intensity points before (left image) and after the intensity change for the range image (right image). Intensity depends on distance to origin but also on surface properties. In the left image we can notice bigger intensity values on leaves that are more distant from the sensor than tree trunks that are closer.

3.2 Filters

In this stage of the processing pipeline, traversability analysis is mainly based based on the following parameters:

- Slope
- Step Height
- Curbs

Since LIDAR loses precision in areas such as intensity the farther the points are from the sensor, filters are applied only within a user-defined, appropriate distance threshold.

When analyzing the point cloud and applying the filters, if the conditions that the filter looks for to flag the points as part of obstacles are met, the respective point intensity is changed to 100 for the points under consideration, flagging them as obstacles for the traversability. This is also taken into account for the elevation map. In this case, the elevation map is nothing more than a point cloud downsampled from the original LIDAR scan, in the form of square points (cells) with features resulting from the points coordinates - elevation - and intensity values - occupancy.

3.2.1 Slope Filter

Another fundamental terrain caracteristic to assess traversability is slope [56]. Navigating steep terrain is much more energy consuming. By going through the point cloud and analyzing defined narrow regions of interest and checking the variance over the three coordinates of the first and last points of the region, one can determine the steepness of the selected area.

$$Slope = \text{atan2}\left(\frac{\Delta z}{\Delta x^2 + \Delta y^2}\right) \tag{3.2}$$

3.2.2 Step Height Filter

Due to the kinematic limitations of the robot, abrupt changes in height between areas are not always traversable, and accurate detection of the occurrence of these features is critical. Downsampling the 3D point cloud into cells allows the creation of an elevation map [57], dividing the world into a 2D grid of uniformly sized cells that store information about the occupied area. In this case, one of the stored fields is the minimum and maximum height

observed in each cell. By evaluating the height difference δ_z within a cell, we can classify this area as an obstacle or free space, depending on the known capabilities of the robot used. Taking this into account, we can define a threshold for δ_z, beyond which the intensity I of the cell will be changed.

$$\delta_z = zmax - zmin \tag{3.3}$$

$$I = \begin{cases} 100, & \text{if } \delta_z > \text{HeightLimit} \\ 0, & \text{if } \delta_z < \text{HeightLimit} \end{cases} \tag{3.4}$$

3.2.3 Curb Filter

Curbs are essential features that distinguish passable paths from undesirable navigation areas [37]. Their usefulness is emphasized in urban scenarios where 90^{o} deviations may occur between the expected vehicle navigation area and restricted areas such as sidewalks. However, in forested areas, it is also common to come across some recognizable man-made paths, created with the intention of driving on them, when driving between raw, deep forest sections. Curb detection is critical because they often serve to delineate trails from the rest of the landscape, and most vehicles are unable to traverse these drastic elevation changes, or they could be damaged by them.

To check for the presence of curbs in the terrain, a region of interest is defined and the variation in range within adjacent points is computed, checking for monotonicity and distance between points. The curb filter works similarly to the step filter in that it looks for distance variation between neighboring regions, but it works by analyzing a smaller range of the immediately neighboring points, rather than all the points that a cell in the elevation map contains. Another difference is that it checks for variation in the elevation gradient between neighboring locations by analyzing a monotonicity in the elevation variance [58]. The results of this filter's application can be seen in Figure 3.3. Most curbs are detected on the path's borders where vegetation abruptly appears.

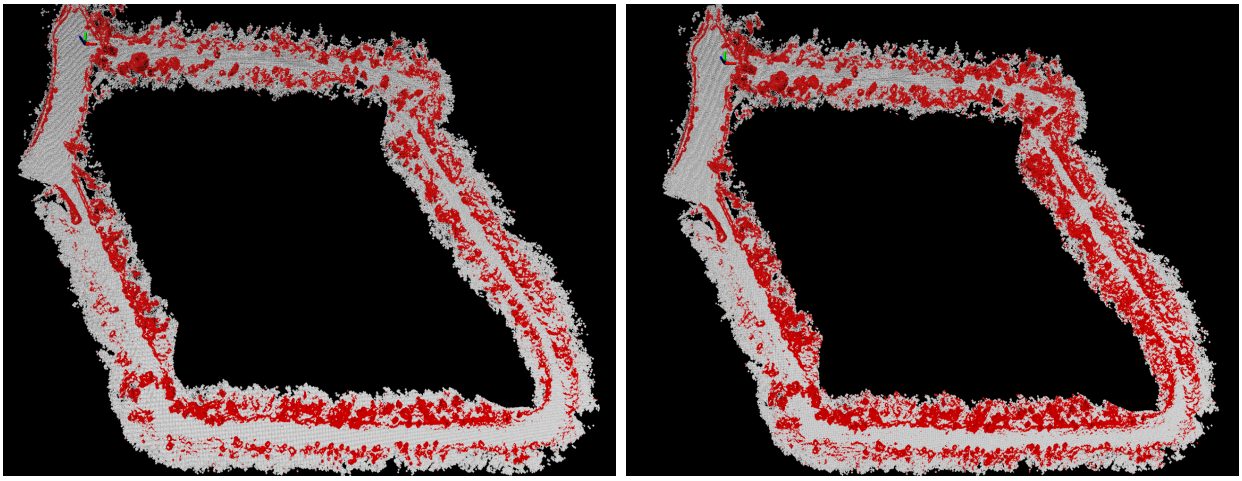


Figure 3.3: Elevation map showing differences in curb filter application. On the left image - no curb filter applied, right image - filter applied. The differences in the application of this filter are mainly on the borders between the human modified path and the deep forest.

3.3 Cloud Occupancy and Elevation Predicition - Bayesian Generalized Kernel Inference

To simplify the calculation of the traversability map, the data obtained from the previous filters are used to classify the cells of the elevation map as obstacles or free areas. If the cell is marked as an obstacle, the intensity of all points within the cell will be changed to the respective value for obstacle or free space, as mentioned before.

To address the problem of sparse data in the point cloud from LIDAR, grid cells are created and assigned near the cells created with previous point observations by predicting their height and occupancy with Bayesian kernel inference. This also increases the smoothness and generation speed of the height map.

This process begins by using the known 2D coordinate data, obtained from the data of the points observed by LIDAR, to compute a covariance matrix with value $k(x^*, x_i)$ and the known data about the height and occupancy y_i within the cells, to obtain an inference kernel that will form the basis for the inference. A training dataset that includes the already observed points will serve as priors in the Bayesian process to predict the occupancy and height of the other cells. The next step of the process is to apply the sparse kernel function given by the sum $\bar{k}(x^*)$ of all values in the covariance matrix to obtain a prediction $\bar{y}(x^*)$ of the height and occupancy of the cell, by translating each of them respectively into:

- Elevation - The z coordinate of the cell - the height in the 2.5D elevation map

- Ocupation - An intensity value for the cell : $\begin{cases} 100, & \text{if occupancy} > 0.5 \\ 0, & \text{if occupancy} < 0.5 \end{cases}$

$$\bar{k}(x^*) = \sum_{i=1}^{N} k(x^*, x_i) \qquad \bar{y}(x^*) = \frac{1}{\bar{k}(x^*)} \sum_{i=1}^{N} (k(x^*, x_i) \cdot y_i)$$

Figure 3.4: Pipeline of Bayesian inference process. Sparse 3D data results in an incomplete elevation map completed with inferred cells that are part of the calculation of the traversability map in place of useless void data.

3.4 Occupancy Update

Occupancy grid maps describe the environment in which the robot moves in cells of equal size. Each cell stores the probability value that it is occupied, providing information about where obstacles are located and which paths can be considered for navigation [39]. In this case, the input data is the filtered grid cells whose intensity has already changed, so a probability-based approach is used to calculate the occupancy value. Each cell of the grid starts with an occupancy value of 0.5 as long as there is no data on it yet. However, when the point cloud of the environment is processed and the elevation map is created, the values are updated based on the intensity information stored on the grid cells.

If a cell was previously marked with an intensity with a value of 100, it will have a higher priority value for the probability of occupancy given by the parameter p, and the log odds of occupancy will then be calculated as follows

$$\log \text{odds} = \ln\left(\frac{p}{1-p}\right) \tag{3.5}$$

Then the logistic regression equation is used to convert to a probability value.

$$\text{occupancy} = \left\lceil 100 \times \left(1 - \frac{1}{1 + \exp(\log\_\text{odds})}\right)\right\rceil \tag{3.6}$$

The goal in creating the occupancy grid is to compute a cost map. Cost maps are used by path planning algorithms to avoid obstacles and choose energy efficient paths for the vehicle. This allows for an increase in range and endurance, which increases effectiveness in

long-term missions such as forest clearing in our case. It is important to know the occupancy value of each location because the first criterion for traversable is the absence of an obstacle. To increase the efficiency, the occupancy values are discretized. Cells with an occupancy probability above a defined threshold are considered occupied (occupancy = 100) while cells with occupancy values below this threshold are considered free cells (occupancy = 0).

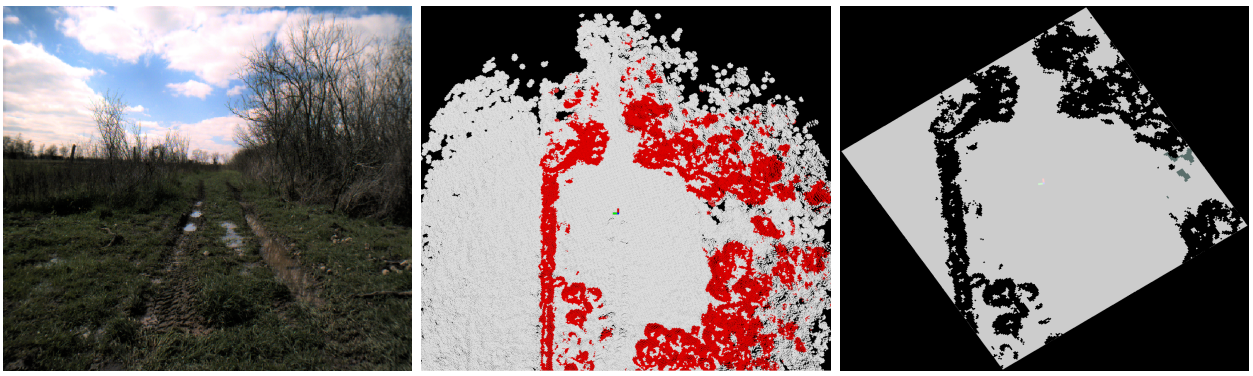


Figure 3.5: RGB image on the left, Elevation Map with obstacle cells in red and the clear path in white in the center. Occupancy grid on the right side. For occupied cells, the total occupancy value (1) appears in black on the grid, free cells are white, while unknown locations that have not been measured appear in gray as the occupancy data keeps the original value of 0.5.

3.5 Elevation Update - Kalman Filter

To update the cell elevation, a Kalman filter is applied to include new measurements. The equations of the filter are shown below. The current stored elevation value is represented by μ_{t-1} and the predicted value by μ_t, the current elevation variance σ_{t-1} and the predicted variance σt, the process error ϵ_t, the measurement noise ξ_t, which depends on the distance δ between the laser and the point, and a confidence factor ρ that is larger or smaller proportional to the times the point was observed, k_t the Kalman gain, $\hat{\mu}_t$ and $\hat{\sigma}_t$ the elevation value and the variance after prediction respectively, μ_t^* the measured z-coordinate of the point.

$$\begin{cases} \overline{\mu}_t = a_t \mu_{t-1} \\ \overline{\sigma}_t = a_t^2 \sigma_{t-1} + \epsilon_t \\ \xi_t = \delta * \rho \\ k_t = \frac{\sigma_t c_t}{c_t \sigma_t + \xi_t} \\ \hat{\mu}_t = \overline{\mu}_t + k_t(\mu_t^* - c_t \overline{\mu}_t) \\ \hat{\sigma}_t = (1 - k_t c_t)\overline{\sigma}_t \end{cases} \tag{3.7}$$

'a' and 'c' are parameters for the filter's prediction and measurement update models. These can have any value and their optimal combination is difficult to determine analytically. After some testing, they were finally left at 1, which gave the best results.

3.6 Traversability Map Building

After all the steps described above, it is possible to create and update the elevation grid map. The information about the elevation and occupancy of each cell will be updated as more observations are obtained, so that a map can be created in dynamic environments.

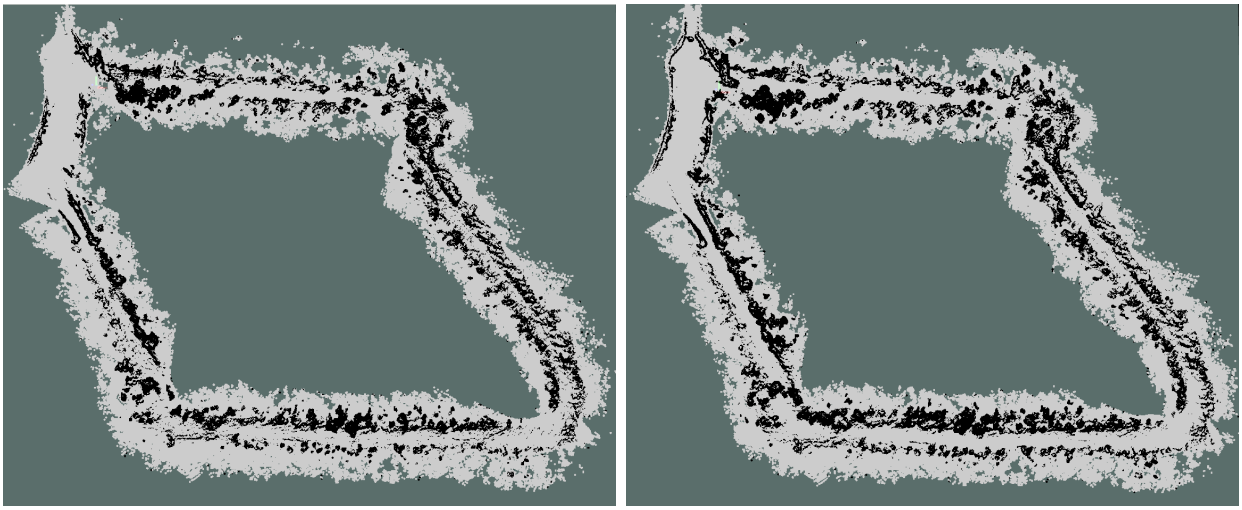


Figure 3.6: Occupancy grid after the first and second rounds to the Choupal dataset. More observations lead to changes in elevation and occupancy data. The map obtained after the second lap is much closer to the expected.

3.7 Cost map and autonomous navigation

With the occupancy grid in place, we can now create a cost map and initiate trajectory planning.

In our approach, the elevation map characterizes the environment by treating each cell as a potential state or pose. Our main goal is to find an optimal path that connects these states while minimizing the associated costs.

To achieve this, we use the probabilistic roadmap technique (PRM), a proven method for navigating unpredictable environments. The PRM workflow begins by generating random configurations, each representing a potential state. These configurations are then connected to their neighboring counterparts based on criteria such as k nearest neighbors or a specified distance threshold. Together, these connections form a roadmap that serves as our navigation guide.

In this process, each potential configuration corresponds to a node. These nodes are placed within the cells of the elevation map. It is important that the placement of the nodes occurs simultaneously with the creation of the elevation map to ensure that the elevation of each node remains up to date as the elevation map changes.

In the next step, the connections between the nodes are made by rectilinear segments called edges. Both the height differences and the distances between cells are evaluated based on a threshold set by the user. When these criteria, which include height and distance, are met, an edge is created. This edge then becomes a valid path segment that potentially contributes to the overall path for transitioning between nodes and ultimately reaching the destination. Our map is constantly updated as new measurement data is acquired. In cases where elevation changes and nodes are no longer neighbors, we immediately remove the corresponding edges.

Once the potential paths are identified, our planning process towards the target node begins, followed by an optimization phase. First, the system searches for the node closest to the robot's current position, while simultaneously checking the surrounding nodes for potential cost improvements. Once the lower cost node is identified, the robot is instructed to navigate to its next destination to continue its path. The search for the next destination is based on a queue and Dijkstra's algorithm [59], which works similarly to a breadth-first search. Our goal is not to identify the nodes with the most edges, but to identify the node closest to the destination to ensure that the shortest path is returned.

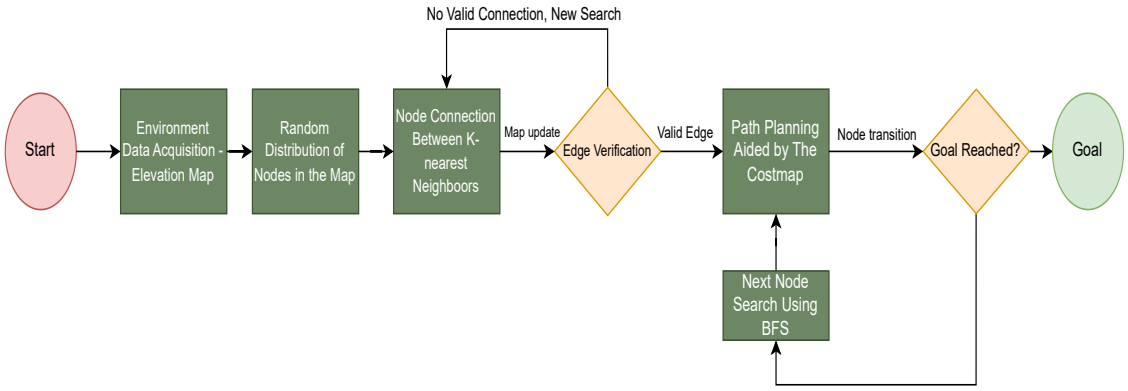


Figure 3.7: PRM Flow Chart.

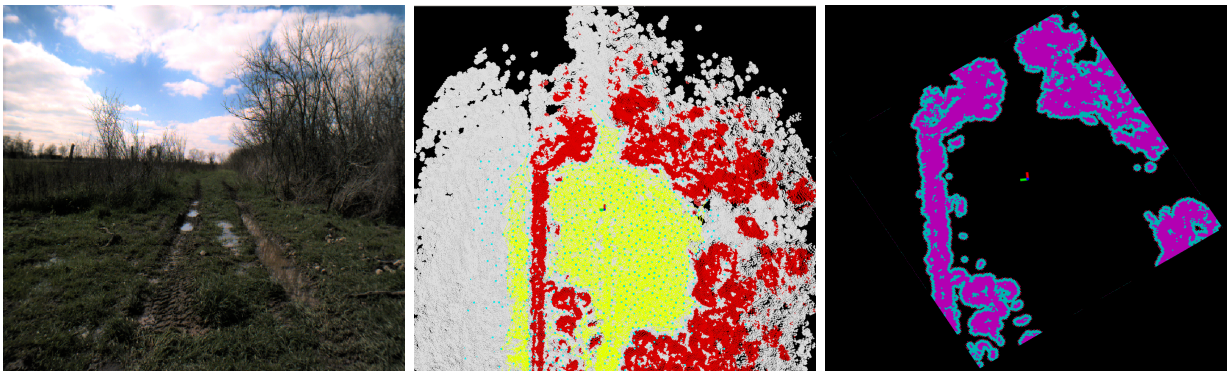


Figure 3.8: From left to right: RGB image, elevation map with PRM, and cost map of the scene. Obstacles and obstacle boundaries are marked with higher costs, which are taken into account during planning.

4 Performance Evaluation

This work is part of the SAFEFOREST project with the goal of finding the best approach for traversability analysis that can be implemented on the INGENIARIUS Ranger (see Figure 4.5). Ideally, we would apply this method online with the vehicle in a forest environment. However, it is not easy to gain access to a vehicle that weighs tons and bring it into our desired test environment.

With this in mind, the performance of the TTA method was evaluated offline. Three different datasets with different characteristics were selected. The criteria for selecting the datasets were the diversity of the environment and the presence of LIDAR scans as well as localization data provided by reliable sensor devices.

The evaluation was performed were performed using ROS melodic on Ubuntu 18.04 in a setup with 16GB RAM, an Intel Core i7-6700 CPU and a NVIDIA RTX 3060 GPU, which allowed us to process all input data in real-time. Rosbag files provided by each dataset included LIDAR point clouds used as input, RGB images used for visual confirmation, and localization data important for scene reconstruction.

Table 4.1 gives an overview of the datasets used in terms of:

- LIDAR Quality - Quality of the LIDAR scan considering the number of points per scan and the horizontal FOV.
- Forest representativeness - Environments classified as forest do not have a strict definition and can vary to a significant degree. The evaluation of this parameter is based on the task of forest cleanup, with the ideal environment being a scenario surrounded by living or dead trees and moving in a completely unworked terrain.

Table 4.1: Evaluation and description of the studied datasets. * Low Grade on the parameter, ** Medium Grade on the parameter, *** Excellent Grade on the parameter.

Author	LIDAR Quality	Forest Representativity	Odometry	Notes
RELLIS 3D	***	*	INS (GPS+IMU)	BEST LIDAR, Less Exigent Environment, Annotated
FRUC CHOUPAL	*	**	RTAB-Map	Cheap LIDAR, Limited FOV, Not Annotated
INGENIARIUS RANGER TRIALS	**	***	INS (GPS+IMU)	Tilted LIDAR, Cropbox, Not Annotated

4.1 RELLIS 3D-Dataset

Rellis 3D is a comprehensive collection of multimodal data acquired in an off-road environment containing 6,235 pixel-wise image annotations and semantic labels for 13,556 full LiDAR point cloud scans. This dataset was acquired at the Rellis campus of Texas AM University.

The dataset contains complete sensor data in ROS bag format, consisting of RGB camera images, high quality LIDAR point clouds, stereo imagery, and an inertial navigation system (INS) with high precision GPS measurements and IMU data.

The dataset is labeled for cathegories that are of interest in forestry, such as: Trunk, log, tree, pile, mud, grass or bush, and others.

This dataset does not include navigation in "deep" forests with tall trees and canopies, and should give us an indication of the algorithm's performance in an off-road environment. The input used to evaluate traversability is the point cloud acquired by the Velodyne HDL-

32E, a LIDAR sensor capable of acquiring data in a horizontal 360º FOV, which allows for a greater number of observations of all cells that remain in view after traversing, resulting in a more accurate and dense elevation map construction.

While this dataset effectively delineates the forest components and the path with minimal overlap, an interesting observation stands out: Certain segments of the free path appear to contract, a phenomenon not entirely consistent with the corresponding RGB images. This narrowing can be attributed to the presence of hanging objects in the trees. Although these objects do not obstruct the path, they are detected as extensions of the obstacles that delineate the path boundaries. The RGB images and corresponding elevation maps of the dataset in question can be seen in Figure 4.1, with this phenomenon clearly visible in the data in the bottom two rows of the figure.

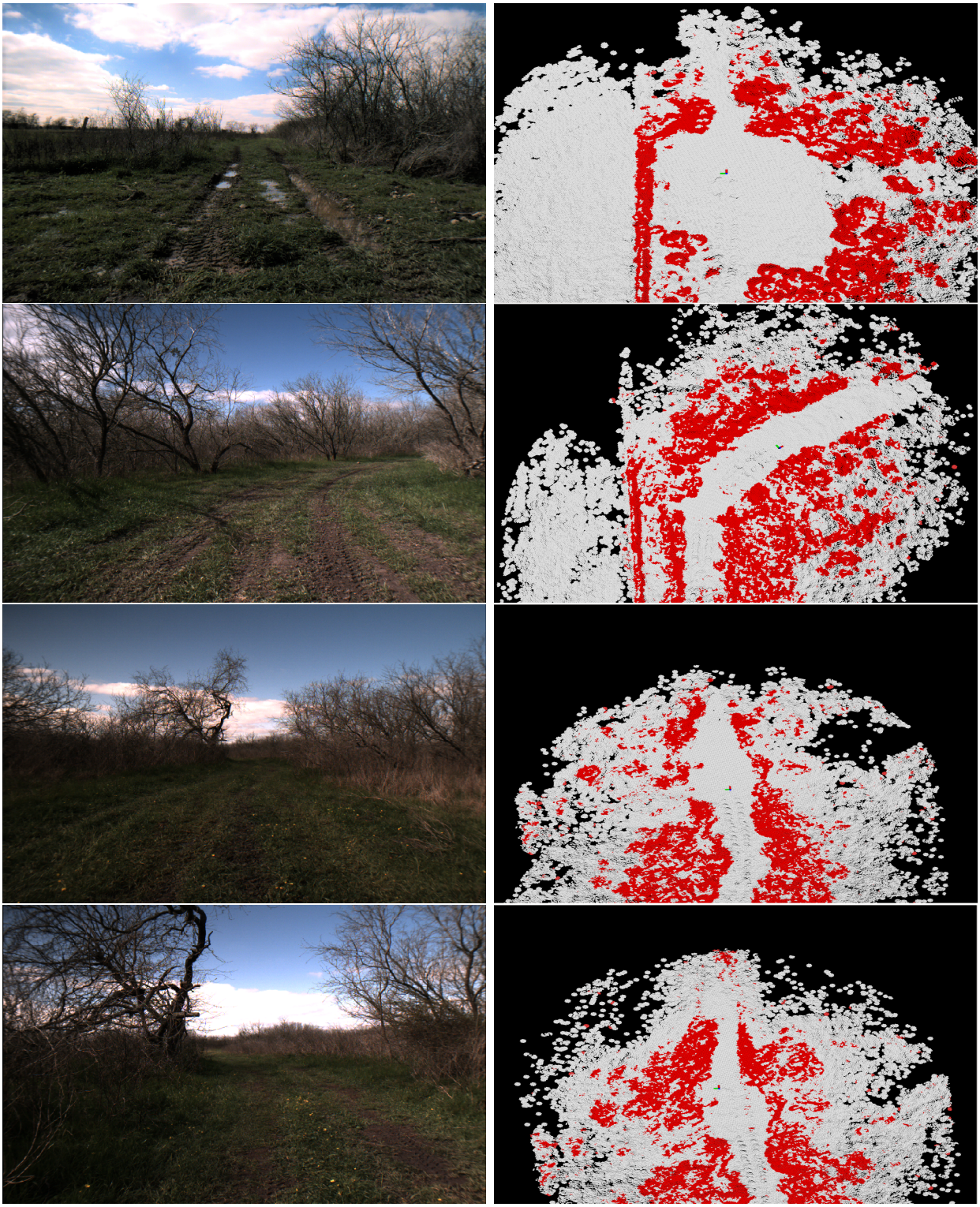


Figure 4.1: RELLIS 3D dataset - The left column shows the recording environment, a hiking trail surrounded by trees on both sides, with the corresponding elevation maps in the right column. In the 2 bottom rows we can see a narrowing of the free path in the elevation map, not visible in the RGB image, caused by hanging objects on the trees.

4.2 Choupal FRUC Dataset

The next dataset was produced by our colleagues in FRUC and was recorded in the Choupal National Woods in Coimbra [60]. This dataset contains a map-referenced localization published as a ROS message of type */navmsgs/odom*. This message can be used to register sensor data in a fixed coordinate system and contains several important features in forestry robotics like trees, bushes, trunks, etc.

The 3D LIDAR point cloud was acquired using a solid-state Livox MID -70 LIDAR, an inferior device compared to the LIDAR used in other datasets, providing fewer points per scan. On the other hand, a different non-repeating petal scan pattern that improves reconstruction results is used (see Figure 4.2).

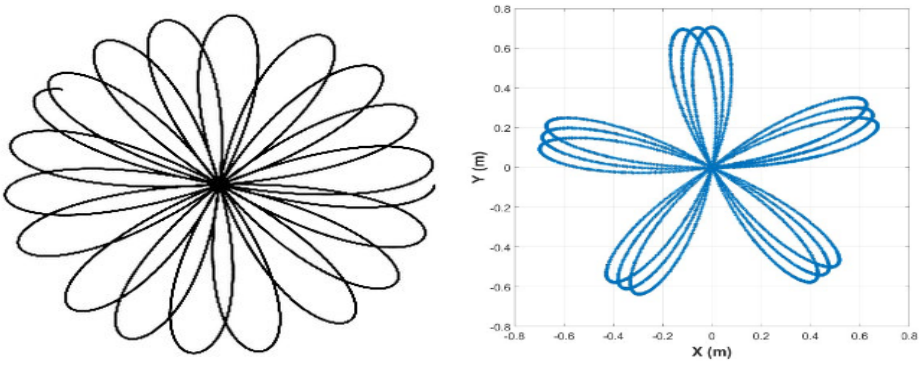


Figure 4.2: Livox Mid LIDAR petal capture pattern in the left image. On the right, we can see the area targeted by 3 consecutive LIDAR scans. Each scan covers a new area.

This LIDAR has not proven to be the most accurate for traversability in a difficult forest context. Due to the lack of a 360^{o} horizontal FOV, the number of observations for each cell is lower than for the mechanical LIDAR and therefore the inference process is not as accurate due to the lower confidence factor of the Kalman filter to update the map with new readings.

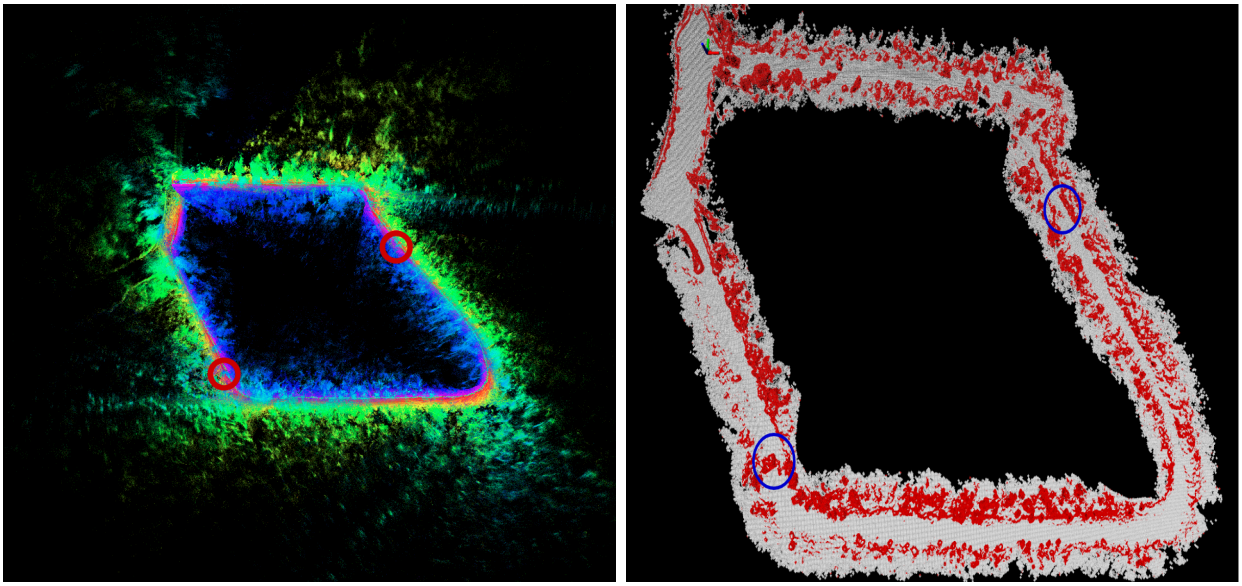


Figure 4.3: Livox Point Cloud on the left and elevation mapping with traversability data output on the right. Unexpected cells with obstacles match with locations with branches along the path. The red circles on the left present locations where canopi is above the trail. The same locations were classified as obstacles in the elevation map and are highlighted with yellow circles.

Nevertheless, the recorded dataset includes two laps around the same circuit, providing more observations and mitigating this problem, obtaining more satisfactory results for the inference and update on the map (see Figure 4.4). Another challenge of the smaller horizontal FOV is that only one side of the obstacle is observed. In urban environments, there is usually no valid path behind cells that are considered obstacles here (e.g., walls, sidewalks). In forested environments, trees are indeed obstacles, but most likely, there is traversable space behind them. When only one side of the tree is observed, the cells behind it rely solely on inference using data from the observed surface of the tree and the nearest clear space and the elevation of the space between the two is the average of the two locations.

An already familiar phenomenon is observed within the results obtained from this dataset. Certain areas that one would expect to be part of a clear path are instead designated as non-traversable. While this discrepancy may not be immediately evident when examining the RGB data, due to the sensor's positioning, a closer analysis of the point cloud reveals that these unexpected non-traversable areas align with locations where tree canopies overlap the path. This discovery confirms that we are grappling with the hanging object problem, as illustrated in Figure 4.3.

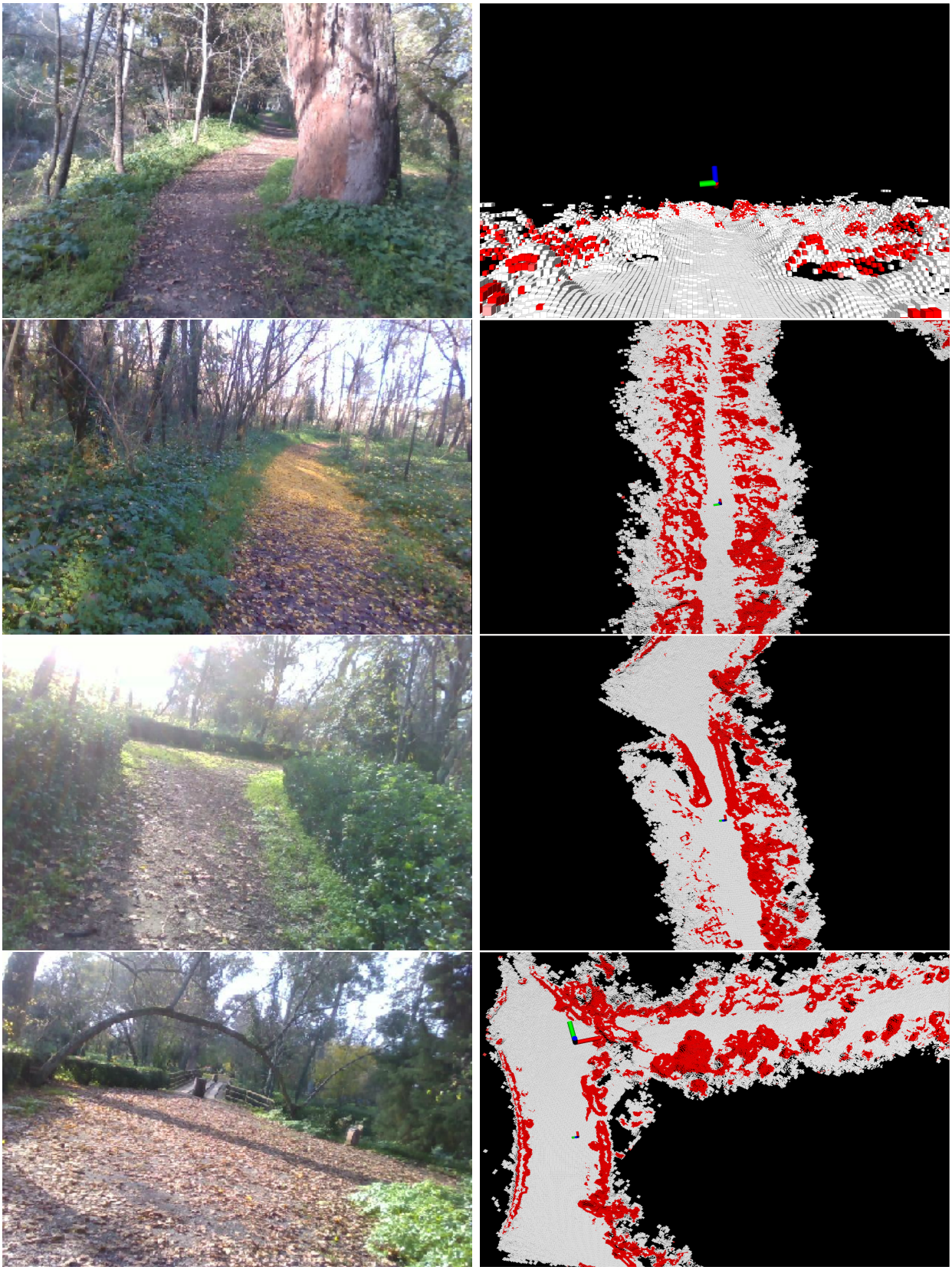


Figure 4.4: RGB image of the scene on the left and elevation map on the right. The elevation map is able to distinguish the path cells as less elevation and describe large obstacles, such as the tree on the right side of the image, which is seen as a high object on the far side of the elevation map. Traversability classification is accurate and corresponds the appearance of the path seen in the camera images on the elevation map.

4.3 Ingeniarius Ranger Trials

The final dataset was acquired by Ingeniarius, a company member of the SAFEFOREST project consorcium, at its HQ in Alfena, Portugal. This dataset contains RGB-D camera data, LIDAR point clouds acquired from a VLS-16 LIDAR with the already reffered "fan" acquisition pattern that allows a 360º FOV, IMU and GPS data.

The goal of these experiments was to obtain a multimodal dataset to assess traversability, either by segmenting RGB images, LIDAR data, or both.

The sensors were mounted on the Ingeniarius Ranger, a heavy tracked vehicle based on a Bobcat and equipped with a mulcher to remove burning material - objects that burn easily, such as dead trees, bushes, or leaves.

Figure 4.5: Ranger Trials - The Ranger is a tracked vehicle equipped with a mulcher for forest cleanup and sensors for autonomous navigation. At the top of the platform, the LIDAR sensor is aligned with the top of the end effector.

To make this cleaning task more efficient, the robot must move autonomously in forest areas. As for data collection LIDAR, the setup consists of 2 sensors, one of which is mounted on the front of the Ranger near the end effector that performs the mulching of the identified fuel. Due to the ability of the VLS-16 LIDAR to acquire data in a horizontal 360º FOV, the input point cloud containing the ambient reconstruction data must be filtered with a crop box that omits points corresponding to the platform structure that has a fixed position relative to the LIDAR sensor. If these readings were kept, the sensor would constantly take into account the presence of an obstacle above the ground and change the elevation map due to the presence of a hanging object, which would affect the elevation values of the cells in

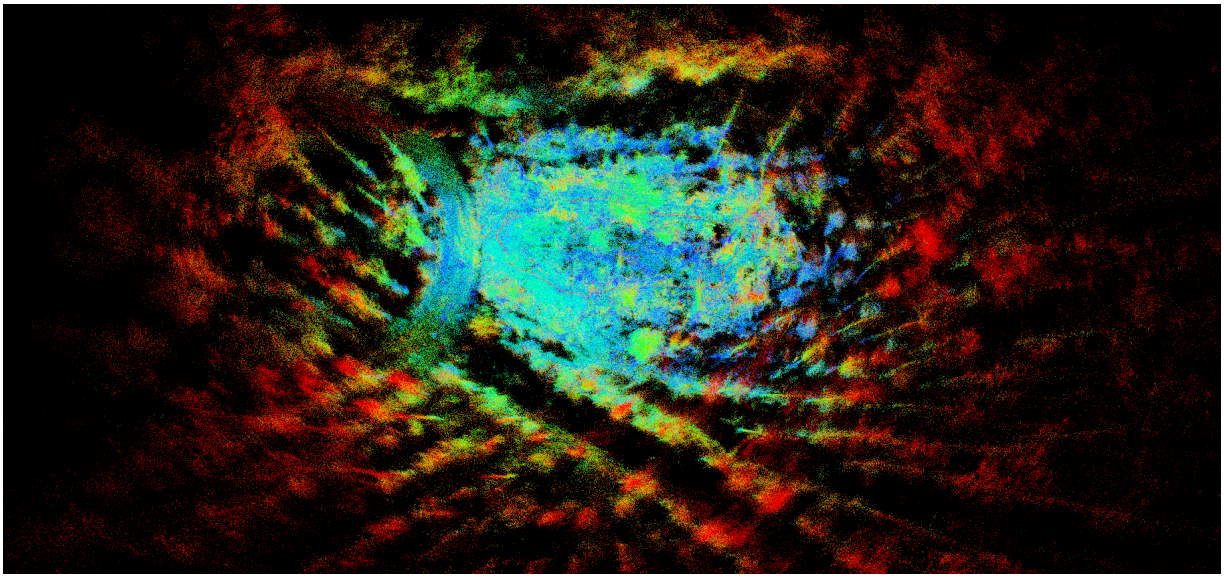


Figure 4.6: LIDAR point cloud of the experimental site, which includes both rear and front LIDAR. The tilt of one of the sensors allows more data to be collected from objects at higher altitudes, such as tree canopies, providing a satisfactory reconstruction of the environment.

front of the trajectory, the traversability classification in the region where the end effector ends and the ground begins, with a large elevation variation on the cells in these areas(see Figure 4.7.

Another process affected is path planning, as the PRM must be set to consider links between nodes that are close enough to each other to balance the efficiency of the path, the computational power required (if it is too close, it must process more edges), and the path options (larger edges could be invalidated by an obstacle that can be easily bypassed with a small deviation).

Another particularity of the sensor setup, is the 30^{o} tilt of the front face LIDAR, which changes the range of the point capture area. In the front direction of the sensor, the detection range is reduced because it is directed towards the ground, the closest "object". In the backward direction, the tilt allows for greater point acquisition at height, enabling the capture of details in tree canopies.

It has been shown that the point cloud from the tilted LIDAR does not perform well with the proposed traversability mapping algorithm(see Figure 4.8. The excessive number of points belonging to canopies and other hanging objects leads to large changes in the observed cells of the elevation map where traversability is evaluated by perceiving obstacles that are in fact inexistent for most vehicles. In addition, the detection of points at higher heights affects the inference of cell height, resulting in many cells being rated as occupied and non

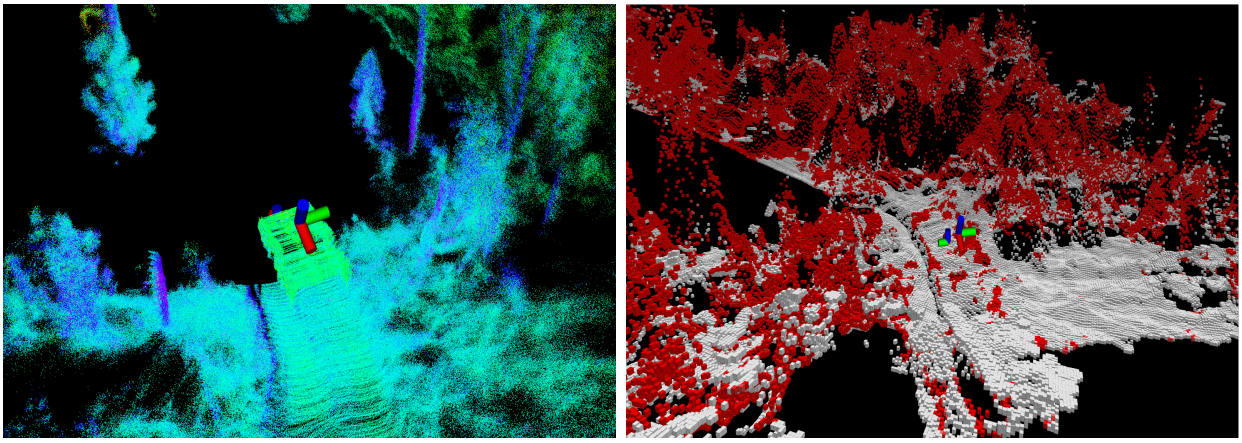


Figure 4.7: The need to filter the point cloud around the ranger. The points corresponding to the surface of the platform would affect the elevation map, leading to the detection of obstacles at the boundary between the sides of the platform and the ground due to the difference in elevation.

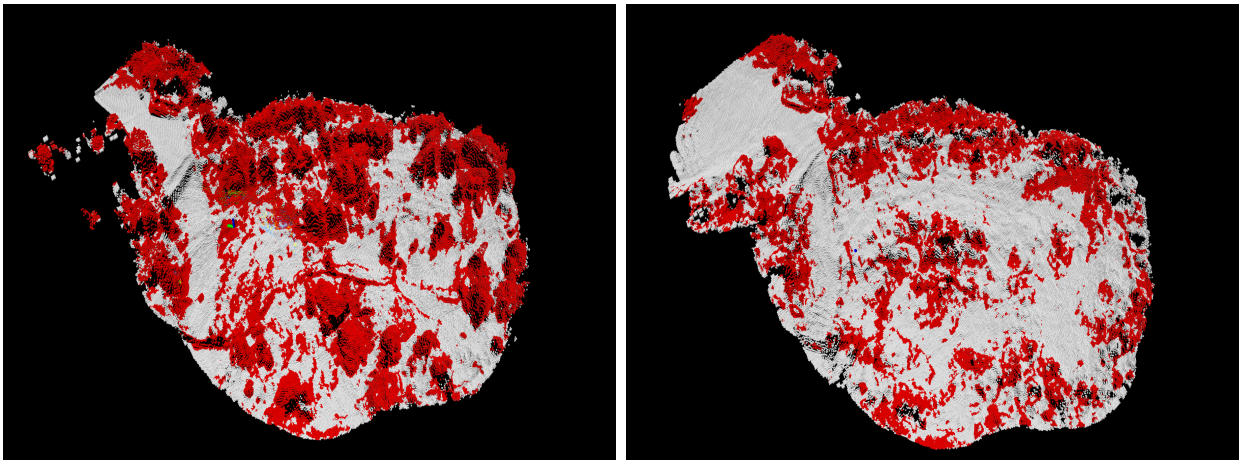


Figure 4.8: Ranger Trials - the left image shows the map with the front tilted LIDAR, the right image shows the map with the back LIDAR, almost parallel to the ground. Larger z-coordinates of the canopy points cause the map with the tilted LIDAR to have larger elevation values within the cell and disparities between cells.

traversable, as the height difference between points in the same cell is too large (see Figure 4.9) .

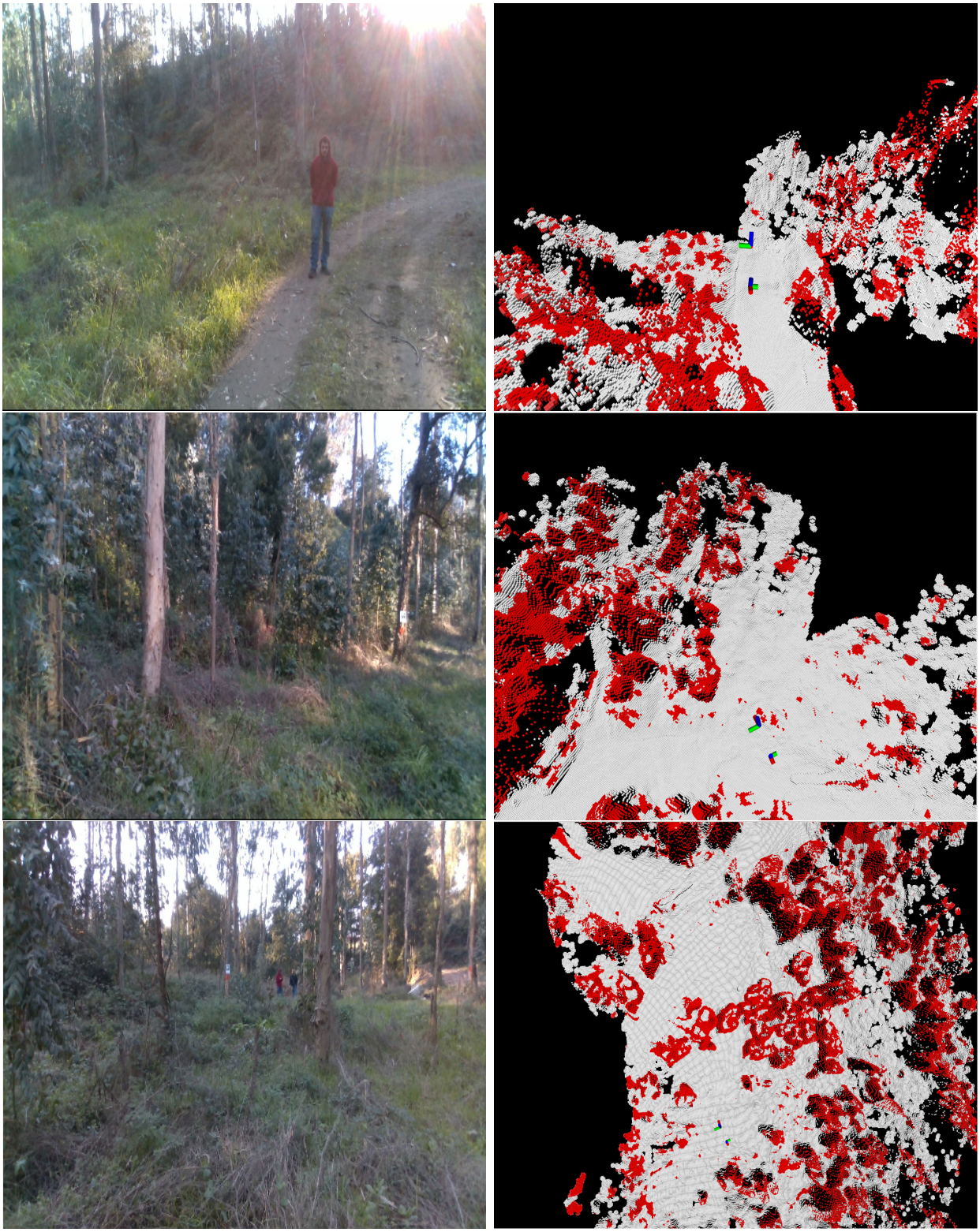


Figure 4.9: The left column shows RGB images of the trials site with the corresponding elevation maps shown on the right column. The Tilted Referential represents the front facing LIDAR aligned with the movement of the robot (direction of the red axis) and the Referential that points in the opposite direction is the back side LIDAR.

4.4 Hanging Obstacles Correction - Point Cloud Filtering

The fact that this method is based on a 2.5D representation of the world has remarkable implications. This approach offers computational advantages by reducing complex 3D environments to a 2D grid. However, it also brings limitations when it comes to accurately capturing certain features of the environment.

In the case of forest environments, a common challenge arises from the presence of hanging objects. Trees with meter-high canopies or branches often influence the cells of the elevation map to which points from these hanging objects are assigned. This influence leads to significant height differences within the affected cells. These variations result from the 2.5D approach of the method, where the map cells are positioned based on the X and Y coordinates of the observed points and their height is determined by assigning the maximum Z value within each cell.

This event causes the traversability assessment to incorrectly classify the cell as an obstacle. In fact, for ground vehicles, we only need to look for obstacles that are between the ground and the height of the vehicle, since there is no risk of colliding with obstacles above.

By determining the height of the vehicle, we can discard all points above the vehicle and perform the same analysis with greater confidence.

In the proposed datasets, all LIDAR sensors were on the platform on which they were mounted. This means that we could consider filtering points with a z greater than 0 with respect to the sensor's coordinate system.

The first dataset did not contain many hanging objects along the expected clear path, but filtering still allows us to see the differences caused by discarding points from small objects that protruded into the road (see Figure 4.10).

The Choupal dataset provided us with a good basis for the study because it contained several tree canopies that crossed the path. By filtering these points (see Figure 4.11), the obstacles appeared more clearly on the elevation map and the classification of traversability was more accurate because broad tree canopies did not blur the obstacle (in the majority trees) and open space boundaries (see Figure 4.12).

The most striking differences occurred in the Ingeniarius data et. Trimming the readings above a certain height (see Figure 4.13) (this time above the X-axis, since the tree tops appear in that direction due to the orientation of the LiDAR) limited the range of the sensor, preserving only the front LIDAR limited distance measurements, but the improvement in precision was remarkable.

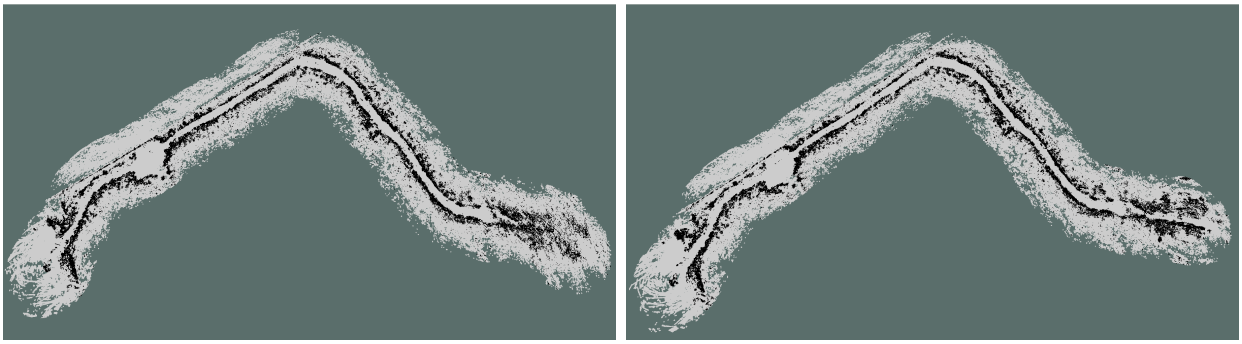


Figure 4.10: Occupancy grid of the RELLIS 3D path. The left image is result of the original cloud. On the right image, the point cloud was cropped in Z to the sensor's height. Shrinkage on the path (occupied edges in some places) due to hanging objects was eliminated.

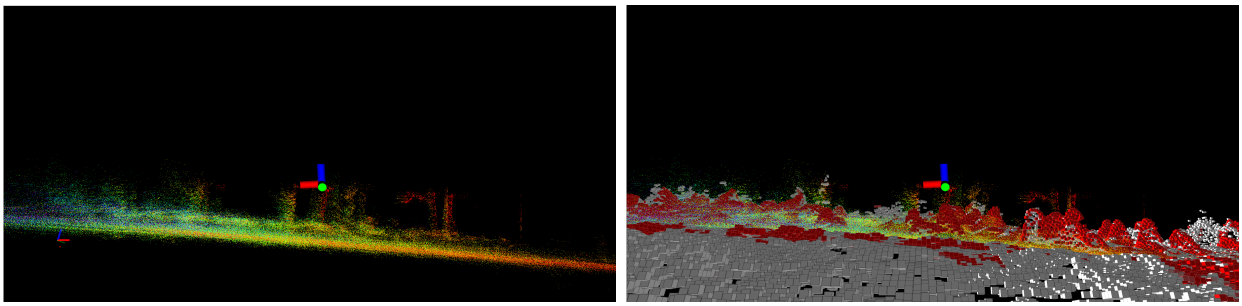


Figure 4.11: The acquired point cloud is cropped at the height of the sensor (left image) because it is mounted on top of the platform. The bottom parts of the trees are better defined and improve the elevation map (right image).

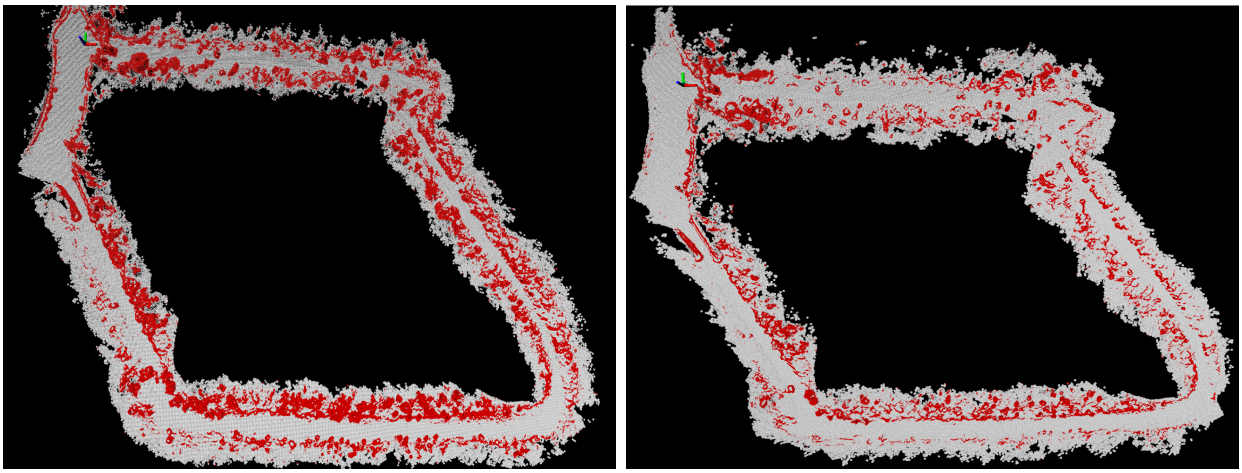


Figure 4.12: Choupal dataset global elevation map before (left) and after filtering (right). The free path that is shown is now closer to the expected.

However, due to the terrain in which the data was captured, a portion of untreated forest with bushes surrounding the trees, and due to the limited field of view, precision was limited as the flagged obstacles spread out from the bushes (see Figure 4.14).

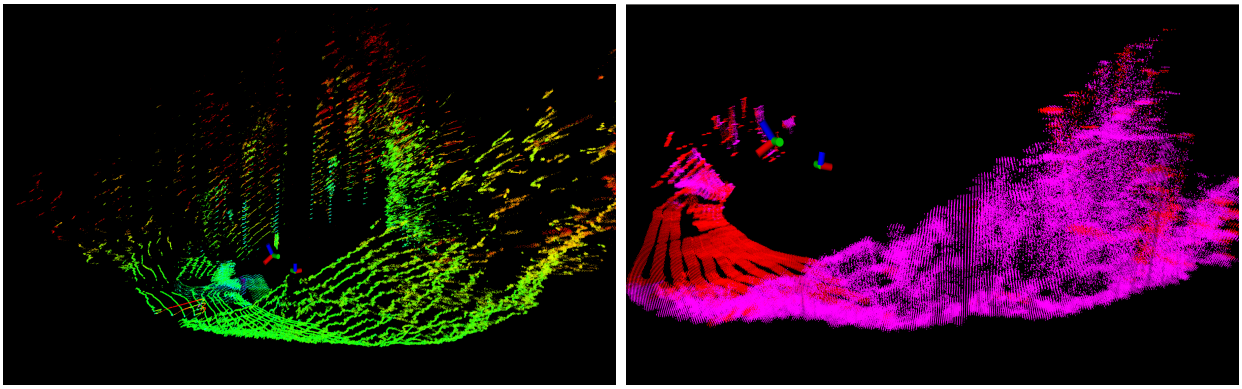


Figure 4.13: Front LIDAR from the Ingeniarius dataset before (left image) and after filtering (right image). The canopies that are behind and higher than the sensor are removed, leaving only data from the regions in front of the sensor and on the sides.

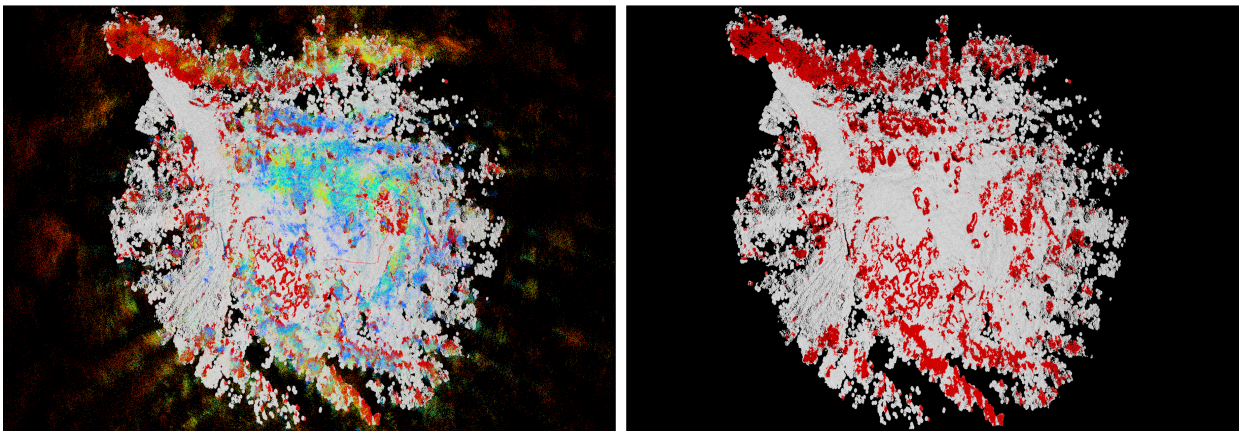


Figure 4.14: Ingeniarius Elevation map with Front LIDAR point cloud was entered, which also shows part of the point cloud from the back LIDAR. During the trial, a person walked next to the ranger. The drift of the LIDAR readings of this moving object allows us to see the trajectory performed in the form of the red line above the elevation map (left image). As expected, the trajectory corresponds mostly to open areas and crosses cells calculated as occupied in zones with denser canopy. On the right image, the elevation map is isolated.

To solve this problem, we increased the resolution of the map by reducing the cell length by half. This allowed us to achieve an improvement in distinguishing between obstacles and free space as seen in Figure 4.15.

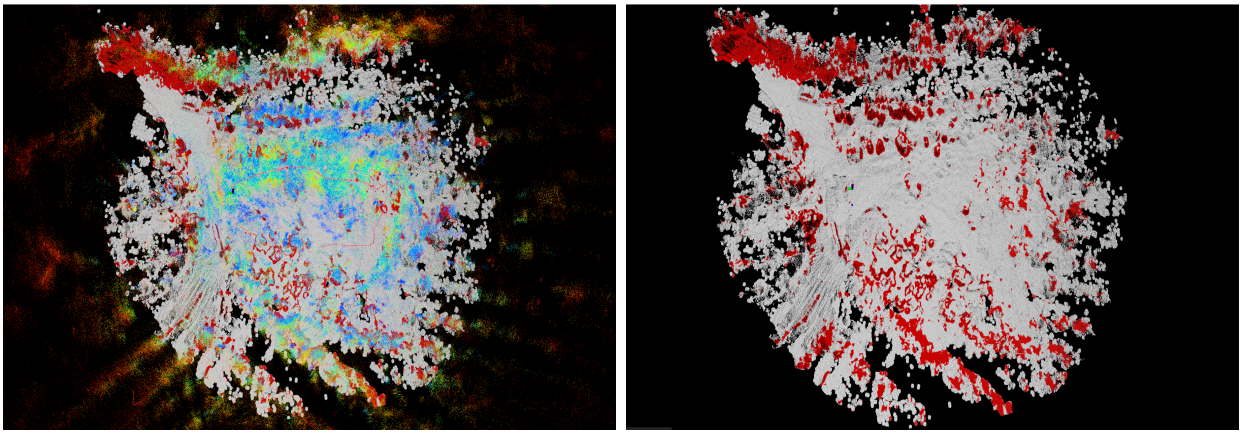


Figure 4.15: Same content as in the previous figure, with the algorithm running with half-length cells. Occupied and free cells are now easier to distinguish, as we can confirm by the simpler visualization of the trajectory through the drift captured by the back LIDAR.

5 Conclusion and Future Work

Having completed our work, we conclude that the proposed objectives have been met.

First, we conclude that the selected TTA method worked well for the data available to us. In order to perform this test in different environments, our only option for traversability analysis was to input LIDAR data, because although RGB data are available in most datasets, annotated data are required to evaluate traversability with the segmentation methods, and these are difficult to find.

We also show that the proposed method, already proven in urban environments, can also be successfully applied in forested environments, albeit with some adjustments.

All three datasets studied were rich in forest features, leading to the conclusion that despite the differences, the most salient features are found in the perceived paths, ranging from off-road trails to man-made paths to wild space in deep forest.

Other differences included the height of trees, the presence of tree canopy above the observed path, or the ability to detect the entire surface of the obstacle, which depended on the FOV of the dataset LIDAR. One can also conclude that the proposed method can be seamlessly applied to different sensory setups with acceptable results.

Despite the good results provided by the method even in forest environments, improvements are needed to achieve optimal analysis and navigation. To further improve this method, some problems or non-ideal aspects should be addressed:

- Hanging Objects - In forest landscapes, trees have many hanging objects that do not hinder navigation, such as branches or leaves that are above the robot's body. Since this method evaluates traversability in cells, we may have points in a cell with free traversable space, but we may also encounter points of these objects that enter the calculation for the height value of the cell and classify the free space as occupied due to the large difference between the minimum and maximum height of the points in the cell. Solutions such as those investigated in this work are only partially suitable, since clipping the point cloud is not ideal, because data is lost, and the resolution of

the elevation map can only be increased to a certain degree, since computational costs increase with the number of cells.

- Morphological Operators - When navigating off-road, the platform and its sensors shake more than in urban environments, leading to misjudgments in some cells and their classification, invalidating paths that should be invalid/valid. On the other hand, using an analysis based on LIDAR can cause problems with certain LIDARs if the obstacles do not have structured geometric features. A tree trunk could appear incomplete if the LIDAR does not have a 360^{o} FOV, and sparse data could result in missing classification on one side of the obstacle that is considered clear on the other side, which does not have real data, only inferred data. Even if LIDAR has the ideal FOV and can detect all sides of the obstacle, an empty area is left in the middle because LIDAR is not able to penetrate surfaces. If the empty area is surrounded, there are no valid paths on that location but it can become a problem if gaps appear, which is not unlikely to happen due to the sparsity of the data.

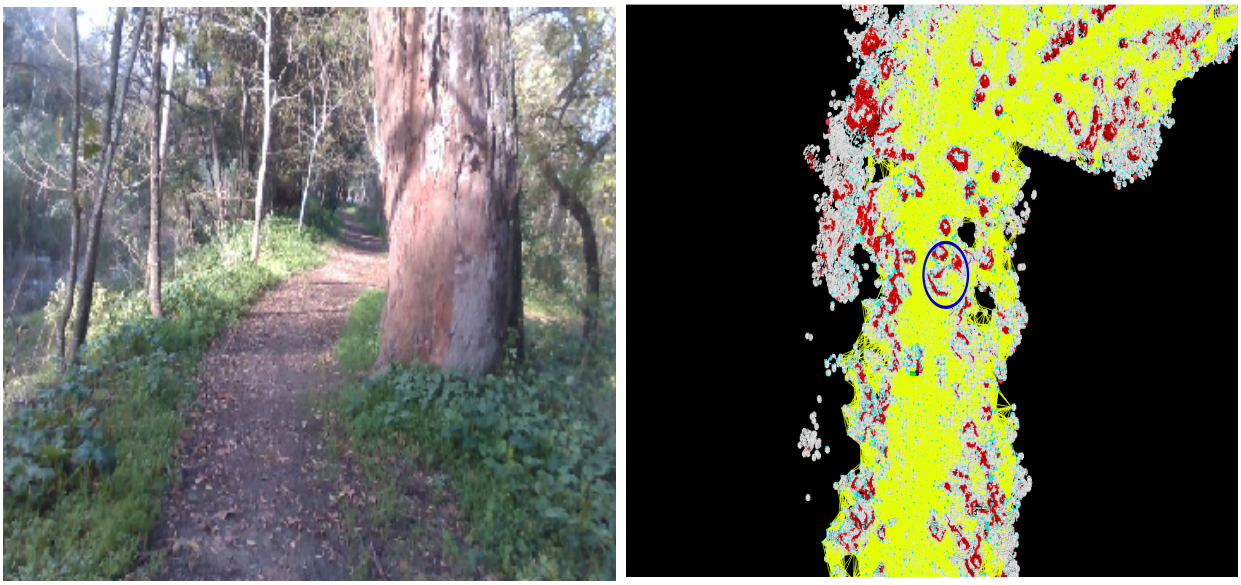


Figure 5.1: Scanning only one side of the object will result in an incorrect estimate of occupancy. If obstacles are classified as free space, they may be considered a good path option by the planner, resulting in collisions or requiring replanning after a close position is reached that allows more accurate measurements. In the lower left portion of the blue ring in the elevation map (right image), we see the side of the tree that was captured by the RGB image. Right next to it, the space is wrongly classified as free, which means that PRM edges are allowed inside the tree.

In urban environments, this is usually not the case because the effects are more clearly defined or easier to observe - if you are driving on a street, there are defined curbs, the surrounding buildings have recognizable walls, the cars are on the same street level and their tops can be easily observed, the poles are thin enough to fill a cell that takes into account the LIDAR readings.

To solve this problem, morphological operators such as erosion, dilation, opening and closing can be useful. These operators are usually used in computer vision applications that use images. Even if we do not use images, the input to the path planning node is the occupancy grid, which can be seen as an image in which each pixel has a value of 0 and is white (free) or has a value of 1 and is black (occupied). With the proper resolution of the grid, we can use, for example, erosion to remove an isolated occupied point from the path, since it is most likely not an obstacle, or use dilation to mark free cells surrounded by occupied cells as part of the obstacle.

- Online Testing - The analysis of the algorithm was based on the results of offline data that had been previously acquired and stored. The performance of the entire process of acquiring, processing, and performing inference is important for the proposed application and may vary with the requirements of an online test.

6 Bibliography

[1] Lara Vilar del Hoyo, María Isabel, and Fiorella Vega. Logistic regression models for human-caused wildfire risk estimation: analysing the effect of the spatial accuracy in fire occurrence data. *European Journal of Forest Research*, 130:983–996, 2011.

[2] Juil Sock, Jun Kim, Jihong Min, and Kiho Kwak. Probabilistic traversability map generation using 3d-lidar and camera. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5631–5637, 2016.

[3] Ji Zhang and Sanjiv Singh. Loam : Lidar odometry and mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, pages 109–111, 01 2014.

[4] H. Lassiter, Travis Whitley, Benjamin Wilkinson, and Amr Abd-Elrahman. Scan pattern characterization of velodyne vlp-16 lidar sensor for uas laser scanning. *Sensors*, 20:7351, 12 2020.

[5] Tao wu, Hao Fu, Bokai Liu, Hanzhang Xue, Ruike Ren, and Zhiming Tu. Detailed analysis on generating the range image for lidar point cloud processing. *Electronics*, 10:1224, 05 2021.

[6] San-Miguel-Ayanz J, Durrant T, Boca R, Maianti P, Liberta' G, Oom D, Branco A, De Rigo D, Ferrari D, Roglia E, and Scionti N. Advance report on forest fires in europe, middle east and north africa 2022. (KJ-NA-31-479-EN-N (online)), 2023.

[7] Carl-Friedrich Schleussner, Inga Menke, Emily Theokritoff, Nicole van Maanen, and Alexandrine Lanson. Climate impacts in portugal. *Climate Analytics, Berlin*, 2019.

[8] Francisco Moreira, O. Viedma, Margarita Arianoutsou, Thomas Curt, Nikos Koutsias, Eric Rigolot, Anna Barbati, Piermaria Corona, Pedro Vaz, Gavriil Xanthopoulos, Florent Mouillot, and Ertugrul Bilgili. Landscape-wildfire interactions in southern europe: Implications for landscape management. *Journal of environmental management*, 92:2389–402, 10 2011.

[9] Cristina Ribeiro, Sandra Valente, Celeste Coelho, and Elisabete Figueiredo. A look at forest fires in portugal: technical, institutional and social perceptions. *Scandinavian Journal of Forest Research*, 30, 11 2014.

[10] European Commission, Joint Research Centre, H Costa, D De Rigo, G Libertà, T Houston Durrant, and J San-Miguel-Ayanz. *European wildfire danger and vulnerability in a changing climate : towards integrating risk dimensions : JRC PESETA IV project : Task 9 - forest fires.* Publications Office of the European Union, 2020.

[11] F.C. Dennis. Fire-resistant landscaping - 6.303, Jul 2018.

[12] Manuela Carreiras, António Dinis Ferreira, Sandra Valente, Luuk Fleskens, Oscar gonzalez pelayo, José Rubio, Cathelijne Stoof, Celeste Coelho, Carla Ferreira, and Coen Ritsema. Comparative analysis of policies to deal with wildfire risk. *Land Degradation and Development*, 25, 01 2014.

[13] Matija Landekić, Ivan Martinić, David Mijoč, Matija Bakarić, and Mario Šporčić. Injury patterns among forestry workers in croatia. *Forests*, 12(10), 2021.

[14] Miloš Gejdoš, Maria Vlckova, Zuzana Allmanová, and Zaneta Balazova. Trends in workplace injuries in slovak forest enterprises. *International Journal of Environmental Research and Public Health*, 16:141, 01 2019.

[15] Micael S. Couceiro, David Portugal, João F. Ferreira, and Rui P. Rocha. Semfire: Towards a new generation of forestry maintenance multi-robot systems. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 270–276, 2019.

[16] Alessandro Giusti, Jérôme Guzzi, Dan C. Cireşan, Fang-Lin He, Juan P. Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, Davide Scaramuzza, and Luca M. Gambardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667, 2016.

[17] Xiaobin Xu, Lei Zhang, Jian Yang, Chenfei Cao, Wen Wang, Yingying Ran, Zhiying Tan, and Minzhou Luo. A review of multi-sensor fusion slam systems based on 3d lidar. *Remote Sensing*, 14(12), 2022.

[18] Paulo Borges, Thierry Peynot, Sisi Liang, Bilal Arain, Matthew Wildie, Melih Minareci, Serge Lichman, Garima Samvedi, Inkyu Sa, Nicolas Hudson, et al. A survey on terrain

traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges. *Field Robot*, 2(1):1567–1627, 2022.

[19] Panagiotis Papadakis. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Eng. Appl. Artif. Intell.*, 26:1373–1385, 2013.

[20] Iram Noreen, Amna Khan, and Zulfiqar Habib. Optimal path planning using rrt* based approaches: A survey and future directions. *International Journal of Advanced Computer Science and Applications*, 7, 11 2016.

[21] Steven C. Martin. *Proprioceptive sensing of traversability for long-term navigation of robots*. PhD thesis, Queensland University of Technology, 2018.

[22] Daniel Maturana, Po-Wei Chou, Masashi Uenoyama, and Sebastian Scherer. Real-time semantic mapping for autonomous off-road navigation. In *Proceedings of 11th International Conference on Field and Service Robotics (FSR '17)*, pages 335 – 350, September 2017.

[23] A. Valada, G. Oliveira, T.Brox, and W. Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *International Symposium on Experimental Robotics (ISER 2016)*, 2016.

[24] Juhana Ahtiainen, Todor Stoyanov, and Jari Saarinen. Normal distributions transform traversability maps: Lidar-only approach for traversability mapping in outdoor environments. *Journal of Field Robotics*, 34, 2017.

[25] Mary B. Alatise and Gerhard P. Hancke. A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access*, 8:39830–39846, 2020.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, pages 234–241. Springer International Publishing, Cham, 2015.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[28] David M. Bradley, Scott Thayer, Anthony Stentz, and Peter Rander. Vegetation detection for mobile robot navigation. 2004.

[29] Tixiao Shan, Jinkun Wang, Brendan Englot, and Kevin Anthony James Doherty. Bayesian generalized kernel inference for terrain traversability mapping. In *Conference on Robot Learning*, 2018.

[30] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1 edition, 2000.

[31] Lupeng Zhou, Jikai Wang, Shiqi Lin, and Zonghai Chen. Terrain traversability mapping based on lidar and camera fusion. In *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*, pages 217–222, 2022.

[32] Anderson Souza and Luiz M. G. Gonçalves. Occupancy-elevation grid: an alternative approach for robotic mapping and navigation. *Robotica*, 34(11):2592–2609, 2016.

[33] Steven W. Chen, Guilherme V. Nardari, Elijah S. Lee, Chao Qu, Xu Liu, Roseli A. F. Romero, and Vijay Kumar. SLOAM: semantic lidar odometry and mapping for forest inventory. *CoRR*, abs/1912.12726, 2019.

[34] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. pages 4758–4765, 10 2018.

[35] Han Wang, Chen Wang, Chun-Lin Chen, and Lihua Xie. F-loam: Fast lidar odometry and mapping. 07 2021.

[36] Jiarong Lin and Fu Zhang. Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3126–3131, 2020.

[37] Boris Jutzi and H. Gross. Normalization of lidar intensity data based on range and surface incidence angle. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, 38, 01 2009.

[38] Jens Behley and Cyrill Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. 06 2018.

[39] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[40] Thomas Collins, J.J. Collins, and Donor Ryan. Occupancy grid mapping: An empirical evaluation. In *2007 Mediterranean Conference on Control Automation*, pages 1–6, 2007.

[41] Víctor Jiménez, Jorge Godoy, Antonio Artuñedo, and Jorge Villagra. Object-wise comparison of lidar occupancy grid scan rendering methods. *Robotics and Autonomous Systems*, 161:104363, 2023.

[42] Ralph Grewe, Matthias Komar, Andree Hohm, Stefan Lueke, and Hermann Winner. Evaluation method and results for the accuracy of an automotive occupancy grid. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, pages 19–24, 2012.

[43] R. Fernandes, Cristiano Premebida, Paulo Peixoto, Denis Wolf, and Urbano Nunes. Road detection using high resolution lidar. *2014 IEEE Vehicle Power and Propulsion Conference, VPPC 2014*, 01 2015.

[44] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3d lidar point clouds. In *2011 IEEE International Conference on Robotics and Automation*, pages 2798–2805, 2011.

[45] Yubo Cui, Jiayao Shan, Zuoxu Gu, Zhiheng Li, and Zheng Fang. Exploiting more information in sparse point cloud for 3d single object tracking. *IEEE Robotics and Automation Letters*, 7(4):11926–11933, 2022.

[46] Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[47] Suprava Chakraborty, Devaraj Elangovan, Padma Lakshmi Govindarajan, Mohamed F. ELnaggar, Mohammed M. Alrashed, and Salah Kamel. A comprehensive review of path planning for agricultural ground robots. *Sustainability*, 14(15), 2022.

[48] Fengqian Dou, Yanjun Huang, Li Liu, Hong Wang, Yu Meng, and Lianqiang Zhao. Path planning and tracking for autonomous mining articulated vehicles. *International Journal of Heavy Vehicle Systems*, 26:315, 01 2019.

[49] Daoliang Li, Peng Wang, and Ling Du. Path planning technologies for autonomous underwater vehicles-a review. *IEEE Access*, 7:9745–9768, 2019.

[50] Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1398–1404 vol.2, April 1991.

[51] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots in cluttered environments. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 572–577 vol.1, 1990.

[52] Nils J. Nilsson. A mobile automaton: An application of artificial intelligence techniques. In *International Joint Conference on Artificial Intelligence*, 1969.

[53] Stephen Cameron. Book reviews : Robot motion planning: by jean-claude latombe published by kluwer academic publishers. *The International Journal of Robotics Research*, 11(4):397–397, 1992.

[54] Steven M. LaValle. Rapidly-exploring random trees : a new tool for path planning. *The annual research report*, 1998.

[55] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[56] Xiangrui Meng, Zhiqiang Cao, Shuang Liang, Lei Pang, Shuo Wang, and Chao Zhou. A terrain description method for traversability analysis based on elevation grid map. *International Journal of Advanced Robotic Systems*, 15(1):1729881417751530, 2018.

[57] J.-S. Gutmann, M. Fukuchi, and M. Fujita. A floor and obstacle height map for 3d navigation of a humanoid robot. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1066–1071, 2005.

[58] Zhao Liu, Jinling Wang, and Daxue Liu. A new curb detection method for unmanned ground vehicles using 2d sequential laser data. *Sensors*, 13(1):1102–1120, 2013.

[59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[60] Mário Cristóvão. FRUC multiple sensor forest dataset including absolute, map-referenced localization. June 2023.