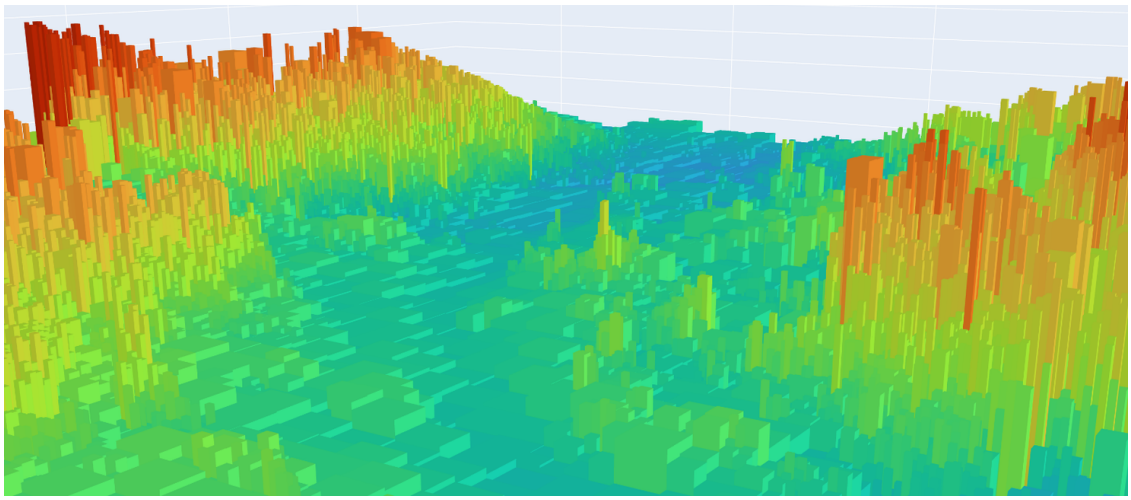


Environmental Perception for Autonomous Forestry Vehicles

Using Sensor Fusion of LiDAR and Stereo Camera for Enhanced Terrain Mapping

Master's thesis in MPSYS and MPCAS

VIKTOR OLSSON

HANNES SKOOG

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Environmental Perception for Autonomous Forestry Vehicles

Using Sensor Fusion of LiDAR and Stereo Camera for Enhanced Terrain Mapping

VIKTOR OLSSON
HANNES SKOOG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2025

Environmental Perception for Autonomous Forestry Vehicles
Using Sensor Fusion of LiDAR and Stereo Camera for Enhanced Terrain Mapping
VIKTOR OLSSON
HANNES SKOOG

Supervisor: Ebba Davidsson, BITADDICT AB
Examiner: Nikolce Murgovski, Electrical Engineering, Chalmers

Master's Thesis 2025
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Image showing a voxel representation of a forest environment, created using fused measurements from two perception sensors.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Environmental Perception for Autonomous Forestry Vehicles
Using Sensor Fusion of LiDAR and Stereo Camera for Enhanced Terrain Mapping
VIKTOR OLSSON
HANNES SKOOG
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Autonomous navigation in forestry environments presents significant challenges due to complex, unstructured terrain with varying visibility conditions. This thesis presents a novel sensor fusion approach integrating LiDAR and stereo camera data for enhanced terrain mapping in forestry applications. The project develops an uncertainty-aware fusion framework based on Kalman filtering that effectively combines the high accuracy of LiDAR with the dense coverage of stereo camera data, while properly accounting for each sensor's unique error characteristics and uncertainties. Additionally, a dynamic voxel-based representation is implemented that adapts map resolution to terrain complexity, optimizing memory usage while maintaining high fidelity in regions of interest. Experimental results demonstrate measurable improvements in various dimensions: the dynamic voxelization reduced memory usage by 31.65% and improved map update time by 44.27% compared to traditional fixed-size voxel grids, while maintaining mapping quality. Testing on real-world autonomous navigation routes showed that the proposed approach enables more complete trajectory following compared to the previous single-sensor approach, achieving path lengths significantly closer to the planned trajectory - for instance, 38.99 m compared to 18.99 m in one test. This work demonstrates that intelligent fusion of complementary sensors, combined with adaptive mapping techniques, can significantly improve terrain perception for autonomous vehicles operating in challenging off-road environments.

Keywords: sensor fusion, LiDAR, stereo camera, terrain mapping, forestry, voxel, Kalman filter, dynamic resolution, autonomous navigation, uncertainty.

Acknowledgements

We want to thank our thesis supervisor at Bit Addict, Ebba Davidsson, for her support, kind words and trust in us throughout the thesis project. All your work to help, give constructive feedback and drive us back and fourth to test sites helped to uplift the thesis to its final form.

We would also like to thank our examiner Nikolce Murgovski for the continued support and supportive feedback during this thesis. Your expertise and valuable insights have helped us enormously.

We would also like to thank the following people for helping us finalise the project: All employees of Bit Addict for creating a welcoming atmosphere and sharing your experience. All people in the BraSatt team for their insights and discussions. Lastly, we would also like to thank friends and family for their feedback and support throughout this journey.

Viktor Olsson, Gothenburg, June 2025
Hannes Skoog, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

k	Index for time step
j	Index for interpolated points

Parameters

h_l	LiDAR mounting height above ground
θ_l	LiDAR mounting angle
v	Voxel size parameter for downsampling and voxel grid
α_L	Base LiDAR point uncertainty (0.03)
β_L	Tunable parameter for longitudinal distance dependency (0.009)
ε_L	Tunable parameter for lateral tilt dependent uncertainty (0.1)
$\sigma^2_{L,xy}$	Positional uncertainty in x and y for LiDAR (0.02)
α_{SC}	Base uncertainty factor for stereo camera (0.3)
β_{SC}	Tunable parameter for stereo camera distance dependency (0.1)
$\sigma^2_{SC,xy}$	Positional uncertainty in x and y for stereo camera (0.02)
q_d	Process noise for deformable terrain (0.0001)
$d_{\max}$	Maximum threshold distance for interpolation (1 m)
M	Grid resolution for LiDAR interpolation (192)
r	Radius for radius outlier filter (0.3)
$N_{\min}$	Minimum neighbors for radius outlier filter (7)

Variables

$\hat{h}_{k\|k-1}$	Predicted height estimate at time step k given measurements up to $k-1$
$\hat{h}_{k\|k}$	Updated height estimate at time step k given measurements up to k
$P_{k\|k-1}$	Predicted state covariance at time step k
$P_{k\|k}$	Updated state covariance at time step k
K_k	Kalman gain at time step k
v_k	Innovation at time step k
S_k	Innovation covariance at time step k
z_k	Measurement vector at time step k
σ^2_{LiDAR}	LiDAR measurement variance based on distance and angle
σ^2_{SC}	Stereo camera measurement variance based on distance
$\sigma^2_{i,j}$	Variance for interpolated LiDAR points
d_1, d_2	Distances to nearest LiDAR points for interpolation
α_j	Normalized distance as spatial weighting factor for interpolation
$\mathbf{T}_{\text{sensor}}$	4×4 transformation matrix to global coordinates
$x_{\text{voxel}}, y_{\text{voxel}}$	Voxel indices in the global grid

Contents

List of Figures

List of Tables

1
Introduction

In 2020, Södra Skogsägarna started a project named BraSatt to develop a new way of scarification and planting saplings to ensure their survival. Currently, Södra's survival rate for planted saplings after 3 years is $70 - 75\%$ [1]. The BraSatt project proposes optimizing and effectivizing forest regeneration by the development of an autonomous machine that will calculate an accessible route through the forest, go along the route and select optimal planting points [2], scarify the soil and then plant the saplings.

The current prototype *BraSatt 01*, shown in Figure 1.1, will be operating in forest environments, creating unique challenges for autonomous navigation. Due to the complex and unstructured nature of these environments, accurate environmental perception is crucial for safe and efficient operation.

Figure 1.1: The BraSatt prototype *BraSatt 01* shown in a rendered picture. The vehicle is approximately 4 m long and 2.4 m high.

1.1 Background

Modern autonomous systems rely on various sensors to perceive surroundings and make real-time decisions [3]. Light Detection And Rangings (LiDARs) are widely used for measuring distances and creating 3D maps [4], while Stereo Cameras (SCs) excel at generating detailed 3D reconstructions based on vision [5]. However, each

sensor type has its limitations. LiDARs provides high-accuracy in-depth measurements but captures limited detail resolution, leaving holes in the data due to sparse scans [6]. In contrast, SCs deliver rich visual details but may struggle with depth estimation over long distances or in low-light conditions [7].

The BraSatt prototype utilizes a front-mounted LiDAR to perceive the terrain in front of the machine and to make decisions on optimal routes. The LiDAR can provide great detail in perceiving 3D environments and is often used in autonomous vehicles [8]. Many autonomous solutions using LiDAR for environment perception are made for road use, to be used in cities, urban areas and highways [9]. For BraSatt, the machine will be operating in forests with boulders, steep terrain, trees and other obstacles, creating new challenges compared to the typical road use [10]. Since the LiDAR use sparse scans (16 scan lines in this case) it is prone to introduce uncertainty in the environmental representation [11].

By introducing a SC, a camera with two sensors able to estimate depth data by triangulation, it can help fill in the gaps where the LiDAR is not able to scan to generate a more robust representation of reality. The richer and less uncertain representation made through fusing the data from both sensors can help the vehicle make better decisions in its path planning. Because of how the vehicle is constructed to handle different terrains, it has a high turn radius. Thus it is crucial to make optimal path decisions as early as possible to efficiently avoid obstacles and steep terrain. Due to the complex and noisy nature of the terrain environments, data from perception sensors may be noisy. Utilizing both sensors at once may therefore yield a more accurate and robust representation.

By fusing data from these two types of sensors, it is possible to represent environments with higher accuracy and robustness, complementing the strenghts of both sensors [12]. This kind of sensor fusion benefits applications such as autonomous driving and robotics by removing uncertainty and noise from sensor data [13].

1.2 Purpose and Aim

The purpose of this thesis is to explore and improve environmental perception for autonomous forestry vehicles operating in unstructured off-road environments. In such settings, traditional single-sensor approaches often struggle with data sparsity, occlusions, and variable terrain.

The aim is to develop a sensor fusion method that combines data from a LiDAR and a SC to produce a robust, high-fidelity terrain representation. The approach should effectively capture ground elevation, account for sensor uncertainties, and introduce an adaptive level of detail based on terrain complexity. A key goal is to ensure that the solution remains computationally efficient and suitable for real-time implementation in embedded systems.

1.3 Scope

This thesis focuses on designing and implementing a sensor fusion framework based on a Kalman filter and dynamic voxelization for terrain mapping in forestry environments. The fusion system integrates data from a Velodyne VLP-16 LiDAR and a ZED 2i stereo camera, mounted on the BraSatt 01 prototype.

The work includes:

- Designing a world representation which captures terrain complexity and is computationally efficient.
- Modeling measurement uncertainty for both LiDAR and stereo camera inputs.
- Implementing a Kalman filter-based height estimation per voxel.
- Evaluating the system using pre-recorded sensor data in ROS2.
- Comparing results against single-sensor and fixed-grid baselines.

1.4 Limitations

The limitations of the project are defined as follows:

- This project will not investigate different methods of SC triangulation. The stereo camera used, ZED2i, already has onboard software for computing a point cloud from two images.
- It will be assumed that the vehicle moves in a static world. I.e. moving agents in the scene will not be explicitly detected, tracked, or identified. This assumption is made since moving objects will rarely be present in the environment in question and is not the main issue at hand.
- The fusion is limited to estimating the height within a 2.5D voxel-based map. It does not model overhanging objects, vegetation classification, or semantic understanding of the terrain.
- The final system will not be tested in a real-time environment, only in a simulated environment.
- The results and methods are tailored to the specific sensor configuration of the BraSatt vehicle, which consists of a Velodyne VLP-16 LiDAR and a ZED2i stereo camera. The methods may not be directly applicable to other sensor configurations without further adjustments.

2
Theoretical Background

This chapter presents a theoretical background on the work presented in this thesis. The Prototype BraSatt 01 is introduced, including its sensors and vehicle dynamics. The world representation for autonomous vehicles is discussed, focusing on the challenges of off-road navigation in forest environments. The chapter also covers the fusion of LiDAR and stereo camera data, highlighting the use of Kalman filtering for height estimation and terrain modeling. Finally, the chapter concludes with a discussion on LiDAR data interpolation techniques.

2.1 Prerequisites on BraSatt 01 Prototype

The *BraSatt 01 prototype* is an articulated terrain vehicle whose purpose is to autonomously follow a predefined route in a forest environment and plant saplings at appropriate planting spots [2]. The vehicle is developed and built to handle forest terrain and can drive over smaller rocks, stumps and other vegetation. It is made up of a front and rear part connected through a coupling which can be controlled by hydraulics to enable steering. This setup enables the vehicle to handle rougher terrain but also yields a slow steering rate [14]. The vehicle has front-mounted LiDAR and SC which will be used in this project.

2.1.1 LiDAR

LiDAR is a sensing technology that uses laser pulses to measure the Time-of-Flight (ToF) of the reflected signal, determining distances to objects and creating precise 3D representations of environments [4]. The accuracy of LiDAR measurements depends on factors such as laser wavelength, pulse repetition rate and environmental conditions.

There are two main types of LiDAR: mechanical and solid-state. Mechanical LiDARs utilizes either a single emitter-receiver module or an array of such modules to achieve a full 360° Field Of View (FOV) around the sensor. In contrast, solid-state LiDARs do not use moving parts, instead measuring in a fixed direction. While solid-state LiDARs offer increased durability and lower failure rates, they typically have a more limited FOV and range compared to mechanical LiDARs.

The *BraSatt 01 prototype* is equipped with the mechanical LiDAR sensor Velodyne VLP-16, which features 16 laser channels and operates at a wavelength of 903 nm

[11]. The VLP-16 can capture up to $\sim 600,000$ points per second when running in Dual Return Mode, distributed across its 16 beams. It provides a vertical FOV of 30° (±15° from the horizontal plane) and a full 360° horizontal FOV. Though due to the mounting position on the vehicle, the effective horizontal FOV is reduced to approximately 175°. The sensor's maximum range is specified as 100 meters under optimal conditions, with an accuracy of ±3 cm.

While LiDARs provides accurate distance measurements, they also have limitations. The sensors are sensitive to adverse weather conditions, such as heavy rain or fog, which can scatter or absorb laser pulses, reducing measurement accuracy [15]. Additionally, near-infrared lasers (e.g., 905 nm) can be affected by certain materials, such as dark or highly reflective surfaces, which may absorb or misdirect the laser pulses.

The sparsity of the VLP-16 with its 16 laser scan lines, visualized in Figure 2.1, introduces additional challenges in capturing detailed environmental features. In complex terrains, such as dense forests, the limited number of scan lines can result in significant gaps in the point cloud data, especially when objects are small or located between the beams. This sparsity makes it difficult to detect the true height of objects and can lead to incomplete representations of the environment. Furthermore, the fixed vertical resolution of the scan lines may cause critical features, such as steep slopes or narrow gaps, to be underrepresented, impacting the accuracy of terrain modeling and obstacle detection [16]. These issues could be mitigated by using a LiDAR with more scan lines, but at the expense of increased cost.

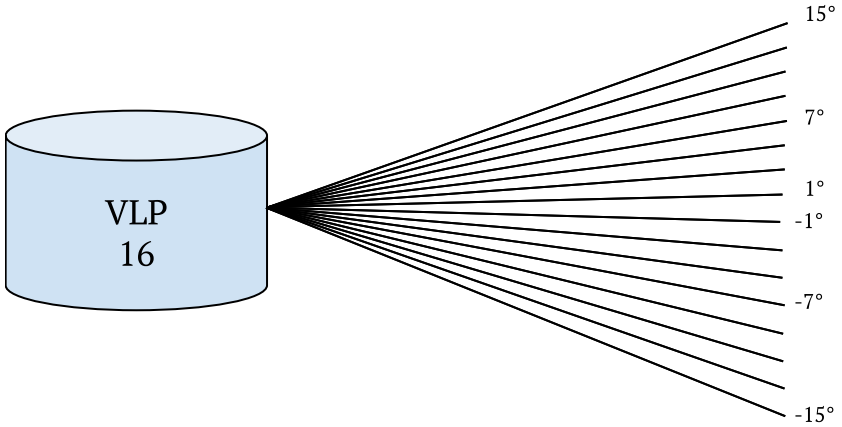


Figure 2.1: Illustration of the VLP-16 LiDAR scan lines. The sensor emits laser pulses in a rotating pattern, capturing distance measurements from the surrounding environment. The resulting point cloud data is used to create a 3D representation of the terrain.

2.1.2 Stereo Camera

A SC is a type of camera with two image sensors slightly separated horizontally, enabling depth estimation through triangulation. By comparing the displacement of

pixels between two images taken at the same instance, it is possible to estimate the depth of a point in 3D space, creating a 3D point cloud of a scene. This capability is valuable for autonomous applications that require environmental perception for navigation.

As shown in Figure 2.2, the SC captures two images from slightly different perspectives. The disparity between corresponding points in the two image planes is used to calculate the distance to the object, creating a 3D point cloud representation of the scene. The triangulation process relies on the known baseline distance d between the two cameras and intrinsic camera parameters such as the focal length of the lenses to compute depth information.

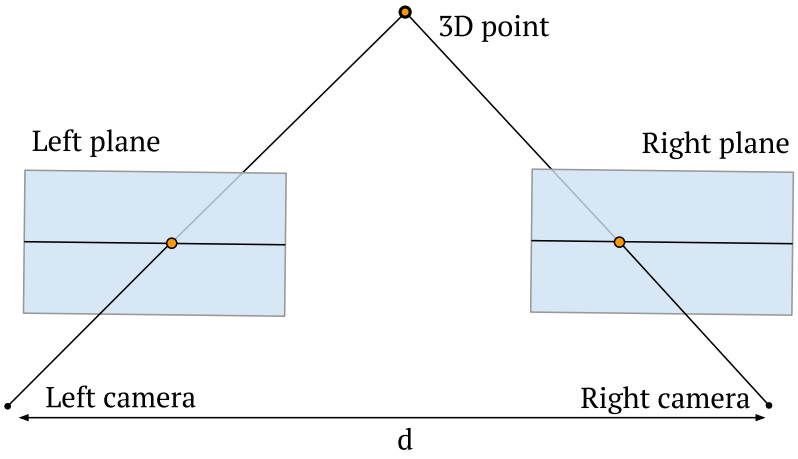


Figure 2.2: Illustration of the triangulation process. The stereo camera captures two images from slightly different perspectives, allowing depth estimation through triangulation. The disparity between corresponding points in the two image planes is used to calculate the distance to the object, creating a 3D point cloud representation of the scene.

One of the key advantages of stereo vision is its ability to estimate depth passively, without the need for active illumination, as required by LiDAR or structured light systems [5]. This makes SC particularly useful in outdoor environments, such as forests, where varying lighting conditions and dense foliage can impact sensor performance. Unlike LiDAR, which may struggle in foggy conditions due to laser scattering, SC rely purely on ambient light and can continue functioning as long as sufficient texture is present in the scene [5]. However, stereo vision has its limitations, such as difficulty in depth estimation for textureless surfaces (e.g., uniform tree trunks) and increased computational demands for disparity calculation.

For the *BraSatt 01 prototype* a ZED 2i stereo camera is used. The ZED 2i has its own Software Development Kit (SDK) which features prebuilt triangulation functionality, allowing real-time depth estimation without manual calibration or external computation [17]. This capability simplifies integration with autonomous systems and provides robust depth perception in complex environments.

Compared to LiDARs, SCs provide dense depth information across the entire image, rather than discrete point measurements [5]. However, stereo depth estimation is less accurate at long distances, where disparity differences become minimal. Each sensor has their own pros and cons, which can be seen in Table 2.1. By combining stereo vision with LiDAR data, it is possible to improve overall robustness, leveraging the strengths of both sensing modalities for enhanced perception in forest environments.

Feature	**LiDAR**	**Stereo Camera**
Accuracy	High for distance measurements	Moderate, depends on texture
Range	Long (up to 100m)	Limited, lower accuracy at distances
FOV	360°(H) × 30°(V)	110°(H) × 70°(V)
Weather	Affected by rain, fog	Less affected, works in ambient light
Resolution	Sparse point clouds	Dense point clouds
Computation	Low for processing point clouds	High for disparity calculation

Table 2.1: Comparison of LiDAR and Stereo Camera: Pros and Cons

2.1.2.1 Filtering of Stereo Camera Data

Due to the nature of triangulated point clouds from a SC, the data can be noisy [7]. Thus, it is often important to filter the data before using it for further processing. One such approach is to use Hidden Point Removal (HPR), described by [18]. HPR determines visible points in a point cloud as viewed from a given viewpoint without reconstructing a surface or estimating normals. The algorithm transforms the point cloud through spherical flipping inversion, then computes the convex hull of the set containing the viewpoint and the transformed points. Points on this convex hull are considered visible. This approach is particularly valuable for filtering occluded points in dense environments like forests, where vegetation can create numerous false readings.

Another approach is also the Radius Outlier Filtering (ROF) which filters outliers based on the number of neighbors within a specified radius [19]. For each point, if the count of neighbouring points within a defined radius falls below a threshold, the point is considered an outlier and is removed from the point cloud. This simple yet effective method helps remove isolated points that often represent noise or measurement errors.

2.1.3 Vehicle Dynamics

While this thesis does not investigate the vehicle dynamics in detail, it is important to understand as a background to why a robust terrain estimation, especially at a distance, is crucial for safe navigation in rough terrain.

The BraSatt 01 prototype is an articulated vehicle, meaning it consists of two segments connected by a coupling joint which steers the vehicle, as shown in Figure 2.3. This design allows for greater maneuverability and flexibility in navigating rough

terrain [14]. The vehicle's steering system is controlled by hydraulics, enabling precise control over the angle of the front and rear segments. The steering angle can be adjusted to optimize the vehicle's trajectory and stability while traversing uneven surfaces.

While the articulated design provides advantages in terms of maneuverability in terrain, it also introduces challenges in vehicle dynamics. The steering system creates a delay in the vehicle's response to steering commands, which can affect its ability to navigate tight turns or obstacles. Additionally, the coupling joint between the two segments creates a large turn radius, making it difficult to navigate tight spaces. This creates a greater demand for a well planned path and an accurate world representation to ensure the vehicle can navigate safely.

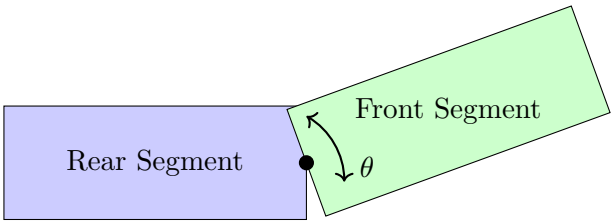


Figure 2.3: Articulated vehicle model showing the coupling joint and steering angle θ between front and rear segments from a top-down view.

2.1.4 Path Planning Algorithm

The path planning algorithm is a crucial component of the autonomous vehicle system, enabling it to navigate through complex environments while avoiding obstacles and adhering to safety constraints. The algorithm is designed to generate a smooth and efficient trajectory for the vehicle based on a predefined route. To make correct choices and avoid obstacles, the algorithm relies on a world representation that accurately captures the terrain and obstacles in the environment.

In each timestep, the path planning algorithm generates a multitude of possible path segment from its current position and evaluates the drivability of each path segment based on the current world representation in form of an elevation map. It then evaluates which path is the best pick by a cost function. The cost function takes into account various factors, including the distance to the planned route, the tilt of the vehicle, the terrain slope and obstables. The path segment with the lowest cost is selected and used to update the vehicle's trajectory.

An efficient and precise world representation is crucial to correctly evaluate the drivability of a path. The algorithm relies on accurate terrain information to make informed decisions about where it is possible to drive and which path is the best choice.

2.2 World Representation for Autonomous Vehicles

Once data is collected from perception sensors, it must be structured into a model that represents the environment in a way that supports autonomous driving decision-making. In road-based autonomous driving, world representation models are often highly detailed and structured, incorporating lane markings, traffic signs, dynamic agents and other road-specific features [20]. However, in off-road and forest environments, the challenges differ significantly. Instead of structured lanes and traffic elements, key factors such as ground elevation, vegetation, rocks and slopes become critical for ensuring safe and efficient navigation.

In such environments, terrain modeling plays a crucial role in path planning and evaluating drivable routes. Various representations exist, ranging from raw point clouds, where the terrain is represented by points in 3D space, or voxel grids where the terrain is represented by discrete cubic units, to more compact and structured models such as elevation maps, which provide a dense and efficient encoding of terrain height variations [9]. These representations are essential for enabling autonomous vehicles to navigate complex environments, identify obstacles and slopes, and make informed decisions about traversable paths.

2.2.1 Elevation Map

Elevation maps are particularly useful for rough terrain navigation, as they enable the identification of traversable and non-traversable regions while maintaining computational efficiency [9]. These maps represent the terrain's height variations in a grid format, where each cell contains the elevation value of the corresponding area. This structured representation allows for efficient path planning and obstacle avoidance algorithms to be applied.

In the context of autonomous vehicles, elevation maps can be generated using data from various sensors, such as LiDAR and stereo cameras [6]. The raw point cloud data from these sensors is processed to create a dense elevation map, which can then be used to identify obstacles, slopes and other terrain features that may impact navigation.

This representation is great for terrain which can be highly variable and cluttered with vegetation. By providing a detailed representation of the ground surface, elevation maps enable autonomous vehicles to navigate complex environments more effectively, ensuring safe and efficient operation [10].

Most elevation map implementations employ uniform grid resolutions across the entire map. Sten et al. [21] demonstrated this approach by constructing a grid-based elevation map that fused interpolated VLP-16 LiDAR data with ZED 2i SC measurements. Their system utilized Kalman filtering to combine sparse LiDAR points (interpolated in one dimension to form a reference ground plane) with denser

SC data, producing a comprehensive terrain model for static environments.

However, uniform grid resolutions present inherent limitations in heterogeneous terrain. A fixed-resolution approach must compromise between computational efficiency and representational accuracy: high-resolution grids capture terrain details accurately but consume excessive computational resources and memory, while low-resolution grids are efficient but may miss critical terrain features. Additionally, uniform resolutions allocate the same computational resources to flat, homogeneous regions as they do to complex, variable terrain — an inefficient use of system resources. To overcome these limitations, it is possible to use more dynamic approaches to terrain representation, such as dynamic voxel-based mapping, which can adapt the resolution of the map based on local terrain characteristics. This allows for a more efficient representation of the environment while maintaining the ability to capture important features and details.

2.2.2 Voxel-based Terrain Representation

Voxel-based mapping is a technique used for representing 3D environments, i.a. in autonomous navigation [22]. By discretizing 3D space into cubic or grid-based units, voxels provide an efficient way to store and process spatial data. Unlike raw point cloud data which is often unordered and computationally expensive to process in real-time, voxel-based representations enable structured storage and efficient querying of height information. While similar to a grid-based elevation map, the dynamic voxel representation employs content-aware grid resolution to be able to capture detailed data while avoiding unnecessary computations and data storing.

2.2.2.1 Dynamic Voxelization

To efficiently model the terrain for autonomous navigation in forests, this thesis presents a novel method to dynamically update the size of the voxels which will adjust the resolution of the map based on local terrain characteristics. This approach is inspired by [23] but differs in implementation and usage. The key aspects of this method are as follows:

- **Adaptive Resolution**: The terrain is represented using voxels whose resolution adapts based on local terrain variability. Areas with low height variance are modeled with larger voxels, while regions with high complexity are refined into smaller voxels to capture more details.
- **Efficient Data Representation**: Voxels are only created where sensor measurements exist, resulting in a memory-efficient and scalable map that reflects the observed environment.
- **Hierarchical Structure**: The voxel grid supports recursive subdivision, enabling selective refinement of complex terrain features without increasing computational load in homogeneous areas.
- **Support for Sensor Fusion**: The approach accommodates integration of heterogeneous sensor data, such as LiDAR and stereo camera measurements, to improve terrain estimation robustness.

2.3 LiDAR and Stereo Camera Fusion

In most autonomous vehicle applications, fusing the data from multiple perception sensors is a common way to create a richer and more robust perception, allowing the system to overcome independent limitations for each sensor [3]. As previously mentioned, LiDARs capture sparse but precise data. By combining data from a LiDAR with data from a SC, which captures depth data across the entire image, the advantages of both sensors are used to create a richer and less uncertain measurement of an environment.

2.3.1 Neural Networks

Neural networks, particularly deep learning models, have become increasingly popular for sensor fusion in autonomous systems due to their ability to learn complex, non-linear relationships from data. Unlike traditional fusion methods such as Kalman filters, which require explicit models of sensor noise and dynamics, neural networks can implicitly learn how to combine inputs from different sensors through supervised or self-supervised training.

In the context of fusing LiDAR and SC data, neural networks can be used to enhance spatial reasoning and semantic understanding. For example, convolutional neural networks (CNNs) can process image-like data such as stereo disparity maps or LiDAR depth images, while more advanced architectures such as PointNet or voxel-based 3D CNNs can directly operate on raw point cloud data [24]. These models can be designed to estimate depth, classify terrain types, or even infer full scene reconstructions by leveraging the complementary strengths of each sensor modality.

Some approaches use early fusion, where raw sensor inputs are combined and fed into a single network, while others adopt late fusion, where features are extracted independently from each sensor before being merged at a higher level [13]. Additionally, attention mechanisms and transformer-based models have recently been applied to dynamically weigh the importance of each sensor input depending on environmental context [25].

While neural network-based fusion can yield high performance and generalization, it often requires large amounts of annotated training data, something we haven't found for this project, and careful validation to avoid overfitting or brittleness in unseen conditions. These methods also tend to be more computationally intensive than traditional filters, which may limit their deployment on resource-constrained platforms. Creating an annotated dataset for the purpose of this thesis is unfeasible in the given time frame. Thus, neural network approaches will not be investigated further.

2.3.2 Kalman Filter

The Kalman Filter (KF) is a mathematical approach for estimating the state of a system by filtering noisy sensor measurements. Developed by Kalman and Bucy in 1960 [26, 27], it is widely used in real-time applications such as aerospace navigation and robotics. The filter operates by recursively updating state estimates based on a predictive model and incoming measurements, minimizing the impact of noise through a least-mean-square optimization approach. It is used for linear systems where sensor inputs can be mapped to internal states. A strength of KF is that it can combine inputs from multiple sensors into a vector of internal states representing the parameters of interest, provided the relationship between inputs and system states remains linear [12].

The KF is built on the premise that we can model a system described by two linear equations. The first is a state-space prediction equation:

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \quad \mathbf{w}_{k-1} \sim \mathcal{N}(0, \mathbf{Q}_{k-1}), \tag{2.1a}$$

where $\mathbf{x}$ is the system state, $\mathbf{A}$ is the process model and $\mathbf{w}$ is process noise, Gaussian distributed with zero-mean and covariance $\mathbf{Q}$. The second equation is the measurement equation:

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k, \quad \mathbf{r}_k \sim \mathcal{N}(0, \mathbf{R}_k), \tag{2.1b}$$

where $\mathbf{z}$ is the measurement from a sensor, $\mathbf{x}$ is the true state, $\mathbf{H}$ is the measurement model matrix and $\mathbf{r}$ is the sensor measurement noise, Gaussian distributed with zero-mean and covariance $\mathbf{R}$.

The KF operates recursively by computing predictions and updates based on the system equations. These steps allow it to estimate the system state while minimizing the effect of noise. The prediction step estimates the system's next state based on the previous state and a mathematical model of the system dynamics. The prediction step

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1}, \tag{2.2a}$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}, \tag{2.2b}$$

computes the state prediction $\hat{\mathbf{x}}_{k|k-1}$ and the uncertainty, described by the covariance $\mathbf{P}_{k|k-1}$, in the state prediction. The update step corrects the predicted state by using the new measurements $\mathbf{z}_k$. The state estimate $\hat{\mathbf{x}}_{k|k}$ and covariance matrix $\mathbf{P}_{k|k}$ is computed as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{v}_k, \tag{2.3a}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \tag{2.3b}$$

where the Kalman gain $\mathbf{K}_k$, Innovation $\mathbf{v}_k$ and Innovation covariance $\mathbf{S}_k$ are defined as

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}, \tag{2.3ca}$$

$$\mathbf{v}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}, \tag{2.3cb}$$

$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k. \tag{2.3cc}$$

The variable $\mathbf{K}_k$ determines how much the new measurement should influence the state update and is used in Equation (2.3a) as a correction term based on the difference (innovation $\mathbf{v}_k$) between the actual measurement and the predicted state. The covariance matrix in Equation (2.3b) is updated to reflect the new uncertainty after incorporating the measurement.

Perception sensors such as LiDARs and SCs are subject to noise from vegetation, sensor limitations and dynamic environmental factors. Individual measurements may not always accurately reflect the true ground height due to outliers and uncertainties. To address these challenges and improve the reliability of height estimation, statistical filtering methods such as the KF are used to estimate the heights within each voxel of the world representation. The KF framework enables the handling of noisy measurements, the fusion of multiple sensor inputs over time and the quantification of uncertainty in the estimated heights.

2.3.3 LiDAR Data Interpolation

Because LiDARs acquire data through discrete scan lines, the resulting point clouds are often sparse. This sparsity can limit the accuracy of downstream processing and environmental interpretation. To address this issue and produce a more continuous representation of the environment, various data interpolation techniques can be applied.

A straightforward approach is linear interpolation, where values such as distance or elevation are estimated between two neighboring points, either along the same scan line or between adjacent lines. While this method is computationally efficient and simple to implement, it performs best in relatively smooth environments where changes are gradual. More advanced techniques include bilinear and bicubic interpolation, which use multiple surrounding points in two dimensions to estimate new values. These methods yield smoother results but require additional computational resources.

When working with irregularly distributions point clouds, interpolation using Triangulated Irregular Networkss (TINs) is a common solution. TIN-based methods rely on Delaunay triangulation [28] to construct triangles from nearby points, enabling interpolation within those triangles. They are widely used in geospatial applications and digital terrain modeling [29]. Deluanay triangulation is used to divide the convex hull of the LiDAR points into triangles such that the circumcircle of each triangle does not contain any of the other points. For each triangle formed by the triangulation, linear interpolation using barycentric coordinates is applied to estimate the height of a predefined grid of interpolated points within the triangle. This process is repeated for all triangles and is illustrated in Figure. 2.4.

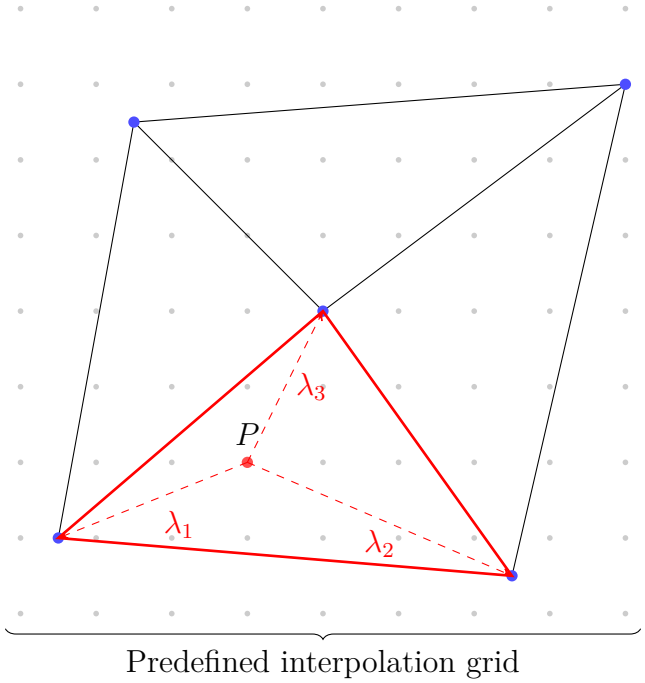


Figure 2.4: Illustration of Delaunay triangulation applied to a set of LiDAR points (blue). A predefined interpolation grid (gray) overlays the point cloud. One triangle is highlighted and a sample interpolated point P (red) within the triangle is shown, along with its barycentric coordinates λ_1, λ_2 and λ_3 used for linear interpolation.

In summary, this chapter presents the theoretical background used to build this thesis. It covers the perception sensors used in the *BraSatt 01 prototype*, including LiDARs and stereo cameras, and their respective advantages and limitations. The chapter also discusses the vehicle dynamics of the BraSatt 01 prototype, highlighting the challenges posed by its articulated design. The challenges of representing the world for autonomous navigation in rough terrain are addressed, with a focus on elevation maps and voxel-based terrain representation. Finally, the chapter explores the fusion of sensor data using Kalman filtering techniques to create a robust world representation for autonomous vehicles.

3
Methodology

This chapter describes the methods used to create the terrain mapping system. The system is designed to efficiently process and fuse data from multiple sensors, including a LiDAR and a SC, to create an accurate elevation map of the terrain. The methods are divided into several sections, each focusing on a specific aspect of the system.

3.1 System Overview

This thesis presents a terrain mapping system that fuses data from multiple sensors to create an accurate elevation map for off-road navigation. The system architecture consists of three main components: data acquisition, data preprocessing and terrain estimation (see Figure 3.1).

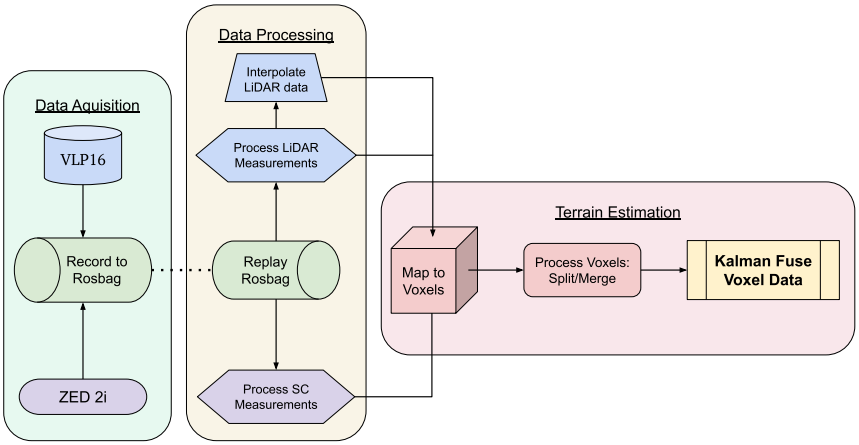


Figure 3.1: System architecture overview. Raw sensor data is preprocessed, transformed to global coordinates and fused using a voxel-based Kalman filter to create a terrain elevation map.

The data acquisition component collects measurements from two primary sensors:

- A LiDAR sensor providing sparse but accurate 3D point clouds
- A SC generating dense 3D point clouds by triangulating stereo images

In the data preprocessing stage, the raw sensor data undergoes several transformations:

- Coordinate transformation to align sensor data in a global reference frame
- Outlier removal and noise filtering
- LiDAR data interpolation to support SC data
- Variance calculation for each measurement and interpolation

The terrain estimation component uses a dynamic hierarchical voxel-based structure where each voxel maintains a Kalman filter to estimate the height of the voxel. The system:

- Maps new measurements to voxels
- Dynamically subdivides the voxels based on measured terrain complexity
- Fuses all new measurements in voxels with Kalman to get height estimate
- Maintains uncertainty estimates for each height measurement

The final output is a probabilistic elevation map that represents the terrain height and associated uncertainty, suitable for autonomous navigation planning.

3.2 Sensor Setup and Data Collection

The sensor setup has both the LiDAR and the SC mounted on the front of the vehicle platform. There are two different sensor mounting setups used in the project. In sensor setup 1 the LiDAR and the SC are tilted down 20° on the *BraSatt 01*. This is done to ensure the LiDAR is used to better capture the terrain closest to the vehicle. The LiDAR is mounted at a height of approximately $h_l = 1.28\,\text{m}$ above ground and the SC 5 cm above the LiDAR. Since the LiDARs vertical field of view is limited to $\pm 15°$, the furthest the LiDAR can measure in the viewing direction is calculated by the equation

$$z(y) = h_l - \tan(\theta_l - 15)y \tag{3.1}$$

where θ_l is the mounting angle, z is the height of the top LiDAR scan line dependent on the distance to the LiDAR y. Setting $z = 0$ and finding y gives $y = 14.63\,\text{m}$. This tilt makes the LiDAR capture the terrain in front of the vehicle well, but also means that effective range is limited greatly, shown in Figure 3.2.

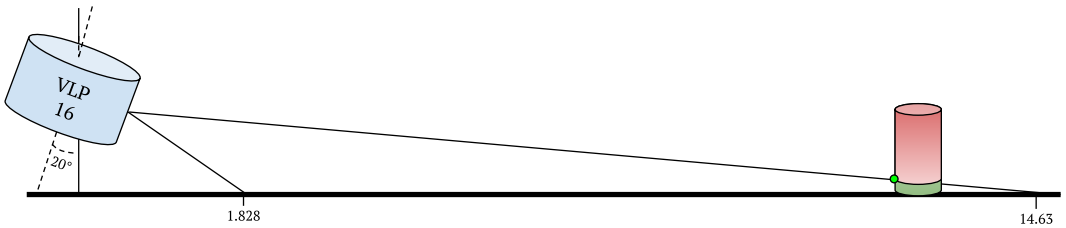


Figure 3.2: Sensor setup 1: The downward tilted angle of the LiDAR makes the maximum visible distance in the viewing direction 14.63 m, with worsening height capturing up to that distance. The cylindrical object shows how the tilt of the LiDAR makes it only capture the height of the green part, while the remaining height of the red part is not captured.

Worth noting is that as an effect of the LiDAR's circular scanning pattern, measurements not directly in the viewing direction will be better captured. As the scan continues in the circular pattern, the tilt will enable it able to measure higher and higher. Thus, the measurements closer to the viewing direction will be very affected by this tilt while measurements further to the sides will accurately capture height. To mitigate these incomplete captures, the behavior is modeled as an uncertainty of the LiDAR (see Section 3.5.1), which can be utilized when fusing the measurements.

The SC is unaffected by this behavior, as the stereo triangulation is not limited by the same angle. The SC can thus be used to capture the terrain further away from the vehicle, but with a lower sensor accuracy than the LiDAR.

In the last stages of the project, in an attempt to utilize the LiDAR better, sensor setup 2 was implemented. The LiDAR and SC were tilted up to a total downward tilt of 15°. This would effectively give the LiDAR unlimited range in the forward direction, with only a small loss of information closest to the vehicle, shown in Figure 3.3. This trade-off is deemed okay, since the terrain directly in front of the vehicle is already well measured and estimated at that point. This change enables us to be more certain of height measurements at a distance with the LiDAR.

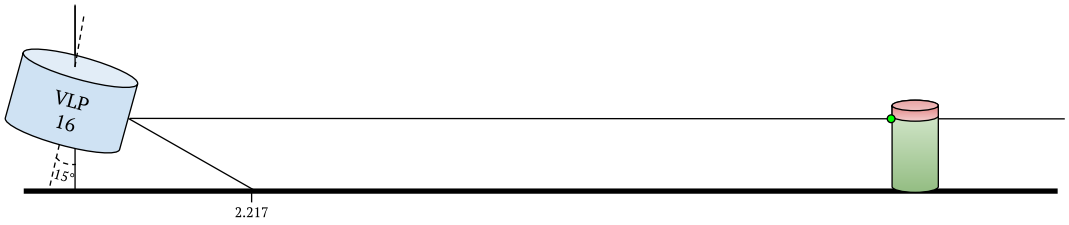


Figure 3.3: Sensor setup 2: The new downward tilted angle of the LiDAR makes the maximum visible distance in the viewing direction theoretically infinite. The cylindrical object shows how the LiDAR with sensor setup 2 is able to capture heights up to its own mounting height, making the red non-captured height much smaller compared to with sensor setup 1.

3.2.1 Data Acquisition

The system operates within the Robot Operation System 2 (ROS2) framework for data collection and processing [30]. Using ROS2, data from the LiDAR and SC are recorded as point clouds to a ROSbag as well as pose data from the vehicle. This ROSbag can later be replayed to simulate the vehicle's operation from the time recorded.

Proper alignment of sensor data is achieved through a combination of hardware and software synchronization mechanisms. Hardware-triggered synchronization ensures precise timing between sensor captures, while ROS2 message timestamps enable temporal alignment of data streams in software. The Real Time Kinematic (RTK)-GPS integration provides consistent alignment with the global reference frame.

3.3 Data Preprocessing

For both the LiDAR and SC, raw point cloud data is published containing local 3D point coordinates. These points are transformed into a global reference frame using pose data recorded as a transformation matrix using Equation (3.2). After the points are transformed, they are appended to separate lists which later is accessed when running the fusion algorithm and reset after the fusion is done.

$$\begin{bmatrix} x_{\text{global}} \\ y_{\text{global}} \\ z_{\text{global}} \\ 1 \end{bmatrix} = \mathbf{T}_{\text{sensor}} \begin{bmatrix} x_{\text{local}} \\ y_{\text{local}} \\ z_{\text{local}} \\ 1 \end{bmatrix} \tag{3.2}$$

where $\mathbf{T}_{\text{sensor}}$ is the 4×4 transformation matrix to global coordinates and rotation created by a pre-built state estimator utilizing fusion of two RTK-GPS devices mounted on the vehicle and the onboard Inertial Measurement Unit (IMU), providing accurate positioning and orientation information.

3.3.1 Stereo Camera Data

The SC system publishes a dense point cloud, P, and RGB images. Since the point cloud contains a great amount of points, it is downsampled by voxel-downsampling. The goal with the downsampling is to preserve the overall structure of the point cloud while reducing its density by aggregating nearby points.

The method partitions the 3D space into a regular grid of voxels of size v, where v is a parameter that controls the resolution of the output. This parameter, $v = 0.2\,\text{m}$, is empirically chosen so that the number of points from the SC are at a manageable size for computation. Each point $\mathbf{p_i} \in P$ is assigned to a voxel based on its spatial coordinates and for each voxel, the center of all contained points is computed. Only this center is retained in the downsampled output.

Formally, the steps are as follows:

1. Each point $\mathbf{p}_i$ is assigned to a voxel with index $\mathbf{v}_i = \left\lfloor \frac{\mathbf{p}_i}{v} \right\rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor operation applied element-wise.
2. All points sharing the same voxel index are grouped together.
3. For each group, the arithmetic mean (centroid) is computed and used as the representative point for that voxel.

This is also visualized in Figure 3.4. This approach preserves geometric features while drastically reducing the number of points, especially in regions with high point density.

To ensure the quality of the SC point cloud data, a noise filtering process is applied to remove outliers and unwanted noise. The filtering algorithm uses the HPR algorithm described by [18]. The algorithm takes as input the current camera center C and the radius parameter R which is defined as

$$R = \|b_{\max} - b_{\min}\| \cdot \alpha, \tag{3.3}$$

where $b_{\max}$ and $b_{\min}$ are the maximum and minimum bounds of the point cloud P, respectively, $\|\cdot\|$ denotes the euclidean norm and $\alpha = 150$ is a scalar scaling factor empirically chosen to ensure sufficient radius for point visibility estimation.

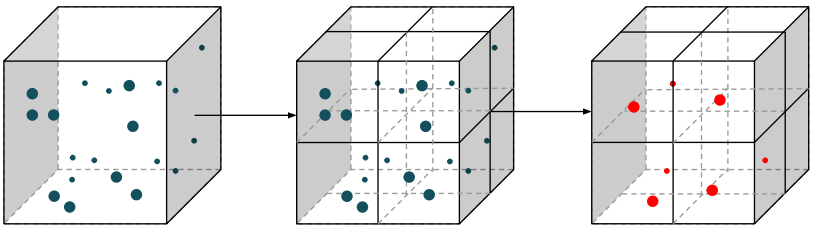
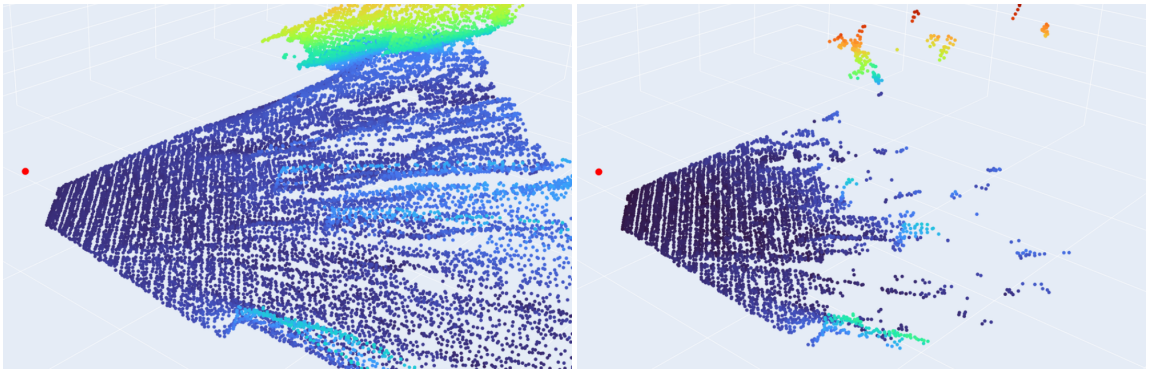


Figure 3.4: Voxel-downsampling of a point cloud. The original point cloud is represented by the blue points and the downsampled points are shown in red.

Following the visibility filtering, a ROF step is applied. Each point retained by the HPR filtering is evaluated based on its local neighborhood: a k-d tree is constructed from the filtered point cloud and for each point, the number of neighboring points within a radius $r = 0.3$ is counted. Points with fewer than $N_{\min} = 7$ neighbors within this radius are considered noise and are removed. Both r and $N_{\min}$ are empirically chosen.
The complete filtering pipeline can thus be summarized in two stages:

1. **Visibility Filtering:** Using the HPR algorithm with camera center C and radius R to remove occluded or non-visible points.
2. **ROF** Removing remaining outliers by retaining only points with at least $N_{\min} = 7$ neighbors within a local radius $r = 0.3$.

After these processes, the downsampled and filtered SC point cloud is transformed with the transformation matrix mapping all points to the global frame. The effect of processing on the SC data is shown in Figure 3.5.



(a) Downsampled point cloud before filtering.

(b) Downsampled point cloud after filtering.

Figure 3.5: Side-view showing the effect of filtering the SC data. Figure 3.5a shows the point cloud before filtering, while Figure 3.5b shows the filtered point cloud. The red dot represents the vehicle position.

3.3.2 Interpolation of Lidar data

The interpolation, shown in Figure 3.6, is created by linearly interpolating between LiDAR scans using a TIN. It is used to support the data collected from the SC, which is inherently more uncertain compared to the LiDAR. The process involves projecting the sparse LiDAR points onto a regular 2D grid to create a continuous height surface that can be sampled at arbitrary locations.

Let the LiDAR point cloud be denoted as a set of N points:

$$\mathbf{P} = \{(x_i, y_i, z_i)\}_{i=0}^{N} \in \mathbb{R}^{N\times 3}. \tag{3.4}$$

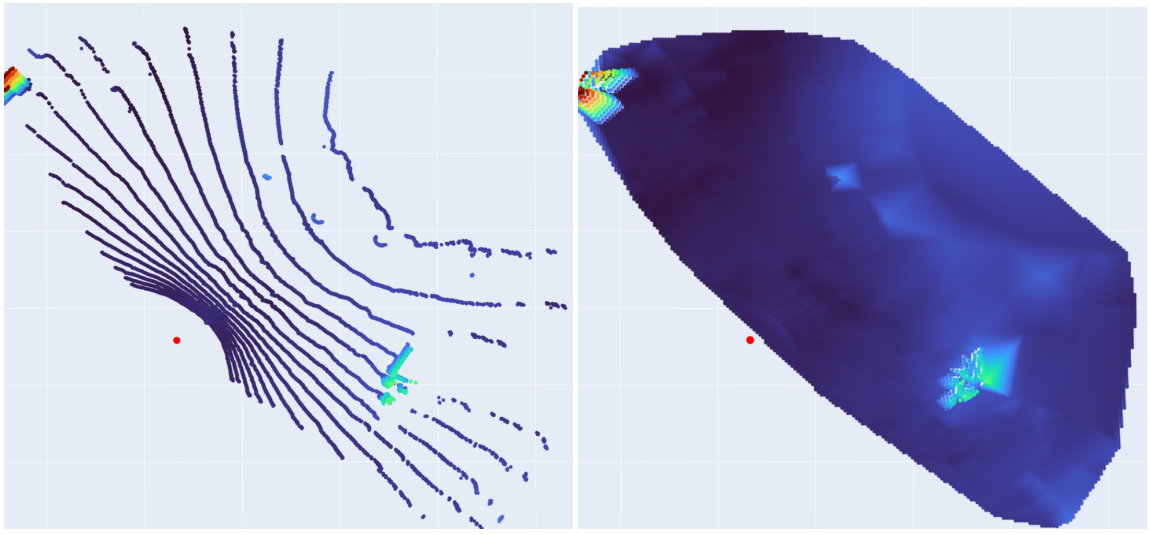
The 2D grid is constructed over the spatial extent of the LiDAR data, with boundaries defined as:

$$x \in [x_{\min}, x_{\max}], \quad y \in [y_{\min}, y_{\max}]. \tag{3.5}$$

The grid is then expressed as:

$$G = \{(x_i, y_j)\}_{i=1,j=1}^{M,M}, \tag{3.6}$$

with M=192 points in each dimension, creating a regular grid with M^2 total points.



(a) LiDAR point cloud. **(b)** Interpolated point cloud.

Figure 3.6: Top-down view showing the effect of interpolating the LiDAR point cloud. Figure 3.6a shows the LiDAR point cloud, while Figure 3.6b shows the interpolated point cloud.

3.4 Voxel-based Terrain Representation

A dynamic voxel-based representation of the world is used to create an elevation map. What this means is that the world representation is built as a quad-tree grid, where grid cells are called voxels and contains the height estimation of the area covered by the respective voxel. The size of the top-level, i.e. the maximum

size, voxels are empirically chosen as $1 \times 1\,\mathrm{m}$. This choice is made both to simplify calculations, since the global coordinates used are in meters and since it captures sufficiently large spaces without over-simplifying areas. The top-level voxels can contain multiple sub-voxels. A global coordinate (x, y, z) can be deterministically mapped to a corresponding top-level voxel index by

$$(x_{\text{voxel}}, y_{\text{voxel}}) = (\lfloor x \rfloor, \lfloor y \rfloor). \tag{3.7}$$

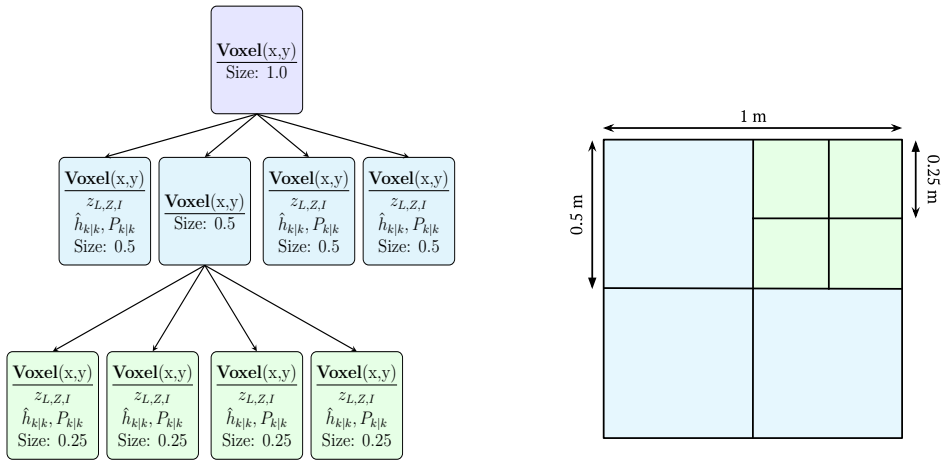
Since the coordinates (x, y) are in meters, this indexation maps all global coordinates to a metric grid. The top-level voxels are saved in a lookup table, where the voxel index is used as a key and the voxel data is the value.

3.4.1 Voxel Data Structure

The voxel-based representation is implemented using a hierarchical data structure resembling a quad-tree, where each parent voxel can be recursively subdivided into sub-voxels down to the minimum size of $0.125 \times 0.125\,\mathrm{m}$, which is empirically chosen inspired by the old fixed size. It is deemed small enough to capture complexity well for the path planning algorithm and big enough to not create unnecessary computations. This structure efficiently represents terrain at varying levels of detail, with fine-grained representation only where needed. Each voxel contains:

- A list of LiDAR measurements within its bounds
- A list of SC measurements within its bounds
- A list of interpolated LiDAR data points within its bounds
- The Kalman filter state and covariance
- References to any sub-voxels (if they exist)
- The voxel's size and center coordinates

This data structure provides memory efficiency, as detailed height information is only stored where terrain complexity requires it. Flat areas are represented with larger voxels, while complex terrain areas utilize smaller, more granular voxels. This quad-tree structure of voxels is shown in Figure 3.7.



(a) Quad-tree view. **(b)** Topdown map view.

Figure 3.7: Voxel quad-tree structure showing the hierarchical representation of terrain. The top-level voxel is subdivided into four sub-voxels, which can further be subdivided into smaller sub-sub-voxels. Each leaf-voxel contains measurements from LiDAR, stereo camera and interpolated data, along with Kalman filter states and covariances.

3.4.2 Recursive Voxel Management

The recursive nature of the voxel splitting and merging allows the system to adapt to changing terrain conditions in real-time. As new measurements are added, voxels may split into smaller sub-voxels to capture increased terrain complexity, shown in Figure 3.8. Conversely, if the Kalman estimated height in previously split voxels become more uniform (possibly due to improved estimation over time), sub-voxels may be merged back into their parent voxel to optimize computational resources, shown in Figure 3.9.

This dynamic management strategy allows the system to:

- Optimize memory usage by maintaining only necessary voxel subdivisions
- Focus computational resources on areas with complex terrain features
- Adapt to changing environmental conditions during vehicle operation

When a voxel is split, four sub-voxels are initialized and saved in the original voxel. The sub-voxel size is set as half the size of the parent voxel, ensuring the hierarchical quad-tree structure. If the sub-voxels of a parent voxel is merged, the sub-voxels are removed and the parent voxel inherits the maximum of the sub-voxels' Kalman states to avoid height degradation by merging. The general algorithm for the voxel management is shown in Appendix A.1.

3.4.3 Split/Merge Variance Calculation

The variance threshold of 0.01 m for voxel splitting was determined empirically to balance computational efficiency with terrain representation accuracy. The variance

is calculated using the standard formula:

$$\sigma^2_{\text{split}} = \frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 \tag{3.8}$$

where z_i represents the height of each measurement in the voxel from LiDAR or SC, $\bar{z}$ is the mean height and n is the number of measurements. This statistical measure effectively identifies voxels with significant terrain height variations that would benefit from more detailed representation, i.e. splitting into sub-voxels.

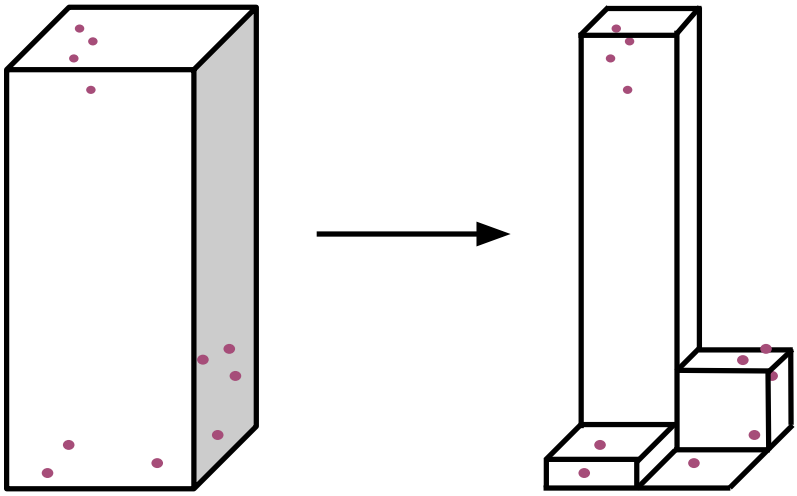


Figure 3.8: Visualization of how a voxel can be split based on the height variance of the measurements in the voxel.

For mergeing, the variance threshold is 0.008 m which was also determined empirically. The variance of the Kalman height estimation is calculated as

$$\sigma^2_{\text{merge}} = \frac{1}{4}\sum_{i=1}^{4}(\hat{h}_{k|k,i} - \bar{\hat{h}}_{k|k,i})^2 \tag{3.9}$$

where 4 is the number of direct sub-voxels in the voxel which is being tested for merge. $\hat{h}_{k|k,i}$ represents the Kalman height estimation of the sub-voxel i and $\bar{\hat{h}}_{k|k,i}$ represents the mean of all sub-voxels Kalman height estimations. This ensures that areas where we estimate low terrain complexity have higher computational efficiency.

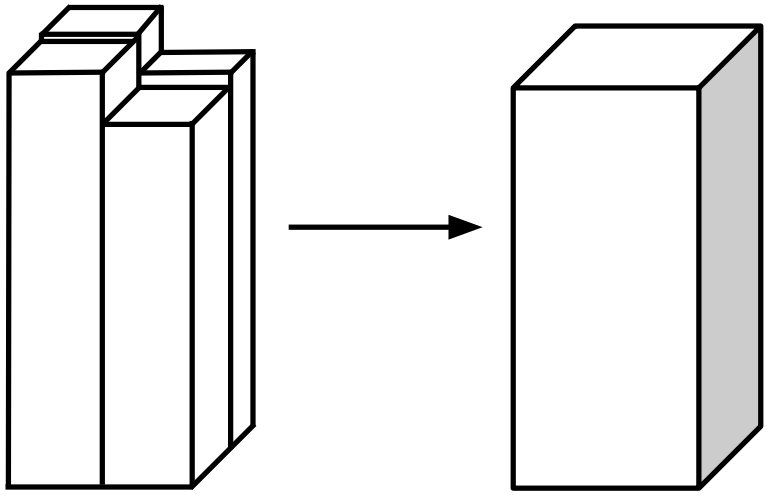


Figure 3.9: Visualization of how sub-voxels can be merged if the variance of the estimated heights are below a threshold value.

3.4.4 Measurement Assignment

When measurements are collected by the sensors, they are assigned to the corresponding voxel based on their global coordinates. The assignment process is as follows:

- Calculate the point variance, explained in Section 3.5.1
- Get the top-level voxel index from the point coordinates by utilizing Equation (3.7)
- If no entry exists for the voxel index in the lookup table, create a new voxel and add it to the table
- If the voxel has sub-voxels, recursively search through the sub-voxels until a leaf voxel that covers the measured point is found
- Add the measurement together with its variance to the voxel

When fusing the measurements in the voxel, the sensor uncertainty is taken into account. LiDAR points, with their high precision, can be used standalone in the elevation estimation. For SC data, which typically has higher uncertainty, the measurements are only used in conjunction with LiDAR measurements or interpolated LiDAR data to help support the SC data.

When running the Kalman filter on the voxel (explained in Section 3.5), all measurements contained in the voxel are condensed to a lower resolution by extracting the top 10 % highest measured z values in the voxel and taking the mean of those heights for both sensors. This is done both as a way to smoothen out potential outliers and also to reduce the number of computations required by the Kalman filter, reducing it down to only one Kalman iteration per voxel per time step. The top 10 % heights are chosen empirically to ensure that the most relevant information is used when estimating the height. Since the voxels are recursively split dependent on the height

variance, we can assume that the top 10 % heights are a good representation of the terrain height in the voxel.

3.4.5 Computational Efficiency Considerations

The multi-resolution approach significantly reduces complexity and computational load compared to using a fixed grid approach, like the one currently being used in the system with 0.1 m squares. The variance-based splitting criteria ensures that computational resources are allocated efficiently, with detailed analysis only occurring in areas where terrain complexity warrants it. This optimization is particularly important for real-time operation on the vehicle's onboard computing system.

An example showing how many voxels are needed for covering different areas by using fixed 0.125 m voxels vs dynamic voxels is shown in Figure 3.10.

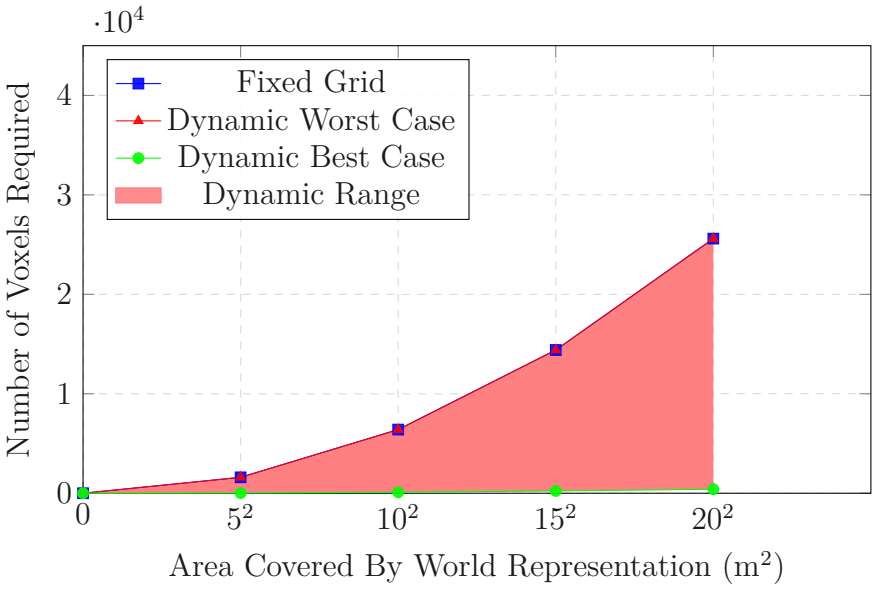


Figure 3.10: Complexity comparison between fixed grid (0.125 m) and dynamic voxel representation (0.125 m − 1 m). The fixed grid requires a constant number of voxels regardless of terrain complexity, while the dynamic voxel representation can adapt to terrain features, resulting in a range of required voxels.

Taking into account that each voxel calculates and saves the Kalman state and variance, the amount of computation and memory needed for creating the world representation as dynamic voxels is significantly lower than the computation and memory needed for a fixed grid representation. The dynamic voxel representation can adapt to terrain features, resulting in a range of required voxels.

3.5 Height Estimation using a Kalman Filter

For estimating the height of a voxel, we treat each voxel as an independent estimation problem and use a Kalman filter to create an estimate of the height. Each voxel

is checked once every second, if the voxel contains new un-fused measurements, the Kalman filter is run. The Kalman filter is used in this project because it provides an efficient and robust method for fusing sensor data with varying levels of uncertainty. By recursively estimating the state (in this case, the terrain height in a voxel) based on a series of noisy measurements, the Kalman filter ensures that the final height estimate is both accurate and reliable.

One of the key advantages of using the Kalman filter for this application is its ability to dynamically weigh measurements based on their uncertainty. For example, LiDAR data, which is highly accurate but sparse, can be given lower uncertainty, while stereo camera data, which is denser but less accurate, can be incorporated with higher uncertainty. This adaptive weighting allows the system to make the best use of all available data, even in challenging environments where sensor performance may vary.

By maintaining an estimate of the uncertainty (variance) alongside the height estimate, the Kalman filter also provides a measure of confidence in the terrain model. This is critical for autonomous navigation, as it allows the system to identify areas of the map that may require further exploration or caution during path planning.

For the basis of deriving our Kalman equations, we assume a static world, which means that at a given coordinate in the world, the height (h) will remain (close to) constant across all timesteps

$$h_k = h_{k-1}. \tag{3.10}$$

Modelling the world with this assumption makes sense for our purpose, since we can safely assume the vehicle will be operating in an enclosed environment without moving agents. We use a single-state system where the only state in a voxel is the height since that is the only thing we are interested in estimating.

3.5.1 Model Design Choices

Since a static world is assumed and the only state is the height h, the process model A is set as $A = 1$, which means that our Kalman prediction follows the convention in Equation (3.10). For the process noise $w_k \sim \mathcal{N}(0, Q)$, we add a small variance to account for deformable terrain such as vegetation, mud or similar, q_d. This parameter is chosen empirically to be small enough to not affect the height estimation significantly, but large enough to account for small variations in the terrain. The final prediction model thus becomes

$$\begin{aligned} h_k = h_{k-1} + w_k, \quad w_k \sim \mathcal{N}(0, Q) \\ \text{where } Q = q_d = 0.0001 \end{aligned} \tag{3.11}$$

For the measurement model, we assume for every measured point (x, y, z) that

$$z = Hh + r, \quad r \sim \mathcal{N}(0, R) \tag{3.12}$$

where z is the measured height at (x, y), h is the real height, r is the sensor noise and R is the sensor covariance matrix. Extracting the z value from the measured

points, the measurement model simply becomes $H = 1$ given a measurement from one sensor. The measurement model is dynamically updated depending on how many sensors are available. For example, if a voxel contains measurements from both LiDAR and SC, $z = [z_{LiDAR} \quad z_{SC}]^\top$, the measurement model would become $H = [1 \quad 1]^\top$ to account for both measurements. This would also mean that the covariance matrix would extend to become

$$R = \begin{bmatrix} \sigma^2_{LiDAR} & 0 \\ 0 & \sigma^2_{SC} \end{bmatrix}. \tag{3.13}$$

3.5.1.1 LiDAR Variance

For the LiDAR, the sensor uncertainty is defined as $\pm 0.03\,\text{m}$ per point [11]. Since the only state variable is the height an additional uncertainty factor is included to account for the positional uncertainty in x and y for each point. Furthermore, the LiDAR is mounted at a downward angle in two different mounting setups (see Figures 3.2, 3.3), this in some cases result in objects at greater distances not being fully captured in the scans.

Sensor setup 1 with the downward tilt of 20°, combined with the circular pattern of the LiDAR scans, results in the height of objects directly in front of the LiDAR not being measured properly. This can be accounted for by an exponential two variable function which ensures that measurements at a distance directly in front of the LiDAR is trusted less. For sensor setup 2 with 15° tilt, this is not needed as the LiDAR has theoretically infinite range up to the height of the LiDAR mounting point. The LiDAR variance for the first sensor setup is expressed as

$$\sigma^2_{LiDAR}(x, y) = \alpha^2_L + \beta_L y^2 \cdot e^{-\varepsilon_L x^2} + \sigma^2_{L,xy} \tag{3.14}$$

and is plotted in Figure 3.11. The parameter $\alpha_L = 0.03$ is the LiDAR point uncertainty, $\beta_L = 0.009$ is a tunable parameter for the longitudinal distance dependency, x is the lateral distance to the point and y is the longitudinal distance to the point, from the LiDAR. $\varepsilon_L = 0.1$ is a tunable parameter for the lateral tilt dependent uncertainty and $\sigma^2_{L,xy} = 0.02$ accounts for the positional uncertainty in x and y.

For the second sensor setup with the mount angle of 15°, the variance is expressed as

$$\sigma^2_{LiDAR} = \alpha^2_L + \sigma^2_{L,xy} \tag{3.15}$$

with no distance dependent factors, trusting the LiDAR equally for all points.

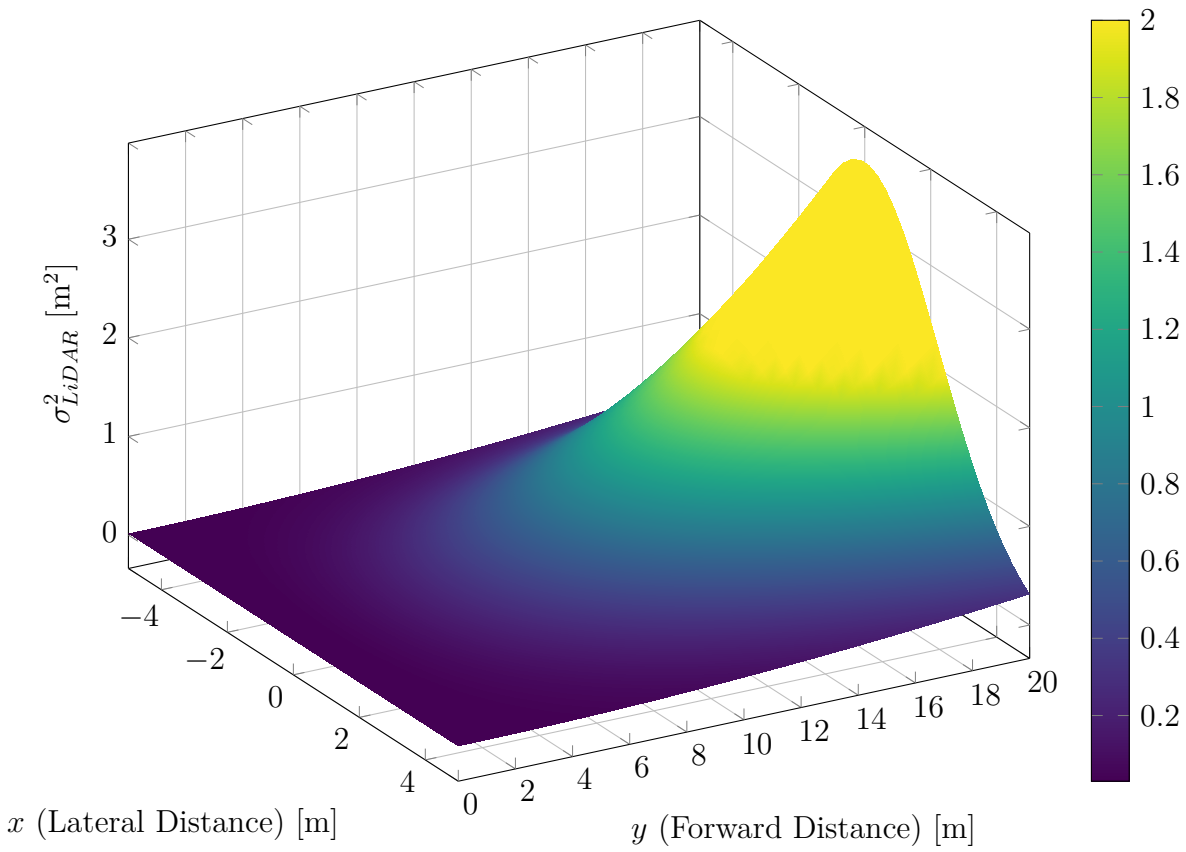


Figure 3.11: 3D surface plot of LiDAR measurement variance $\sigma^2_{LiDAR}(x, y)$, showing higher uncertainty directly in front of the sensor.

3.5.1.2 Stereo Camera Variance

For the SC, the uncertainty increases quadratically with the distance to the triangulated point. This is because the SC relies on triangulation in stereo images to generate depth information, explained in Section 2.1.2. As with the LiDAR an additional factor is included to account for positional uncertainty in x and y, this time scaled by the distance to the voxel to account for the distance dependency. The SC covariance is given by

$$\sigma^2_{SC}(x, y) = \alpha^2_{SC} + (\beta_{SC}(||(x, y)|| - 1))^2 + \sigma^2_{SC,xy}||(x, y)|| \tag{3.16}$$

and is shown plotted in Figure 3.12. The parameters in the SC covariance equation are defined as follows: $\alpha_{SC} = 0.3$ is a base uncertainty factor, $\beta_{SC} = 0.04$ is a tunable parameter controlling the quadratic distance dependency, $||(x, y)||$ is the distance from the SC to the point in the x, y-plane and $\sigma^2_{SC,xy}$ accounts for the positional uncertainty in x and y, scaled by the distance to the point. This formulation ensures that both distance-related and variance-related uncertainties are appropriately captured in the covariance calculation.

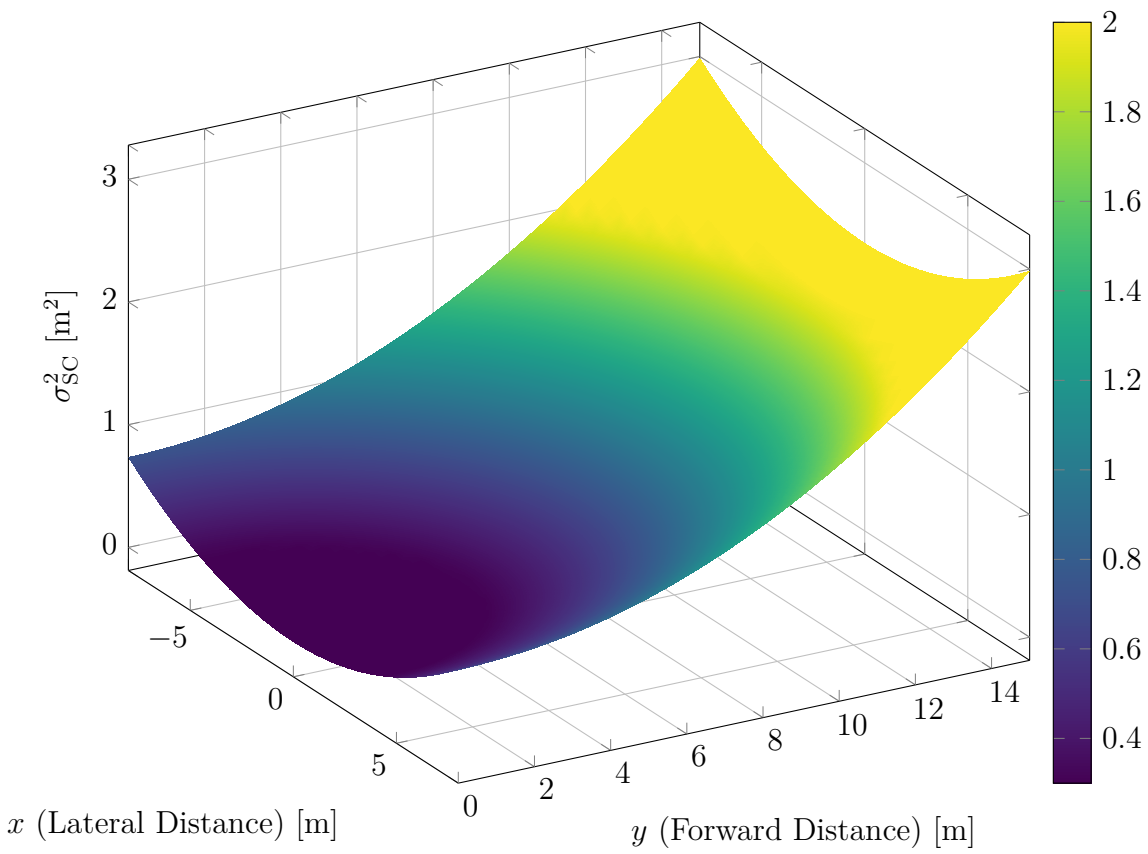


Figure 3.12: 3D surface plot of SC measurement variance $\sigma^2_{SC}(x, y)$ as a function of lateral and forward distance. Variance increases quadratically with distance due to triangulation uncertainty and linearly due to positional uncertainty.

3.5.1.3 LiDAR Interpolated Variance

To quantify the spatial variance of interpolated points, σ_i^2 relative to nearby LiDAR measurements, a method based on nearest-neighbor search and configurable noise modeling is used. First, a 2D k-d tree is constructed from the x, y components of the available LiDAR point cloud to enable fast nearest-neighbour queries. For each interpolated point j, the two nearest LiDAR points are identified and their distances d_1 and d_2 are retrieved, example shown in Figure 3.13. A maximum threshold distance $d_{\max} = 1$, empirically defined, is used to normalize the distance:

$$\alpha_j = \frac{\min(d_1, d_2, d_{\max})}{d_{\max}}. \tag{3.17}$$

This normalized distance $\alpha_j \in [0, 1]$ serves as a spatial weighting factor, where smaller distances indicate greater confidence in the local measurement.

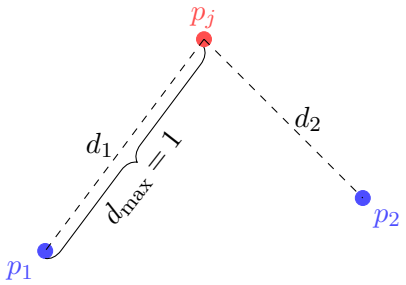


Figure 3.13: Visualization of selection of d_1, d_2.

Finally, the total variance is obtained by adding the weighting factor, α_j, to the intrinsic LiDAR variance, σ^2_{LiDAR} (obtained from Equation (3.14)):

$$\sigma^2_{i,j} = \sigma^2_{LiDAR,j} \cdot (\alpha_j + 1). \tag{3.18}$$

This approach ensures that points further from any reliable LiDAR observation are assigned a higher variance (shown in Figure 3.14), reflecting reduced confidence in the interpolated data.

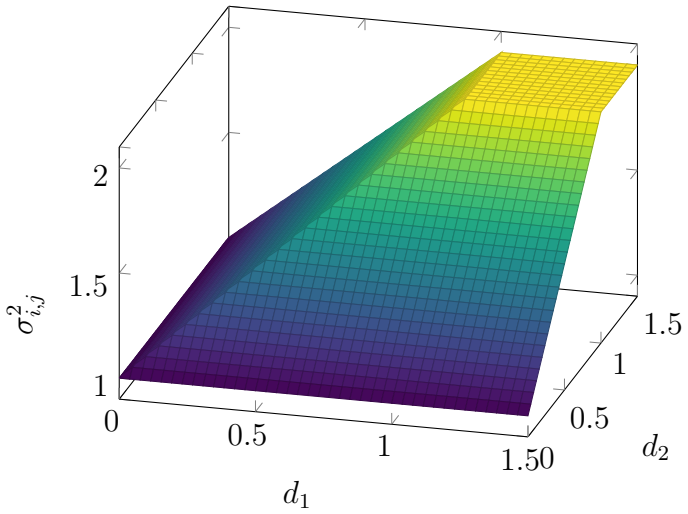


Figure 3.14: Interpolation variance as a function of distances d_1, d_2.

3.5.2 Kalman Equations

With our model design, the Kalman prediction becomes as follows

$$\hat{h}_{k|k-1} = \hat{h}_{k-1|k-1} \tag{3.19}$$
$$P_{k|k-1} = P_{k-1|k-1} + Q \tag{3.20}$$

where Q is the process noise described by $Q = 0.0001$ and the update step is given by

$$\hat{h}_{k|k} = \hat{h}_{k|k-1} + K_k v_k \tag{3.21}$$
$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^\top \tag{3.22}$$

where

$$K_k = P_{k|k-1}H^\top S^{-1} \tag{3.23}$$

$$S_k = HP_{k|k-1}H^\top + R \tag{3.24}$$

$$v_k = z_k - H\hat{h}_{k|k-1} \tag{3.25}$$

where H is a vector dynamically updated by 1s dependent on the size of z_k. Through this design, measurements from multiple sources are fused to improve the accuracy and reliability of the state estimation. The Kalman fusion process is visualized in Figure 3.15.

R is the sensor covariance matrix described in Section 3.5.1. The covariance update, given by Equation (3.22), reflects the certainty of the height estimate. A lower covariance indicates higher confidence in the estimate, as it incorporates the measurement uncertainty and the model's predictive capability.

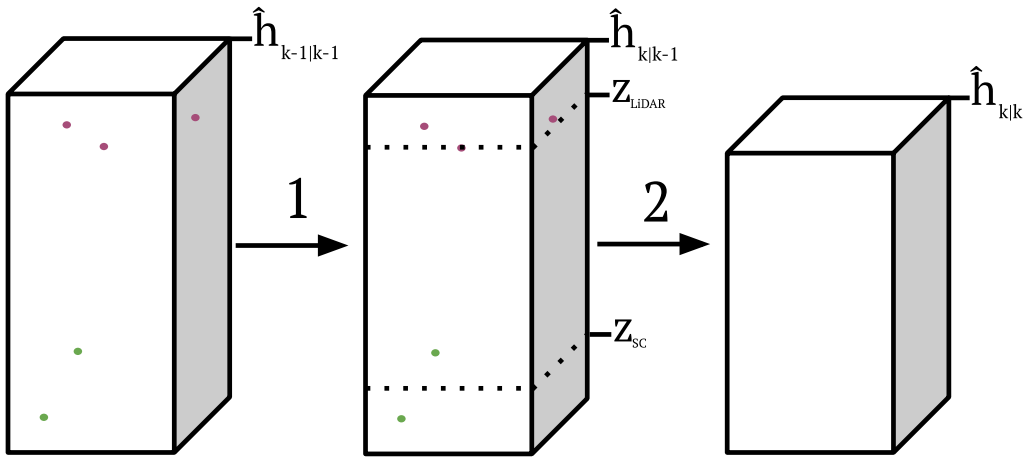


Figure 3.15: The Kalman filter height estimation process. The process starts with step 1, the prediction step, where the previous state estimate is propagated forward in time. Then, measurements from sensors are fused in step 2 to update the state estimate and covariance, resulting in a refined height estimate for the voxel.

3.6 Integration to Elevation Map

Putting it all together, the voxel-based height estimates are integrated into a structured elevation map that can be used for path planning and navigation. The elevation map is constructed by aggregating the height estimates from all leaf-voxels, i.e. voxels which do not have sub-voxels. Looping through these voxels and remappning them to global points in a point cloud creates a 3D point cloud of the terrain. This point cloud is then used as the world representation which the path planning algorithm can be run on.

The path planning can be evaluated by running a simulation of the vehicle on the generated point cloud. The path planning algorithm will then iteratively find the

best path in the simulated world based on a cost function, described briefly in Section 2.1.4. The covariance from the Kalman filter can be used as a good indicator of how certain we are of the height in the area covered by the voxel and can thus be used in the cost function to determine how much risk is acceptable in the path planning.

In summary, this chapter has described the methods of this thesis, including how data is collected and processed, how the voxel-based world representation is created and how the Kalman filter is used to estimate the height in the voxels. The integration of these components allows for a robust and efficient terrain representation that can be used for autonomous navigation and path planning in complex environments.

4

Results and Analysis

This chapter presents the experimental results and analysis of our LiDAR-SC fusion system for terrain mapping in forestry environments. We evaluate the system's performance across multiple dimensions, including sensor-specific accuracy, fusion effectiveness and the efficiency of our dynamic voxel-based representation.

4.1 Sensor Fusion Evaluation

In this section, the performance of the LiDAR and stereo camera sensors is analyzed individually and in combination. We assess the coverage, uncertainty and overall mapping quality achieved through our fusion approach. The performance is evaluated using several recorded scenarios:

Scenario 1: A flat parking lot with no vegetation in the path, allowing for a direct comparison of sensor performance. Obstacles are placed out and measured.

Scenario 2: A forested area with varying terrain complexity, the type of terrain where the vehicle is meant to drive, where the fusion approach is expected to demonstrate its advantages in challenging conditions.

Scenario 3: Using sensor setup 2. Driving at a steady pace towards a measured trash can standing on a flat parking lot. There are two variants of this scenario, in scenario 3a, the vehicle starts at a distance of 15 m from the trash can and drives forward until it stops at a distance of 3 m from the trash can. In scenario 3b, the vehicle starts at a distance of 30 m from the trash can and drives forward until it stops at a distance of 2 m from the trash can.

Scenario 4: Using sensor setup 1. Driving at a steady pace towards a measured pile of tires standing on a gravel road. The vehicle starts at a distance of 10 m from the pile of tires and drives forward until it stops at a distance of 2.7 m from the pile of tires.

(a) Scenario 1: Flat parking lot with obstacles.

(b) Scenario 2: Forested area with varying terrain.

(c) Scenario 3: Flat parking lot with obstacle.

(d) Scenario 4: Flat gravel road with obstacles.

Figure 4.1: Images showing what the camera sees in the four scenarios used for evaluating the sensor fusion.

4.1.1 LiDAR Performance

The LiDAR sensor provides sparse but accurate measurements of the terrain surface. Figures 4.4a, 4.4b shows the coverage pattern of LiDAR measurements, demonstrating both its strengths and limitations. In scenarios 1 and 3, sensor setup 2 is used, described in Section 3.2. This means that the LiDAR is mounted with the tilt angle 15°, which makes the effective range better. As can be seen in Figure 4.3b, the LiDAR is better able to capture accurate heights at a distance. For scenario 1 and 3, the uncertainty is therefore constant for all LiDAR measurements.

Scenarios 2 and 4 are recorded with sensor setup 1, meaning the tilt angle of 20° is used along with the distance dependent variance model described in Equation (3.14). This setup clearly limits the accuracy of LiDAR measurements at greater distances, as the variance increases with distance, which can be seen in Figure 4.3c. This requires us to trust the SC measurements more at distances, which is not ideal due to the noisy nature of the triangulated data.

4.1.2 Stereo Camera Performance

After filtering, the SC provides denser point clouds but with higher uncertainty, particularly at greater distances. Figures 4.4c, 4.4d illustrates the coverage pattern of the SC, showing the more uniform distribution of measurements compared to

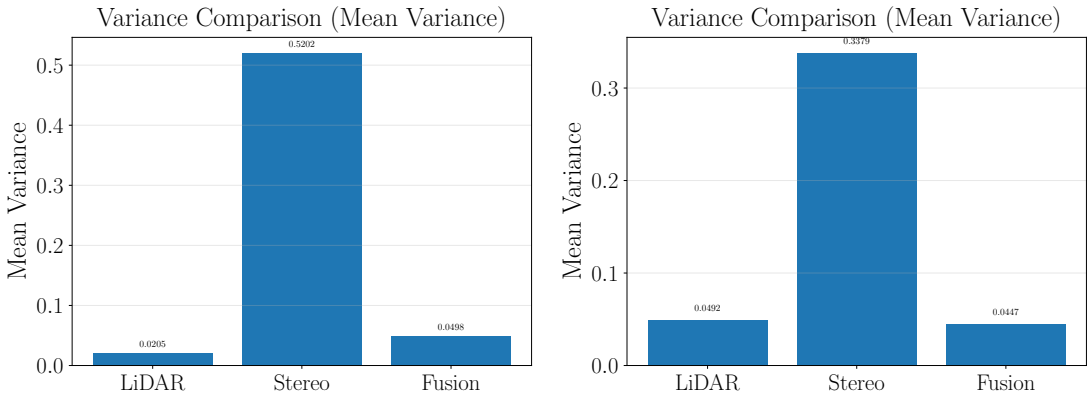
LiDAR.

The SC's variance is characterized by increasing quadratically with distance, as shown in Figure 3.12. At ranges beyond 10 m, the uncertainty becomes significantly higher than LiDAR when using sensor setup 2, limiting its effectiveness for precise terrain mapping at greater distances. The mean variance of the SC is also higher than the LiDAR variance, as can be seen in Figure 4.2.

4.1.3 Fusion Performance

The integration of LiDAR and SC data through our Kalman filter-based fusion approach yields significant improvements in both coverage and accuracy. As can be seen in Figures 4.4e, 4.4f, the fused map combines the coverage of both sensors, providing a more complete representation of the terrain. The data captured in the plot is from one fusion step. When the vehicle moves forward, new measurements will be introduced in new uncovered spots while previously estimated voxels will maintain their height estimation, creating an even richer map.

The variance also improves thanks to the Kalman filter, which effectively combines the measurements from both sensors while accounting for their respective uncertainties as can be seen in Figure 4.2. The mean variance of all voxels is in some cases higher than the mean variance of the LiDAR points, which has the lowest total variance. This can be explained by some voxels only containing measurements from SC and interpolated LiDAR data which have higher variance, which results in a higher total variance. The Kalman filter effectively reduces the uncertainty in regions where both sensors provide measurements, while maintaining the lower uncertainty of the LiDAR in regions where stereo data is highly uncertain. It is also worth noting that the data plotted is from one fusion step, meaning that the Kalman filter has not yet had time to converge. The variance will therefore be lower in the final map.



(a) Scenario 1 mean variance comparison.

(b) Scenario 2 mean variance comparison.

Figure 4.2: Figure showing the mean variance of the LiDAR, SC and Fusion approach from one fusion step.

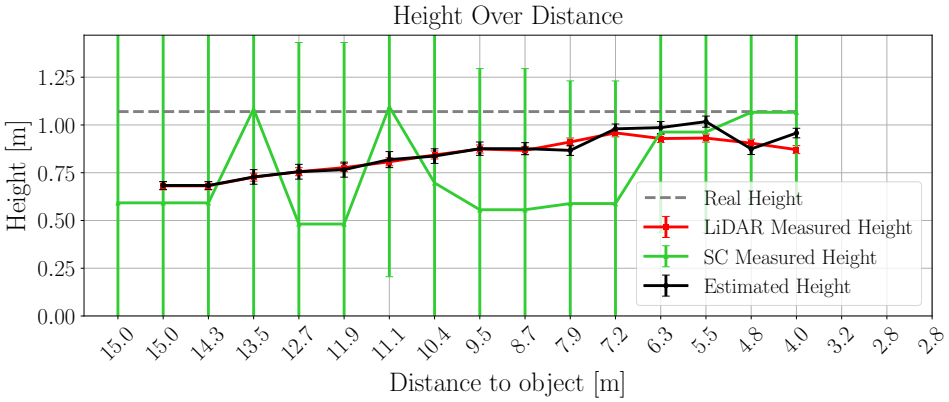
Height estimation accuracy

The accuracy of the height estimation is tested in scenarios 3 and 4 using the different sensor setups described in Section 3.2. Driving towards the measured objects and recording the height measurements from the LiDAR, SC and Fusion as well as the variances in the area covering the objects yields a view of how accurate the height estimation is with our approach. Shown in Figure 4.3 is plots from these tests.

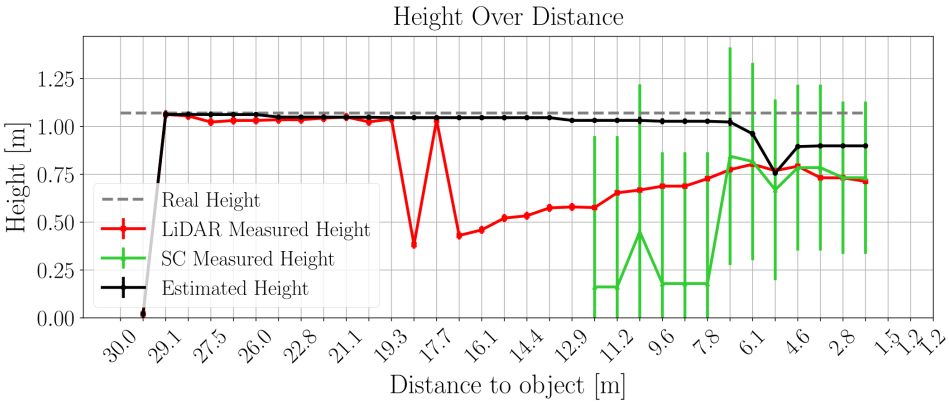
In scenario 3, sensor setup 2 is used, meaning the LiDAR variance is modelled to be constant. The measurements from the LiDAR are therefore trusted more than the SC measurements and thus the estimated height more closely follows the LiDAR measurements, effectively eliminating the noise from the SC measurements. Though, the LiDAR measurements are quite far away from the true height in this case. This is likely due to the top-most LiDAR scan line missing the trash can and going above it. Thus the measurements from the LiDAR in this case are captured by the second top-most scan line. This is a clear limitation of using a sparse LiDAR for measuring height in this way.

In scenario 3b, the LiDAR is able to accurately capture the true height of the obstacle from 30 m distance. At around 18 m from the obstacle, the top-most LiDAR scan line reaches above the trash can, possibly due to a slight ground slope. The second scan line hits the trash can at 0.4 m above the ground, which is what the LiDAR then reports as the height. However, our estimation in the area remain close to the true height. In the top-level voxel covering the obstacle, previous correctly estimated sub-voxels by the old measurements remain, as the new measurements from the second scan lines are only hitting the front of the obstacle. These previously estimated voxels aren't updated with new measurements until the vehicle gets closer.

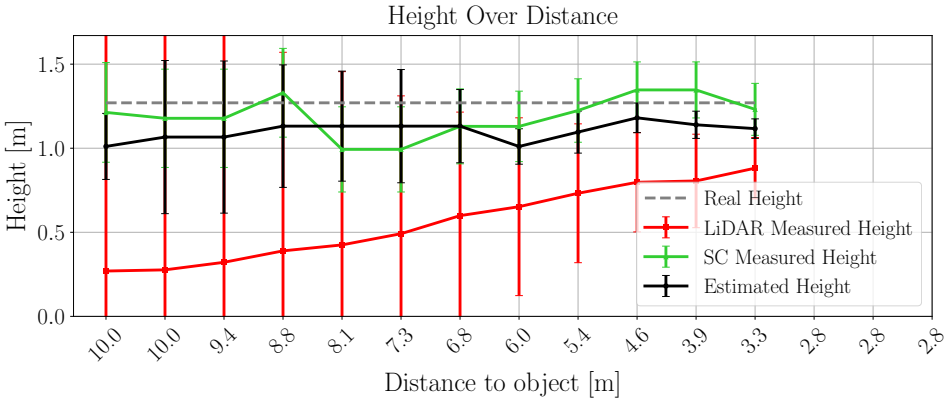
In scenario 4, sensor setup 1 is used along with the distance dependent variance model. The LiDAR measurements are therefore not trusted as much at a distance, which makes the Kalman filter trust the SC measurements more. As can be seen in Figure 4.3c, the LiDAR measurements are far from the true height, creating a scenario similar to the one shown in Figure 3.2. In this case, the SC measurements are much closer to the true height and thus the estimated height is therefore improved greatly by smart fusing of measurements from both sensors, compared to only using the LiDAR.



(a) Scenario 3a height accuracy. Plotted is the height estimation from the LiDAR, SC and Fusion and the real height of the object. The error bars show the variance for each sensor.

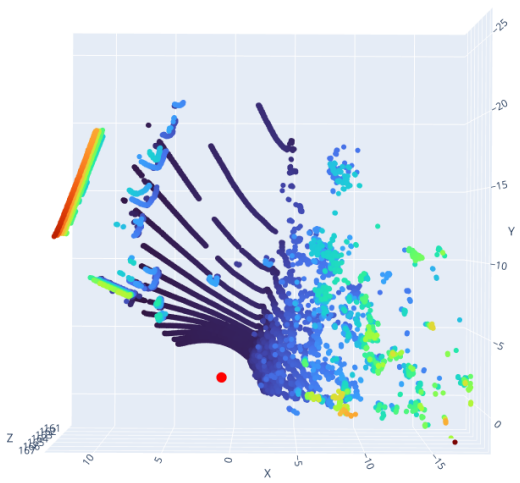


(b) Scenario 3b height accuracy. Plotted is the height estimation from the LiDAR, SC and Fusion and the real height of the object. The error bars show the variance for each sensor.

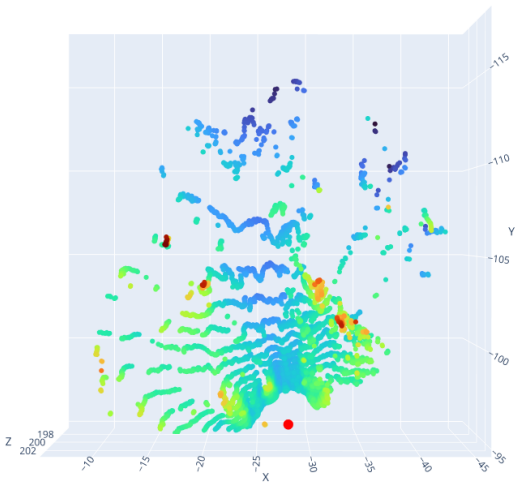


(c) Scenario 4 height accuracy. Plotted is the height estimation from the LiDAR, SC and Fusion and the real height of the object. The error bars show the variance for each sensor.

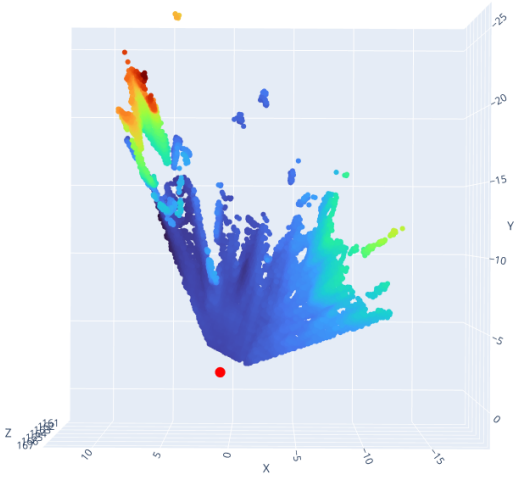
Figure 4.3: Height accuracy evaluation in two scenarios, created by driving towards measured objects and recording the heights and variances from the different sensors as well as the fusion.



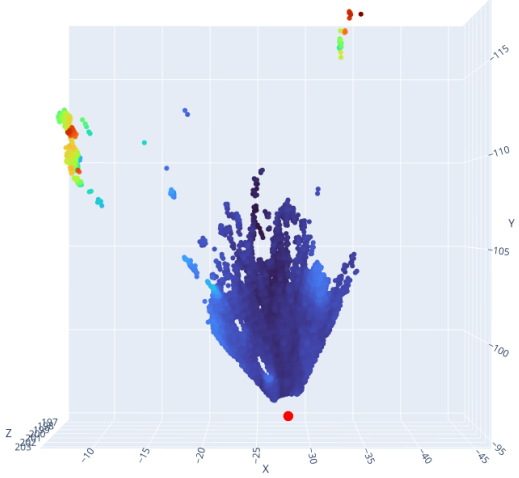
(a) Scenario 1 LiDAR coverage.



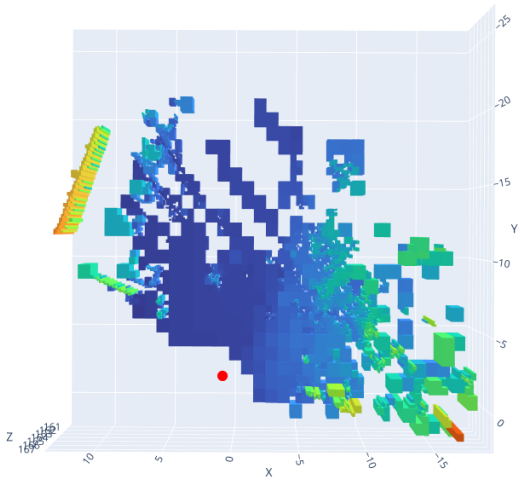
(b) Scenario 2 LiDAR coverage.



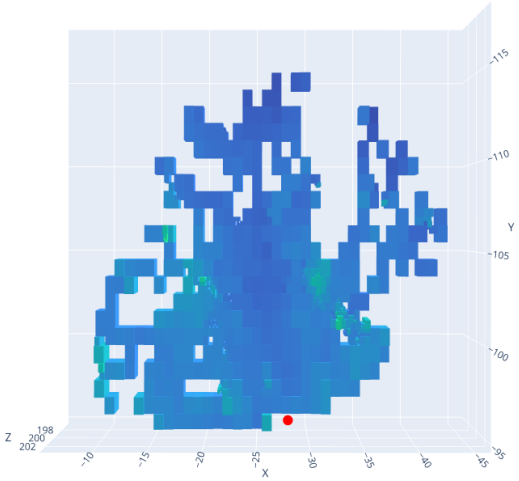
(c) Scenario 1 SC coverage.



(d) Scenario 2 SC coverage.



(e) Scenario 1 Fusion coverage.



(f) Scenario 2 Fusion coverage.

Figure 4.4: The data from LiDAR, SC and Fusion viewed from top down in scenario 1 and 2. The red dot represents the vehicle position and the color map shows the height of the terrain, darker blue means lower terrain and brighter colors and red means higher terrain.

4.2 Dynamic Voxelization Analysis

This section describes the performance of our dynamic voxelization approach, which adapts the map resolution based on terrain complexity. The voxelization process is designed to optimize memory usage and computational efficiency.

4.2.1 Adaptive Resolution Performance

To demonstrate the benefits of dynamic voxelization, we evaluate the system in two distinct scenarios with varying terrain characteristics, as illustrated in Figure 4.5:

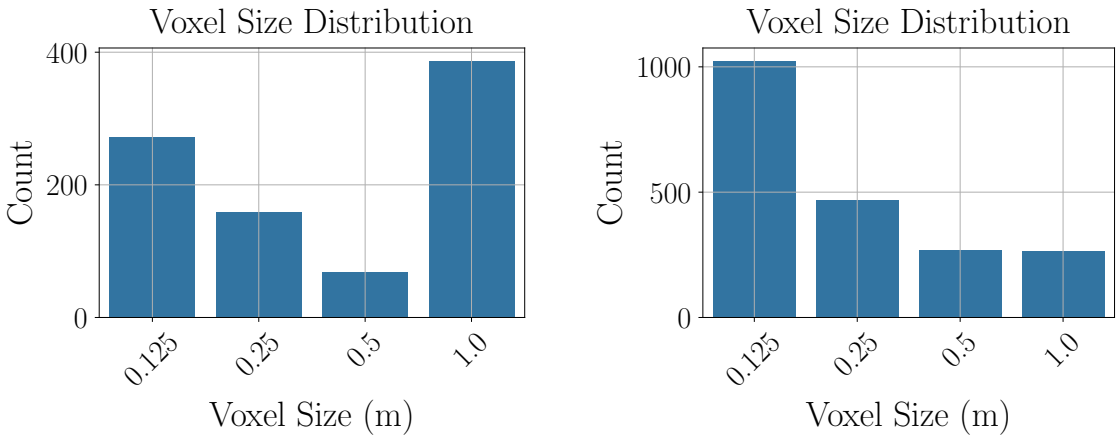
Scenario 4: Described in Section 4.1.
Scenario 5: A forested area with varying terrain complexity, the type of terrain where the vehicle is meant to drive, where the fusion approach is expected to demonstrate its advantages in challenging conditions.

(a) Scenario 4: Flat gravel road with obstacles.

(b) Scenario 5: Forested area with varying terrain at standstill.

Figure 4.5: Images showing what the camera sees in the two scenarios. The left image is from the flat gravel road, while the right image is from the forested area.

The adaptability of our approach is illustrated in Figure 4.6, which shows the distribution of voxel sizes for both scenarios when at standstill. In scenario 5, a higher number of small voxels is observed, reflecting the need for finer resolution in areas with greater terrain complexity. Despite this, the total number of top-level voxels remains nearly constant between the scenarios, indicating that the system maintains global structure while adjusting resolution locally.



(a) Distribution of voxel sizes from scenario 4, with the total number of top-level voxels being 545.

(b) Distribution of voxel sizes from scenario 5, with the total number of top-level voxels being 567.

Figure 4.6: Distribution of voxel sizes showing how the system adapts resolution to terrain complexity. Smaller voxels are used in areas with high variation while larger voxels represent uniform regions.

Table 4.1 provides a comparison between the dynamic voxelization approach and a traditional fixed-size voxel grid using 0.125 m resolution, with measurements collected from scenario 4 when driving forward. The table presents two key performance metrics that are critical for real-time autonomous operation:

- **Memory Usage (MB):** quantifies the total memory allocation required to maintain the terrain representation. The dynamic voxelization approach consumed 26.57 MB compared to 38.88 MB for the fixed voxelization, representing a 31.66 % reduction in memory footprint.
- **Mean Update Time (ms):** measures the average time required to process new sensor data and update the terrain representation during operation. This includes all computational steps: mapping measurements to voxels, performing Kalman filter updates and executing split/merge operations where needed (for the dynamic approach). The dynamic approach required 607.80 ms per update compared to 1090.43 ms for the fixed approach, yielding a 44.27 % improvement in computational efficiency. Lower update times directly translate to higher refresh rates for the world model.

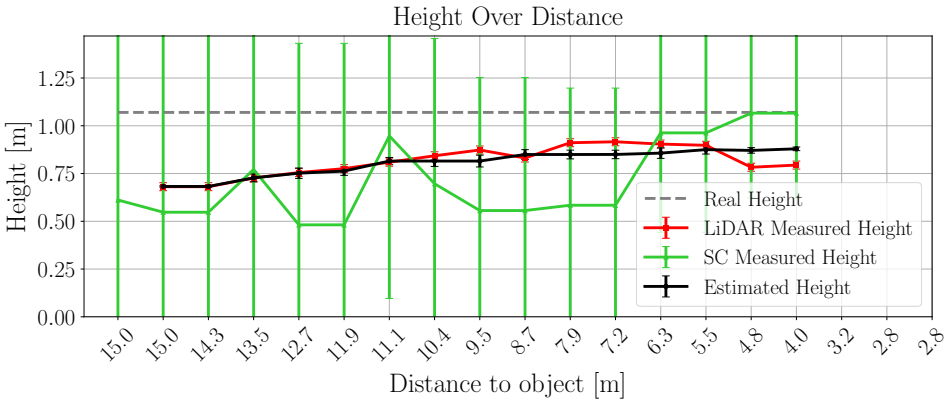
Metric	**Dynamic Voxelization**	**Fixed Voxelization**
Memory Usage (MB)	26.57	38.88
Mean Update Time (ms)	607.80	1090.43

Table 4.1: Performance comparison of running scenario 4: Dynamic vs Fixed Voxelization

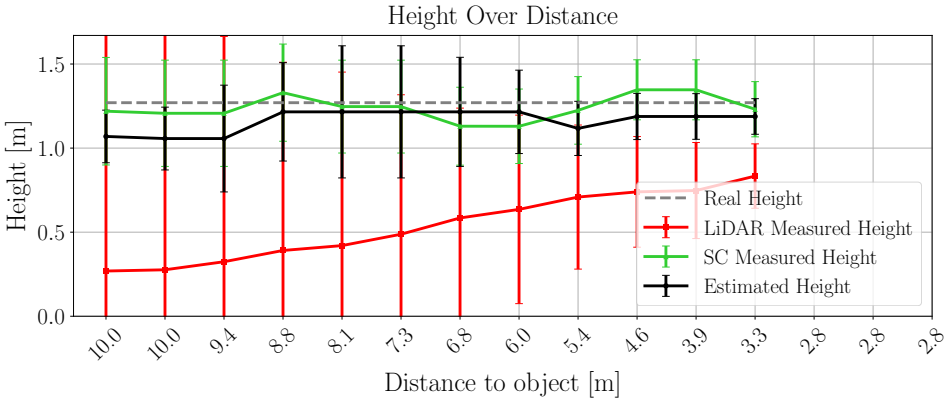
4.2.2 Dynamic Voxel Height Accuracy

The accuracy of the height estimation when using fixed voxels is tested on scenario 3a and 4 in the same manner as is done in Section 4.1.3 and is plotted in Figure 4.7. The results vary slightly and may be hard to spot when just looking at the plots, the exact metrics for the scenarios can be seen in Table 4.2.

In scenario 3a, the RMSE is slightly better for the dynamic voxels but for scenario 4, the RMSE is slightly better for the fixed voxels. The two approaches are expected to have similar performance in this case, as the purpose of the dynamic voxelization is not to improve the accuracy of the height estimation, but rather to improve the computational efficiency and memory usage.



(a) Scenario 3a height accuracy when using fixed voxels. Plotted is the height estimation from the LiDAR, SC and Fusion and the real height of the object. The error bars show the variance for each sensor.



(b) Scenario 4 height accuracy when using fixed voxels. Plotted is the height estimation from the LiDAR, SC and Fusion and the real height of the object. The error bars show the variance for each sensor.

Figure 4.7: Height accuracy evaluation in two scenarios using fixed 0.125 m voxels, created by driving towards measured objects and recording the heights and variances from the different sensors as well as the fusion.

The performance of these test can vary greatly, as with the fixed voxelization, the

resolution of the world is changed and thus the measurements may end up in different voxels compared to the dynamic case. Since the minimum requirement for running the height estimation in a voxel is that it has LiDAR measurements or interpolated LiDAR and SC measurements, the height estimation may vary in uncertain ways.

	Scenario 3a	**Scenario 4**
RMSE fixed (m)	0.2724	0.1256
RMSE dynamic (m)	0.2576	0.1763

Table 4.2: RMSE height estimation comparison of running scenario 3a and 4: Dynamic vs Fixed Voxelization

4.3 Terrain Navigation Evaluation

To evaluate the practical benefits of our approach, we conducted a navigation test in two challenging forest environments. While the path planning algorithm isn't a result from this master thesis, it is ultimately what our world representation is made for. Testing how vehicle runs when using our final world representation is a test showing the potential of running this approach online on the vehicle for usage in the autonomous navigation. The test compared the path planning performance using maps generated from:

- **Old approach:** Aggregated and filtered LiDAR-only data in fixed 0.1 m resolution grid.
- **Our approach:** Fused sensor data with dynamic voxelization.

The vehicle was first manually driven through the test area and data from both sensors was recorded. From the recorded sensor data, we could generate one world representation with the old approach and one with our approach. These world representations were then used as environments in a vehicle simulation program where the path planning could be tested offline. The vehicle's autonomous navigation was evaluated in the simulation by comparing the planned path(the manually driven route) to the trajectory taken by the vehicle in the simulation.

This was tested on two scenarios, scenario 2 (see Section 4.1), named route 1 and another similar scenario, named route 2, recorded from the same forest environment. Metrics from these tests can be seen in Table 4.3. The planned path length is the length of the manually driven path. The path lengths for our and the old world representations are how long the simulated vehicle was able to drive in each world representation without stopping due to finding no drivable paths. The RMSE is the root mean square error between the planned path and the driven paths in simulation.

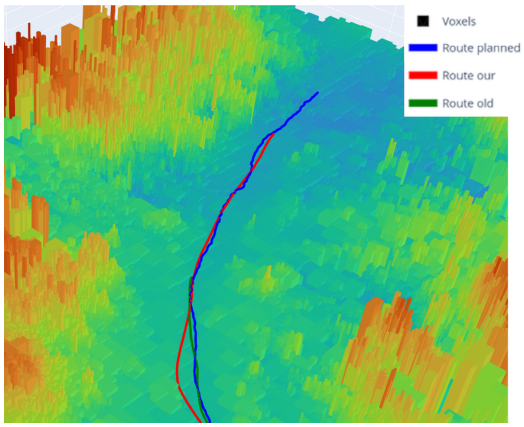
Table 4.3: Path planning performance comparison between our approach and the old approach in two scenarios.

	Route 1	**Route 2**
Planned length (m)	45.007	24.807
Our path length (m)	38.994	20.997
Old path length (m)	18.995	21.497
Our RMSE (m)	0.397	0.385
Old RMSE (m)	0.313	0.437

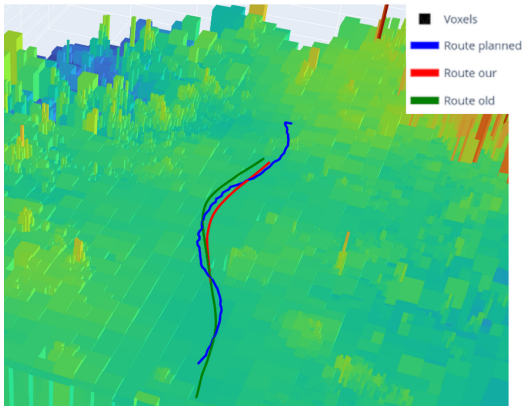
Since the planned path was recorded by manually driving through the forest, we are certain that this is a drivable path. The goal is to keep as close to the planned path as possible while also being able to find drivable path segments along the entire planned paths. Figure 4.8 shows the paths generated by each approach in a section of the test area containing slopes, vegetation and obstacles.

In route 1, the simulated vehicle was able to drive 105.3 % longer with our world representation while also keeping the RMSE at a similar score. Having a longer total path means that with our world representation the vehicle is able to find drivable path segments for a longer stretch. In the old approach, the vehicle stops after 18.995 m due to not finding any drivable paths, potentially due to sensor noise. This is a great result and shows that our approach is able to filter out potential noise that the old approach is unable to. We have a slightly higher RMSE, which can be explained by the vehicle making a slight left turn at the start of the scenario (seen in Figure 4.8a) when running with our world representation. It is unclear why this happens in this case, visual inspection of the map doesn't show any obstacles in the way here but the path planning algorithm still decides that this is the better path to take. This would have to be investigated further to draw most robust conclusions. The overall RMSE is still very similar between the cases and finding longer drivable routes is overall a more desirable goal.

In route 2, the simulated vehicle is able finish the entire planned path in both world representations, but with our world representation keeping the RMSE down. This again shows how our approach has similar or improved performance compared to only using the old single-sensor approach. Worth noting that the path planning stops the vehicle if the length of the planned route is too short to keep planning paths, thus it may seem like both approaches stops prematurely, but this is the longest possible path which can be produced in this case.



(a) Planned and generated paths shown in our world representation for forest route 1.



(b) Planned and generated paths shown in our world representation for forest route 2.

Figure 4.8: Path planning performance comparison. The world representation is from our approach. The blue line represents the planned path, while the green line represents the path generated by the old approach. The red line shows the path generated by our approach.

5

Conclusion and Discussion

This thesis has presented a novel sensor fusion approach that integrates LiDAR and SC data for enhanced terrain mapping in forestry environments. By leveraging the complementary strengths of these sensors and implementing a dynamic voxel-based representation, our system effectively addresses critical challenges in autonomous navigation for forestry vehicles, achieving improved terrain perception in complex environments.

5.1 Summary of Contributions

The primary contributions of this work include a robust sensor fusion framework that effectively combines the high accuracy of LiDAR measurements with the dense coverage of SC data, resulting in more complete and accurate terrain maps. We have developed a dynamic voxelization approach that adapts map resolution based on terrain complexity, optimizing memory usage while maintaining high fidelity in regions of interest. Additionally, we introduced an uncertainty-aware fusion algorithm based on Kalman filtering that properly accounts for sensor-specific error characteristics, particularly the distance-dependent uncertainties of both LiDAR and SC measurements. Our approach is validated through comprehensive evaluation across multiple real-world scenarios, demonstrating quantifiable improvements in mapping accuracy, coverage and computational efficiency compared to single-sensor approaches.

5.2 Key Findings

Our experimental results demonstrate several important findings that highlight the advantages of combining complementary sensors with adaptive mapping techniques for robust and efficient terrain perception in autonomous forestry operations.

5.2.1 Dynamic Voxelization Benefits

The dynamic voxelization approach developed in this thesis shows improvement over traditional fixed-grid elevation maps. By adaptively adjusting voxel resolution based on terrain complexity, our system achieved remarkable gains while preserving mapping quality in critical areas. Quantitative evaluation revealed a 31.65 % reduction in memory usage compared to fixed-size voxel grids of equivalent coverage (see Table 4.1). This memory efficiency is particularly significant for autonomous forestry vehicles with limited computational resources.

Beyond memory savings, the dynamic approach demonstrated a 44.27 % improvement in map update time, allowing for more responsive real-time operation. This performance enhancement stems from the optimized data structure that concentrates computational resources on complex terrain regions while using larger voxels for homogeneous areas. The adaptive resolution also proved particularly valuable in forestry environments, where terrain complexity varies dramatically between open areas and dense vegetation or obstacle-rich regions.

5.2.2 Sensor Fusion Effectiveness

The Kalman filter-based sensor fusion approach successfully leveraged the complementary strengths of LiDAR and SC sensors, achieving substantial improvements in terrain mapping capabilities while highlighting important considerations regarding computational complexity. Our evaluation demonstrates that the fusion approach provides enhanced terrain coverage and uncertainty quantification that neither sensor achieves independently, with measurable benefits for autonomous navigation in complex forest environments.

Analysis of accuracy metrics reveals that fusion effectiveness improve or maintain around the same accuracy compared to single-sensor use. However, the true value of fusion extends beyond pure accuracy metrics: the approach provides comprehensive uncertainty maps that enable more informed navigation decisions, effectively trading marginal accuracy improvements for substantial gains in system robustness and reliability.

The computational overhead of multi-sensor fusion, while non-trivial, yields valuable dividends in practical applications. Navigation performance evaluation revealed that the fusion approach enabled 105.3 % longer drivable paths in Route 1 scenarios, demonstrating superior obstacle avoidance and terrain assessment capabilities despite comparable RMSE values (0.397 m vs 0.313 m). This trade-off between computational complexity and enhanced navigation capability represents a strategic advantage for autonomous forestry operations, where conservative path planning often outweighs marginal accuracy gains in ensuring safe and reliable operation.

While our system does show promise in these tests, more extensive real-world evaluations are needed to fully understand the trade-offs between computational complexity and practical navigation performance. Future work should focus on optimizing the fusion algorithm for specific forestry tasks, potentially incorporating task-specific heuristics to further improve efficiency without sacrificing robustness. More tuning of the system could also be done to improve the performance of the Kalman filter, such as tuning the process noise and measurement noise parameters. The current system uses a constant process noise and measurement noise, which may not be optimal for all scenarios. Future work could explore adaptive noise estimation techniques to dynamically adjust these parameters based on environmental conditions and sensor performance.

5.3 Limitations and Future Work

Despite the promising results, several limitations and opportunities for future research remain. The current system relies on accurate sensor pose estimation to work accurately. Future work could explore integration with Simultaneous Localization And Mapping (SLAM) techniques to improve robustness in GPS-denied environments, something which can be prevalent in rural forest planting sites [31]. SLAM can be very useful in forest environments where the GPS signal is weak and the terrain is complex. The uncertainty models, while effective, could be further refined through online calibration that adapts to changing environmental conditions such as lighting, precipitation, or seasonal vegetation changes.

Different sensors could also be investigated for future use. While more expensive, a LiDAR with higher resolution in scan lines could be used to improve the mapping quality. An algorithm to calculate variance dependent on the distance between LiDAR scan lines could also be used to improve estimation when scan lines reach above objects. The SC used in the project, the ZED 2i, is running an deprecated SDK version, which could not be upgraded in this project due to higher level dependency issues. Upgrading the SDK and using StereoLabs new depth mode which uses neural networks to improve depth estimation could be used to improve the mapping quality significantly [32].

From an algorithmic perspective, the dynamic voxelization approach could be extended in several ways:

- Incorporating semantic information to adapt resolution based on object classification rather than purely geometric features.
- Implementing temporal process models that increase estimation uncertainty when voxels go unobserved for extended periods.
- Developing context-aware splitting and merging criteria that consider the vehicle's task and planning requirements.
- Exploring reinforcement learning approaches for optimizing voxel resolution based on navigation performance feedback.
- Evaluating alternative voxel shapes, such as hexagonal prisms.

With these suggestions, future work can build on the foundation established in this thesis to further enhance terrain mapping capabilities for autonomous forestry vehicles. By addressing the limitations identified and exploring new avenues for improvement, we can continue to advance the state of the art in sensor fusion and terrain perception for complex off-road environments.

5.4 Sustainability Aspects

This thesis contributes to the BraSatt project's goal of improving sustainability in the forestry industry by optimizing forest regeneration. With current sapling survival rates of only 70-75% after three years, our robust terrain mapping system

enables the *BraSatt 01* vehicle to navigate complex terrain more effectively, make better routing decisions, and ultimately create optimal conditions for planting. The system's obstacle detection capabilities help preserve biodiversity by avoiding young trees and sensitive areas, while more efficient navigation routes directly reduce fuel consumption and greenhouse gas emissions.

Beyond environmental benefits, this technology addresses the societal impact of automation in forestry. The manual planting process is monotonous, time-consuming, and physically demanding. By transforming this labor into machine operator roles utilizing human-machine interaction, we create safer working conditions while enabling enhanced data collection for continuous improvement in forestry practices. This transformation presents dual perspectives: while requiring operator licensing and education may reduce low-skill job opportunities, the creation of safer, higher-quality work environments and improved forest regeneration outcomes may outweigh these concerns.

As the forestry sector transitions toward more environmentally conscious practices, technologies like those presented in this thesis play an increasingly crucial role in balancing productivity with ecological responsibility, contributing to both immediate operational improvements and long-term forest sustainability.

5.5 Practical Implications

The developed system has significant practical implications for autonomous forestry operations. The improved terrain mapping capability enables safer and more efficient navigation through complex forest environments, potentially reducing operational costs and environmental impact. The system's ability to maintain high-quality maps with reduced computational resources makes it suitable for deployment on practical forestry vehicles with limited onboard computing capacity. The improved real-time performance allows for more responsive navigation, which is critical when operating in dynamic environments where terrain conditions may change rapidly.

In conclusion, this work demonstrates that intelligent fusion of complementary sensors, combined with adaptive mapping techniques, can improve terrain perception for autonomous vehicles operating in challenging off-road environments. Our approach strikes an effective balance between map quality, computational efficiency and practical applicability, addressing a critical need in the advancement of autonomous forestry operations. The integration of uncertainty quantification throughout the mapping pipeline enables not only better terrain representation but also more informed decision-making for autonomous navigation systems, ultimately leading to more robust and reliable operation in complex, unstructured environments.

Bibliography

[1] Södra, "Södra's Test Run of New Planting Machine Shows Promising Results," 2025, accessed: 2025-02-11. [Online]. Available: https://www.sodra.com/en/global/about-sodra/press/press-releases-global/sodras-test-run-of-new-planting-machine-shows-promising-results/

[2] O. Christenson and J. Lundgren, "Building a computer vision system for autonomous tree planting: Using yolo and u-net to find planting spots in clear-felled areas," Master's Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2022. [Online]. Available: https://hdl.handle.net/20.500.12380/304950

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

[4] J. Shan and C. K. Toth, *Topographic Laser Ranging and Scanning: Principles and Processing*. CRC Press, 2008.

[5] D. Beltran and L. Basañez, "A comparison between active and passive 3d vision sensors," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 725–730, 2012.

[6] M. Himmelsbach, F. v. Hundelshausen, and H.-J. Wuensche, "Lidar-based 3d object perception," in *Proceedings of 1st international workshop on cognition for technical systems (CoTeSys-2010)*, 2010, pp. 1–7.

[7] D. Pasechnik, "Triangulation methods and uncertainty," https://mrcal.secretsauce.net/triangulation.html, 2023, accessed: 2025-05-24.

[8] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[9] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[10] M. Wermelinger, F. Pomerleau, and R. Siegwart, "Navigation in forest environments: Fusing and evaluating uav imagery and lidar data," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2194–2200.

[11] V. LiDAR, "Velodyne vlp-16 user manual," 2016. [Online]. Available: https://velodynelidar.com/products/puck/

[12] W. Elmenreich, "An introduction to sensor fusion."

[13] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Deep sensor fusion for multiple modalities in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3927–3939, 2019.

[14] D. Weir and J. Zellner, "An introduction to the operational characteristics of all-terrain vehicles," SAE International, SAE Technical Paper 860225, 1986. [Online]. Available: https://doi.org/10.4271/860225

[15] A. Filgueira, H. Gonzalez-Jorge, S. Lagüela, and J. Armesto, "Performance of lidar technology in adverse weather conditions," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2017, pp. 1–8.

[16] K. Lim, P. Treitz, M. Wulder, B. St-Onge, and N. Flood, "Lidar remote sensing of forest structure," *Progress in Physical Geography*, vol. 27, no. 1, pp. 88–106, 2003.

[17] Stereolabs, "Zed 2i stereo camera," 2025. [Online]. Available: https://www.stereolabs.com/store/products/zed-2i

[18] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," *ACM Trans. Graph.*, vol. 26, no. 3, p. 24–es, Jul. 2007. [Online]. Available: https://doi.org/10.1145/1276377.1276407

[19] P. Szutor and M. Zichar, "Fast radius outlier filter variant for large point clouds," *Data*, vol. 8, no. 10, 2023. [Online]. Available: https://www.mdpi.com/2306-5729/8/10/149

[20] Y. Guan, H. Liao, Z. Li, J. Hu, R. Yuan, Y. Li, G. Zhang, and C. Xu, "World models for autonomous driving: An initial survey," *arXiv preprint arXiv:2403.02622*, 2024.

[21] G. Sten, L. Feng, and B. Möller, "Enhancing off-road topography estimation by fusing lidar and stereo camera data with interpolated ground plane," *Sensors*, vol. 25, no. 2, 2025. [Online]. Available: https://www.mdpi.com/1424-8220/25/2/509

[22] J. A. Smith and J. B. Doe, "A novel approach to biomedical research," *Journal of Biomedical Innovations*, vol. 15, no. 3, pp. 123–130, 2025, accessed: 2025-05-16. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10708681/

[23] X. Xu, L. Chen, C. Cai, H. Zhan, Q. Yan, P. Ji, J. Yuan, H. Huang, and Y. Xu, "Dynamic voxel grid optimization for high-fidelity rgb-d supervised surface reconstruction," 2023. [Online]. Available: https://arxiv.org/abs/2304.06178

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] D. Xu, Y. Li, X. Huang, H. Zhao, T. Darrell, and F. Yu, "Deepfusion: A unified multi-sensor multi-task architecture for autonomous driving," in *CVPR*, 2022.

[26] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[27] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Transaction of the ASME, Series D,Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, 03 1961.

[28] B. Delaunay, "Sur la sphère vide. a la mémoire de georges voronoï," *Izv. Math*, no. 6, 1934.

[29] Z. Li, C. Zhu, and C. Gold, *Digital Terrain Modeling: Principles and Methodology*, 1st ed. CRC Press, 2004. [Online]. Available: https://doi.org/10.1201/9780203357132

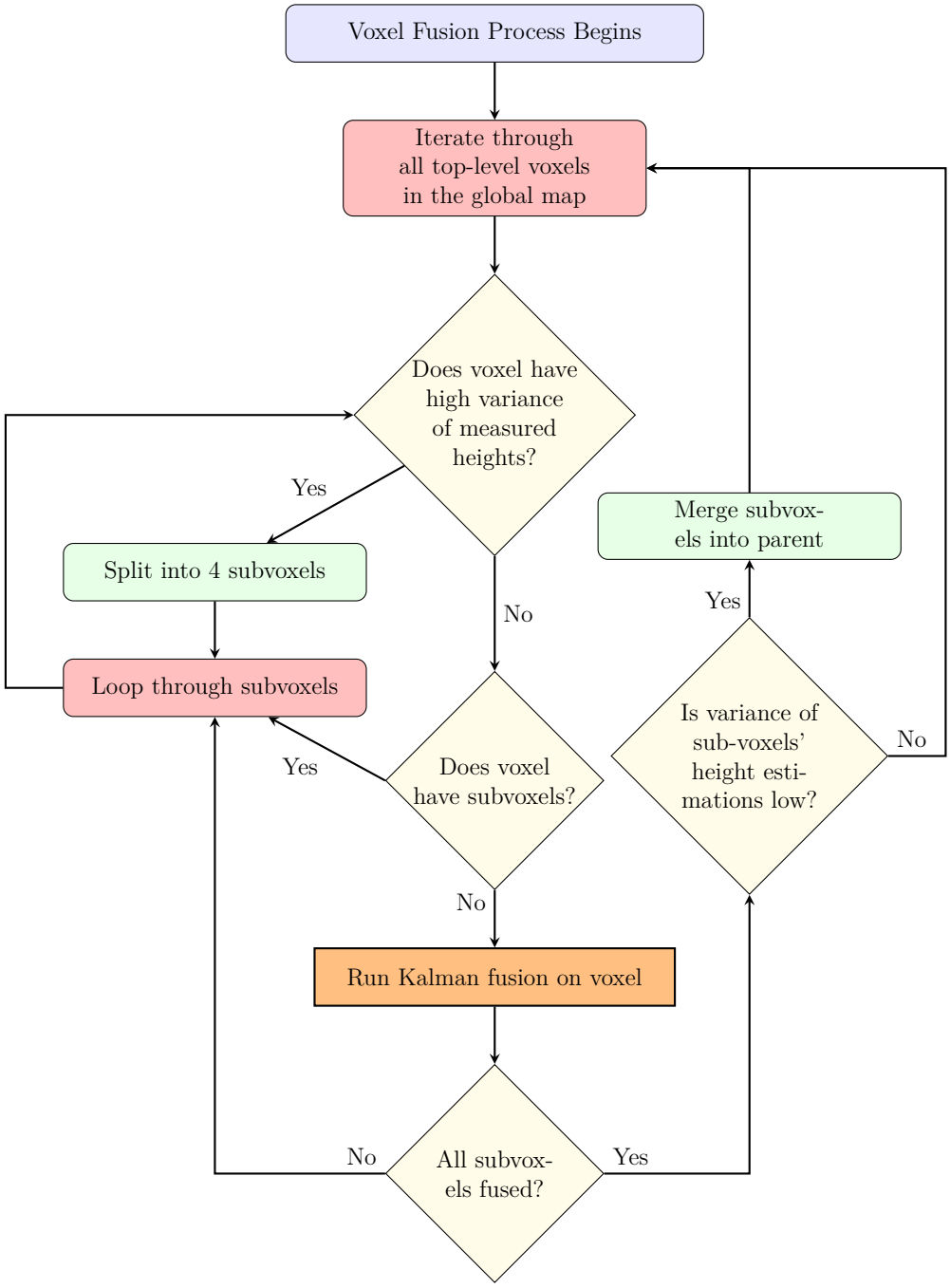
[30] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Software*, 2009. [Online]. Available: https://www.ros.org/

[31] W. Khaksar and R. Astrup, "Multi-sensor terrestrial slam for real-time, large-scale, and gnss-interrupted forest mapping," 2023. [Online]. Available: https://arxiv.org/abs/2310.01064

[32] Stereolabs. (2025, March) Stereolabs unveils zed sdk 5 with terra ai, revolutionizing vision-based sensing. Accessed: 2025-05-16. [Online]. Available: https://www.stereolabs.com/en-se/blog/introducing-zed-sdk-50

A
Appendix 1

A.1 Voxel Algorithm



DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY