UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Semantic Localization and Mapping in Forests

Guilherme Vicentim Nardari

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Guilherme Vicentim Nardari

Semantic Localization and Mapping in Forests

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Roseli Aparecida Francelin Romero

USP – São Carlos
April 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

```
        Vicentim Nardari, Guilherme
V633s      Semantic Localization and Mapping in Forests /
        Guilherme Vicentim Nardari; orientadora Roseli
        Aparecida Francelin Romero. -- São Carlos, 2023.
           102 p.


           Tese (Doutorado - Programa de Pós-Graduação em
        Ciências de Computação e Matemática Computacional) --
         Instituto de Ciências Matemáticas e de Computação,
        Universidade de São Paulo, 2023.


           1. Localização. 2. Mapeamento. 3. Segmentação
        Semântica. 4.  Aprendizado de Máquina. 5. Robótica.
        I. Aparecida Francelin Romero, Roseli, orient. II.
        Título.
```

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Guilherme Vicentim Nardari

Localização e Mapeamento Semânticos em Florestas

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Roseli Aparecida Francelin Romero

USP – São Carlos
Abril de 2023

Dedicated to my mother Silvia.

ACKNOWLEDGEMENTS

I would like to thank the most important people in my life who provided me with unwavering support and encouragement throughout my academic journey. To my family, thank you for always believing in me and pushing me to pursue my dreams. Your constant love and motivation kept me focused and determined to achieve my goals. To Carol, thank you for being my rock and my motivation. This journey would have been incomplete without your enduring love and support. Thank you from the bottom of my heart. To my friends, thank you for offering a listening ear and being a source of inspiration when the going got tough. Your presence made the journey less lonely and more meaningful.

Moving to a different country can be challenging. Thankfully, during my time at the University of Pennsylvania, I had an exceptional group of people that offered me support, guidance, and friendship. Your tireless efforts, dedication, and expertise have helped us to achieve our goals and reach new heights in the field of robotics. I am grateful for the countless hours we all spent making this work possible. Above all, I will cherish the memories of the laughter, camaraderie, and fun that we shared with you during our time together. Your friendship and support will always be remembered fondly. Thank you again for the incredible journey we have had together.

To my advisor Roseli, thank you for your guidance, encouragement, and support in shaping this work. Your mentorship has been invaluable in helping me grow and develop as a researcher.

Finally, I would like to thank the São Paulo Research Foundation (FAPESP) that supported this work via grants #2017/17444-0 and #2018/24526-5.

RESUMO

NARDARI, G. V. **Localização e Mapeamento Semânticos em Florestas**. 2023. 102 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Enquanto dados de sobrevoo podem fornecer informações gerais sobre uma floresta, no interior da mata é possível identificar plantas do sub-bosque, medir o diâmetro e contar os troncos de cada árvore. Atualmente, essas medições dependem de expedições humanas, que podem ser lentas, caras e até perigosas. Portanto, robôs capazes de navegar e extrair dados do interior da mata de forma autônoma têm o potencial de revolucionar a forma como as florestas são monitoradas em todo o mundo, aumentando a quantidade e qualidade das informações obtidas. No contexto de florestas, algoritmos clássicos desenvolvidos para ambientes urbanos podem falhar devido à falta de sinal confiável de GPS, terrenos irregulares, plantas e folhas que cobrem o terreno, além das árvores que balançam com o vento. Isso ocorre porque as suposições feitas pelos algoritmos clássicos podem não ser válidas nesse ambiente. No entanto, informações semânticas, como classes e formas de objetos esperados no ambiente são uma opção promissora para aumentar a robustez e o desempenho de sistemas autônomos. Nesta tese é apresentado um *framework* que utiliza informações semânticas derivadas de algoritmos de aprendizado de máquina dos dados de sensores carregados por um veículo aéreo não tripulado. O framework desenvolvido é capaz de identificar árvores e modelá-las como cilindros, criando um mapa semântico. A formulação adotada possibilita a incorporação de estimativas ruidosas que podem ser refinadas com a chegada de novas leituras dos sensores e de medidas externas para aumentar a robustez do sistema. A partir do mapa semântico gerado, é proposto um algoritmo capaz de gerar descritores únicos de locais em florestas que visualmente são extremamente similares. Tais descritores permitem o reconhecimeno de locais já visitados, e podem ser utilizados pelo *framework* para reduzir o erro acumulado nas estimativas de localização. Os resultados obtidos em experimentos em ambientes simulados e em florestas de Pinus do mundo real, demonstram que os métodos desenvolvidos geram mapas semânticos que melhoram a qualidade das estimativas de localização do robô e geram mapas informativos. Ademais, a representação semântica dos dados obtidos pelos sensores é mais eficiente computacionalmente, pois resume os dados brutos em um modelo geométrico semântico com poucos parâmetros.

Palavras-chave: Localização, Mapeamento, Segmentação Semântica, Aprendizado de Máquina.

ABSTRACT

NARDARI, G. V. **Semantic Localization and Mapping in Forests**. 2023. 102 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

While overhead data can provide general information about a forest, inside the forest, we can identify understory plants and measure the diameter and count of the trunks of each tree. Currently, specialists rely on human expeditions to get these measurements, which can be slow, expensive, and dangerous. For this reason, robots that can autonomously navigate and extract data from inside the forest could revolutionize how we monitor forests worldwide and the amount of information we have about them. In forestry, the lack of reliable GPS signal, uneven terrain covered by plants and leaves, and trees with branches moving with the wind are a few of the challenges posed. These factors can create shortcomings for classic algorithms as some assumptions may not be valid in this environment. Semantic information, such as classes and forms of objects expected in the environment is a promising way to increase the robustness and performance of autonomous systems. In this context, this thesis introduces a framework that uses 3D data provided by LiDAR or stereo cameras to identify semantic information using neural networks. This information is used to identify trees and model them as cylinders, creating a semantic map. Our formulation allows the incorporation of noisy estimates that can be refined with the arrival of new sensor readings and external measurements to increase the framework's robustness. Using the semantic map generated by our framework, we propose an algorithm capable of generating unique forest location descriptors that are visually highly similar. These descriptors can be used to identify previously visited locations and feedback to reduce accumulated errors in location estimates. We present several experiments in simulated environments and real-world Pine forests, demonstrating that our method generates semantic maps that improve the quality of the robot's location estimates and generate informative maps with information on the individual count and the trunk diameter of each tree. Furthermore, the semantic representation of the data obtained by the sensors is much more computationally efficient, as it summarizes the raw data in a semantic geometric model with few parameters.

Keywords: Localization, Mapping, Semantic Segmentation, Machine Learning.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

t — Time step

k — Keyframe, a subset of the time steps

$\mathcal{X}$ — Set of robot poses in the SLAM Problem

x_t — Unknown variable at time step t

$\mathcal{O}$ — Set of sensor observations

o_t — Observation at time step t

$\mathcal{L}_t$ — Set of landmark models at time step t

l^i — A single landmark model.

D_π — Point to plane distance

D_c — Point to cylinder distance

$\mathcal{M}$ — A set of accumulated landmarks (semantic map)

$\mathcal{S}_t$ — A subset of $\mathcal{M}$ at time step t

p_i — A single feature point

$\mathbf{G}$ — A set of feature points labeled as ground

$\mathbf{T}$ and $\mathbf{H}$ — SE3 transformation

$\mathcal{G}$ — A graph representation of a set of points

$\mathcal{V}$ — Vertices of a graph

$\mathcal{E}$ — Edges of a graph

DT — Delaunay Triangulation

$\mathcal{H}$ — A set of geometric primitives derived from a tessellation

I_i — An image from camera i

d — Disparity map between an image pair

S — Semantic segmentation mask with a label per pixel

CONTENTS

CHAPTER 1

INTRODUCTION

Forests are of paramount importance to our society. They directly influence Earth's climate, house most of the planet's biodiversity, and enable multiple economic activities. For this reason, researchers, government, and industry spend enormous efforts to understand these environments and estimate metrics such as timber inventory, fuel volume, health, impacts of deforestation, and biodiversity.

To this end, satellite images are an essential tool, especially in fighting deforestation and continuously monitoring large areas. From this data, we can detect illegal activities and metrics about the health of the forest (BOYD; DANSON, 2005). However, the resolution of the images provided by satellite can be meters per pixel. For this reason, overhead flights with UAVs are becoming a standard approach to complement satellite data with more detail. UAVs can carry different sensors such as LiDAR, multispectral cameras, and ultrasound. From these sensors, it is possible to estimate metrics such as canopy height and leaf area and recognize species (ALMEIDA *et al.*, 2019).

While this technology dramatically improves what is possible to measure in a forest, we still depend primarily on human expeditions to extract data about their interior. For example, estimating the exact number of trees from over-canopy data is challenging since the tree crowns can be very close and blended. It is also challenging to estimate the diameter of tree trunks from overhead since the canopy blocks most sensor readings. Moreover, many forests will have understory species, smaller trees that are not visible over the canopy because more prominent individuals block them (MIRANDA *et al.*, 2021). While Terrestrial Laser Scanning (TLS) has facilitated interior mapping (CONTO *et al.*, 2017), the setup required for this technology still slows down the process for large areas.

For this reason, mobile robots can be an essential tool for frequently obtaining rich information about an area's interior. The robots can carry sensors such as LiDAR, cameras, and microphones and capture mosquitoes, leaves, and water samples. This data can be analyzed automatically or by specialists to gather information about forests at a speed and quality never seen before. The robots could also execute predefined missions without requiring humans to enter the area of interest, which can be dangerous due to the terrain and animals present in the forest.

In practice, all these technologies should be used together since, even with robots, covering the interior of the forest will take more time than the alternatives. We envision a system where satellite, overhead UAVs, and under-canopy robots work together to create a digital twin of a forest that is continuously updated at different frequencies and resolutions.

While robots are starting to become part of our everyday lives, most works consider urban or indoor settings. However, some valid assumptions in these environments do not hold in a forest. For example, planar features and edges are not as reliable when twigs and leaves cover the ground and branches constantly move with the wind. Moreover, forest can cause perceptual aliasing, that is, different places that generate a similar footprint resulting in wrong data associations. In this work, we propose algorithms designed with forests in mind. We argue that semantic information can provide more reliable features for these algorithms and create more meaningful maps that help us understand the forests.

The main objective of this thesis is to address this gap by developing a robot and algorithms that can be deployed to gather data about the interior of the forests as a semantic map. Semantic maps associate metric data to object-level representations of the elements present in an environment, which can be more lightweight, robust and informative that other representations. To this end, we propose a semantic localization and mapping framework that can incorporate different sensor measurements and approximates the trunk shapes with cylinder models that are constantly updated as new measurements arrive. Like any iterative state estimation approach, our method will accumulate errors and drift over time, even with more reliable features. For this reason, we propose a novel framework that computes unique descriptors for different forest regions using the detected trees. These detections can be used to incorporate loop closure constraints that help mitigate the drift in forests.

It is necessary to mention that rainforests like the Amazon concentrate a large portion of the world's biodiversity and must be protected to avoid the advancement of climate change. However, these forests are very dense and challenging to access and traverse with a mobile robot, especially an UAV. In this project, we propose methods motivated by such environments, but our development and experiments are performed in more homogeneous environments such as Pine forests. These forests are also important for carbon storage and have a significant commercial interest. In practice, we believe that most of the limitations and assumptions our algorithms make for Pine are also valid in any forest. We believe this research is a first step towards enabling mobile robots in rainforests.

1.1 Contributions

- SLOAM, a framework for LiDAR-based odometry and mapping in forests that extracts semantic information from the sensor data to identify and model individual trees, creating a lightweight, informative map that also improves the robot state estimation. We also demonstrate how SLOAM can be integrated into an autonomous UAV system to perform large-scale missions under the forest canopy.
- A novel method for place recognition in forests that handles perceptual aliasing. Our method uses the position of trees derived from the semantic map to create geometric shapes that uniquely represent the different areas of the forest.
- A theoretical formulation based on a factor graph for SLOAM, capable of incorporating different sensors, measuring and updating tree models, and incorporating loop closure constraints. This formulation ties together our first two contributions, creating a more robust framework for state estimation in forests.
- A neural network architecture that mixes supervised and self-supervised learning to jointly estimate a semantic segmentation mask and depth. The proposed model can output semantic Pseudo-LiDAR measurements that can be directly used by SLOAM as a cheap and lightweight replacement for the LiDAR.

1.2 Thesis Outline

This thesis is structured as follows: Chapter 2 presents the current state of the art on the localization and mapping problem and the place recognition task. Chapter 3 introduces the main theoretical concepts used in our work, such as sensing modalities, map representations, SLAM, and machine learning for object detection. Chapter 4 presents a formal definition of the semantic SLAM problem, introduces SLOAM and its factor graph formulation, and shows different results in simulation and in real-world experiments. In Chapter 5, we present a novel method for place recognition in forests using only the position of trees, results in real-world and simulated data, and its integration in the SLOAM factor graph formulation. In Chapter 6, we present the neural network architecture for semantic pseudo-LiDAR from stereo. Finally, Chapter 7 wraps up the thesis with a discussion on the results and limitations of our work and possible future directions. Published papers and scientific outreach interviews are listed in the Appendix section.

CHAPTER

2

RELATED WORK

For an autonomous robot to efficiently navigate an unknown environment, it must observe its surroundings using sensors to estimate ego-motion and create a representation of the space. The problem of simultaneously creating a map to represent the space through sequential sensor measurements while estimating the robot's position with respect to this map is known as SLAM.

2.1 Localization and Mapping

SLAM has been widely studied in the literature (CADENA *et al.*, 2016). Despite the field's maturity, there are still many open challenges in SLAM. As new sensors appear and embedded computers become more powerful, new possibilities emerge for research. In Table 1, we present an overview of the recent SLAM works related to this thesis.

Many works rely on cameras to perform SLAM due to their affordable price, availability, and weight. ORB-SLAM (MUR-ARTAL; MONTIEL; TARDOS, 2015) utilizes ORB feature points to perform data association between measurements. The system uses a graph-based back-end for pose optimization and mapping. Moreover, reobserved locations are detected via a bag-of-features approach and integrated as loop-closure constraints to reduce drift. ORB-SLAM is robust and optimized for real-time robotics applications, making it a popular approach for many other robotics works that need a SLAM solution. Since visual odometry and SLAM are popular topics, many other libraries such as VINS-Mono (QIN; LI; SHEN, 2018) have been proposed in the literature, and we point the reader to (ROSINOL *et al.*, 2020) where the authors review open-source libraries for visual-based SLAM methods.

LiDARs, on the other hand, are newer and more expensive sensors but capture large amounts of information due to their measurement range. LiDAR Odometry and Mapping (LOAM) (ZHANG; SINGH, 2014) was a seminal work in LiDAR SLAM. The authors developed a custom spinning mechanism using a 2D LiDAR and a motor for rotational motion with an encoder that measures the rotation angle. With this device, they could capture 3D data using a simpler sensor. To perform data association, LOAM relies on geometric features, such as points on corners and planar surfaces extracted from the sensor point cloud. The original LOAM formulation relies on direct nonlinear optimization of point-to-point and point-to-plane distances. This formulation is effective but does not incorporate other sensor measurements or loop closure detections to correct the inevitable accumulated drift of the pose estimation. Lio-SAM (SHAN *et al.*, 2020) proposed an extension of LOAM with a Smoothing and Mapping (SaM) framework, supporting IMU preintegration, GNSS, loop closure while running real-time on an edge device. Similar to the original work, Lio-SAM relies on geometric features to perform data association, which works well in most structured and urban settings but does not exploit the environment's semantics and can suffer in outdoor environments such as a forest.

The assumption made by LOAM and Lio-SAM that geometric features will provide reliable points for data association is reasonable in most structured environments such as indoor and urban settings. However, if more knowledge about the environment can be provided to the SLAM algorithm, we can add more robustness in state estimation and, consequently, better maps. For example, by considering that a ground plane will be visible by the robot at every sensor observation, LeGO-LOAM (SHAN; ENGLOT, 2018) demonstrates how a simple assumption that is reasonable in many applications can increase robustness. Ground plane detection enables LeGO-LOAM to selectively sample features on the ground or other parts of the observation and constrains the pose estimation optimization more effectively.

Most of the methods in the literature create metric maps that represent the environment as occupied or free space, which is helpful for planning paths and obstacle avoidance but not an optimal representation. For instance, when someone asks for directions, humans will not give instructions in meters, e.g., walk 100m, turn left, walk 20m and turn right. Instead, we remember reference points or landmarks and high-level structures that help us navigate, e.g., "walk straight until you see the supermarket and turn right." Locally, we still need to consider metric information to plan our next steps (are there obstacles on the sidewalk that I have to avoid?), but it is not feasible to maintain this information for long-term navigation.

A robot can benefit from semantic information on many levels. The growth of learning-based methods for machine perception, especially in computer vision (VOULODIMOS *et al.*, 2018), and specialized hardware to run such algorithms on embedded systems (CHEN *et al.*, 2020) enabled robots to consider semantic information to reason about the environment in more complex and meaningful ways. Semantic maps are more compact than their metric counterparts since objects' shapes can be approximated by geometric primitives such as cylinders, cuboids, or quadrics instead of maintaining all the feature points that are part of the object (CHEN *et al.*, 2020b; YANG; SCHERER, 2019; NICHOLSON; MILFORD; SÜNDERHAUF, 2018). This property is essential for large-scale operations where memory space is limited (LIU *et al.*, 2022a). For self-driving cars, semantic information is indispensable for planning and control since the autonomous driving system needs to recognize other cars, pedestrians, cyclists, and other structures to make safe decisions, such as stopping at a cross-walk or switching lanes (SCHWARTING; ALONSO-MORA; RUS, 2018).

Focused on self-driving vehicles, SuMA++ (CHEN *et al.*, 2019) utilizes semantic labels to reject features from moving objects and constrain data association. For example, points from different observations should be associated if they are from the same type of object, and this class of object is not expected to be moving, increasing the quality of the features and the robustness of LiDAR state estimation. The authors use a voxel map to represent the environment and enrich each cell with semantic labels. A flood-fill algorithm refines the semantic labels of the voxels that may be inconsistent due to segmentation mistakes. Similar to classic approaches, this method still relies purely on feature points, and despite incorporating semantic labels into the map, it still uses a dense representation and does not identify or model individual objects.

Most of the modern machine learning approaches for perception come from the computer vision literature. For this reason, most of the semantic segmentation and instance detection methods are designed for cameras. Kimera is a complete SLAM system that combines metric and semantic information to create mesh maps from images and depth maps enriched with semantic labels (ROSINOL *et al.*, 2020). Kimera is composed of 4 main modules. VIO for state estimation, robust pose graph optimization, a mesh builder based on feature triangulation and a semantic module operates on 2D images but applied to the 3D meshes.

Since bounding boxes are a common output for these methods, different works have proposed the use of 2D bounding boxes in images to locate and model objects in 3D space using a robot. CubeSLAM (YANG; SCHERER, 2019) fits 3D Cuboids from a single image bounding box detection. These cuboid measurements are integrated into a SaM framework that combines ORB features with object models in 2D and 3D to optimize the robot pose and the cuboids jointly. Similarly, QuadricSLAM (NICHOLSON; MILFORD; SÜNDERHAUF, 2018) fits 3D quadrics from 2D bounding box detections on RGB images. Using a SaM approach, they can continuously update the quadric models as new measurements arrive. These methods are general since they can approximate the shape of an object from bounding boxes. In some applications like forest inventory, where we are interested in detecting objects and measuring a property such as the diameter, more knowledge about the application can be incorporated to obtain better estimates.

One of the contributions of this thesis is SLOAM, a framework that performs SLAM while detecting and modeling trees in a forest using LiDAR observations.

2.2 Place Recognition

The first and most challenging part of loop closure is place recognition, that is, recognizing that the robot has returned to a known location. SLAM algorithms often rely on point-to-point or point-to-geometric shape distance to perform data association. Geometric or intensity-based features such as ORB and SIFT (LOWE, 2004; RUBLEE *et al.*, 2011) can vary significantly with illumination changes, weather conditions, and dynamic environments that constantly change.

In most cases, feature associations are never reconsidered even as new measurements that could reduce ambiguity arrive. Wrong associations may add noise to the pose estimation step, reducing the performance of the SLAM algorithm. The probabilistic framework proposed in (BOWMAN *et al.*, 2017) addresses these limitations by solving data association as an expectation maximization (EM) problem of the measurement likelihoods. In other words, instead of taking the most likely association, their method maintains a probability distribution that is updated iteratively with new measurements. Their framework also considers semantic object detections and updates semantic labels of objects observed by the robot. The main drawback of this approach is the computational load required to perform these updates for every new observation, making this method hard to scale. For this reason, most works opt to make hard assignments but try to improve data association robustness by using better features.

Recent works have explored learnable or object-based descriptors to increase robustness, especially for global localization methods requiring these features to be consistent, even if the viewpoints differ from the last time the robot visited the same place. In Table 2, we present an overview of global localization methods related to this thesis.

The Local Semantic Tensor (LoST) descriptor computes semantically-consistent keypoints extracted from a convolutional neural network's hidden layers that can be used for keypoint matching (GARG; SUENDERHAUF; MILFORD, 2018). The authors show that this approach performs well in a global localization task in urban settings, even associating opposite viewpoints of the same scene. In (CHEBROLU *et al.*, 2019), the authors utilize semantic segmentation to select robust and unique points that help a ground robot localize using a database of images taken from a UAV. The proposed system is intended for outdoor crop fields, showing that these features are robust even after multiple sessions spanning several weeks.

In (MILLER *et al.*, 2021), the authors perform semantic segmentation in both LiDAR and RGB cameras captured by a ground vehicle and combine both sensor measure-

Table 1 – Overview of related SLAM works.

Title	Sensor	Semantic	Feature Type	Optimization back-end	Object Modelling	Loop Closure	Contribution
Ours	**LiDAR** **Stereo**	**Yes**	**Semantic**	**SaM**	**Yes**	**Yes**	Semantic SLAM framework based on a Factor Graph formulation
(ZHANG; SINGH, 2014)	LiDAR	No	Geometric	Nonlinear optimization	No	No	SLAM framework for LiDAR based on geometric features
(SHAN; ENGLOT, 2018)	LiDAR	No	Geometric	SaM	No	Yes	State estimation aided by ground plane detection
(SHAN *et al.*, 2020)	LiDAR	No	Geometric	SaM	No	Yes	SAM framework with IMU, GPS and Loop Closure integration
(MUR-ARTAL; MONTIEL; TARDOS, 2015)	Mono Stereo RGBD	No	Geometric	Pose Graph	No	Yes	Real-time SLAM system for cameras
(BOWMAN *et al.*, 2017)	Mono	Yes	Semantic	SaM	No	Implicit	Semantic SLAM with data association uncertanty
(CHEN *et al.*, 2019)	LiDAR	Yes	Semantic	SaM	No	No	Data association with semantic constraints
(NICHOLSON; MILFORD; SÜNDERHAUF, 2018)	Mono	Yes	Geometric	SaM	Yes	Implicit	3D quadrics from 2D bounding box detections
(YANG; SCHERER, 2019)	Mono	Yes	Geometric	SaM	Yes	No	3D cuboids from 2D bounding box detections
(ROSINOL *et al.*, 2020)	Mono Stereo RGBD	Yes	Geometric	SaM	No	Yes	3D meshes enriched with semantic labels
(QIAN *et al.*, 2021)	Mono Stereo	Yes	Geometric	Pose Graph	No	Yes	Object-level data association

ments and semantic labels to create a semantic scan. Assuming that the objects observed by the ground vehicle can also be observed by satellite imagery, they can perform global localization without GNSS. Other works combine semantic and topological information to compute robust descriptors. X-view (GAWEL *et al.*, 2018) and (LIN *et al.*, 2021) perform segmentation on images and creates a dense graph based on the centroid of each object. Both works propose a descriptor based on random walks on the graph to represent the object distribution in space. They demonstrate that it can represent environments uniquely enough for the global localization task even if the viewpoint is extremely different, as long as enough object overlap is available.

While most works are concerned with urban settings that contain a wide variety of semantic information, forests are especially challenging due to perceptual aliasing, the high similarity between different parts of the environment. For example, if the only semantic objects available are trees, the nodes of the graph-based methods would all be the same. For this reason, methods that focus on the topology are better suited to robustly detect previously seen locations in these environments.

GLARE (HIMSTEDT *et al.*, 2014; KALLASI; RIZZINI, 2016) encapsulate geometric relationships based on the neighborhood of keypoints or landmarks. The neighborhood of each element is defined by a user-defined threshold on the euclidean to other points. The GLARE descriptor of an element descriptor is given by a 2D histogram of each neighbor's distances and relative angles. Finally, a global descriptor can be estimated by averaging the local descriptors of each keypoint.

The work parallel to this thesis of Li *et al.* (LI *et al.*, 2020) detects trees in a sparse forest in accumulated point clouds captured by a LiDAR using a clustering algorithm. The authors represent each tree as a 2D point in space, from which a Delaunay triangulation can be computed. The triangles define local descriptors for the trees that can be used for global localization. Our method, presented in Chapter 5 also leverages the Delaunay triangulation, but proposes a composition of triangles to encode local regions of the observation that proves to be more robust on noisy estimates.

2.3 Final Considerations

In this chapter, we introduced the pivotal works of the SLAM literature. We showed how this field is improving by incorporating machine learning algorithms to increase performance and robustness in different environments. Moreover, we presented related works on the place recognition problem, especially for environments with high perceptual aliasing. In this thesis, we present methods that leverage semantic information. We show that semantic features and models lead to better results in forests, where the usual assumptions made in urban environments do not hold.

Table 2 – Overview of related works in global localization.

Title	Sensor/Input	Semantic	Topology	Method Summary
Ours	**Semantic Map**	**Yes**	**Yes**	Polygon descriptors derived from the position of semantic landmarks
(KALLASI; RIZZINI, 2016)	LiDAR	No	Yes	Descriptors based on relative distances and angles between keypoints
(GAWEL *et al.*, 2018)	Mono	Yes	Yes	Semantic graph-based descriptor for localization under view-point changes
(GARG; SUENDERHAUF; MILFORD, 2018)	Mono	Yes	Yes	Derives semantically-consistent keypoints that are robust to viewpoint variation
(CHEBROLU *et al.*, 2019)	Mono	Yes	No	Ground robot localization based on aerial images using semantic features
(MILLER *et al.*, 2021)	LiDAR	Yes	No	Localization on satellite imagery using semantic segmentation of RGB and LiDAR
(LI *et al.*, 2020)	LiDAR	Yes	Yes	Triangle-based descriptors from trees in a forest
(ZHANG; LI; MA, 2021)	Mono	Yes	Yes	Semantic keypoints and local topological features
(LIN *et al.*, 2021)	RGB-D	Yes	Yes	Semantic graph-based descriptor; Edit distance for matching

CHAPTER

3

BACKGROUND

This work aims to generate an efficient representation of a forest for large-scale robotics operations while also being useful for downstream tasks such as forest inventory, wildfire prevention, and preservation. We propose methods motivated by the challenges and necessities of deploying a UAV flying under the forest canopy to solve these tasks. However, with the scale and complexity of these environments, this task may be split into multiple agents such as humans, UAVs, and ground robots, which, individually or in collaboration, can all benefit from our methods.

These agents must carry different sensors, combine and reason about their data to make decisions and estimates. In this chapter, we will introduce the theoretical concepts and sensors that the methods proposed in this thesis and the UAV system used in our experiments utilize to enable the semantic localization and mapping task.

3.1 Sensors for 3D Perception

LiDAR

Light Detection And Ranging (LiDAR) is a class of sensors that emit near-infrared light (700nm-2000nm wavelength) and estimate the time the reflection of the emitted signal takes to return to the sensor. In other words, a LiDAR is an active time-of-flight sensor. Depending on the frequency of the light the sensor emits, it may be more or less subject to the effect of the sunlight, making it suitable for indoor or outdoor applications.

We refer to 2D LiDAR as the sensors that return a single planar set of readings of the environment, which is called a beam. A popular example is the Hokuyo-UTM [1]. 3D LiDARs, on the other hand, contain multiple layered beams, capturing much more information about the surroundings. We illustrate in Figure 1 a 2D and a 3D LiDAR.

[1] <https://www.hokuyo-aut.jp/>

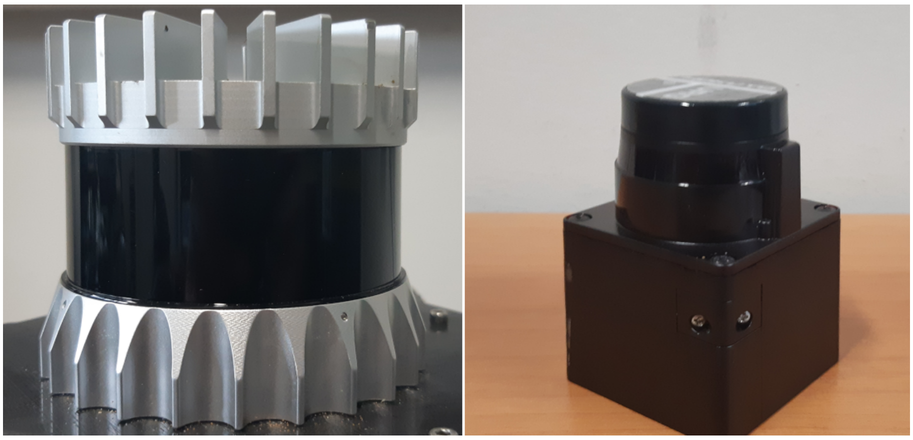


Figure 1 – Ouster OS1 3D LiDAR (left) and a Hokuyo URG-04LX-UG01 2D LiDAR (right).

These sensors have seen increasing interest from the robotics community, especially for self-driving car research and industry (RORIZ; CABRAL; GOMES, 2021).

Most LiDARs such as the Ouster [2] or Velodyne [3] utilize a spinning mirror mechanism to emit light in different directions. However, having a moving part inside the sensor is not desirable. This mechanism is subject to mechanical failure and creates vibration, affecting the quality of other sensors' measurements and even the dynamics of the robot, especially UAVs. A new class of sensors called solid-state LiDARs is emerging to address these problems. These sensors do not have any moving parts. Instead, they use mirrors and lenses to emit the same light source in different directions. For this reason, these sensors usually have a smaller field of view but solve the mentioned limitations of the spinning LiDARs.

Structured light sensors, such as the Microsoft Kinect™, can also be considered a solid-state LiDAR. These sensors flash a pattern of infrared light into space and measure the displacement and distortion of the pattern when it hits objects to compute the depth of that region. The main disadvantages of this technology are the limited sensing range, the field of view, and the sensitivity to sunlight depending on the frequency of light the sensor emits.

LiDARs are a powerful class of sensors since they can capture a large amount of information per observation with ranges up to kilometers. Their size, weight, and price can make this technology prohibitive in some applications that require small robots or risky operations where the robot may crash frequently. However, as the market for this technology grows, they are expected to be better and more accessible in the near future. Another limitation of LiDARs is that they do not provide color or texture information, limiting algorithms to the shape of features and objects. For this reason, some works combine this technology with cameras to compute a colored point cloud (VECHERSKY *et*

[2] <https://ouster.com/>
[3] <http://velodyne.com/>

al., 2018). Nevertheless, payload, computational burden, and cost constrain the use of both sensors simultaneously in some applications. For this reason, much of the literature focuses on stereo cameras as an alternative source of close-range depth while simultaneously capturing rich texture and color information.

Inertial Measurement Unit

An Inertial Measurement Unit (IMU) combines gyroscopes, accelerometers, and other optional sensors that can estimate the robot's pose with respect to an inertial frame. The gyroscope can measure angular velocity in the inertial frame of the sensor. The accelerometer measures the rate of velocity change generated by external forces, which can be used to estimate the acceleration of the IMU in one direction. A magnetometer or compass can measure the intensity and direction of the magnetic fields around the IMU, providing an absolute reference for yaw measurements. Typically, IMUs will contain three sensors of each type to provide measurements in three rotation axes (roll, pitch, and yaw).

Instead of operating on raw measurements, most applications of IMUs utilize it as part of an Attitude and Heading Reference System (AHRS), where the raw sensor measurements are fused to provide high-frequency (100Hz+) attitude data. However, these measurements are subject to external noise, such as vibration from spinning LiDAR, UAV rotors, or magnetic interference. Moreover, the integration of the raw sensor measurements will quickly accumulate drift over time.

A standard solution to address this problem is using GNSS to provide an absolute reference for the 3D position of the system. Since GNSS provides independent measurements, it can minimize the accumulated drift of the IMU estimates. However, GNSS can only provide reliable estimates in open spaces. Even with partially occluded environments such as under tree canopy, the GNSS measurements can become unreliable (ZHENG; WANG; NIHAN, 2005; CARREIRAS; MELO; VASCONCELOS, 2013).

Stereo Cameras

A stereo camera is defined by two sensors with overlapping fields of view, separated by a known fixed distance (stereo baseline). Given a pair of rectified images from these sensors, one can compute a disparity map from one image to the other. That is, for every pixel in image A, the number of pixels that we have to move across the epipolar line in image B to find the corresponding pixel. See (HARTLEY; ZISSERMAN, 2003) for more details on the theory of multiple view geometry.

The ground truth distance of the objects in the scene is a function of the disparity and can be computed if the camera's intrinsic and extrinsic parameters are known. This powerful framework provides dense depth estimates without needing expensive or

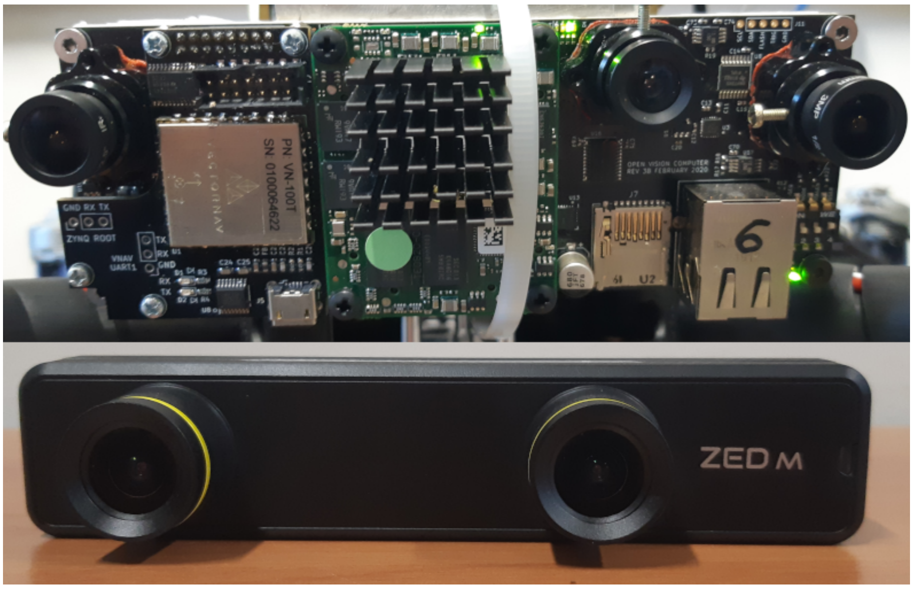


Figure 2 – The **Open Vision Computer (top)**, is an open-source sensor that contains stereo gray-scale global shutter sensors with 120mm baseline, an RGB rolling shutter sensor (center), and a Vectornav VN100 IMU. **ZED Mini (bottom)** a commercial solution with stereo RGB rolling shutter sensors separated by a 63mm baseline and an IMU. For both alternatives, most of the processing has to be done onboard a host computer such as an Intel NUC or an NVIDIA Jetson.

complex sensors. For this reason, many works use stereo cameras for 3D object reconstruction (ACKERMANN; GOESELE, 2015), state estimation (SUN *et al.*, 2018), and object detection (PON *et al.*, 2020).

The minimum and maximum depth that stereo cameras can estimate are defined primarily by the camera's baseline and field of view. If the baseline is small, the camera can estimate the depth of objects close to the sensor but will quickly drop the quality for objects further away. Conversely, the overlap will be small if the objects are too close to the camera and the baseline is large. State-of-the-art commercial stereo cameras such as the Stereolabs ZED 2 report a depth range of 0.2m to 20m with a 120mm baseline, while the Stereolabs ZED Mini has a depth range of 0.1m to 15m with a 63mm baseline. In practice, we observe that this range may have to be reduced for mapping applications to avoid adding noise to the system. Both sensors are shown in Figure 2. In Figure 3 we illustrate a single observation at the same location provided by a Hokuyo URG-04LX-UG01 2D LiDAR, an Ouster OS1-16 3D LiDAR sensor and a ZED Mini$^{\text{TM}}$ stereo camera.

Since most disparity estimation algorithms rely on matching blocks of pixels based on feature similarity, repetitive patterns and textureless regions can cause wrong associations and, consequently, wrong disparity estimates. Recent works propose using learning-based disparity estimation algorithms that learn feature descriptors that leverage the scene context to be more robust to these scenarios (XU; ZHANG, 2020). This approach has become a standard since convolutional neural networks can be highly optimized for

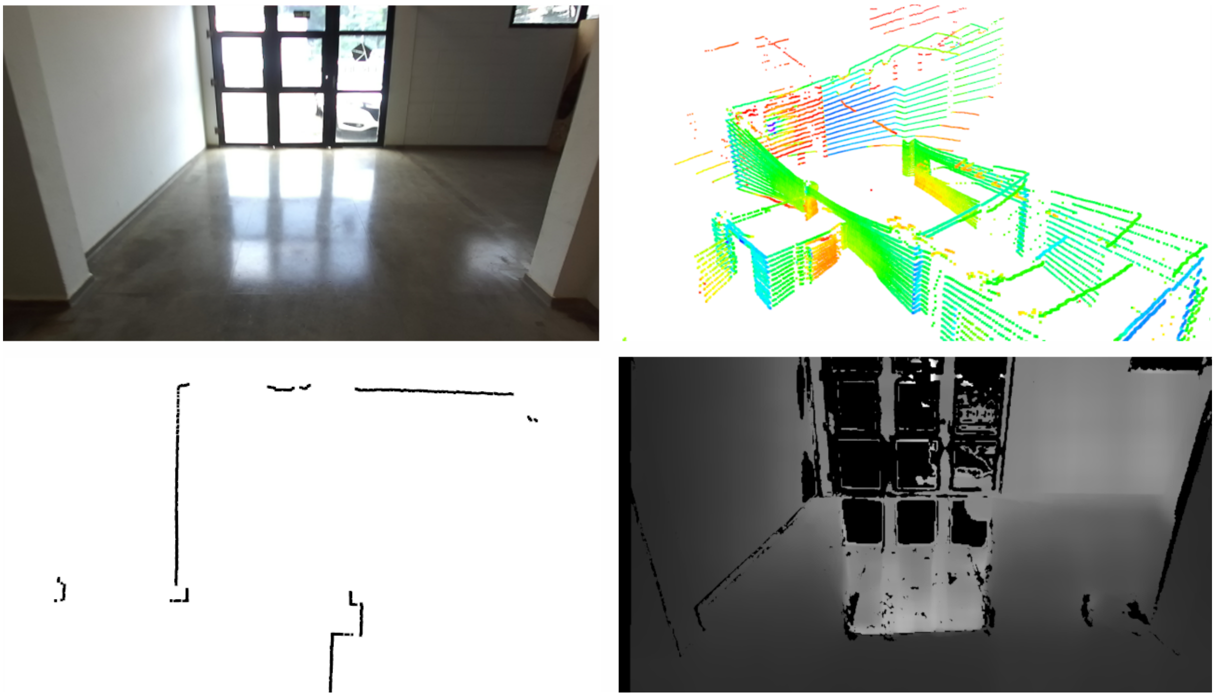


Figure 3 – Different sensor observations at the same location. The top left panel shows an RGB image. The top right panel shows an Ouster OS1-16 point cloud, where the colors represent the intensity of the light return. The bottom left corner shows a Hokuyo 04LX-UG01 laser scan. The bottom right panel shows a depth map computed by a stereo camera. The glass door shows an important limitation of these sensors. Since the light emitted by the laser sensors will pass through the glass, it will appear as an empty space. Moreover, the sunlight coming from outside creates degenerate cases for the stereo depth estimation algorithm, causing issues with the estimates on the glass door and the reflection on the ground.

Graphics Processing Units (GPUs) and the same network weights can be shared for other relevant applications, as discussed in Chapter 6.

3.2 Map Representations

Most robotic systems rely on some environment representation to localize and plan paths for robot navigation. Occupancy grids are a simple and popular approach. The space is represented by a 2D matrix, where each grid cell can assume three different values (occupied, open, and unknown) and has a fixed resolution, e.g., 0.05m per cell. Some robots and applications require a 3D representation of the environment, and this concept of an occupancy grid, then expanded to three dimensions, is called a voxel grid. The memory requirements of these methods increase with the size of the area that the robot can operate and the resolution, which can be prohibitive in large-scale applications.

Another popular approach, especially in the LiDAR SLAM literature, is to store the map directly as a point cloud. While this representation facilitates matching new sensor observations into the existing map for estimation, this representation can be extremely

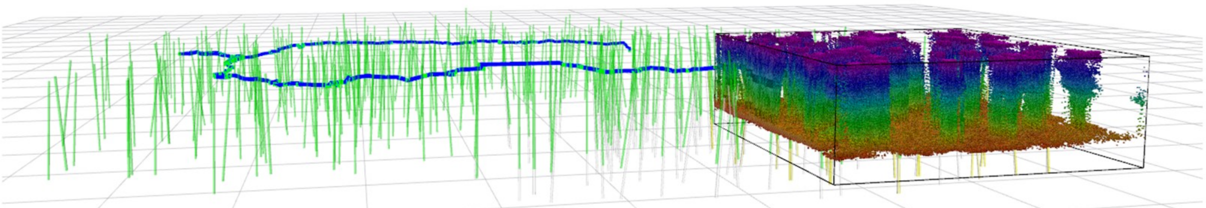


Figure 4 – **Voxel map vs Semantic Map**. The voxel map contains dense metric information that an autonomous system can use for planning. However, a semantic representation is more suitable for large-scale maps since they can encapsulate the relevant information with few parameters that describe the object models. This representation can alleviate the memory requirements and enables more informative planning strategies, such as planning trajectories to reduce the uncertainty an object model or predicting the path of dynamic obstacles such as pedestrians. Figure adapted from (LIU *et al.*, 2022a).

memory intensive in large-scale applications without some optimization (ZHANG; SINGH, 2014; SHAN *et al.*, 2020; XU; ZHANG, 2021).

Instead of operating solely on metric information of the space the robot observed, a more natural approach is to store high-level information about the environment (i.e., semantic information) and have local metric maps for planning (LIU *et al.*, 2022a). This approach is illustrated in Figure 4. Semantic maps enable algorithms to interact with the environment and make decisions in more complex settings. For example, instead of "there is a 180x30cm obstacle in front of me", a semantic map can provide "a person is crossing the street on the crosswalk." or "this set of points belong to a moving car, so it is not reliable as a reference for ego-motion estimation." Moreover, semantic maps aggregate metric information in more compact representations, such as a model of an object instead of a set of 3D points, which is more memory efficient and more suitable for large-scale applications. This thesis proposes algorithms that can compute such semantic maps in real-time with different sensor modalities, creating informative maps that can disambiguate measurements even under large perceptual aliasing and create storage-efficient maps for large-scale operation.

3.3 Data Segmentation

Given a sensor observation (e.g., an image) the goal of the semantic segmentation task is to assign each individual measurement (i.e., a pixel) a semantic label for objects and other uncountable elements such as the sky or the ground. With similar inputs, the goal of the instance segmentation task is to identify individual objects in the observation (individual cars, persons, trees), creating masks or bounding boxes for each object. The union of both tasks is called Panoptic Segmentation (KIRILLOV *et al.*, 2019). In other words, semantic segmentation adds labels to individual pixels, instance segmentation labels, or group pixels that belong to different objects. In contrast, panoptic segmentation will

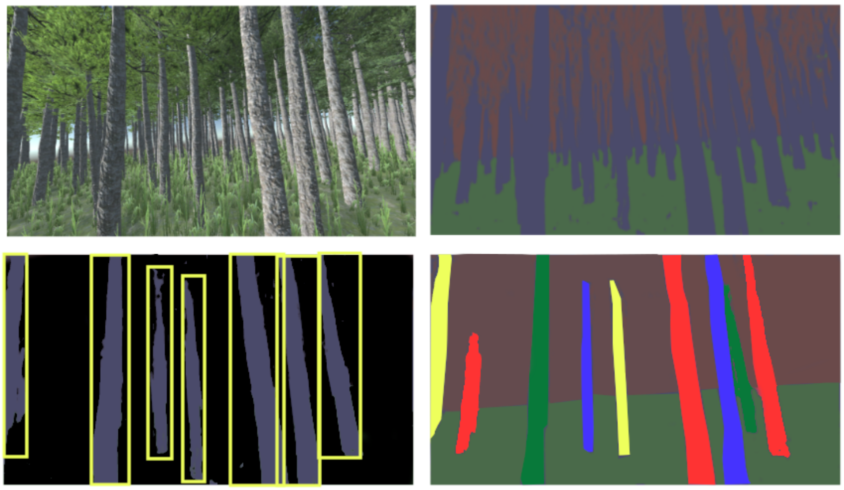


Figure 5 – The different segmentation tasks. Given an input such as an RGB image (top left panel), the semantic segmentation task (top right panel) will assign a semantic label for each pixel. In the instance segmentation task (bottom left panel), the objective is to detect every individual countable object). Finally, in the panoptic segmentation task (bottom right panel), we are interested in individual countable objects and uncountable elements such as ground and sky.

add a semantic label to each pixel and an instance id. When the element is uncountable such as the sky, the instance id is *null*. This definition can be generalized to other types of sensors, such as 3D LiDARs, labeling or grouping 3D points instead of pixels. We illustrate the output of each of the different segmentation tasks in Figure 5.

In this thesis, we utilize a combination of semantic segmentation and heuristics to perform panoptic segmentation in LiDAR and image data.

3.4 Simultaneous Localization and Mapping (SLAM)

For a sequence of sensor observations $\mathcal{O} \triangleq \{o_t\}_{t=1}^{T}$, the SLAM problem consists of estimating the unknown variable $\mathcal{X} \triangleq \{x_t\}_{t=1}^{T}$ that represents the robot pose. In three dimensions, the robot pose can be represented by a rotation (roll, pitch, yaw) and a translation (x, y, z) relative to some reference coordinate system, also referred to as a reference frame. Moreover, the SLAM problem can incorporate other relevant measurements, such as landmarks $\mathcal{L} \triangleq \{l_i\}_{i=1}^{N}$. Landmarks are objects or points of interest that the robot can observe in the environment. They can be used to anchor the robot pose estimates while simultaneously having its properties, such as location and size, refined as new measurements of the same landmark are captured. The SLAM problem can be summarized by

$$\arg\max_{\mathcal{X},\mathcal{L}} p(\mathcal{X}, \mathcal{L} | \mathcal{O}).$$

We refer to the algorithm that will solve this equation as the SLAM back-end. The main types of back-ends in the literature are filter-based approaches (AULINAS *et al.*, 2008) that solve only for the current estimate x_t and z_t or least-squares approaches that formulate SLAM as a maximum a posteriori estimation (MAP) problem. The latter is not only concerned with the current observation but also with reducing the uncertainty of past measurements (THRUN; MONTEMERLO, 2006; OLSON; LEONARD; TELLER, 2006; KAESS *et al.*, 2012).

Most state estimation algorithms assume that subsequent sensor observations will have enough overlap so that repeating patterns can be identified and used as a reference to estimate the relative motion between observations. Traditional SLAM approaches rely on features and models such as lines, planes, and edges to identify these patterns and perform data association. These patterns are widely used and are consistent enough if the environment is structured, such as an urban setting. However, in environments such as a highly unstructured forest, with moving leaves and the ground covered with underbrush that masks the actual shape of the ground, these features are not as reliable, adding a significant amount of noise to the state estimation algorithm. For this reason, we propose using semantic information to improve the robustness of data association in these environments.

Once data association is solved, the problem is reduced to finding the relative transformation that minimizes some distance between the associated features. With known poses, the sensor measurements can be accumulated in a common map frame.

The increased success of machine learning approaches for semantic segmentation and methods that can run in real time on edge devices enabled using more informative features to become viable for real-time SLAM.

A central approach in this thesis is representing the SLAM optimization problem as a factor graph (KAESS *et al.*, 2012). This representation models the unknowns (i.e., robot and landmark poses) as nodes of a graph that are connected by probabilistic knowledge about them (factors). In Figure 6, we illustrate a factor graph where subsequent poses are connected by the lines representing odometry measurement factors. The connection between poses x_2 and x_5 is a particular case where a loop closure happens—the optimization algorithm can use the loop constraint to minimize the accumulated drift. Moreover, we can incorporate semantic information to create constraints between poses and landmarks the robot observes during its trajectory. For example, the landmark l_1 was observed by $x_0, x_1, and x_4$. The multiple observations of the same landmark can help the algorithm refine the object's parameters while simultaneously using this to anchor the pose estimation.

In this thesis, we propose different sub-modules for localization and mapping that together compose our factor graph formulation. We show in chapter 4 that using semantic features and object models to improve the robustness of SLAM algorithms in

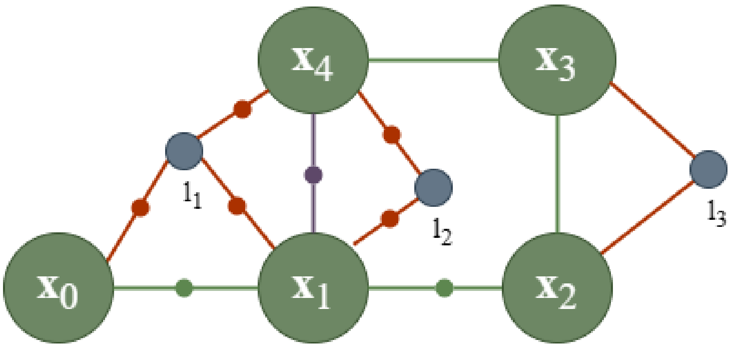


Figure 6 – Example of a Factor Graph with constraints between agent poses (green), poses and landmarks (red) and loop closure (purple).

such unstructured settings can improve estate estimation and the quality of the resulting maps. Moreover, we show in Chapter 5 that it is possible to derive geometric descriptors from semantic object models to reliably identify previously seen locations in forests to create loop closure constraints.

3.5 Final Considerations

This chapter introduced the primary sensors and concepts used in this work. The methods we present in this thesis leverage machine learning algorithms that can run efficiently on limited computing to perform segmentation and use this information to select features and model objects. In our experiments, the primary sensing sources are a 360 degrees LiDAR and a stereo camera. With a stream of sensor data, we can use the semantic information to generate semantic maps that are informative and more efficient than their metric counterparts while increasing the robustness of robot localization in forests.

CHAPTER

4

SEMANTIC SLAM IN FORESTS

This chapter presents a framework that can run onboard an autonomous agent such as a UAV and create high-quality and informative maps from under forest canopy data. One of the primary challenges in forests is that geometric features are not as reliable as in urban settings due to plants, leaves, and branches on the ground and the tree branches that move with the wind. To increase robustness in these environments, we leverage semantic information to select more reliable features and use geometric priors that better approximate the structure of the environment to improve the robustness of state estimation. At the same time, our map stores the number of individual trees, trunk growth direction, and their DBH, which are useful for forest management and preservation.

Some Figures and Tables presented in this chapter were adapted from our publications "SLOAM: Semantic LiDAR Odometry and Mapping for Forest Inventory" (CHEN *et al.*, 2020b) and "Large-scale Autonomous Flight with Real-time Semantic SLAM under Dense Forest Canopy" (LIU *et al.*, 2022a) with permission from IEEE. The reference implementation is open-sourced and can be found at github.com/kumarRobotics/sloam. Complementary videos are available online video A, and video B.

4.1 Problem Formulation

Let $\mathcal{L} \triangleq \{l^i\}_{i=1}^{N}$ be the set of objects available in an environment. An agent traverses this environment with sensors, collecting a sequence of observations $\mathcal{O} \triangleq \{o_t\}_{t=1}^{T}$, where $o_t \subseteq \mathcal{L}$. The semantic localization and mapping problem consist of estimating the sensor state trajectory $\mathcal{X}$, the number of objects N, and the model parameters and classes of each object l^i. Under the specific assumptions on the model parametrization and class of objects l^i made in this work, solving the semantic localization and mapping problem will yield the tree count and their corresponding DBH for an area of interest of a forest covered by the robot.

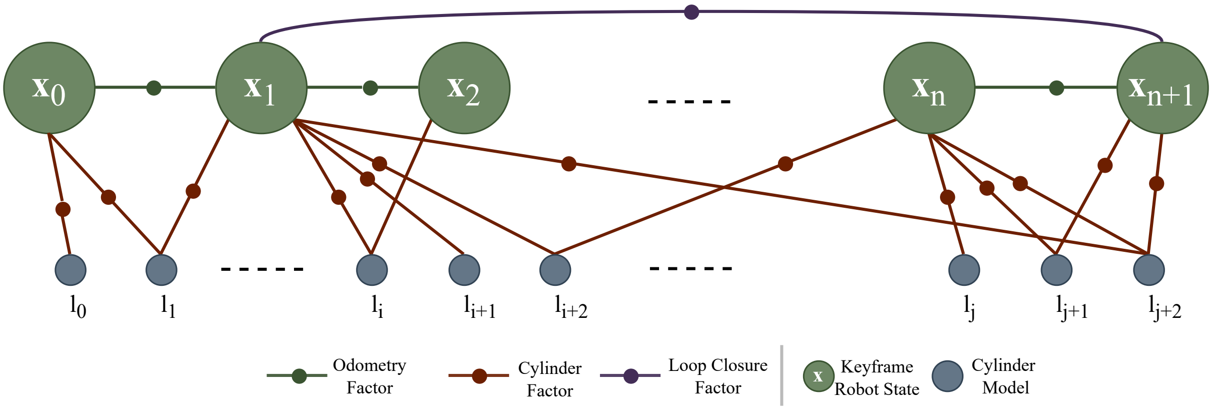


Figure 7 – We propose a modular framework based on a factor graph to couple different sources of measurements, such as LiDAR, stereo odometry, GPS, and loop closure constraints while also enabling our cylinder models of trees to be updated as new observations arrive.

4.2 A Framework for Semantic Localization and Mapping

To achieve reliable state estimation and mapping in forests using a UAV, we propose a factor graph formulation that combines different sensor measurements to update the robot pose while also updating our tree cylinder models as new observations arrive. The factor graph is composed of two main sub-modules. SLOAM, our framework for semantic odometry and mapping, and Urquhart tessellations for loop closure detection. In this thesis, we describe each sub-module of the factor graph in-depth and provide experiments for the individual parts, enabling us to compare their performance with other state-of-the-art methods for LiDAR state estimation and place recognition.

This chapter focuses on the LiDAR odometry and mapping problem, presenting the SLOAM framework and our extensions to integrate external odometry measurements and the factor graph. In Chapter 5, we present a novel method that leverages the semantic maps computed by SLOAM to identify previously seen locations of the forest. The factor graph can directly use this detection to reduce the accumulated drift. The factor graph is illustrated in Figure 7. This flexible framework supports other measurements, such as IMU and GNSS when available, that could further improve the estimates.

Panoptic Segmentation

The first step to incorporating semantic information into our framework is to develop algorithms that recognize objects in the environment from sensor data. We combine a convolutional neural network for semantic segmentation with heuristics for post-processing to identify individual trees and the ground surface. Most architectures for this task were designed initially for images, and to use them with LiDAR. We convert the point clouds into a range image using a spherical projection, as depicted in Figure 8. Models that

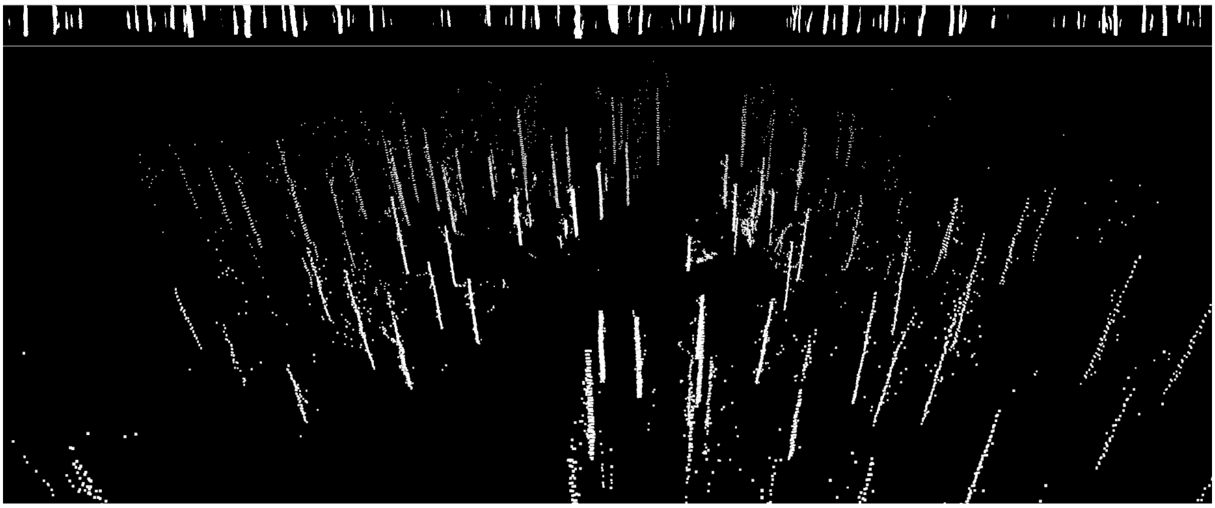


Figure 8 – A 3D point cloud is converted to the more efficient range image representation that serves as input to our semantic segmentation model that outputs a mask of where the tree trunks are (top image). We convert the points labeled as a trunk back to a 3D representation (bottom image) from which SLOAM can extract semantic features and model the trees.

operate on this representation are also more efficient than methods that operate directly on point clouds, such as PointNet++ (QI *et al.*, 2017). We model the segmentation problem as a per-pixel binary classification where each 3D point from the LiDAR (range image pixel) can be labeled as a tree or background.

To run in real time onboard the UAV computer, the neural network model has to be lightweight, which limits the size of the architecture. In this work, we utilize two model architectures to perform semantic segmentation depending on the computational constraints. In these first experiments where every LiDAR sweep is processed, we use a simplified variation of ERFNet (ROMERA *et al.*, 2017) with fewer layers as detailed in Table 3. This architecture can run at 100Hz on the CPU onboard the UAV. However, we observe that the outputs would not be accurate along object edges, considering leaves and small branches as part of the trunk, consequently adding noise to the entire framework. For this reason, in scenarios where we do not need to run inference on every LiDAR sweep, we utilize RangeNet++ (MILIOTO *et al.*, 2019), a more robust architecture designed for LiDAR data. This model can run at only 2Hz while using three CPU cores instead of one.

Table 3 – ERFNet inspired architecture used for semantic segmentation.

Layer	Type	Filter
1	Downsampler block	16
2-3	2x Non-bt-1D (no dilation)	16
4	Downsampler block	32
5-6	2x Non-bt-1D (no dilation)	32
7	Deconvolution	32
8-11	4x Non-bt-1D (no dilation)	32
12	Deconvolution	Num. Classes

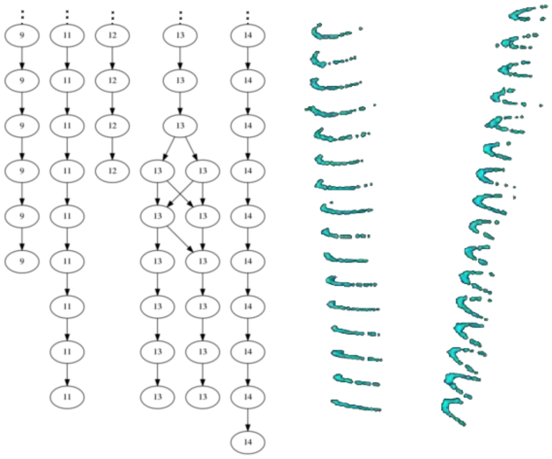


Figure 9 – **Left**: Trellis graph with 5 detected trees (beams closer to the ground at the top, higher beams at the bottom). Tree 13 exhibits a fork structure, which is a valuable information for foresters. **Right**: Front and sideways view of LiDAR points of a tree trunk from which the Trellis graph is derived.

Individual Tree Detection

Assuming that the LiDAR measurements are gravity aligned, trees will grow from the bottom of the LiDAR sweep to the top. Using this insight, we propose a heuristic that defines a Trellis graph (FORNEY, 1973) to detect individual trees from the segmentation. In this representation, each LiDAR beam represents a slice of the graph. For each slice, a group of points that are close enough in space define a node on the graph. Assuming the sweep is gravity aligned and trunks are continuous, we can expect that a node will be available in the following slices that are part of the same tree trunk.

Once the Trellis graph is built, we can identify each tree instance by starting from vertices on the initial slice and finding the shortest path through the graph using a greedy algorithm. We illustrate the resulting graph in Figure 9. Each tree instance in the graph initializes the cylinder model parameters of a new landmark l^i that will be used during the least squares optimization.

Ground Segmentation

The points labeled as the background will contain the ground, shrubs, branches, canopy, and leaves. To extract points $\mathbf{G}$ that belong to the ground surface, we sample the lowest points around the LiDAR sweep using a circular grid with a user-defined parameter to control the number of cells. This parameter can be defined based on the expected variation of the ground (the closer to a plane, the fewer cells are necessary). Without this heuristic, sampling in irregular or sloped terrain would yield an incomplete representation of the actual ground surface.

Model parametrization

Ground

We model the ground as a plane parameterized by $\boldsymbol{\pi} = (\boldsymbol{\omega}, \beta)$, where $\boldsymbol{\omega}$ is the normal of the plane, and β is the offset such that the plane is defined by $\{\mathbf{x} | \langle \mathbf{x}, \boldsymbol{\omega} \rangle + \beta = 0\}$. Given a point $\mathbf{p}$ and plane $\boldsymbol{\pi}$, let $\mathbf{x}_0$ be a point on the plane. We can then define a point to plane distance:

$$D_\pi(\boldsymbol{\pi}, \mathbf{p}) = \frac{\langle -(\mathbf{p} - \mathbf{x}_0), \boldsymbol{\omega} \rangle}{||\boldsymbol{\omega}||}. \tag{4.1}$$

Cylinders

Let $\mathbf{c} = (\rho, \alpha, \kappa)$ be the parameters of a cylinder model. The first parameter ρ is a $3D$ point that represents the cylinder root. That is, the lowest point of the tree that was observed by the LiDAR. α is a ray starting at the tree root that represents the growth direction of the cylinder. Finally, κ represents the radius of the cylinder model. Given a point $\mathbf{p}$ and cylinder $\mathbf{c}$, we can project the feature point $\mathbf{p}$ into the cylinder axis α,

$$\mathbf{p}_{proj} = \frac{(\mathbf{p} - \rho) \cdot \alpha}{\alpha \cdot \alpha} \alpha \tag{4.2}$$

With $\mathbf{p}_{proj}$ we can compute the distance to the cylinder as

$$D_c(\mathbf{c}, \mathbf{p}) = \left\| \mathbf{p} - \mathbf{p}_{proj} \right\|_2 - \kappa. \tag{4.3}$$

That is, the euclidean distance between p and the projection p_{proj} with a margin given by the cylinder radius κ.

Using the lowest observed point as the cylinder root, we assume that the sensor observed the lowest part of the trees, which may not be accurate due to the distance between the tree and the sensor's field of view. For this reason, we can increase the robustness of the cylinder model by computing the intersection between the cylinder axis and the closest ground model. This step increases the consistency of the root estimates across different observations and consequently reduces the drift of the system. This operation is only used in the experiments in subsection 4.3.

Odometry and Mapping

After performing panoptic segmentation on a new LiDAR sweep o_t, we obtain a set of landmarks (cylinders) $\mathcal{O}_t$ and a set of ground points $\mathbf{G}_t$. At $t = 1$, we initialize the coordinate system at the origin and the map $\mathcal{M}$ with the first set of landmarks $\mathcal{M} = \mathcal{O}_1$. For subsequent sweeps, we perform object-based data association between cylinders in $\mathcal{O}_t$

and $\mathcal{O}_{t-1}$ or $\mathcal{M}$ depending on the step. A subset of feature points $\{\mathbf{p}_j\}_{j=1}^{\delta}$ extracted from each cylinder of the sweep that has an association to the map adds a cost in the pose optimization based on the point to cylinder distance in Eq. 4.3.

We present two different approaches to create ground constraints. Similar to LOAM (ZHANG; SINGH, 2014) for each ground point of the current sweep $\mathbf{p}_i \in \mathbf{G}_t$, we find the subset of the closest ground points in $\mathbf{G}_{t-1}$ to $\mathbf{p}_i$ and define a local plane π_i to create a point to plane cost as defined by Eq. 4.1. This approach has the advantage that each point to plane cost will consider a small region of the ground. Even if the ground is irregular, it assumes that only a small patch can be approximated reliably by a plane. On the other hand, this requires the algorithm to compute a plane model for each feature, and if the ground is flat, most of these models will be very similar.

Alternatively, we can compute a fixed number of plane models per observation using the grid cells defined to sample the ground points uniformly. Depending on the expected irregularity of the ground surface, the grid must contain more cells to approximate the surface better. However, if the ground is mostly flat, the algorithm can use only a few plane models, saving computational time. In this approach, we perform data association by matching each feature point to the nearest plane in $t-1$ using the centroid of the points from the cell that defines the plane.

Finally, we can formulate the nonlinear least-squares objective function to estimate the motion between the sweeps z_{t-1} and z_t, T_t by

$$\underset{\mathbf{T}_t}{\arg\min} \quad \lambda_c \sum_{i=1}^{N_t} \sum_{j=1}^{\delta_i} D_c(\mathbf{c}_j, \mathbf{p}_j) + \lambda_g \sum_{l=1}^{\gamma} D_\pi(\boldsymbol{\pi}_l, \mathbf{p}_l), \tag{4.4}$$

where $\lambda_c = \frac{\gamma}{\sum_{i=1}^{N_t} \delta_i}$ and $\lambda_g = \frac{1}{\lambda_c}$ balance the frequency between different features.

Results

We evaluate SLOAM in two different experiments in a pine forest in the state of New Jersey, US. The environment is depicted in Figure 10. In the first experiment, a human carries the LiDAR sensor around a dense part of the forest. This dataset contains rotations and instabilities caused by the human operator dodging obstacles such as brushes on the ground. In the second experiment, the sensors are onboard the UAV that is flown manually by a human pilot. Since GNSS is not reliable under the canopy, the UAV flight controller can only rely on the IMU measurements to stabilize the robot. In this setup, the controller will try to maintain the robot upward, but the altitude has to be maintained by the pilot. This limitation causes the robot to move up and down as the pilot tries to correct the altitude on the controller, as seen in the sideways trajectory in Figure 11, adding additional complexity for state estimation. In this dataset, the UAV starts from

Figure 10 – Wharton State Forest, New Jersey, US. The environment where our real-world experiments are performed (left). Additionally, we measured individual trees with a tape measure to compare human and SLOAM count and diameter measurements (right).

hover and flies in a 65m trajectory for two minutes until it loops back to the initial position and lands. Since the start and end marks are different in the z axis, we offset the goal coordinate by 1 meter instead of using the origin.

The simplified ERFNet segmentation network is trained on 544 scans from which 16 are extracted from the handheld dataset and the remaining are from 5 other regions of the same forest. No data from the UAV flight was used for training.

We benchmark SLOAM against A-LOAM[1], an open source implementation of LOAM, a Intel RealSense T265 stereo camera with odometry measurements provided by their proprietary VIO software and GICP (SEGAL; HAEHNEL; THRUN, 2009), available through PointCloud Library (PCL). In order to increase the speed of GICP, we apply a voxel grid filter to reduce the number of points the algorithm has to consider.

To compare the proposed method with the benchmarks, we qualitatively evaluate the resulting accumulated point cloud maps to observe if any tree ghosting or duplication occurs. Quantitatively, we estimate the accumulated pose drift by computing the difference between the start and end in the UAV experiment, which performs a loop. Since SLOAM explicitly estimates the radius of each tree, we also quantitatively evaluate the DBH estimation compared to human field measurements using a tape measure.

In Figure 11 we present the resulting trajectories of each method for the UAV experiment. This dataset contains sharp rotations and altitude variability due to the robot being controlled by a human pilot. Combined with the vibration caused by the UAV rotors, these factors add noise, especially to the IMU measurements. Moreover, the features under the canopy are less reliable since the algorithm may track points on branches and plants on the ground that can move with the wind. Additionally, there could be patches of light where the sunlight can pass through the canopy, creating abrupt illumination changes that

[1] <https://github.com/HKUST-Aerial-Robotics/A-LOAM>

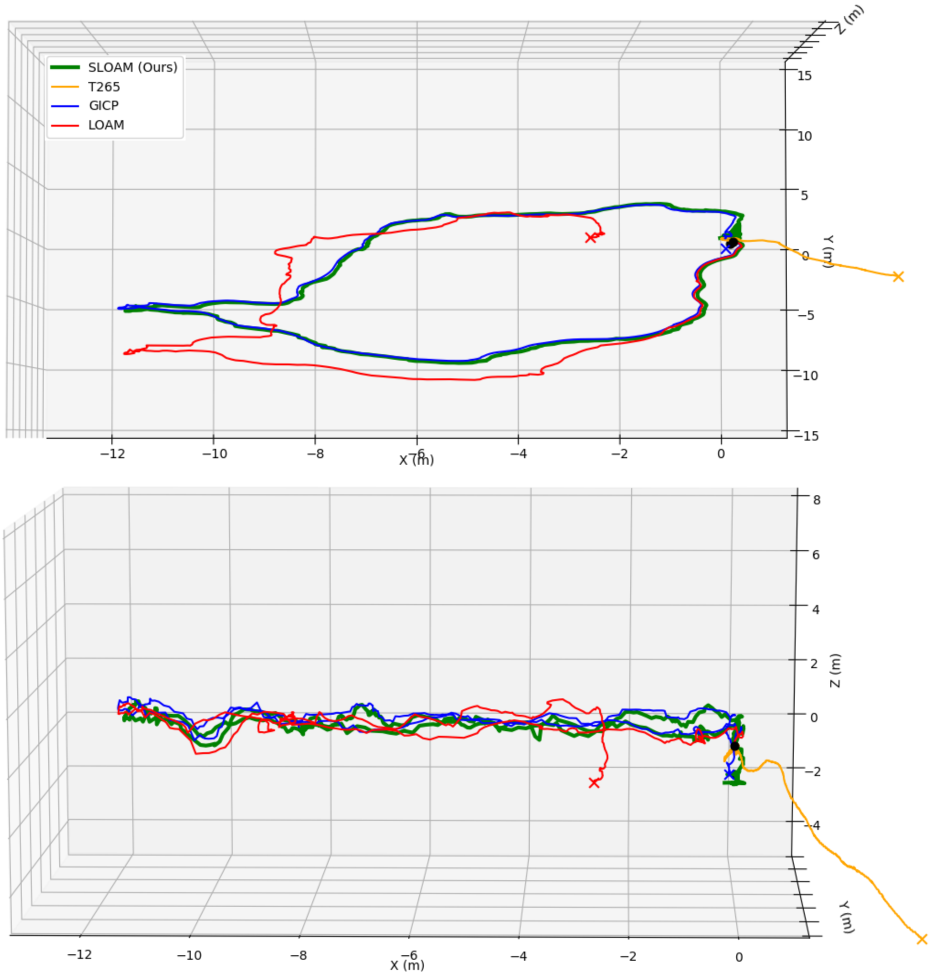


Figure 11 – Top-down (top) and sideways (bottom) views of the trajectories of benchmark methods in the UAV loop trajectory experiment. GICP and SLOAM produce similar trajectories and overall low odometry drift. Meanwhile, A-LOAM drifts significantly, and the RealSense camera completely fails.

may be an issue if the sensor can not adjust the exposure.

Method	Distance from the origin (m)	Error
Ours	0.37	0.58%
GICP	0.41	0.63%
A-LOAM	2.75	4.24%
T265 (VIO)	> 100	> 100%

Table 4 – Error: Distance relative to trajectory length in UAV experiment loop.

LOAM relies on geometric features to perform data association. As explained in Chapter 3, these methods do not perform well in forests since edges and planar surfaces are not well defined in this environment. This noise in data association propagates to the pose estimation, as seen in the looping trajectory (see Figure 11). As presented in Table 4, the distance from the goal according to LOAM is 2.75 meters, while our method achieves an error of 0.37 meters. Consequently, we observe in Figure 12 that the feature points (green dots in the bottom figure) look like a random sample, illustrating how looking for corners and planes is not as viable in this kind of environment.

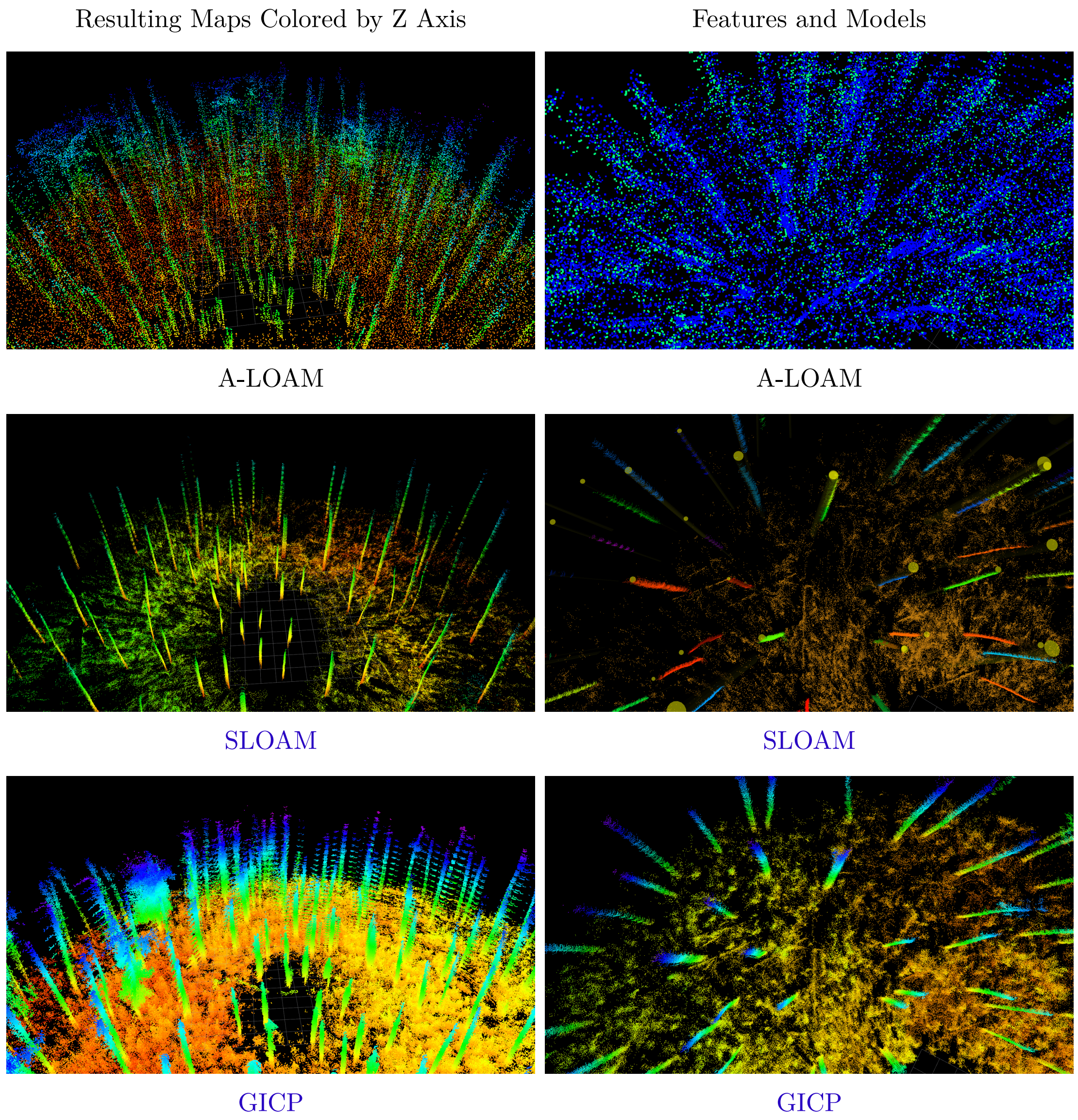
GICP does not make any distinctions between the points to make associations. Since it relies purely on the euclidean distance between points, it requires the source, and target LiDAR sweeps to have a small difference in motion to perform reliable data association without resulting in a local minima solution. For this reason, in the hand-carry dataset (Figure 12 right), GICP can compute a relatively clean map when compared to LOAM. Moreover, the pose drift in the UAV dataset is slightly larger than SLOAM. However, even with a similar drift, we can observe in Figure 12 left that in the UAV dataset, the more aggressive motion creates duplicate trees, especially in regions that are further away from the sensor. These artifacts would inflate the individual count and make it challenging to estimate tree diameters reliably when post-processing the results. SLOAM outperforms both A-LOAM and GICP because our semantic features are more reliable than texture-based lines and planes. Specifically, data association is more robust for both ground and tree features since it inherently filters out noise. The resulting cost function is more informative than the other approaches due to landmark shapes. Moreover, SLOAM uses a point to cylinder cost function while A-LOAM and GICP rely on point-to-plane and point-to-line cost functions forcing a false planar model onto the cylinders resulting in tree trunks that look like flat surfaces.

A by-product of our method is that the resulting maps can be used directly to estimate the properties of the forests. This also provides us with another metric to estimate the quality of the resulting map. We compare the diameter estimates with measurements made by humans for the UAV experiment. We manually measured 35 trees that were inside the field of view of the robot during the flight, from which SLOAM identified 29 individuals as presented in Table 5 with an average error of 1.7 centimeters, which is within the expected margins of the industry.

Detected Trees	Mean	Median	Max	Min
29	1.70	1.52	3.55	0.25

Table 5 – SLOAM DBH error metrics in the UAV experiment with respect to human measurements.

This comparison assumes that the human measurements are correct. To estimate how much we can expect this ground truth to be reliable, we executed another expedition in the same forest with two humans carefully measuring the DBH of 1539 trees. In this comparison, the mean DBH error mas 0.314 centimeters, and the median was 0.254 cm. This result suggests that the expected noise in the ground truth is not the only source of error in the SLOAM results, and there is still room for improvement.



Resulting Maps Colored by Z Axis

Features and Models

A-LOAM

A-LOAM

SLOAM

SLOAM

GICP

GICP

Figure 12 – The assumptions made by GICP and A-LOAM about the geometry of the points during pose estimation result in blurry point clouds and flattening of the tree trunks. Meanwhile, SLOAM exploits semantic information to extract more reliable features and simultaneously models the trees as cylinders, resulting in cleaner and informative maps while having more robustness to aggressive motion.

4.3 Incorporating External Odometry Measurements

SLOAM relies purely on semantic information extracted from a LiDAR to estimate the agent's pose and a map. The problem with this approach is that in the case of segmentation and object detection, if LiDAR failures, the entire system will crash. This limitation is fundamental to consider when designing an autonomous aerial system for large-scale missions, where the odds of encountering edge cases or sensor failure increase.

To incorporate SLOAM into a system that performs such missions under forest canopy, we use S-MSCKF (SUN *et al.*, 2018) VIO as a source of high frequency odometry measurements and run SLOAM only in keyframes. This VIO method is more reliable than the results found in the previous experiments due to the quality of the sensors and strong padding of the UAV's vibration that reduces the IMU noise. Although drift will still occur due to the limitations of VIO in this type of environment, as long as the estimates are locally consistent and smooth, this measurement can be used to provide SLOAM with an initial guess. We chose VIO due to the number of solid works in the literature with open-sourced code. However, this formulation can be adapted to any other source of odometry such as wheel encoders and LiDAR as long as the local consistency and smoothness assumptions hold.

To integrate both measurements, at every keyframe k, we store a tuple of SLOAM and VIO poses $(\mathbf{T}_k^{\text{SLOAM}}, \mathbf{T}_k^{\text{VIO}})$, where the first SLOAM pose is set to odometry pose at the time of the first keyframe. That is, $\mathbf{T}_1^{\text{SLOAM}} = \mathbf{T}_1^{\text{VIO}}$. The initial guess of relative motion between keyframes estimated by the VIO is $\mathbf{T}_k^{\text{REL}} = (\mathbf{T}_{k-1}^{\text{VIO}})^{-1} \cdot \mathbf{T}_k^{\text{VIO}}$. The estimated relative motion can then be combined with the previous SLOAM pose $\mathbf{T}_{k-1}^{\text{SLOAM}}$ to form $\mathbf{T}_k^{\text{GUESS}} = \mathbf{T}_k^{\text{REL}} \cdot \mathbf{T}_{k-1}^{\text{SLOAM}}$, which is used to initialize a new SLOAM iteration.

We create a new keyframe when VIO estimates τ meters of translation movement, where τ is defined by the user. Since SLOAM relies on nearest-neighbor matching, if the motion between two keyframes is large, the association may fail or give false positive matches. To address this, for a new keyframe and its set of landmarks $\mathcal{L}_k^{ROB}$ in the robot frame, we leverage the initial guess $\mathbf{T}_k^{\text{GUESS}}$ to transform these cylinders to the map frame and perform data association between the new measurements and the existing map. To speed up the search for matches, we narrow the candidates from the map by filtering landmarks whose distances to the guess pose are smaller than a threshold ψ

$$\mathcal{S}_k = \{ \left\| \mathbf{l}^{SLOAM} - \mathbf{T}_k^{\text{GUESS}} \right\|^2 < \psi \, : \, \mathbf{l}^{SLOAM} \in \mathcal{M} \}, \tag{4.5}$$

We use a KD-Tree where cylinders are indexed by their roots to perform this operation.

For the ground measurements of a new observation, we only consider the $k-1$

planes $\mathbf{G}_{k-1}^{SLOAM}$ estimated via the grid sampling as presented in Subsection 4.2. Similar to the cylinders, we use the initial guess to perform data association in the map frame.

With the initial guess available, we can split the pose optimization problem defined in Equation 4.4 in two separate steps. One that relies only on the tree cylinders

$$\underset{\mathbf{T}_k^{\text{CYLINDER}}}{\arg\min} \sum_{i=1}^{N_k} \sum_{j=1}^{\delta_{i,k}} D_s(\mathbf{c}_j^{SLOAM}, \mathbf{p}_j), \tag{4.6}$$

where $\mathbf{T}_k^{\text{CYLINDER}}$ is an $\mathbf{SE}(3)$ transformation with three degrees of freedom (translation Z, Pitch, and Roll are fixed), and another that uses only the ground

$$\underset{\mathbf{T}_k^{\text{GROUND}}}{\arg\min} \sum_{l=1}^{\gamma_k} D_\pi(\boldsymbol{\pi}_l, \mathbf{p}_l), \tag{4.7}$$

where $\mathbf{T}_k^{\text{GROUND}}$ is an $\mathbf{SE}(3)$ transformation with three degrees of freedom (translation X, Y and Yaw are fixed). By doing this, we can constrain parts of the initial pose in case of partial failure. For example, we can constrain the altitude of the UAV even in regions where no trees are detected by using ground constraints. The final pose estimate estimated by SLOAM is given by

$$\mathbf{T}_k^{\text{SLOAM}} = \mathbf{T}_k^{\text{GUESS}} \cdot \mathbf{T}_k^{\text{GROUND}} \cdot \mathbf{T}_k^{\text{CYLINDER}}. \tag{4.8}$$

Note that since the features and models of the keyframe k are already in the map frame according to $\mathbf{T}_k^{\text{GUESS}}$, the output of each optimization step is a refinement with respect to the initial guess.

LeGO-LOAM (SHAN; ENGLOT, 2018) proposes a similar two-step optimization formulation, where the authors report 35% reduction in computation and similar accuracy when compared to estimating the full pose in one problem. LeGO-LOAM uses the output of the ground optimization to constrain the X,Y and Yaw parameter estimation. However, since we have an initial guess from the odometry, we solve 4.6 and 4.7 independently.

This integration is part of a system described in depth in (LIU *et al.*, 2022a). We refer the reader to the paper for more information on the other parts of the system such as the custom simulator, trajectory planning, obstacle avoidance and the robot's hardware.

Results

We present two experiments in different regions of the same pine forest as the previous results and evaluate the influence of the VIO and SLOAM integration on the robustness of the state estimation module for an autonomous UAV system. For segmentation, we utilize RangeNet++ trained on simulator data and fine-tuned on 50 (37 training and

13 validation) manually annotated LiDAR sweeps from a different pine forest in Arkansas, US. Note that the semantic segmentation model did not have access to data from the forest where the experiments were conducted.

In the first scenario, the robot executes an autonomous flight mission where the objective is to perform two square loops and return home. However, the autonomous system has access only to the VIO estimates. For this reason, the UAV believes the mission was successful when in practice, we observe that the robot landed far from the takeoff position. We run SLOAM on the data recorded from this mission and use GNSS as a high-level reference for the actual trajectory. It is only possible to qualitatively compare the impact of this integration since the GNSS sensor reported a ~10 m standard deviation for X and Y position estimates, which is expected under the forest canopy. From Figure 13, we can observe that the SLOAM + VIO trajectory is much closer to the GNSS measurements than pure VIO. Nevertheless, even with noisy measurements, an external source of odometry can reduce the computational load of running a semantic framework as part of an autonomous system in real time and increase the system's robustness in this environment.

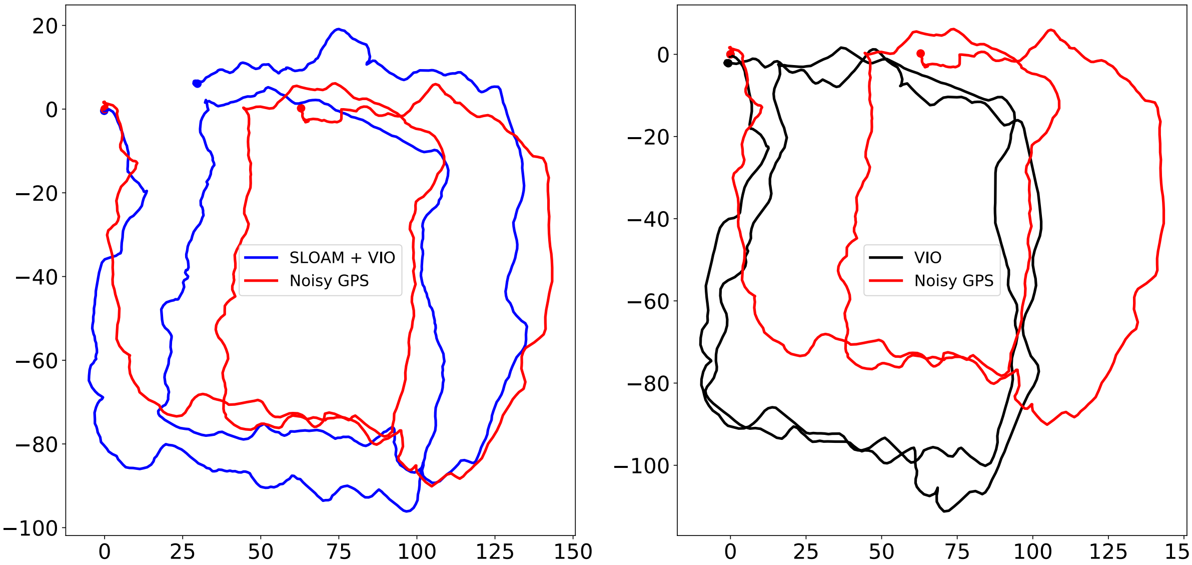


Figure 13 – Top-down view of the 800m UAV flight. The robot flies autonomously, attempting to perform two squared loops having access only to the VIO estimates. While the autonomy stack believes the robot achieved its goal, both SLOAM and the noisy GNSS show that the robot was far from the take-off position.

In the second experiment, similar to the loop experiment in Subsection 4.2, the UAV is controlled by a human pilot throughout a 1.1km trajectory, depicted in Figure 14 to guarantee it will return to the take-off position. In this flight, the GNSS sensor was not able to get a lock. Consequently, our only reference for the quality of each trajectory is the difference between the starting and end pose of the robot according to VIO and SLOAM + VIO. From the drift estimates in Table 6 we can observe that the XY drift is reduced significantly with SLOAM, but it is still present. The most important impact is

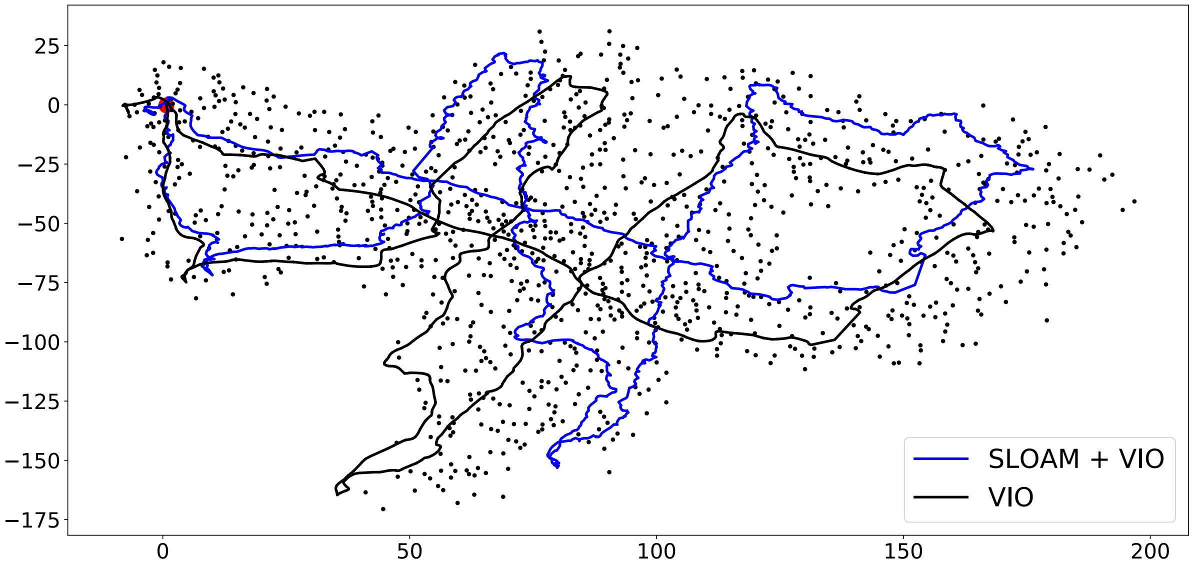


Figure 14 – Top-down view of the 1.1 km trajectory. In this scenario, the robot is manually piloted so that we guarantee it will return to its initial position and we can estimate the final trajectory drift. Black points illustrate the tree cylinders detected by SLOAM.

on the Z-axis, where VIO drifted more than 6 meters downward while SLOAM almost completely removed this error.

Table 6 – Distance of the final estimated pose from the beginning of the trajectory on the 1.1 km loop.

Method	XY Drift (m)	Z Drift (m)	Total Drift (m)
VIO	7.92	-6.27	10.10
SLOAM + VIO	3.93	0.71	3.99

4.4 SLOAM as a Factor Graph

Integrating other sensor measurements with SLOAM is an essential step towards a more robust autonomous system. This section presents an extension to the external odometry formulation using a Factor Graph formulation solved via incremental SaM (KAESS *et al.*, 2012) for state estimation and mapping. The Factor graph is a modular framework that can naturally incorporate external odometry measurements, semantic landmarks, loop closure, and GNSS in a single framework. Our implementation is based on GTSAM (DELLAERT, 2012), a popular library for defining and solving factor graphs. The factor graph enables the SLOAM framework to consider multiple sources of measurements and their uncertainty, which is especially important as we can update the tree cylinders as new observations arrive in a probabilistic way and quantify the certainty of our estimates.

Similar to the external odometry formulation, the factor graph operates on keyframes created with respect to a minimum motion threshold defined by the user. This regime reduces the number of variables in the optimization problem, balancing the quality of the solution, memory, and computational cost (SHAN *et al.*, 2020).

Keyframe pose estimates are constrained by the relative motion between them with a binary factor $f_{\text{pose}}(x_{k-1}, x_k; \mathbf{T}_k^{\text{REL}})$ (DELLAERT; KAESS *et al.*, 2017). Notice that since the source of the pose guess and the relative poses are the same (odometry), without any other factors, the solution of the factor graph would be the same as the odometry.

Cylinder Factor

For every keyframe k, we estimate ground models $\mathbf{G}_k$ and cylinder models $\mathcal{L}_k$. For every observed tree, we perform data association with the map via nearest neighbors as described in Equation 4.5. Instead of directly running pose optimization with these associations as in previous formulations, the landmark measurements are added to the factor graph problem, together with the keyframe pose as a binary factor$f_{\text{land}}(l^i, x_k; m_k^i)$.

In f_{land}, l^i is the current factor graph estimate of the ith landmark indexed with respect to the order they where added to the map. x_k is the current pose estimate, given by the previous keyframe pose estimate and the relative motion estimated by the odometry $x_k = x_{k-1} \cdot \mathbf{T}_k^{\text{REL}}$ and m_k^i is new landmark measurement. In other words, the cylinder factor describes the measurement error between the current model estimate and a new observation of the same tree as the difference of each model parameter transformed by the pose estimate. This dependency on the pose estimate also causes the cylinder model to constrain the robot poses during optimization. Our custom factor f_{land} models the measurement likelihoods with a Gaussian noise model, calculated as

$$f_{\text{land}}(\mathbf{T}_k, l^i; m_k^i) = exp\{-\frac{1}{2}\left\|E(\mathbf{T}_k, l^i, m_k^i)\right\|_{\Sigma}^2\} \tag{4.9}$$

where $E(\cdot,\cdot,\cdot)$ can be split into three independent terms

$$E(l, m)_\kappa = l_\kappa - m_\kappa$$

estimates the error between the current estimate and a new measurement for the cylinder radius κ, and does not depend on the pose.

$$E(\mathbf{T}, l, m)_\alpha = (\mathbf{T}_R \cdot l_\alpha) - m_\alpha$$

estimates the cylinder axis α error considering the effect of the robot rotation. Finally,

$$E(l, \mathbf{T}, m)_\rho = (\mathbf{T} \cdot l_\rho) - m_\rho$$

considers the full $\mathbf{SE}(3)$ pose for the cylinder root ρ error.

With this factor, the cylinder models and their uncertainty are updated as new measurements arrive. Similar to the previous SLOAM formulations, the cylinder factors also contribute to the pose optimization problem, creating additional constraints to the robot pose at every keyframe. Our factor graph formulation has been used as the foundation and extended for an active mapping system (LIU *et al.*, 2022b). The authors demonstrated that this factor graph is an effective way to perform robust state estimation in real time in the real world with multiple robots.

4.5 Final Considerations

This chapter introduced the theoretical foundation for a factor graph formulation for semantic SLAM in forests. This formulation relies on SLOAM, a framework for odometry and mapping under the forest canopy that automatically estimates timber inventory using LiDAR measurements. We show that by using semantic information, namely tree and ground features and models, this method matches or outperforms other LiDAR-based frameworks that do not incorporate such information. We present results showing that the quality of pose estimates and the resulting map is improved with SLOAM while generating useful information about the environment.

In large-scale experiments, we observe that depending solely on semantic constraints can be problematic since these methods cannot compute an estimate when segmentation or detection failure occurs. For this reason, we presented an extension for SLOAM using external odometry measurements to increase robustness. We show that even if VIO drifts over time, as long as the estimates are locally consistent, it can be used by SLOAM and the factor graph as an initial guess for the pose optimization problem. At the same time, SLOAM provides refinements based on measurements from keyframes. This integration is also beneficial in an autonomous system with limited computational resources since the panoptic segmentation and semantic modeling task can be expensive to perform at the same rate as the sensor measurements.

One crucial missing piece for the factor graph is detecting previously seen locations and incorporating such detections into the state estimation problem. In the next chapter, we present our method for loop closure detection based on the semantic maps computed by SLOAM and how it can be incorporated to mitigate drift. Later, in Chapter 6 we present an experiment in simulation using the complete factor graph formulation.

CHAPTER

5

PLACE RECOGNITION UNDER PERCEPTUAL ALIASING

The semantic SLAM framework proposed in this thesis computes a semantic map $\mathcal{M} \triangleq \{l^i\}_{i=1}^N$, where each landmark l is modeled by a cylinder parameterized by their root ρ, a 3D point representing the position of the lowest part of the tree, a ray α that captures the direction of the trunk and a radius κ.

In general, forests have many similar regions. In environments such as Pine forests, where trees have approximately the same age and are all from the same species, their radius and growth direction are very similar. In this case, their position is the only information that reliably differentiates trees. While object properties could be more informative in other environments, we show that in this worst-case scenario, using the position of the landmarks is enough to identify seen places even under high perceptual aliasing.

Images and Tables presented in this chapter were adapted from our published work "Place Recognition in Forests with Urquhart Tessellations" with permission from IEEE. The reference implementation is open-sourced and can be found at github.com/gnardari/urquhart.

5.1 Problem Formulation

Similar to the previous chapter, we represent an observation $\mathcal{O}_k$ of a place in the environment as sets of semantic objects $\mathcal{L}_k$ that are visible from a reference frame T_k. However, for the place recognition problem, we consider that this observation can come from a single sensor reading $\mathcal{O}_k = \mathcal{L}_k = \{l^i\}_{i=1}^{N_k}$ or a union of a sequence of $2c$ observations $\mathcal{O}_k = \bigcup\limits_{i=-c}^{c} \mathcal{L}_{k+i}$ to increase the area being represented by our algorithm.

Due to sensor noise and occlusion, some landmarks in the environment may not be detected by the system or detected in one observation and then not detected in another. For this reason, our method has to be robust to detection failure. Moreover, each observation will have different landmark position estimates due to the different noise sources (sensor, segmentation, instance detection, modeling), and our algorithm has to be able to handle these inconsistencies.

Our primary motivation for doing place recognition is to mitigate drift by incorporating loop closure into the factor graph formulation of SLOAM introduced in Chapter 4. To this end, we must not only identify a previously seen location but also be able to compute a transformation $\mathbf{H}_{a,b}$ that maps the observation b to a that will be added as a constraint to our SLAM framework.

Problem (Place recognition under perceptual aliasing). Given observations $\mathcal{O}_a$ and $\mathcal{O}_b$, determine if the corresponding observations overlap $\mathcal{O}_a \cap \mathcal{O}_b \neq \emptyset$ and if so, estimate the associated rigid transformation $\mathbf{H}_{a,b}$ where

$$\mathbf{T}_a = \mathbf{H}_{a,b}\mathbf{T}_b. \tag{5.1}$$

5.2 Urquhart Tessellations

Our method encodes the topology of the semantic map via geometric shapes. To achieve this, we switch back and forth between a graph-based and a geometric interpretation of these shapes. To this end, we represent every landmark as a point in $\mathbb{R}^2$. For trees, this point comes from the 2D projection of the root ρ onto the ground plane.

The most basic primitive of our formulation is the edge. Geometrically, an edge $e = (p_i, p_j)$ is a line segment bounded by a pair of points p_i, $p_j \in \mathbb{R}^2$. The length of this primitive encodes the euclidean distance between p_i and p_j. A polygon L is a closed set defined by the region enclosed by the edges constructed via consecutive point pairs (p_i, p_{i+1}) in the sequence of points $(p_1, \ldots, p_n)$ where $p_1 = p_n$. Conversely, the polygon L is defined by a sequence of edges $(e_1, \ldots, e_n)$, where each edge e_i is constructed based on the original point sequence.

Let P be a set of at least 3 discrete points $p \in \mathbb{R}^2$ in general position. A tessellation is a finite set of polygons $\{L_1, \ldots L_n\}$ which covers the convex hull $\mathcal{Q}(P)$ without gaps or overlaps. More precisely, $\bigcup\limits_{i=1}^{n} L_i = \mathcal{Q}(P)$ and $int(L_i) \cap int(L_j) = \emptyset \;\; \forall\, i \neq j$, where $int(\cdot)$ denotes the interior of a polygon.

A triangle is a special case of the polygon where $n = 3$, and the circumcircle is the circle that passes through the points of the triangle. A triangulation is a tessellation where all elements are triangles. The Delaunay triangulation $DT(P)$ is a triangulation where no point $p \in P$ is in the circumcircle of any triangle of $DT(P)$ (DELAUNAY *et al.*, 1934). In our application, P is given by the 2D projections of the root of each landmark in O_k.

We interpret the Delaunay triangulation as a graph, where the edges are given by the triangles, in order to construct the Urquhart graph (URQUHART, 1980). Let $\mathcal{G}_\mathcal{D} = \{\mathcal{V}_\mathcal{D}, \mathcal{E}_\mathcal{D}\}$ be the graph representation of $DT(P)$, where $\mathcal{V}_\mathcal{D}$ is the union of the triangle points, and $\mathcal{E}_\mathcal{D}$ is the union of the triangle edges. The set of the longest edges of each triangle in $DT(P)$ is defined by $\Omega = \{\underset{e \in L}{\arg\max} \|e\| : L \in DT(P)\}$. The Urquhart graph of $DT(P)$ is a graph $\mathcal{G}_U = \{\mathcal{V}_U, \mathcal{E}_U\}$ where $\mathcal{V}_U = \mathcal{V}_\mathcal{D}$ and $\mathcal{E}_U = \mathcal{E}_\mathcal{D} \setminus \Omega$. $\mathcal{G}_U$ is a sub-graph of $\mathcal{G}_\mathcal{D}$ where the longest edges of each triangle are removed.

We then convert the Urquhart graph $\mathcal{G}_U$ back into an Urquhart tessellation $U(P)$ using cycle detection. A simple cycle c of an arbitrary graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is a non-empty sequence of edges $\mathcal{E}_c = (e_1, \ldots, e_n) \subseteq \mathcal{E}$ with a vertex sequence $\mathcal{V}_c = (v_1, \ldots, v_n) \subseteq \mathcal{V}$ such that $v_1 = v_n$, and $v_i = v_j \iff i, j \in \{1, n\}$, i.e. there are no repeated vertices except for the first and the last. The simple cycles of a graph correspond to polygons of a tessellation.

The cycle basis $\mathcal{C}$ of a graph $\mathcal{G}$ is the minimal set of simple cycles such that for all cycles $c \in \mathcal{G}$, $\exists\, c_i, c_j \in \mathcal{C}$, such that $c = c_i \Delta c_j$ where Δ represents the symmetric difference operation. Intuitively, any cycle can be computed with elements of the cycle basis. The cycle basis of a graph corresponds to the tessellation. As a result, we can convert the graph $\mathcal{G}_U$ into the tessellation $U(P)$.

These tessellations motivate a hierarchy of geometric primitives $\mathcal{H}(P)$ that encompass local to global information. The first level $\mathcal{H}^0(P)$ is the set of all edges $\mathcal{E}_D$. The second level $\mathcal{H}^1(P)$ is given by the triangles of the Delaunay triangulation $DT(P)$. The third level $\mathcal{H}^2(P)$ is given by the polygons of the Urquhart tessellation $U(P)$.

We define a function $\phi^i(\cdot) : \mathcal{H}^{i+1}(P) \to \mathcal{H}^i(P)$, to map from higher to lower levels, where

$$\phi^i(s) = \{l : l \cap s = l, l \in \mathcal{H}^i(P)\}, \tag{5.2}$$

e.g. ϕ^0 maps triangles of the Delaunay triangulation to its corresponding edges, ϕ^1 maps polygons of the Urquhart tessellation to its corresponding triangles. An example of the different structures computed with this method is given in Figure 15.

Method

The geometric primitives derived from the Urquhart Tessellation naturally encode the topology of an observation of the environment by connecting nearby landmarks and their relative distance. Similarly, the parallel work of Li et al (LI *et al.*, 2020) demonstrates that triangles provided by the Delaunay Triangulation of the landmark positions $DT(P)$ can be used directly for place recognition.

Triangles encode a small portion of the observation, subsequently, the odds of encountering similar structures increases with the scale of the region being mapped. On the

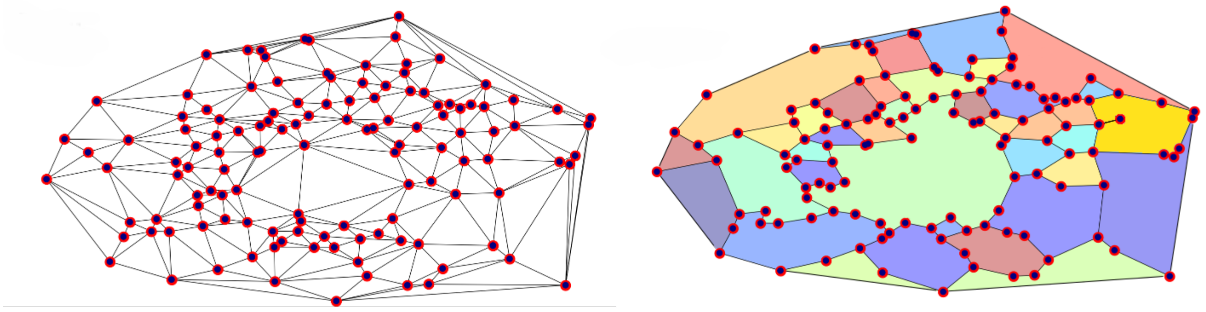


Figure 15 – Given a set of 2D points derived from our tree cylinder models, the Delaunay Graph (left) defines a set of $\mathcal{H}^1$ polygons (triangles). The Urquhart Graph (right) composes these triangles to generate a new set of $\mathcal{H}^2$ polygons (colored elements). The set of all available polygons $\mathcal{H}$ can be used to identify previously seen locations and correspondences between points.

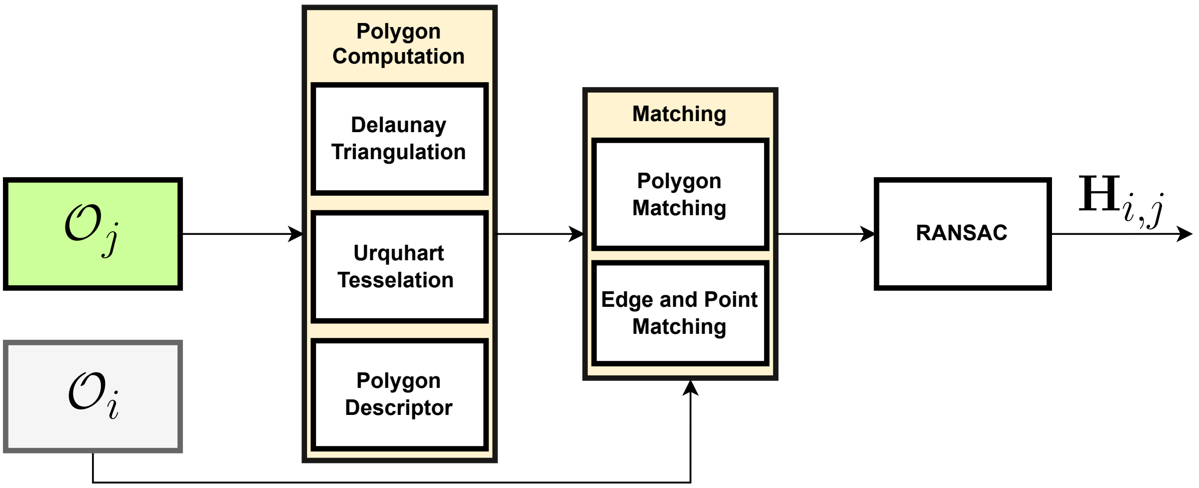


Figure 16 – Place recognition pipeline with Urquhart Tessellations.

other hand, for two $\mathcal{H}^2$ polygons L_m and L_n from different regions to have similar metric properties, it would require that the triangles that compose L_m and L_n also have similar metric properties and are arranged in space similarly, such that $\bigcup \phi^1(L_m) \approx \bigcup \phi^1(L_n)$. For this reason, the Urquhart Tessellation $U(P)$ creates polygons that are less likely to repeat than $DT(P)$, decreasing the probability of false-positive correspondences.

The number of triangles computed by $DT(P)$ has an upper bound of $2n - 2 - b$ where n is the number of points, and b are the points that lie in the convex hull of P $\mathcal{Q}(P)$ (DELAUNAY *et al.*, 1934). By joining triangles to create more complex polygons, we reduce the odds of false positives and the average number of shapes that must be compared to identify a loop.

A trade off of our method is that it requires extra computation to get $\mathcal{H}^2$ from $\mathcal{H}^1$. In Algorithm 1 we summarize our approach to compute the polygons given a new set of points, where we loop through the elements of $\mathcal{H}^1$ to compute the set of longest edges Ω while also updating $\mathcal{C}_U$ as triangles are combined to efficiently compute both the Urquhart

Algorithm 1 – Urquhart Graph with Cycle Detection

```
input: $\mathcal{G}_D$, $\mathcal{H}^1$                                  ▷ Delaunay graph and triangles
$\mathcal{C} = \mathcal{H}^1$
$\mathcal{G}_U = \mathcal{G}_D$
for each triangle $L \in \mathcal{H}^1$ do
    $e_L = \arg\max_{e \in L} \|e\|$
    Find $L_{neigh} \in \mathcal{C}$, $L_{neigh} \neq L$, $e_L \in L_{neigh}$   ▷ A neighboring triangle that shares $e_L$.
    Drop $e_L$ from $\mathcal{G}_U$
    $\mathcal{C}_L = L \Delta L_{neigh}$
end for
return $\mathcal{G}_U$, $\mathcal{C}$
```

graph $\mathcal{G}_U$ and its cycle basis $\mathcal{C}_U$.

Descriptor

Our $\mathcal{H}^2$ polygons can be composed by a different number of points, edges and triangles. The same region can generate slightly different polygons due to a missed landmark or position noise. For this reason, we need a robust and efficient descriptor to compare the geometric shapes.

We borrow techniques from the shape retrieval literature and, for each polygon and triangle $L \in \mathcal{H}(t)$, we compute a descriptor based on their centroid distance (ZHANG; LU *et al.*, 2001). Let $N = \{p : p \in L\}$ be the set of points that compose a polygon. The centroid $c = (c^x, c^y)$ of L is computed by

$$c^x = \frac{1}{|N|}\sum_{n=1}^{|N|} p_n^x, \quad c^y = \frac{1}{|N|}\sum_{n=1}^{|N|} p_n^y.$$

Since the size of N can vary for different polygons, we sample a constant number points relative to their perimeter length P. The *step* size between sampled points is a user-defined parameter, where $0 < step < 1.0$ such that $step * P$. This operation creates a new set of points M with the same number of elements regardless of the size of N. If *step* is large, the samples will be spaced out, smoothing out the surface descriptor, while a small number will capture more details of the shape, such as sharp corners. An optimal value of *step* balances the two properties such that the descriptor is robust to some noise but capture enough information about the polygon to avoid false-positive associations.

The new centroid distance with sampled points, as illustrated in Figure 17 is

$$F(L) = \{\|p_n - c\|^2 \, : \, p_n \in M\}.$$

$F(L)$ is translation invariant since it uses distance relative to the centroid. However,

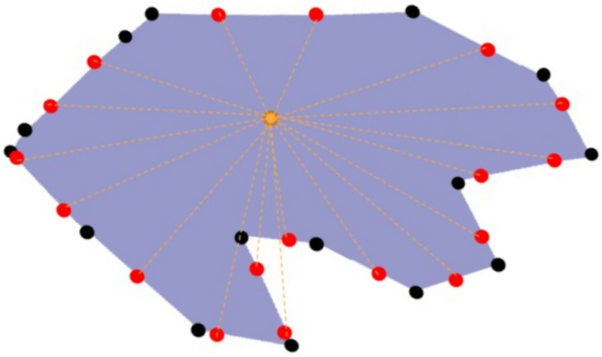


Figure 17 – We compute the polygon centroid (yellow) as the mean of the original points (black). Then, we sample a new set of points from the perimeter (red) from which the centroid distance descriptor is computed. With this approach, the descriptor size will be independent of the number of points that define the shape.

the order of the descriptor can vary depending on where sampling starts. Similar to GLAROT (KALLASI; RIZZINI, 2016), it is possible to test every configuration of a pair of descriptors, and choose the permutation with the smallest distance, but this can be inefficient, as we may have to match many polygons per observation.

Instead, following the shape retrieval literature (ZHANG; LU *et al.*, 2001), we compute a Discrete Fourier Transform (DFT) of $F(L)$ to obtain a new descriptor $\hat{F}(L) = DFT(F(L))$ that transforms the centroid distance to the frequency domain. The lower frequency values of the DFT contain information about the general features of the shape, while higher frequency descriptors contain information about the details of the polygon. The magnitude of this descriptor $\bar{F}(L) = \left|\hat{F}(L)\right|$ has the property that it will be the same regardless of the order of the input, making the descriptor more robust to the sampling order.

Matching

We store $\bar{F}(L)$ for all polygons in $\mathcal{H}$, including triangles. Given two polygons of any dimension $L_n \in \mathcal{H}_i$ and $L_m \in \mathcal{H}_j$, they are considered a match if

$$\left\|\bar{F}(L_n) - \bar{F}(L_m)\right\|^2 < \tau. \tag{5.3}$$

To increase robustness and speed, we only compare polygons if $|N_n| - |N_m| \leq 3$. That is, if the difference in number of points that define the polygons is smaller than or equal to 3. With this approach, we are able to identify correspondences in overlapping observation even with noisy measurements, as illustrated in Figure 18.

The pair of polygons $L_n \in \mathcal{H}_i^2$ from observation i and $L_m \in \mathcal{H}_j^2$ from observation j define the subsets of triangles $\phi^1(L_n) \subseteq \mathcal{H}_i^1$ and $\phi^1(L_m) \subseteq \mathcal{H}_j^1$. Let x bet the number of triangles in $\phi^1(L_n)$ with correspondences in $\phi^1(L_m)$. If the ratio between x and the

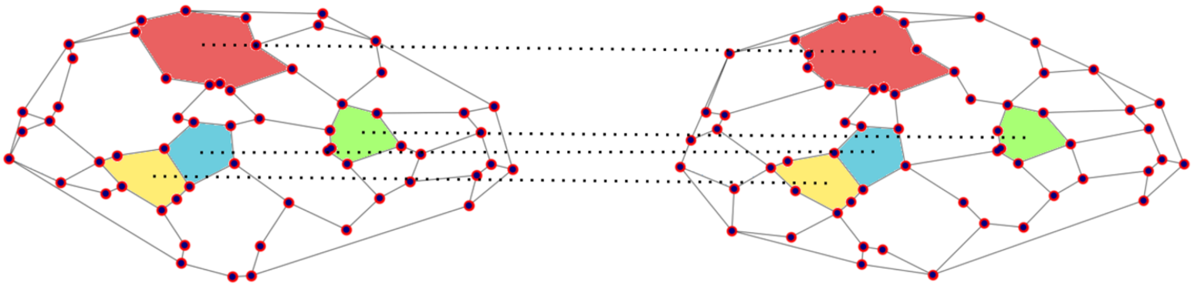


Figure 18 – Example of two observations of the sample place, with inconsistencies in trees detected and their position. Even with noisy measurements, our method can identify enough corresponding polygons.

number of elements of $\phi^1(L_n)$ is greater than a threshold η, the triangle correspondences are considered valid. That is,

$$\begin{cases} \text{match}, & \text{if } \dfrac{x}{\|\phi^1(L_n)\|} > \eta \\ \text{not match}, & \text{otherwise.} \end{cases}$$

If L_n and L_m match, we repeat the process comparing the subsets of triangles they define. For a pair of matching triangles $L_k \in \phi^1(L_n) \in \mathcal{H}_i^1$ and $L_l \in \phi^1(L_m) \in \mathcal{H}_j^1$, we associate edges based on their lengths $\bar{\bar{\mathcal{E}}}_k = \{\|e\| : e \in L_k\}$, and $\bar{\bar{\mathcal{E}}}_l = \{\|e\| : e \in L_l\}$ by

$$\arg\min_k \left\| \bar{\bar{\mathcal{E}}}_m - \chi_k \bar{\bar{\mathcal{E}}}_n \right\|^2 ,$$

where χ is a permutation matrix reordering the elements of $\bar{\bar{\mathcal{E}}}$. Intuitively, the matrix that generates the smallest difference between the lengths of the edges is the best assignment between them. Since the number of elements to compare is small, this comparison is not as computationally expensive as testing permutations for an arbitrary polygon. Finally, we extend this assignment to point correspondences by matching points that share corresponding edges.

Euclidean Transformation Estimation

We run our experiments in $\mathbb{R}^2$, and assume that the data will not suffer from shearing or scale variations to reduce $\mathbf{H}_{i,j}$ to a Euclidean transformation with 3 degrees of freedom that we estimate using RANdom SAmple Consensus (RANSAC).

If the Euclidean distance between corresponding points after the transformation is below a threshold d, we consider the correspondence an inlier. If the ratio of inliers to outliers is above a threshold r or the maximum number of iterations s is reached, the algorithm stops and returns the estimate which has the most inliers.

For the $\mathbb{R}^3$ case, the assignments found by our algorithm can be propagated to the entire object, and an optimization-based approach can be used to align the instances, e.g., as an alternative to the data association methods presented in Chapter 4. We present in Section 5.3 an approach to create loop closure constraints between 3D poses in our factor graph formulation of SLOAM using Urquhart Tessellations.

Results

Simulation Experiments

We simulate a $1km^2$ or approximately 247 acres forest. To ensure a density of trees that is consistent with a real world forest, the set of 2D landmarks is generated by Poisson-Disc sampling through Bridson's algorithm with a minimum distance between points of $7m$ (BRIDSON, 2007). This algorithm will create a regular pattern across the environment, which is not a realistic representation of the distribution of trees in a real forest. To make the distribution closer to reality, each point in the set is perturbed with Gaussian noise with 0 mean and $3m$ standard deviation.

Each simulated observation will have a radius of $50m$, similar to a real-world sensor such as an Ouster or Velodyne, and will capture approximately 80 trees. The average distance from a tree to its nearest neighbor in the simulated forest is $3.4m$, while in the accumulated map from our real-world dataset, this distance is $3.2m$.

Due to sensor noise and detection failure, the input observations may be noisy, and different measurements from the same place will not be perfectly consistent. It is essential to measure how our method and the benchmarks handle this noise to check their suitability in real-world loop closure applications.

To simulate real-world inconsistencies, we model the landmark detection noise $\delta(l)$ with a Bernoulli distribution as in (WANG *et al.*, 2011). Let $\delta(l)$ be the Boolean random variable representing the successful detection of landmark l,

$$\delta(l) = \begin{cases} 1 & \text{if landmark } l \text{ is detected,} \\ 0 & \text{otherwise.} \end{cases}$$

The distribution of $\delta(l)$ has a success probability ω, that is, $\delta(l) \sim Ber(\omega)$. We define simulated observation under the presence of detection noise

$$\bar{\mathcal{O}}_t \triangleq \{l : \delta(l) = 1\}_{l \in \mathcal{S}_t} \subseteq \mathcal{S}_t,$$

where $\mathcal{S}_t$ is a subset of the ground truth semantic map.

Due to sensor noise and uncertainty in the landmark projection on the ground plane, the tree root projection $\bar{\rho}$ may vary in different observations. We model this noise as $\bar{\rho} = \rho + \epsilon$, where ϵ is a Gaussian random variable with zero mean and variance Σ, $\epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma)$. The 2D observation of a submap $\mathcal{S}_t$ including both forms of sensor noise is

$$\bar{\mathcal{O}}_t \triangleq \{\delta(\bar{\rho})\}_{l \in \mathcal{S}_t}.$$

The simulated robot does a circular path 4 times. Each observation is rotated by a random angle sampled from a uniform distribution in the range $\{0, \frac{\pi}{2}\}$, and every landmark l in the observations subject to position noise and detection noise. Excluding the trivial match where $i = j$, we consider a match a true positive if $\|\mathbf{T}_a - \mathbf{H}_{a,b}\mathbf{T}_b\|^2 < 10$ and the rotation difference is smaller than 20^o, which are similar constrains to related work (GAWEL *et al.*, 2018). We consider a false negative when not enough matches are found but the distance between the ground truth poses is smaller than the lidar radius.

For the simulated experiments, we run all combinations of the detection success probability ω in the range $\{0.8, 0.9, 0.95, 1.0\}$ and the standard deviation $\sigma = \sqrt{\Sigma}$ of the position estimation error ϵ in the range $\{0.0, 0.1, 0.2, 0.3, 0.4\}$ totaling 20 different configurations.

As defined in Equation 5.3, our method requires a threshold parameter τ that sets the maximum difference between polygons to be considered a match. We refer the reader to our paper (NARDARI *et al.*, 2020), where we evaluate multiple configurations of τ and define that $\tau = 5$ is the value that achieves the best balance between precision and recall, quantified by the F1-Score.

Finally, we evaluate the performance of all methods with respect to the F1-Score for each combination of noise using the same range for σ and δ.

Both the position of detection noise impact the performance of every method. Li et al. can handle more detection failures than both our method and GLAROT. As triangles capture a smaller portion of the observation, they are more likely to have consistent polygons, even with a large percentage of unobserved landmarks. However, with the smallest possible position noise ($10cm$), the performance of Li et al. significantly drops. The main factor for this is that the descriptor relies on the area of the triangles, which has high sensitivity to noise.

We used the parameters configuration for all methods that achieved the best F1 score in our simulated environment. However, GLAROT was not designed for such a high-density and high similarity environment, which implied either high precision and low recall or low precision and high recall. As a result, GLAROT performs poorly even in the scenario with no noise due to false-positive associations.

Table 7 – F1-scores for each method in the simulation place recognition experiment. We simulate the observations of a robot with different levels of landmark position noise and detection success probability. In most cases, our method is more robust the benchmarks.

Detection Success Prob.	100%			95%			90%			80%		
Position Noise	Ours	GLAROT	Li et al.	Ours	GLAROT	Li et al.	Ours	GLAROT	Li et al.	Ours	GLAROT	Li et al.
0cm	**1.00**	0.52	**1.00**	**1.00**	0.27	**1.00**	0.95	0.12	**0.99**	0.32	0.01	**0.75**
10cm	**1.00**	0.39	0.07	**0.99**	0.12	0.01	**0.92**	0.04	0.00	0.32	0.01	0.00
20cm	**0.99**	0.17	0.00	**0.98**	0.06	0.00	**0.82**	0.02	0.00	0.23	0.00	0.00
30cm	**0.97**	0.04	0.00	**0.76**	0.01	0.00	0.45	0.00	0.00	0.08	0.00	0.00
40cm	0.66	0.01	0.00	0.30	0.00	0.00	0.12	0.00	0.00	0.01	0.00	0.00

For our method, as the polygons in $\mathcal{H}^2$ capture a larger area of the observation, these elements are more likely to be altered as landmarks are not detected. For this reason, while it is more robust than the other methods, we observe that our approach is more sensitive to detection failures.

While our method is generally more robust than the benchmarked methods, we observe that the $\mathcal{H}^2$ polygons are more sensitive to detection failure. Since these polygons capture a larger area of the observation, their shape is more likely to be altered if landmarks are not detected.

Real-World Experiments

In this experiment, we fly the robot twice across the same plot of a commercial Pine tree forest. For both flights, we split the raw data into subsets for every minute of the flight. The sub-maps have partial overlap, position noise, and detection failure caused by the expected causes of noise that were already discussed, making this task challenging for methods that are not robust to these factors. To extract the landmark observations $\mathcal{O}_t$, we run SLOAM on each subset t individually.

The map is initialized with the first sub-map $\mathcal{O}_0^M = \mathcal{O}_0$, and subsequent sub-maps are matched to the map in chronological order such that $\mathcal{O}_{t+1}^M = \mathcal{O}_t^M \bigcup \mathbf{H}_{M,t+1}\mathcal{O}_{t+1}$, where $\mathbf{H}_{M,t+1}$ is given by the different place recognition methods and the union $\bigcup$ is the output of Density Based Spatial Clustering of Application with Noise (DBSCAN) (ESTER *et al.*, 1996), clustering trees closer than $0.5m$ after the alignment into a single landmark. We use this approach to remove duplicates instead of correspondences since some landmark matches may not have been detected. For every new observation that is integrated into the map, we recompute a new descriptor for the map.

To quantify the performance of the different methods, we require some kind of reference or ground truth. The forest canopy causes GNSS to have errors ranging up to tens of meters, preventing it from being used as ground truth. On the other hand, despite the small drift observed in past experiments with SLOAM, its estimates may not be perfect. Moreover, since each sub-map is a separate run, there will be inconsistencies in the location of trees due to the order trees are added to the different sub-maps. To compute a reference transformation, we use the SLOAM estimates as an initial guess to

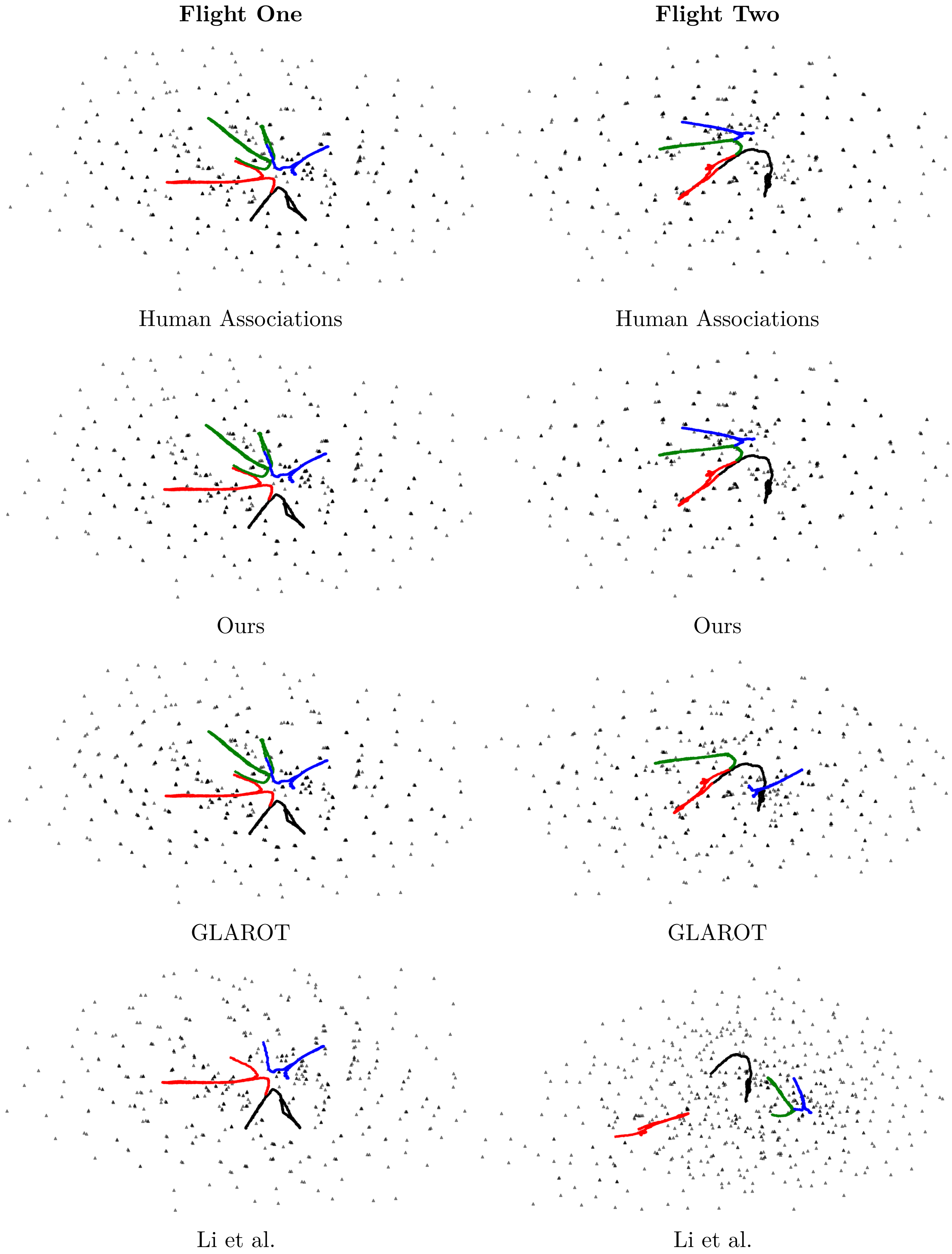


Figure 19 – Accumulated sub-maps for flight one (left column) and flight two (right column). Colors represent the robot trajectory in different sub-maps, and black triangles represent the landmarks. We iteratively merge pairs of sub-maps until the entire trajectory is accumulated into a single map. Our method closely approximates the results obtained with human associations.

Table 8 – Landmark alignment error. We compare the euclidean distance between manually associated landmarks using the transformations computed with correspondences provided by each method.

Experiment / Method	Flight 1			Flight 2		
	Mean	Min.	Max.	Mean	Min.	Max.
Human Associations	0.17	0.00	0.54	0.21	0.01	0.67
Ours	**0.19**	**0.00**	**0.61**	**0.23**	**0.01**	**0.83**
GLAROT	0.20	0.01	0.90	30.15	0.03	89.22
Li et al.	-	-	-	42.38	4.79	125.71

align subsequent sub-maps. For sub-map $t + 1$, its initial alignment is given by the last SLOAM pose in sub-map t. Using this pose, we manually annotate tree correspondences between pairs of pre-aligned sub-maps, and use RANSAC to compute the final reference transformation that will be used to compare the different approaches.

For all methods, we compute a translation error based on the Euclidean distance and a rotation error based on the absolute difference for each transformation between subsequent sub-maps when compared with the human associations. For our method, the translation error on flight 1 is 0.43 meters, and the rotation error is 0.32 degrees. For GLAROT, the translation and rotation errors are 0.33 meters and 1.54 degrees, respectively, and Li et al. fails to find correspondences for one of the sub-maps. On flight two, the translation and rotation errors are 0.26 meters and 0.33 degrees for our method, while the others have translation errors greater than 15 meters and 200 degrees. In Figure 19, we show that on flight 1, our method and GLAROT closely approximate the result with human associations, and Li et al. also approximates it except for the sub-map that it fails to align. On flight 2, our method is similar to the reference, while the benchmarks are significantly off.

With the transformations computed with manual annotations, we select tree associations considered inliers by RANSAC to evaluate the quality of landmark alignments, totaling 142 and 162 trees for flights 1 and 2 respectively. The distance between these landmarks after alignment is computed on pairs of subsequent sub-maps. In Table 8, we show the mean, minimum and maximum distances over all pairs compared to the human associations. Similar to our previous results, our method closely approximates the reference on both flights, while GLAROT achieves good results in flight one but fails in flight two, and Li et al. fails in both flights.

5.3 Loop Closure Factor

The Urquhart Tessellations proposed in this chapter can identify previously seen locations in forests relying only on a 2D projection of the position of trees. However, as our robot operates in 3D space, the method has to be extended to be incorporated into

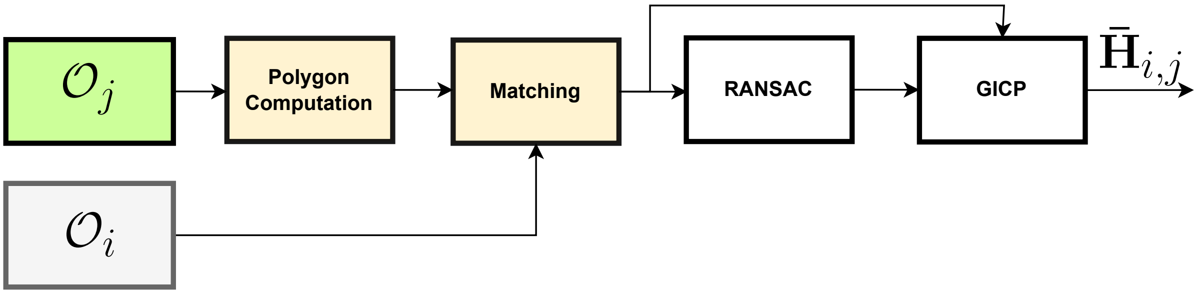


Figure 20 – We extend the proposed place recognition pipeline using GICP on the cylinder features of landmarks associated with our 2D Urquhart Tessellations. With this approach, we can create 3D pose constraints for our factor graph formulation without increasing the computational complexity of the descriptor.

the factor graph formulation.

While it is possible to generalize the 2D geometric tessellations to N dimensions, this increases their computational cost. For this reason, we maintain the 2D descriptors for place recognition and data association. Instead, for a pair of observations $\mathcal{O}_i$ and $\mathcal{O}_j$, if a loop is identified with our method, we use the 3D locations of the roots of associated cylinders to estimate an initial alignment of the keyframes with RANSAC.

From the RANSAC inlier matches, we initialize a pair of point clouds using the associated cylinder features. Then, we apply GICP to get the refined transformation $\mathbf{T}_{i,j}^{\text{LOOP}}$ between the two frames. Since GICP can get stuck in local minima if the initial rotation between the point clouds is large, this two step solution is more robust without making any assumption about the reliability of the pose estimates associated with these observations. The output transformation $\bar{\mathbf{H}}$ creates a factor between the poses associated with the observations $f_{\text{pose}}(x_i, x_j, \bar{\mathbf{H}}_{i,j}^{\text{LOOP}})$.

5.4 Final Considerations

This chapter presented a novel approach for place recognition under high perceptual aliasing. We show that using the position of trees derived from the semantic map computed by our method introduced in Chapter 4, we can detect previously visited locations more robustly than other methods designed for similar constraints. Moreover, we present how this method can be incorporated into the factor graph formulation to help reduce the accumulated drift when the robot revisits places.

Up to this point, our methods used LiDAR as the primary sensing source for semantic mapping. However, this sensor is expensive and heavy. As we work towards denser and more diverse forests such as rainforests, we would like to be able to use the algorithms proposed in this thesis on smaller and cheaper platforms. In the next chapter, we present results in simulation on using a stereo camera as a substitute for the LiDAR and how this sensor can seamlessly be incorporated into the factor graph framework by computing pseudo-LiDAR measurements.

CHAPTER

6

TOWARDS REAL-TIME SEMANTIC PSEUDO-LIDAR

Autonomous robots must understand their surroundings and make real-time decisions while having tight computational constraints. For this reason, it is essential that the algorithms that compose the system share as much information as possible to solve different but related tasks.

We have shown in this thesis how semantics can improve LiDARs state estimation, mapping, and solving global localization under high perceptual aliasing. However, LiDARs are expensive and relatively heavy sensors compared to cameras. The weight is especially significant when designing UAVs where payload directly impacts flight time. In this chapter, we propose a learning-based approach to the stereo depth estimation problem that simultaneously outputs semantic labels and stereo disparity estimates. This method can obtain pseudo-LiDAR readings and substitute the semantic pipeline used in previous sections by a single model while saving on computation and hardware weight. Finally, we demonstrate that these measurements can be used by our factor graph to perform semantic mapping in a simulated forest.

Classic feature matching-based algorithms rely on edges and corners to select keypoints, which are unreliable under illumination changes, which frequently happen in forests due to the sunlight passing through canopy gaps. Neural networks can learn more robust descriptors that give cues about the geometry and structure of a scene even in such scenarios (KENDALL *et al.*, 2017; CHANG; CHEN, 2018). When designing systems that use stereo cameras, estimating the depth and semantic labels for the image observations is often desirable for the downstream task.

Tackling both tasks with a single model can improve computational efficiency and the model's performance. For example, SegStereo (YANG *et al.*, 2018) concatenates semantic and correlation features to compute a disparity cost volume. Moreover, based

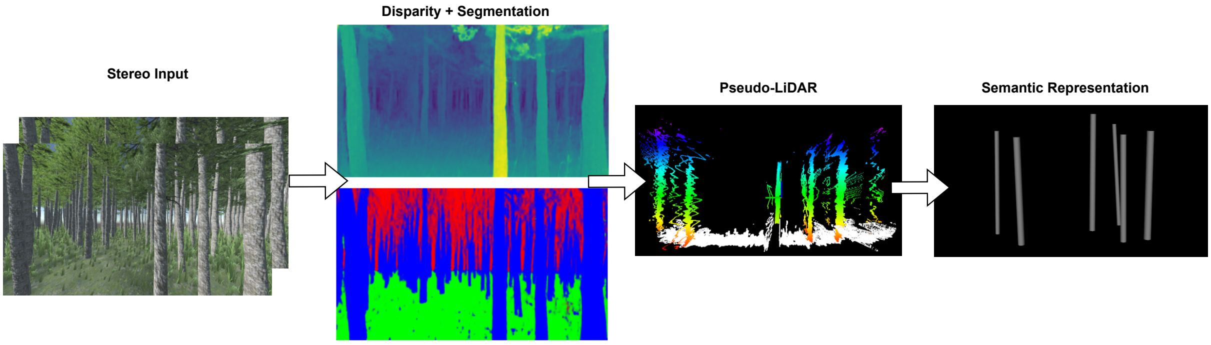


Figure 21 – Given a pair of images from a stereo camera, our neural network will simultaneously estimate disparity and semantic labels for each pixel. We can then project the pixels of interest to 3D to get Pseudo-LiDAR measurements. These outputs are useful for our main task of semantic mapping where the robot could identify and model individual trees in real time.

on the disparity predicted by the model, SegStereo warps the right image I_r with the estimated disparity to reconstruct the left frame I_l'. From I_l', the left segmentation ground truth can be used to compute a loss term that is backpropagated through the disparity network.

DispSegNet (ZHANG *et al.*, 2019) and RTS^{2}Net (DOVESI *et al.*, 2020) propose architectures where semantic embeddings are used to refine the disparity estimates of the model. SGNet (CHEN *et al.*, 2020a) proposes a confidence module composed of 3D convolutions that leverages the consistency between the semantic and disparity inner product left and right feature correlations. In addition, the authors utilize category-dependent residuals for the disparity (WU *et al.*, 2019). The intuition is that semantic categories contain well-defined boundaries that should exist in the disparity map. However, this is not true for all cases. For example, the boundary between roads and crosswalks is not necessarily visible in the disparity map. For this reason, the authors limit this loss to a pre-defined list of classes where this assumption holds.

In this context, we propose a novel network architecture for joint semantic segmentation and disparity estimation using supervision only for the first task. With this approach, we can incorporate loss constraints that consider our object of interest (tree trunks) to improve the disparity results.

6.1 Problem Formulation

Given a pair of rectified RGB images $(I_l, I_r)^k$ at keyframe k, we estimate the disparity map D^k that displaces each pixel (u, v) in I_r to its corresponding coordinates (u', v') in I_l, and a semantic mask S^k. Our final goal will be to identify individual trees to create a semantic representation of the keyframe $\mathcal{L}^k = \{l_1, l_2, \ldots, l_n\}$, where l is a tree landmark modeled as a cylinder. Each step of our framework to achieve this is illustrated in Figure 21.

6.2 Semantic Feature Extraction and Segmentation

The semantic branch of our architecture is a MobilenetV3 (HOWARD *et al.*, 2019) backbone for efficient feature extraction for disparity and segmentation. We use Lite R-ASPP architecture as the segmentation head, as in the original paper. We modify the original architecture to increase the dimensionality of our feature maps. This increases the overall computational time of the backbone but is necessary to maintain enough resolution for the iterative disparity estimation.

The segmentation branch considers only the left rectified view, using supervised examples automatically computed by the simulator. The semantic loss function $\mathbf{L}_{sem}$ is defined by

$$\mathbf{L}_{sem} = 0.5\,\mathrm{CE}(S_l, \hat{S}_l) + 0.5\,\mathrm{DICE}(S_l, \hat{S}_l), \tag{6.1}$$

where S_l and $\hat{S}_l$ are the ground truth and predicted semantic segmentation mask, respectively. DICE refers to the multi-class Dice Loss, and CE is the Cross-Entropy loss.

6.3 Deep Disparity Estimation

Feature Aggregation

Feature pyramids have become a standard for disparity estimation. By leveraging feature maps from different intermediate layers, the model can capture local and global information about the input. We define three feature maps extracted from the MobileNetV3 backbone F_1, F_2, F_3 of resolutions $1/8, 1/4,$ and $1/2$ of the original image input.

Different approaches have shown that aggregation of features from different resolutions benefits disparity estimation. AANet (XU; ZHANG, 2020) constructs separate cost volumes for each feature resolution and aggregates their output. Similar to SENet (HU; SHEN; SUN, 2018) and RTStereo (CHANG; CHANG; CHEN, 2020), the proposed approach combines features from our backbone via cross-scale feature aggregation. An attention mechanism weights this operation before computing the cost volume. The network learns the attention weights A on the smallest scale features (F_1 for our architecture). Then, for a feature map F_i, we match the scale of the other maps for aggregation via the following rule

$$\begin{cases} F_j' = \phi(up(F_j)), & \text{if } i > j \\ F_j' = \phi'(F_j), & \text{otherwise.} \end{cases}$$

up is the bilinear upsampling operation, ϕ and ϕ' are convolutions followed by batch norm but with strides 1 and 2 respectively. With this approach, we redefine each feature map as a combination of the resized counterparts. E.g., for F_2, the aggregated feature map is $F_2 = \phi(\phi(up(F_1)) + F_2 + \phi'(F_3)) \cdot A + F_2$. This operation increases the representability of each feature map with a small computational footprint.

Cost Volume

One of the most expensive steps of disparity estimation is the cost volume computation, where every disparity hypothesis inside a pre-defined range $[0, D_{max}]$ for each pixel is considered. We propose to compute the total cost volume on a lower scale representation of the input images and compute residuals (adjustments with respect to the lower resolution disparity estimate) on larger scales to speed up this step (WANG *et al.*, 2019b; CHANG; CHANG; CHEN, 2020). We compute the full disparity in our model using $F1$ (1/8) and residuals with $F2$ (1/4) and $F3$ (1/2). See the architecture design overview in Figure 22 for reference.

A critical part of the cost volume computation is cost representation. Some models estimate pixel similarity by concatenating the feature vectors, creating a 4D cost volume (C, D, H, W). In contrast, others perform a reduction such as a norm or correlation by inner product (DOSOVITSKIY *et al.*, 2015), resulting in a 3D cost volume (D, H, W). In the first case, the cost volume is followed by an aggregation step that requires 3D convolutions, which is a more expensive operation than its 2D counterpart. For this reason, we use a correlation-based cost volume. However, if computation is not a constraint, 4D volumes are considered more informative and are expected to lead to better results (KENDALL *et al.*, 2017; CHANG; CHEN, 2018; ZHANG *et al.*, 2020).

Disparity

As proposed in (KENDALL *et al.*, 2017), we model disparity estimation as a regression problem, where the disparity estimate $\hat{d}$ is given by

$$\hat{d} = \sum_{d=0}^{D_{max}} d \times \sigma(c_d), \tag{6.2}$$

where σ is the softmax function and c_d is the matching cost for each disparity candidate d.

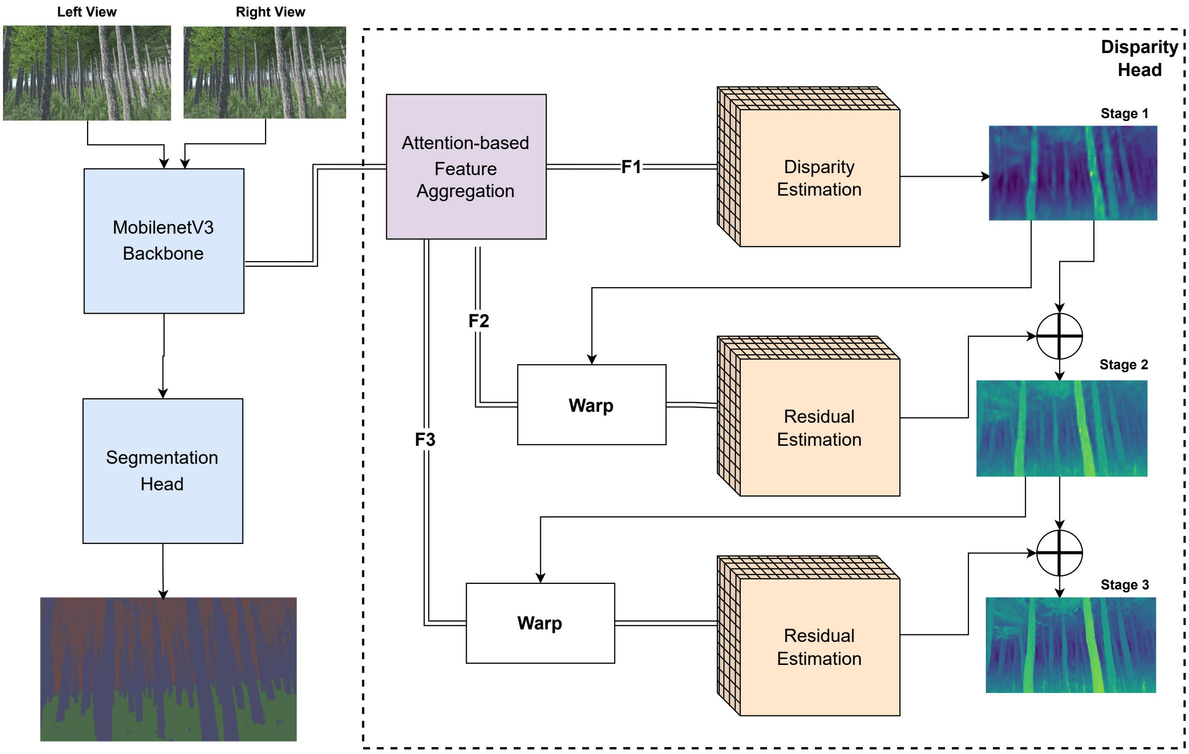


Figure 22 – Overview of our Neural Network Architecture. Sharing a MobilenetV3 backbone, the segmentation head outputs a mask for the left image view. Meanwhile, the disparity head uses semantic features of different resolutions to estimate a disparity map efficiently in three stages that trade-off speed and accuracy.

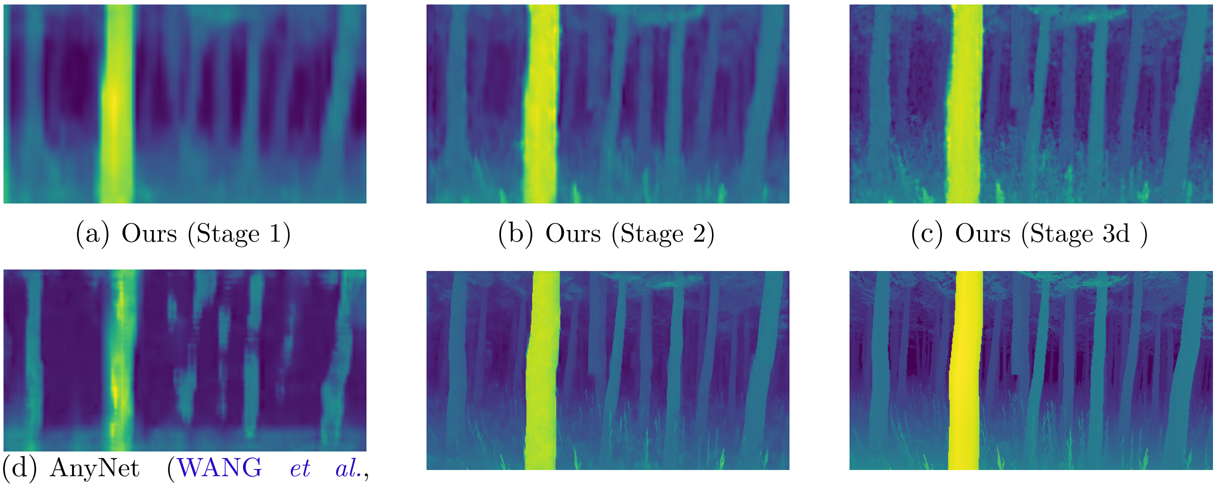
Self-supervised Loss

Most works using CNNs to estimate disparity rely on pseudo or sparse ground truth to learn in a supervised fashion. However, this data is not trivial to capture and is not always available. For this reason, we rely on the reprojection error to learn the disparity between the rectified images. Given the rectified image pair I_l, I_r and the disparity $\hat{d}$, we can reconstruct the left image from the right image by warping the right pixels with the disparity estimate $\hat{I}_l = \text{warp}(I_r, \hat{d})$. Finally, as in Monodepth (GODARD *et al.*, 2019), the final loss is given by a weighted sum of SSIM(WANG *et al.*, 2004) and the L1 loss function, with α set to 0.15

$$\mathbf{L}_{disp} = (1 - \alpha)\,\frac{(1 - \text{SSIM}(I_l, \hat{I}_l))}{2} + \alpha \left| I_l - \hat{I}_l \right|. \tag{6.3}$$

Semantic Smoothness Loss

The ground and the canopy are irregular surfaces in forests due to the noise created by leaves, branches, and foliage. On the other hand, the tree trunk, our main object of interest, can be approximated by a smooth surface. Similar to (HEISE *et al.*, 2013; GODARD; AODHA; BROSTOW, 2017), we incorporate a second loss term based on



(a) Ours (Stage 1)

(b) Ours (Stage 2)

(c) Ours (Stage 3d)

(d) AnyNet (WANG *et al.*, 2019b) (Stage 3)

(e) AANet (XU; ZHANG, 2020)

(f) Ground Truth

Figure 23 – Disparity results from each stage of our network, AnyNet and AANet.

disparity gradients ∂d weighted by an edge-aware term using image gradients ∂I_l. However, in our formulation, we incorporate the semantic mask $\hat{S}^l$ to only consider pixels labeled as a tree trunk,

$$\mathbf{L}_{smooth} = \frac{1}{N}\sum_{i,j}\left|\partial_x(\hat{d}_{ij}\hat{S}^l)\right| e^{-\left\|\partial_x I^l_{ij}\right\|} + \left|\partial_y(\hat{d}_{ij}\hat{S}^l)\right| e^{-\left\|\partial_y I^l_{ij}\right\|}. \tag{6.4}$$

We perform joint learning by a simple summation of task-specific losses $\mathbf{L}_{sem} + \sum_{i=1}^{3}\mathbf{L}^i_{smooth} + \mathbf{L}^i_{disp}$. However, many works investigate better approaches for multi-task learning to ensure the model will converge for both tasks, that the features learned are the best possible for both scenarios, and minimize the trade-offs these approaches make (KENDALL; GAL; CIPOLLA, 2018; NAKAMURA; GRASSI JR; WOLF, 2022). Integrating these methods could further improve the accuracy of our model without compromising the speed requirements.

6.4 Experiments

To design and compare our architecture, we capture data using a pine forest simulated in Unity3D. In the simulator, we define a UAV carrying a stereo camera with a baseline of 12cm, 80 degrees of field of view, and image resolution of 640 x 360. The simulator provides ground truth semantic segmentation labels and depth, from which we derive the disparity using the camera intrinsic and extrinsic parameters.

Using the autonomy stack proposed in (LIU *et al.*, 2022a), we define multiple missions using waypoints relative to the take-off position, which are completed autonomously. This creates data from trajectories that are closer to what a real robot would do, with different orientations and speeds. We define a mission to capture the training set, com-

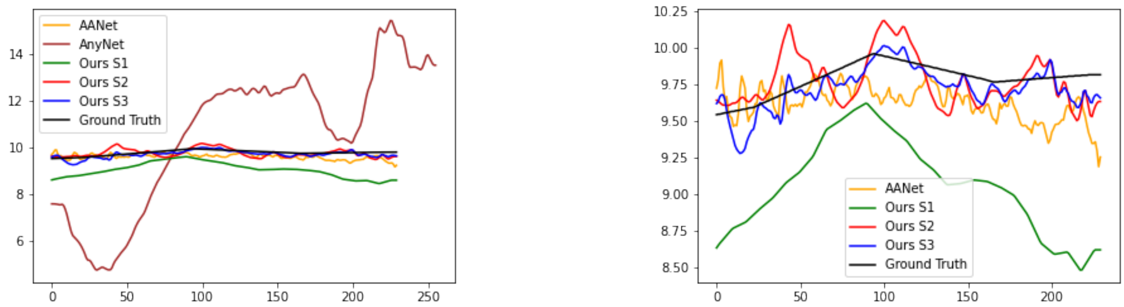


Figure 24 – Disparity estimates across a vertical line on a tree trunk from each method. We observe that AnyNet outputs noisy estimates while our method computes increasingly better estimates at each stage. AANet is the closest to the ground truth but is computationally more expensive.

posed of 590 image pairs, and a different mission for testing with 149 images. Unlike in autonomous vehicle datasets where the car is aligned with the ground plane, the robot tilts and yaws during the mission, increasing the variability of the data.

In Figure 23, we illustrate how the trade-offs made by each model affect the quality of the disparity estimate. Since the first stage of our model computes the disparity using a low-resolution feature map, the disparity is blurry, but the structure of the nearest tree trunks is captured. Meanwhile, we can observe that the twigs and leaves on the ground look blurry. Subsequently, stages two and three refine the disparity map sharpening edges and defining smaller structures on the ground. Similarly, in Figure 24, we observe that the quality of the disparity estimate for each stage across a tree trunk follows the same pattern.

AnyNet uses 1/16 of the input resolution to compute the initial disparity map. This resolution works for datasets such as Kitti (GEIGER *et al.*, 2013), where the image width is large (1382). However, for our dataset, we can observe that the low resolution creates a very noisy and incomplete disparity map, from which the later stages cannot recover due to the smaller disparity residual search range. AANet, on the other hand, aggregates cost maps at different resolutions to estimate a single disparity map. This approach is visually the closest to the ground truth but comes at the cost of increased computation.

Unlike the benchmarks, our model can also output a semantic mask for the input. This output is essential for downstream tasks such as semantic mapping. We illustrate in Figure 21 how the masked disparity can be used to create Pseudo-LiDAR measurements and separate point clouds for the ground and tree points.

Finally, since we have ground truth disparity measurements in the simulator, we calculate metrics for the disparity error, displayed in Table 9. The End-Point Error (EPE) computes the absolute per-pixel difference between model output and ground truth disparity. The 3-Pixel Error (3PE) denotes the percentage of estimated disparity pixels

Table 9 – Disparity EPE, 3PE, and model runtime for each stage of our method trained jointly for both tasks, only for disparity (Ours D), and the benchmarks. We use the ground truth disparity to compute the errors and to train the benchmarks, while our method is learns disparity without supervision.

	Ours S1	Ours S2	Ours S3	Ours D S1	Ours D S2	Ours D S3	AANet	AnyNet S3
3PE	0.070	0.036	0.035	0.056	0.0328	0.033	0.012	0.493
EPE	1.080	0.762	0.688	1.040	0.760	0.689	0.380	4.230
Trunk EPE	1.199	0.762	0.669	1.159	0.768	0.672	0.302	4.559
Ground EPE	0.803	0.706	0.689	0.741	0.686	0.682	0.433	2.438
Runtime (ms)	11.345	12.253	13.094	11.345	12.253	13.094	36.567	6.59

whose absolute difference from the ground truth is larger than 3. We also report these metrics stratifying by the ground truth semantic labels. Moreover, we report median inference time for a single image pair running on an NVIDIA 3060 RTX laptop GPU.

The proposed method balances between cost map resolution and speed while learning two tasks. Simultaneously, our architecture maintains an inference speed performance acceptable for real-time semantic mapping. When using the same architecture, the model trained only for disparity estimation performs slightly better but still underperforms AANet, which computes cost maps at higher resolutions. On the other hand, AnyNet performs worse than all methods due to the low horizontal resolution during cost map computation.

Semantic Mapping with Stereo and Factor Graph SLOAM

To demonstrate how the Pseudo-LiDAR measurements derived from the proposed model can be integrated into the factor graph SLOAM framework, we define a loop trajectory in the Unity3D forest, executed autonomously by the simulated UAV.

The network's disparity output is transformed to depth and projected to 3D using the camera parameters. With the semantic segmentation mask, two separate point clouds are created, one for the points labeled as a tree and another for the ground points, as shown in Figure 26. In this figure, we can observe that the measurements are noisier around the edges, where the disparity changes drastically. This noise is a known limitation of CNN-based approaches that have to reason about 3D data using 2D images and convolutions. Some works have proposed losses that consider the 3D projection of the network's estimates to explicitly account for this effect when using pseudo-LiDAR (WANG *et al.*, 2019a), which could further to be applied and improve our proposal.

In Figure 25, we show the estimated trajectories using LiDAR and pseudo-LiDAR in a simulated loop experiment. The colored triangles illustrate the trees mapped via each method. While the pose estimates are similar, in this experiment is illustrated the most significant trade-offs of using cameras: the horizontal field of view and depth range. It holds to note that the LiDAR has a 360^o horizontal field of view, our cameras have 80^o.

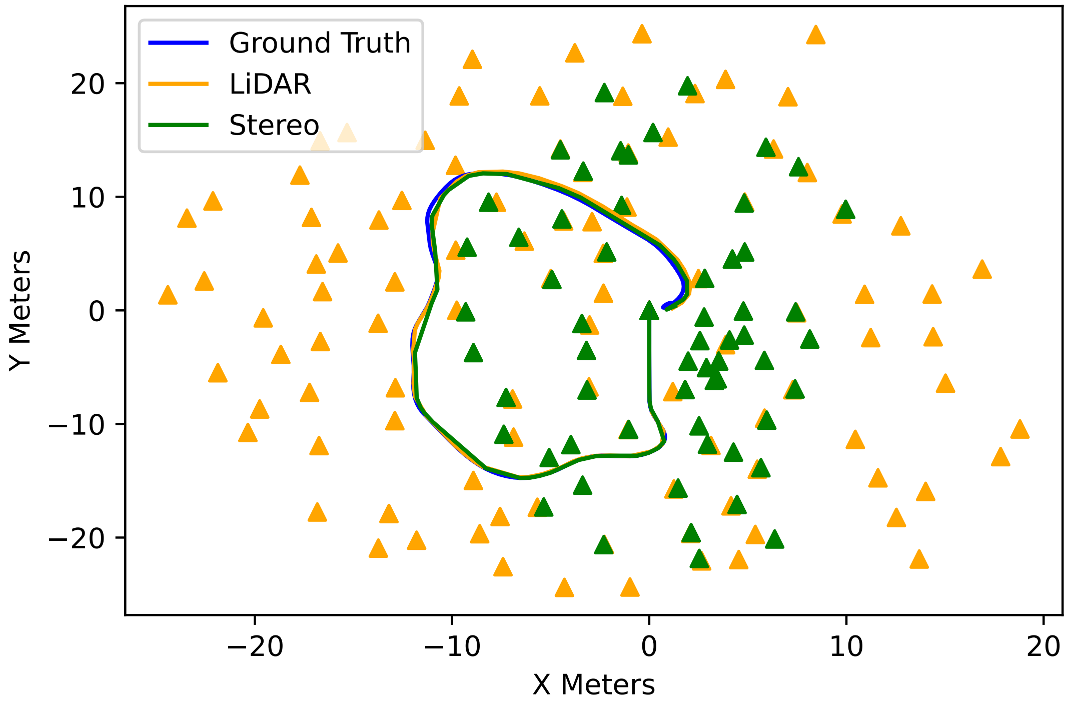
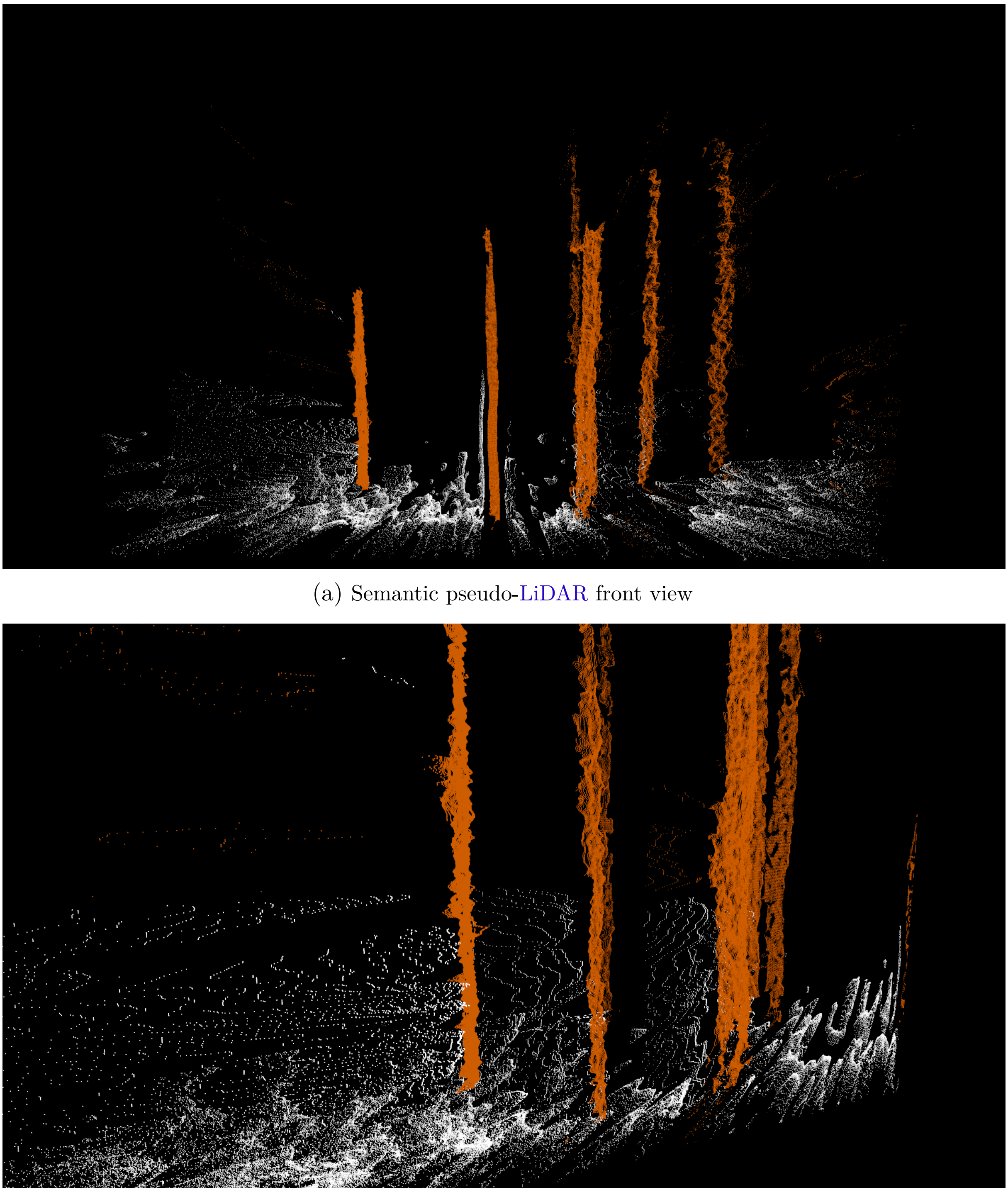


Figure 25 – A loop trajectory in simulation using our proposed model as source of semantic pseudo-LiDAR compared with a simulated LiDAR sensor. Both sensor sources as integrated with our factor graph formulation for semantic SLAM. We observe that the trajectories are similar, but the number of trees observed with the camera is much smaller. This is expected since the field of view of the LiDAR is larger.

The stereo system captures reliable depth up to 5 meters while the LiDARs captures a 25 meters radius in this experiment. On the other hand, cameras have a larger vertical field of view and capture texture and color information. This information could enable other tasks, such as species recognition, which would be hard to do with LiDAR point clouds.

6.5 Final Considerations

LiDARs capture large amounts of 3D information but are heavy, expensive, and do not capture the texture and color of the images. In this chapter, we proposed a neural network that simultaneously estimates disparity and a semantic mask. To achieve real-time performance, we make several speed-accuracy trade-offs. First, we use a single model to learn related tasks, which is known to impact the performance of the model when compared to specialized models. Also, we show that there is an accuracy and speed trade-off between the iterative disparity computation approach and aggregating features to compute a single disparity. The results indicate that we can learn features for both tasks by training the proposed neural network architecture on a mixture of supervised learning for semantic segmentation and self-supervised for disparity estimation, saving



(a) Semantic pseudo-LiDAR front view

(b) Semantic pseudo-LiDAR side view

Figure 26 – Tree trunk point cloud (brown) and ground point cloud (white) derived from a stereo image pair using the proposed CNN model.

computational resources for downstream tasks while maintaining acceptable performance for our end goal of semantic mapping in forests.

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

The preservation of forests is essential for the maintenance of the Earth's climate and biodiversity. In this context, multiple stakeholders seek solutions to accelerate and improve what we know about these environments. More data about forests can help governments define policies and laws for preservation. Local communities can learn more about possible sustainable economic activities derived from the rich diversity of plants available in the forest. Companies can compensate for their emissions by incentivizing governments and forest owners to preserve their areas by investing in carbon credit. For all of these activities to happen, it is necessary to know what is inside a forest, and autonomous robots carrying sensors could be a game changer.

In this thesis, we proposed different methods that run onboard UAVs to gather valuable information about forests. In Chapter 4, we introduced SLOAM, a framework that leverages semantic information to increase the robustness of localization and mapping under forest canopy without GNSS. While semantic features and models increase the robustness of state estimation and quality of forest maps, we argue that to cover large regions of a forest, a robot must consider multiple sensing sources to be more robust to the different types of failure. To this end, we presented a factor graph formulation of SLOAM, capable of integrating external odometry estimates, semantic landmarks and loop closure constraints.

Analogously most of the state estimation algorithms, the SLOAM computes its estimates by associating data between subsequent sensor measurements. This process is subject to error that is accumulated over time. Adding loop closure constraints to our factor graph can help mitigate this. Since GNSS is unreliable under the dense forest canopy, and different regions of the forest look very similar, we proposed in Chapter 5 a novel approach for "fingerprinting" different regions of a forest using only SLOAM's semantic map to derive the position of trees. Our method defines unique geometric shapes by combining local regions of trees inside one or more sensor observations. We show that even under

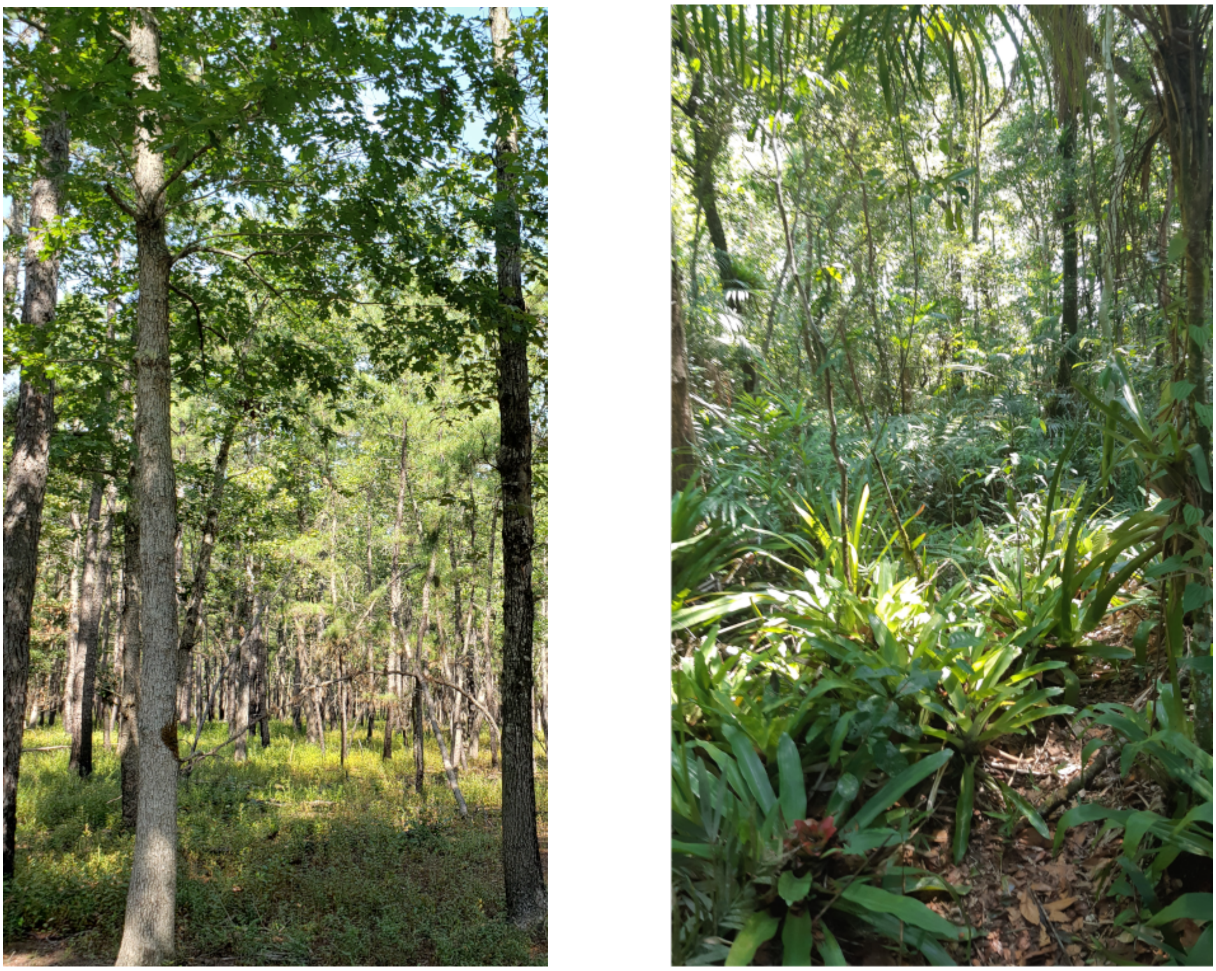


Figure 27 – Example of a Pine forest (left) where most of our experiments were performed, and Sub-tropical forest (right) where we believe future work should strive for. Tropical forests are denser and more diverse, creating additional challenges and opportunities for future research.

significant position or landmark detection perturbations, our method can reliably detect previously seen regions.

There are many variations of forests, some are homogeneous and sparse while others are diverse and dense. In Figure 27 we illustrate a Pine and Tropical forest. Most experiments presented in this thesis considered homogeneous forests such as Pine or Eucalyptus. Even if these environments are relatively simpler when compared to tropical forests, these forests are very challenging for an autonomous robot, and systems that navigate in any forests do not exist. While our motivation for the algorithms proposed in this thesis is to have them work in any type of forests, there are many gaps that have to be addressed before deploying UAVs in Rainforests. In Chapter 6 we presented a neural network that estimates depth and semantic labels given a pair of stereo rectified images. While there is a clear trade-off between LiDAR and cameras, the latter are cheaper and lightweight, which could enable smaller and more scalable platforms for forest mapping in the future.

Another direction for future work is related to extracting semantic information from sensor data. The cornerstone of this thesis is semantic data. However, to extract this information, all methods proposed in this thesis belong to the supervised learning paradigm. They require labeled data during the training phase before being deployed to identify the objects of interest. This approach requires labeled data during the training phase. While there are many datasets for urban navigation with labeled data, there are none for forestry. For this reason, much effort was put into data labeling during the development of this work. However, there is a considerable context change for the models trained on Pine forests data to Rainforests, making it only possible to reutilize the models by retraining with new examples in the new domain. To increase the usability of our methods, future work could explore unsupervised or few-shot learning methods to reduce the dependency on large labeled datasets. For example, the segmentation models could explore ways to incorporate the geometric priors, such as the tree cylinder models, to its loss function to learn with weak supervision.

While SLOAM was designed for forestry applications, the factor graph formulation presented in this thesis is modular and generalizable for other domains. The work presented in (LIU *et al.*, 2022b) builds on our formulation to incorporate 3D bounding boxes to capture objects such as cars in urban settings. Future works could explore using more general geometric priors for arbitrary objects or incorporating other sensors into the state estimation framework to increase robustness.

Finally, another promising direction for future work is to deploy teams of robots to improve the scalability of our methods. Since forests are enormous, given the limited autonomy of UAVs, it would be impossible to cover the entire area with one robot in a feasible time. For this reason, strategies for multi-robot collaboration would be essential to make these systems viable in the real world.

BIBLIOGRAPHY

ACKERMANN, J.; GOESELE, M. A survey of photometric stereo techniques. **Foundations and Trends® in Computer Graphics and Vision**, Now Publishers Inc. Hanover, MA, USA, v. 9, n. 3-4, p. 149–254, 2015. Citation on page 38.

ALMEIDA, D. d.; BROADBENT, E. N.; ZAMBRANO, A. M. A.; WILKINSON, B. E.; FERREIRA, M. E.; CHAZDON, R.; MELI, P.; GORGENS, E.; SILVA, C. A.; STARK, S. C. *et al.* Monitoring the structure of forest restoration plantations with a drone-lidar system. **International Journal of Applied Earth Observation and Geoinformation**, Elsevier, v. 79, p. 192–198, 2019. Citation on page 23.

AULINAS, J.; PETILLOT, Y.; SALVI, J.; LLADÓ, X. The slam problem: a survey. **Artificial Intelligence Research and Development**, IOS Press, p. 363–371, 2008. Citation on page 42.

BOWMAN, S. L.; ATANASOV, N.; DANIILIDIS, K.; PAPPAS, G. J. Probabilistic data association for semantic slam. In: IEEE. **2017 IEEE international conference on robotics and automation (ICRA)**. [S.l.], 2017. p. 1722–1729. Citations on pages 30 and 31.

BOYD, D.; DANSON, F. Satellite remote sensing of forest resources: three decades of research development. **Progress in Physical Geography**, Sage Publications Sage CA: Thousand Oaks, CA, v. 29, n. 1, p. 1–26, 2005. Citation on page 23.

BRIDSON, R. Fast poisson disk sampling in arbitrary dimensions. **SIGGRAPH sketches**, v. 10, p. 1278780–1278807, 2007. Citation on page 68.

CADENA, C.; CARLONE, L.; CARRILLO, H.; LATIF, Y.; SCARAMUZZA, D.; NEIRA, J.; REID, I.; LEONARD, J. J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. **IEEE Transactions on robotics**, IEEE, v. 32, n. 6, p. 1309–1332, 2016. Citation on page 27.

CARREIRAS, J.; MELO, J. B.; VASCONCELOS, M. J. Estimating the above-ground biomass in miombo savanna woodlands (mozambique, east africa) using l-band synthetic aperture radar data. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 5, n. 4, p. 1524–1548, 2013. Citation on page 37.

CHANG, J.-R.; CHANG, P.-C.; CHEN, Y.-S. Attention-aware feature aggregation for real-time stereo matching on edge devices. In: **Proceedings of the Asian Conference on Computer Vision (ACCV)**. [S.l.: s.n.], 2020. Citations on pages 77 and 78.

CHANG, J.-R.; CHEN, Y.-S. Pyramid stereo matching network. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 5410–5418. Citations on pages 75 and 78.

CHEBROLU, N.; LOTTES, P.; LÄBE, T.; STACHNISS, C. Robot localization based on aerial images for precision agriculture tasks in crop fields. In: IEEE. **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.], 2019. p. 1787–1793. Citations on pages 30 and 33.

CHEN, S.; XIANG, Z.; QIAO, C.; CHEN, Y.; BAI, T. Sgnet: Semantics guided deep stereo matching. In: **Proceedings of the Asian Conference on Computer Vision**. [S.l.: s.n.], 2020. Citation on page 76.

CHEN, S. W.; NARDARI, G. V.; LEE, E. S.; QU, C.; LIU, X.; ROMERO, R. A. F.; KUMAR, V. Sloam: Semantic lidar odometry and mapping for forest inventory. **IEEE Robotics and Automation Letters**, IEEE, v. 5, n. 2, p. 612–619, 2020. Citations on pages 29 and 45.

CHEN, X.; MILIOTO, A.; PALAZZOLO, E.; GIGUERE, P.; BEHLEY, J.; STACHNISS, C. Suma++: Efficient lidar-based semantic slam. In: IEEE. **2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2019. p. 4530–4537. Citations on pages 29 and 31.

CHEN, Y.; ZHENG, B.; ZHANG, Z.; WANG, Q.; SHEN, C.; ZHANG, Q. Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 53, n. 4, p. 1–37, 2020. Citation on page 28.

CONTO, T. de; OLOFSSON, K.; GÖRGENS, E. B.; RODRIGUEZ, L. C. E.; ALMEIDA, G. Performance of stem denoising and stem modelling algorithms on single tree point clouds from terrestrial laser scanning. **Computers and Electronics in Agriculture**, Elsevier, v. 143, n. November, p. 165–176, 2017. ISSN 01681699. Citation on page 23.

DELAUNAY, B. *et al.* Sur la sphere vide. **Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk**, v. 7, n. 793-800, p. 1–2, 1934. Citations on pages 62 and 64.

DELLAERT, F. **Factor graphs and GTSAM: A hands-on introduction**. [S.l.], 2012. Citation on page 58.

DELLAERT, F.; KAESS, M. *et al.* Factor graphs for robot perception. **Foundations and Trends® in Robotics**, Now Publishers, Inc., v. 6, n. 1-2, p. 1–139, 2017. Citation on page 59.

DOSOVITSKIY, A.; FISCHER, P.; ILG, E.; HAUSSER, P.; HAZIRBAS, C.; GOLKOV, V.; SMAGT, P. V. D.; CREMERS, D.; BROX, T. Flownet: Learning optical flow with convolutional networks. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 2758–2766. Citation on page 78.

DOVESI, P. L.; POGGI, M.; ANDRAGHETTI, L.; MARTÍ, M.; KJELLSTRÖM, H.; PIEROPAN, A.; MATTOCCIA, S. Real-time semantic stereo matching. In: IEEE. **2020 IEEE international conference on robotics and automation (ICRA)**. [S.l.], 2020. p. 10780–10787. Citation on page 76.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Kdd**. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231. Citation on page 70.

FORNEY, G. D. The viterbi algorithm. **Proceedings of the IEEE**, IEEE, v. 61, n. 3, p. 268–278, 1973. Citation on page 48.

GARG, S.; SUENDERHAUF, N.; MILFORD, M. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. **Proceedings of Robotics: Science and Systems XIV**, 2018. Citations on pages 30 and 33.

GAWEL, A.; DON, C. D.; SIEGWART, R.; NIETO, J.; CADENA, C. X-view: Graph-based semantic multi-view localization. **IEEE Robotics and Automation Letters**, IEEE, v. 3, n. 3, p. 1687–1694, 2018. Citations on pages 32, 33, and 69.

GEIGER, A.; LENZ, P.; STILLER, C.; URTASUN, R. Vision meets robotics: The kitti dataset. **The International Journal of Robotics Research**, Sage Publications Sage UK: London, England, v. 32, n. 11, p. 1231–1237, 2013. Citation on page 81.

GODARD, C.; AODHA, O. M.; BROSTOW, G. J. Unsupervised monocular depth estimation with left-right consistency. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 270–279. Citation on page 79.

GODARD, C.; AODHA, O. M.; FIRMAN, M.; BROSTOW, G. J. Digging into self-supervised monocular depth estimation. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 3828–3838. Citation on page 79.

HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. [S.l.]: Cambridge university press, 2003. Citation on page 37.

HEISE, P.; KLOSE, S.; JENSEN, B.; KNOLL, A. Pm-huber: Patchmatch with huber regularization for stereo matching. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2013. p. 2360–2367. Citation on page 79.

HIMSTEDT, M.; FROST, J.; HELLBACH, S.; BÖHME, H.-J.; MAEHLE, E. Large scale place recognition in 2d lidar scans using geometrical landmark relations. In: IEEE. **2014 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.], 2014. p. 5030–5035. Citation on page 32.

HOWARD, A.; SANDLER, M.; CHU, G.; CHEN, L.-C.; CHEN, B.; TAN, M.; WANG, W.; ZHU, Y.; PANG, R.; VASUDEVAN, V. *et al.* Searching for mobilenetv3. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 1314–1324. Citation on page 77.

HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 7132–7141. Citation on page 77.

KAESS, M.; JOHANNSSON, H.; ROBERTS, R.; ILA, V.; LEONARD, J. J.; DELLAERT, F. isam2: Incremental smoothing and mapping using the bayes tree. **The International Journal of Robotics Research**, Sage Publications Sage UK: London, England, v. 31, n. 2, p. 216–235, 2012. Citations on pages 42 and 58.

KALLASI, F.; RIZZINI, D. L. Efficient loop closure based on falko lidar features for online robot localization and mapping. In: IEEE. **2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2016. p. 1206–1213. Citations on pages 32, 33, and 66.

KENDALL, A.; GAL, Y.; CIPOLLA, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 7482–7491. Citation on page 80.

KENDALL, A.; MARTIROSYAN, H.; DASGUPTA, S.; HENRY, P.; KENNEDY, R.; BACHRACH, A.; BRY, A. End-to-end learning of geometry and context for deep stereo regression. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2017. p. 66–75. Citations on pages 75 and 78.

KIRILLOV, A.; HE, K.; GIRSHICK, R.; ROTHER, C.; DOLLÁR, P. Panoptic segmentation. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 9404–9413. Citation on page 40.

LI, Q.; NEVALAINEN, P.; QUERALTA, J. P.; HEIKKONEN, J.; WESTERLUND, T. Localization in unstructured environments: Towards autonomous robots in forests with delaunay triangulation. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 12, n. 11, p. 1870, 2020. Citations on pages 32, 33, and 63.

LIN, S.; WANG, J.; XU, M.; ZHAO, H.; CHEN, Z. Topology aware object-level semantic mapping towards more robust loop closure. **IEEE Robotics and Automation Letters**, IEEE, v. 6, n. 4, p. 7041–7048, 2021. Citations on pages 32 and 33.

LIU, X.; NARDARI, G. V.; OJEDA, F. C.; TAO, Y.; ZHOU, A.; DONNELLY, T.; QU, C.; CHEN, S. W.; ROMERO, R. A. F.; TAYLOR, C. J. *et al.* Large-scale autonomous flight with real-time semantic slam under dense forest canopy. **IEEE Robotics and Automation Letters**, IEEE, 2022. Citations on pages 13, 29, 40, 45, 56, and 80.

LIU, X.; PRABHU, A.; CLADERA, F.; MILLER, I. D.; ZHOU, L.; TAYLOR, C. J.; KUMAR, V. Active metric-semantic mapping by multiple aerial robots. **arXiv preprint arXiv:2209.08465**, 2022. Citations on pages 60 and 89.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, Springer, v. 60, n. 2, p. 91–110, 2004. Citation on page 30.

MILIOTO, A.; VIZZO, I.; BEHLEY, J.; STACHNISS, C. Rangenet++: Fast and accurate lidar semantic segmentation. In: IEEE. **2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2019. p. 4213–4220. Citation on page 47.

MILLER, I. D.; COWLEY, A.; KONKIMALLA, R.; SHIVAKUMAR, S. S.; NGUYEN, T.; SMITH, T.; TAYLOR, C. J.; KUMAR, V. Any way you look at it: Semantic crossview localization and mapping with lidar. **IEEE Robotics and Automation Letters**, IEEE, v. 6, n. 2, p. 2397–2404, 2021. Citations on pages 30 and 33.

MIRANDA, A.; CATALÁN, G.; ALTAMIRANO, A.; ZAMORANO-ELGUETA, C.; CAVIERES, M.; GUERRA, J.; MOLA-YUDEGO, B. How much can we see from a uav-mounted regular camera? remote sensing-based estimation of forest attributes in south american native forests. **Remote Sensing**, MDPI, v. 13, n. 11, p. 2151, 2021. Citation on page 23.

MUR-ARTAL, R.; MONTIEL, J. M. M.; TARDOS, J. D. Orb-slam: a versatile and accurate monocular slam system. **IEEE transactions on robotics**, IEEE, v. 31, n. 5, p. 1147–1163, 2015. Citations on pages 27 and 31.

NAKAMURA, A. T. M.; GRASSI JR, V.; WOLF, D. F. Leveraging convergence behavior to balance conflicting tasks in multi-task learning. **Neurocomputing**, Elsevier, v. 511, p. 43–53, 2022. Citation on page 80.

NARDARI, G. V.; COHEN, A.; CHEN, S. W.; LIU, X.; ARCOT, V.; ROMERO, R. A.; KUMAR, V. Place recognition in forests with urquhart tessellations. **IEEE Robotics and Automation Letters**, IEEE, v. 6, n. 2, p. 279–286, 2020. Citation on page 69.

NICHOLSON, L.; MILFORD, M.; SÜNDERHAUF, N. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. **IEEE Robotics and Automation Letters**, IEEE, v. 4, n. 1, p. 1–8, 2018. Citations on pages 29 and 31.

OLSON, E.; LEONARD, J.; TELLER, S. Fast iterative alignment of pose graphs with poor initial estimates. In: IEEE. **Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.** [S.l.], 2006. p. 2262–2269. Citation on page 42.

PON, A. D.; KU, J.; LI, C.; WASLANDER, S. L. Object-centric stereo matching for 3d object detection. In: IEEE. **2020 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2020. p. 8383–8389. Citation on page 38.

QI, C. R.; YI, L.; SU, H.; GUIBAS, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2017. p. 5099–5108. Citation on page 47.

QIAN, Z.; PATATH, K.; FU, J.; XIAO, J. Semantic slam with autonomous object-level data association. In: IEEE. **2021 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2021. p. 11203–11209. Citation on page 31.

QIN, T.; LI, P.; SHEN, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. **IEEE Transactions on Robotics**, IEEE, v. 34, n. 4, p. 1004–1020, 2018. Citation on page 27.

ROMERA, E.; ALVAREZ, J. M.; BERGASA, L. M.; ARROYO, R. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 19, n. 1, p. 263–272, 2017. Citation on page 47.

RORIZ, R.; CABRAL, J.; GOMES, T. Automotive lidar technology: A survey. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, 2021. Citation on page 36.

ROSINOL, A.; ABATE, M.; CHANG, Y.; CARLONE, L. Kimera: an open-source library for real-time metric-semantic localization and mapping. In: IEEE. **2020 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2020. p. 1689–1696. Citations on pages 27, 29, and 31.

RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. Orb: An efficient alternative to sift or surf. In: IEEE. **2011 International conference on computer vision**. [S.l.], 2011. p. 2564–2571. Citation on page 30.

SCHWARTING, W.; ALONSO-MORA, J.; RUS, D. Planning and decision-making for autonomous vehicles. **Annual Review of Control, Robotics, and Autonomous Systems**, Annual Reviews, v. 1, p. 187–210, 2018. Citation on page 29.

SEGAL, A.; HAEHNEL, D.; THRUN, S. Generalized-icp. In: SEATTLE, WA. **Robotics: science and systems**. [S.l.], 2009. v. 2, n. 4, p. 435. Citation on page 51.

SHAN, T.; ENGLOT, B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: IEEE. **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2018. p. 4758–4765. Citations on pages 28, 31, and 56.

SHAN, T.; ENGLOT, B.; MEYERS, D.; WANG, W.; RATTI, C.; RUS, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In: IEEE. **2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2020. p. 5135–5142. Citations on pages 28, 31, 40, and 59.

SUN, K.; MOHTA, K.; PFROMMER, B.; WATTERSON, M.; LIU, S.; MULGAONKAR, Y.; TAYLOR, C. J.; KUMAR, V. Robust stereo visual inertial odometry for fast autonomous flight. **IEEE Robotics and Automation Letters**, IEEE, v. 3, n. 2, p. 965–972, 2018. Citations on pages 38 and 55.

THRUN, S.; MONTEMERLO, M. The graph slam algorithm with applications to large-scale mapping of urban structures. **The International Journal of Robotics Research**, SAGE Publications, v. 25, n. 5-6, p. 403–429, 2006. Citation on page 42.

URQUHART, R. Algorithms for computation of relative neighbourhood graph. **Electronics Letters**, IET, v. 16, n. 14, p. 556–557, 1980. Citation on page 63.

VECHERSKY, P.; COX, M.; BORGES, P.; LOWE, T. Colourising point clouds using independent cameras. **IEEE Robotics and Automation Letters**, IEEE, v. 3, n. 4, p. 3575–3582, 2018. Citation on page 37.

VOULODIMOS, A.; DOULAMIS, N.; DOULAMIS, A.; PROTOPAPADAKIS, E. Deep learning for computer vision: A brief review. **Computational intelligence and neuroscience**, Hindawi, v. 2018, 2018. Citation on page 28.

WANG, C.; ZENG, Y.; SIMON, L.; KAKADIARIS, I.; SAMARAS, D.; PARAGIOS, N. Viewpoint invariant 3d landmark model inference from monocular 2d images using higher-order priors. In: IEEE. **2011 International Conference on Computer Vision**. [S.l.], 2011. p. 319–326. Citation on page 68.

WANG, Y.; CHAO, W.-L.; GARG, D.; HARIHARAN, B.; CAMPBELL, M.; WEINBERGER, K. Q. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 8445–8453. Citation on page 82.

WANG, Y.; LAI, Z.; HUANG, G.; WANG, B. H.; MAATEN, L. V. D.; CAMPBELL, M.; WEINBERGER, K. Q. Anytime stereo image depth estimation on mobile devices. In: IEEE. **2019 international conference on robotics and automation (ICRA)**. [S.l.], 2019. p. 5893–5900. Citations on pages 78 and 80.

WANG, Z.; BOVIK, A. C.; SHEIKH, H. R.; SIMONCELLI, E. P. Image quality assessment: from error visibility to structural similarity. **IEEE transactions on image processing**, IEEE, v. 13, n. 4, p. 600–612, 2004. Citation on page 79.

WU, Z.; WU, X.; ZHANG, X.; WANG, S.; JU, L. Semantic stereo matching with pyramid cost volumes. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 7484–7493. Citation on page 76.

XU, H.; ZHANG, J. Aanet: Adaptive aggregation network for efficient stereo matching. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 1959–1968. Citations on pages 38, 77, and 80.

XU, W.; ZHANG, F. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. **IEEE Robotics and Automation Letters**, IEEE, v. 6, n. 2, p. 3317–3324, 2021. Citation on page 40.

YANG, G.; ZHAO, H.; SHI, J.; DENG, Z.; JIA, J. Segstereo: Exploiting semantic information for disparity estimation. In: **Proceedings of the European conference on computer vision (ECCV)**. [S.l.: s.n.], 2018. p. 636–651. Citation on page 75.

YANG, S.; SCHERER, S. Cubeslam: Monocular 3-d object slam. **IEEE Transactions on Robotics**, IEEE, v. 35, n. 4, p. 925–938, 2019. Citations on pages 29 and 31.

ZHANG, D.; LU, G. *et al.* A comparative study on shape retrieval using fourier descriptors with different shape signatures. In: **Proc. of international conference on intelligent multimedia and distance education (ICIMADE01)**. [S.l.: s.n.], 2001. p. 1–9. Citations on pages 65 and 66.

ZHANG, F.; QI, X.; YANG, R.; PRISACARIU, V.; WAH, B.; TORR, P. Domain-invariant stereo matching networks. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2020. p. 420–439. Citation on page 78.

ZHANG, J.; SINGH, S. Loam: Lidar odometry and mapping in real-time. In: BERKELEY, CA. **Robotics: Science and Systems**. [S.l.], 2014. v. 2, n. 9, p. 1–9. Citations on pages 27, 31, 40, and 50.

ZHANG, J.; SKINNER, K. A.; VASUDEVAN, R.; JOHNSON-ROBERSON, M. Dispsegnet: Leveraging semantics for end-to-end learning of disparity estimation from stereo imagery. **IEEE Robotics and Automation Letters**, IEEE, v. 4, n. 2, p. 1162–1169, 2019. Citation on page 76.

ZHANG, K.; LI, Z.; MA, J. Appearance-based loop closure detection via bidirectional manifold representation consensus. In: IEEE. **2021 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2021. p. 6811–6817. Citation on page 33.

ZHENG, J.; WANG, Y.; NIHAN, N. L. Quantitative evaluation of gps performance under forest canopies. In: IEEE. **Proceedings. 2005 IEEE Networking, Sensing and Control, 2005.** [S.l.], 2005. p. 777–782. Citation on page 37.

GLOSSARY

CNN Convolutional Neural Network.

DBH Diameter at Breast Height.

DBSCAN Density Based Spatial Clustering of Application with Noise.

DFT Discrete Fourier Transform.

DGPS Differential GPS.

EKF Extended Kalman Filter.

EM expectation maximization.

FCN Fully Convolutional Network.

GICP Generalized Iterative Closest Point.

GNSS Global Navigation Satellite Systems.

GPU Graphics Processing Unit.

IMU Inertial Measurement Unit.

LiDAR Light Detection and Ranging.

LOAM LiDAR Odometry and Mapping.

MAP maximum a posteriori estimation.

OVC Open Vision Computer.

RANSAC RANdom SAmple Consensus.

RTK Real-Time Kinematics.

SaM Smoothing and Mapping.

SLAM Simultaneous Localization and Mapping.

SLOAM Semantic LiDAR Odometry and Mapping.

TLS Terrestrial Laser Scanning.

UAV Unmanned Aerial Vehicle.

UGV Unmanned Ground Vehicle.

VIO visual-inertial odometry.

APPENDIX A

PUBLICATIONS AND OUTREACH

First Author or Equal Contribution

- Chen, S. W.*, Nardari, G. V.*, Lee, E. S., Qu, C., Liu, X., Romero, R. A. F., and Kumar, V. (2020). Sloam: Semantic lidar odometry and mapping for forest inventory. IEEE Robotics and Automation Letters, 5(2), 612-619. **Presented at ICRA 2020.**
- Nardari, G. V., Cohen, A., Chen, S. W., Liu, X., Arcot, V., Romero, R. A., and Kumar, V. (2020). Place recognition in forests with Urquhart tessellations. IEEE Robotics and Automation Letters, 6(2), 279-286. **Presented at ICRA 2021.**
- Liu, X.*, Nardari, G. V.*, Ojeda, F. C., Tao, Y., Zhou, A., Donnelly, T., Qu, C., Chen, S. W., Romero, R. A. F., Taylor, C. J., and Kumar, V. (2022). Large-scale autonomous flight with real-time semantic slam under dense forest canopy. IEEE Robotics and Automation Letters, 7(2), 5512-5519. **Presented at ICRA 2022.**
- Nardari, G. V., Romero, R. A., Guizilini, V. C., Mareco, W. E., Milori, D. M., Villas-Boas, P. R., and Santos, I. A. D. (2018, November). Crop anomaly identification with color filters and convolutional neural networks. In 2018 Latin American Robotic Symposium (LARS) - (pp. 363-369). IEEE.

Other Publications

- Liu, X., Chen, S. W., Nardari, G. V., Qu, C., Ojeda, F. C., Taylor, C. J., and Kumar, V. (2022). Challenges and Opportunities for Autonomous Micro-UAVs in Precision Agriculture. IEEE Micro, 42(1), 61-68.
- Ranieri, C. M., Nardari, G. V., Pinto, A. H. M., Tozadore, D. C., and Romero, R. A. F. (2018, November). LARa: A robotic framework for human-robot interaction

on indoor environments. In 2018 Latin American Robotic Symposium (LARS) - (pp. 376-382). IEEE.

- Tozadore, D. C., Ranieri, C. M., Nardari, G. V., Romero, R. A., and Guizilini, V. C. (2018, October). Effects of emotion grouping for recognition in human-robot interactions. In 2018 7th Brazilian Conference on Intelligent Systems (BRACIS) - (pp. 438-443). IEEE.

Scientific Outreach

Television

- Bom dia Brasil (Globo)
- SBT Brasil (SBT)
- Jornal regional (Record Ribeirão Preto)

Websites

- Agência FAPESP
- UOL
- Jornal da USP
- G1 São Carlos
- Rede Brasil Atual
- A Cidade ON
- EESC USP

Magazines

- Revista FAPESP
- Qualé

ICMC USP
SÃO CARLOS