Efficient and Elastic LiDAR Reconstruction for Large-Scale Exploration Tasks

Yiduo Wang

St Cross College

University of Oxford

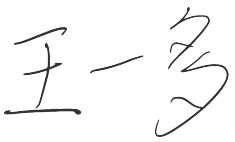
A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2022

Declaration

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.



Yiduo Wang,
St Cross College

Acknowledgements

I would like to express my sincerest gratitude towards my supervisor, Professor Maurice Fallon, for his support, guidance, leadership and constant encouragement throughout my DPhil. His expertise in the field, his dedication to research, and his pursuit of perfection have not only guided me but also inspired me in the last four years. It has been a truly incredible experience working as one of his students.

I would also like to thank Dr Milad Ramezani and Dr Sundara Tejaswi Digumarti for their support and inspiration through this DPhil project, both professionally and personally. Their knowledge, advice, and sharing the stress of programming, debugging and late night paper writing are the boost that pushes this DPhil project over the finishing line. Furthermore, thank you to my colleague Matias Mattamala for working together with me on mathematics and publications, as well as for being a good friend.

I am grateful to the Dynamics Robot Systems group and the Oxford Robotics Institute as a whole, for the amazing research environment and DPhil experience. I am also grateful to my college advisor Dr Timothy Pound for supporting and encouraging me throughout my DPhil, and advising me on how to be a better DPhil researcher.

I would like to gratefully acknowledge the collaborations we had with researchers from around the world. In particular, I would like to thank Professor Stefan Leutenegger, the Smart Robotics Lab led by him, and his students Nils Funk and Sotiris Papatheodorou for designing and developing essential components of the proposed system of this DPhil project.

I would like to further thank my examiners, Prof Nick Hawes and Prof Margarita Chli, for taking time out of their busy schedules reviewing my thesis, providing feedback and participating in my viva examination. Their professional insight and expertise in both the field of robotics as well as thesis writing are immensely valuable.

Last but not least, I would like to express my forever appreciation to my parents for their unconditional love and support. I would like to thank all my friends for their company through this journey. In particular, I am truly grateful to my friend, Oliwier Melon, for being a good colleague and a even better friend, for advising me on how to improve my writing and presentation skills, and for helping me when I suffered from COVID symptoms.

Abstract

High-quality reconstructions and understanding the environment are essential for robotic tasks such as localisation, navigation and exploration. Applications like planners and controllers can make decisions based on them. International competitions such as the DARPA Subterranean Challenge demonstrate the difficulties that reconstruction methods must address in the real world, e.g. complex surfaces in unstructured environments, accumulation of localisation errors in long-term explorations, and the necessity for methods to be scalable and efficient in large-scale scenarios.

Guided by these motivations, this thesis presents a multi-resolution volumetric reconstruction system, *supereight*-Atlas (SE-Atlas). SE-Atlas efficiently integrates long-range LiDAR scans with high resolution, incorporates motion undistortion, and employs an Atlas of submaps to produce an elastic 3D reconstruction.

These features address limitations of conventional reconstruction techniques that were revealed in real-world experiments of an initial active perceptual planning prototype. Our experiments with SE-Atlas show that it can integrate LiDAR scans at $60\,\mathrm{m}$ range with $\sim 5\,\mathrm{cm}$ resolution at $\sim 3\,\mathrm{Hz}$, outperforming state-of-the-art methods in integration speed and memory efficiency. Reconstruction accuracy evaluation also proves that SE-Atlas can correct the map upon SLAM loop closure corrections, maintaining global consistency.

We further propose four principled strategies for spawning and fusing submaps. Based on spatial analysis, SE-Atlas spawns new submaps when the robot transitions into an isolated space, and fuses submaps of the same space together. We focused on developing a system which scales against environment size instead of exploration length. A new formulation is proposed to compute relative uncertainties between poses in a SLAM pose graph, improving submap fusion reliability. Our experiments show that the average error in a large-scale map is approximately $5\,\mathrm{cm}$.

A further contribution was incorporating semantic information into SE-Atlas. A recursive Bayesian filter is used to maintain consistency in per-voxel semantic labels. Semantics is leveraged to detect indoor-outdoor transitions and adjust reconstruction parameters online.

Contents

List of Figures

List of Tables

List of Abbreviations

AABB Axis-Aligned Bounding Box

AEROS AdaptivE RObust least-Squares

AGP Art Gallery Problem

BA Bundle Adjustment

BCH Baker-Campbell-Hausdorff

BIM Building Information Model

CAD Computer-Aided Design

CNN Convolutional Neural Network

CoS Confirmation of Status

DARPA Defense Advanced Research Projects Agency

DOF Degree of Freedom

DPhil Doctor of Philosophy

DRS Dynamic Robot Systems

DRS Group Dynamic Robot Systems Group

DSM Digital Surface Model

ECMR European Conference on Mobile Robots

EKF Extended Kalman Filter

ESDF Euclidean Signed Distance Function

ESM Efficient Segmentation and Mapping

ETH Zurich Eidgenössische Technische Hochschule Zurich

FCN Fully Convolutional Network

FoV Field of View

GNC Graduated Non-Convexity

HVG Hashing Voxel Grid

ICL Imperial College London

ICP Iterative Closest Point

ICRA International Conference on Robotics and Automation

IEEE Institute of Electrical and Electronics Engineers

IEEE/RSJ Institute of Electrical and Electronics Engineers/Robotics Society of Japan

IMU Inertial Measurement Unit

IRLS Iteratively Reweighted Least Squares

IROS International Conference on Intelligent Robots and Systems

iSAM incremental Smoothing and Mapping

KPConv Kernel Point Convolution

LiDAR Light Detection And Ranging

LOAM LiDAR Odometry And Mapping

MAV Micro Aerial Vehicle

MIT Massachusetts Institute of Technology

MPLS Mathematical Physical and Life Sciences Department

MSCKF Multi-State Constraint Kalman Filter

NBV Next-Best-View

NCD Newer College Dataset

OCEKF Observability Constrained Extended Kalman Filter

ORCA Offshore Robotics for Certification of Assets

ORI Oxford Robotics Institute

RA-L Robotics and Automation Letters

RADAR Radio Detection And Ranging

RAS Robotics and Automation Society

RGB-D Red Green Blue Depth

ROS Robotic Operating System

RRT Rapidly-exploring Random Tree

RSJ Robotics Society of Japan

SAM Smoothing and Mapping

SDF Signed Distance Function

SE-Atlas *supereight* Atlas

SfM Structure from Motion

SLAM Simultaneous Localisation And Mapping

SPLAM Simultaneous Planning, Localisation, And Mapping

SRL Smart Robotics Lab

SubT Subterranean

TSDF Truncated Signed Distance Function

TSIF Two State Implicit Filter

TSP Travelling Salesman Problem

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

UKF Unscented Kalman Filter

V4RL Vision For Robotics Lab

VILENS Visual Inertial Lidar/LEgged Navigation System

VIO Visual Inertial Odometry

VO Visual Odometry

VTOL Vertical Take Off and Landing

List of Symbols

Reference Frame

Symbol	Description
$\mathcal{B}$	Robot Base frame
$\mathcal{L}$	LiDAR frame
$\mathcal{M}$	Map frame
$\mathcal{O}$	Odometry (Odom) frame
$\mathcal{S}$	Submap frame

Estimations and Measurements

Symbol	Description	
ρ	2D pixel corridinate	$\mathbb{R}^2$
$\mathbf{p}$	Single point in point cloud	$\mathbb{R}^3$
$\mathbf{C}$	Point cloud — a set of $\mathbf{p}$	
$\mathbf{x}$	Single robot pose	$\mathbf{SE}(3)$
$\mathbf{X}$	Robot pose set — a set of $\mathbf{x}$	
$\mathbf{r}$	Single ray in ray-casting	$\mathbb{R}^3$
$\mathbf{R}$	Ray set in ray-casting — a set of $\mathbf{r}$	
l	Single semantic class	
$\mathbf{I}$	All the semantic information within an image	
Z	Single pixel semantic observation	
$\mathbf{v}$	Single voxel	$\mathbb{R}^3$
$\mathbf{V}$	Voxel set — a set of $\mathbf{v}$	

Variables

Symbol	Description	
$\mathbf{R}$	Rotation	$\mathbf{SO}(3)$
$\mathbf{T}$	Transformation	$\mathbf{SE}(3)$
$\mathbf{t}$	Translation	$\mathbb{R}^3$
P	Probability	$\mathbb{R}$
H	Entropy based on P	$\mathbb{R}$
L	Log-odds of P	$\mathbb{R}$
g	Information gain for Next-Best-View planning	$\mathbb{R}$

Symbol	Description	
$\mathcal{I}$	Volumetric information for Next-Best-View planning	$\mathbb{R}$
c_{pos}	Position cost for Next-Best-View planning	$\mathbb{R}$
c_{tra}	Traversal cost for Next-Best-View planning	$\mathbb{R}$
u	Utility value for Next-Best-View planning	$\mathbb{R}$
θ	Scalar angle	$\mathbb{R}$
d	Distance	$\mathbb{R}$
N	Natural number counter	$\mathbb{N}$
R	Spatial overlap ratio	$\mathbb{R}$
r	Voxel resolution	$\mathbb{R}$
σ	Standard distribution	$\mathbb{R}$
λ	Scalar threshold	$\mathbb{R}$
Σ	Covariance of pose transformation	
E	Expectation	
ξ	Perturbation to pose transformation	
π	Projection function from 3D point to 2D pixel	
M	Number of semantic classes	$\mathbb{R}$
O	The big O notation for algorithm complexity	

1 Introduction

1.1 Motivations

For a robot to explore and understand its surrounding, actively planning and mapping the environment is an essential component of an navigation system. An active mapping system can assist a variety of tasks, such as regular inspection and monitoring of industrial facilities in remote offshore platforms, and search and rescue missions in dangerous disaster sites. Reducing human labour in these inconvenient or unpleasant scenarios with autonomous systems has been a ongoing theme in the field of robotics. Though there have been a multitude of existing works investigating this topic, latest developments in hardware, including perception sensors and robot platforms, open up interesting new directions of research.

Lower cost and denser LiDAR has grown more and more common on the market thanks to the on-going development of self-driving cars. LiDAR provides much longer sensing range ($100\sim200\,\mathrm{m}$ for terrestrial LiDAR) and higher accuracy at long distances compared to the limited sensing range of Red Green Blue Depth (RGB-D) cameras, which is typically $\sim3\,\mathrm{m}$. Many of the LiDAR sensors on the market also have full 360° horizontal Field of View (FoV), which is beneficial for efficient mapping of large open space.

In addition, different types of robot platforms usually require tailored systems to fully exploit their advantageous properties while avoiding their disadvantages. Unmanned Aerial Vehicle (UAV) has 6 Degree of Freedom (DOF) but limited payload capacity, which require computationally light-weight navigation systems; wheeled and tracked robots drive on the ground and have limited traversability, which require conservative path planners. Quadruped robots have experienced significant improvement in reliability and mobility in recent years, such as ANYmal B and C from ANYbotics, with real-world testing in facilities such as industrial plants [1, 2]. These platforms have advantages in all-terrain traversability and stability, as well as considerable payload capability.

1.2 Objectives

At the beginning of this Doctor of Philosophy (DPhil) project, an active mapping system has been developed as the framework and foundation of this study. It was deployed in real-world experiments to assess the challenges and limitations of conventional reconstruction techniques.

Long-term exploration in large-scale environments poses the challenge of maintaining global consistency in reconstruction. Accumulating odometry error is unavoidable during long-term exploration. In the context of SLAM, odometry drift is usually corrected by using loop closures. Rigidity in typical reconstruction representations, such as volumetric [3] and surface mesh [4], limits the accuracy in reconstruction when the map is built on the fly by an exploring robot. It is difficult to incorporate loop closure corrections in such a reconstruction to maintain global consistency.

Another challenge is finding a suitable trade-off between the resolution/scale of the reconstruction, and the speed/efficiency of the mapping system. Some state-of-the-art systems find it difficult to maintain both long sensing range and high resolution in online exploration tasks, and some compromise the reconstruction quality in order to achieve real-time capability [5, 6]. However, being able to recover precise high-resolution

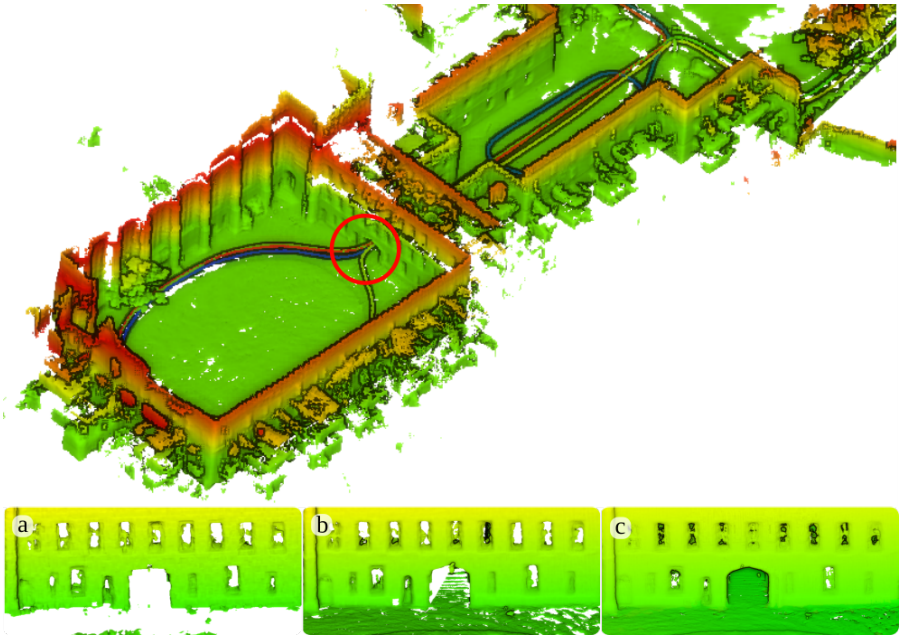


Figure 1.1: Exploration trajectory and 3D reconstruction result from the proposed elastic *supereight* multi-resolution Truncated Signed Distance Function (TSDF) pipeline on Newer College Dataset (NCD). The close-ups focus on the narrow tunnel on the opposite side of the Quad area from experiment's start. (a - the first submap 40 m away; b - just before going through the tunnel; c - revisiting after a large loop closure.)

geometric information is essential for robot path planning and obstacle avoidance, especially when traversing through tunnels and door ways such as the scenario presented in Fig. 1.1.

This study aims to address these challenges of dense LiDAR reconstruction in large-scale long-term exploration tasks. We have proposed a system named SE-Atlas. It is an elastic and efficient LiDAR reconstruction pipeline which has been tested using multiple outdoor exploration datasets in large-scale environments.

In addition, the accuracy, scalability and memory efficiency of SE-Atlas is further improved by analysing spatial overlap and presenting a new formulation for relative uncertainty between SLAM pose graph nodes.

Lastly, semantic information is also integrated into the reconstruction pipeline and the LiDAR map. As an extension to SE-Atlas, an external semantic segmentation module is used to introduce semantic labels into the elastic large-scale LiDAR reconstruction. This additional information then assists the

development of the functionality to detect transitions between indoor and outdoor environments. A probabilistic formulation for fusion across semantic classes is also employed to ensure consistency among submaps.

1.3 Contributions

The main contributions of this DPhil study are:

- An elastic 3D reconstruction system that uses submaps to support corrections to its underlying shape, e.g. from loop closures, and improvements to memory efficiency and scalability of this system via pose graph clustering and submap fusion strategies based on probabilistic, semantic and spatial understanding. – Chapter 4, Chapter 5
- A new formulation for relative uncertainty derived from the work of Mangelson et al. [7] and GTSAM [8], and a formal treatment of uncertainty in submap fusion. – Chapter 4
- The incorporation of LiDAR into *supereight*, a state-of-the-art reconstruction pipeline which achieves multi-fps ($3\,\mathrm{Hz}$) long-range ($60\,\mathrm{m}$), high-resolution ($\sim 5\,\mathrm{cm}$) dense LiDAR integration, which is more detailed than previously existing approaches. These parameters enable high-precision motion planning and long range autonomy. – Chapter 4
- Extension of a state-of-the-art dense semantic mapping framework [9] to maintain a collection of semantic submaps and to probabilistically fuse voxel labels across overlapping submaps, which then enables on-the-fly detection of indoor-outdoor transitions and online adjustment of reconstruction parameters in different environments. – Chapter 5
- An active perception planning system using LiDAR by solving the Next-Best-View (NBV) problem based on information gain metrics and understanding of the environment, capable of efficiently scanning an object or area of interest while traversing unstructured environments. – Chapter 3

- Experimental validation in both simulation and real-world trials, of the proposed system in a multitude of different scenarios, such as industrial facilities, large-scale outdoor environments and indoor multi-story multi-room exploration tasks. – Chapter 4, Chapter 5

1.4 Publications and Highlights

Below is a summary of the peer-reviewed publications produced during this DPhil degree, including four first-author and one non-first-author peer-reviewed articles. It should be noted that RAS Special Edition is an extended version of the ECMR-2021 paper.

1.4.1 List of First-Authored Publications

ICRA-2020 [10]: Y. Wang, M. Ramezani, and M. Fallon. "Actively Mapping Industrial Structures with Information Gain-Based Planning on a Quadruped Robot". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 8609–8615.

ICRA-2021 [11]: Y. Wang, N. Funk, M. Ramezani, S. Papatheodorou, M. Popovic, M. Camurri, S. Leutenegger, and M. Fallon. "Elastic and Efficient LiDAR Reconstruction for Large-Scale Exploration Tasks". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2021, pp. 5035–5041.

ECMR-2021 [12]: Y. Wang, M. Ramezani, M. Mattamala, and M. Fallon. "Scalable and Elastic LiDAR Reconstruction in Complex Environments Through Spatial Analysis". In: *Proc. of the European Conference on Mobile Robotics (ECMR)*. Aug. 2021.

RAS-2022 [13]: Y. Wang, M. Ramezani, M. Mattamala, S. T. Digumarti, and M. Fallon. "Strategies for Large Scale Elastic and Semantic LiDAR Reconstruction". In: *J. of Robotics and Autonomous Systems (RAS)* [2022].

1.4.2 List of Co-Authored Publications

IROS-2020 [14]: M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon. "The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2020, pp. 4353–4360.

IROS-2022 [15]: Y. Tao, M. Popović, Y. Wang, S. T. Digumarti, N. Chebrolu, and M. Fallon. "3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2022.

1.5 Thesis Roadmap

This thesis is laid out as follows:

Chapter 2 – Background: Overview of fundamental information for this thesis, including notation and frame definition, hardware description, and prerequisites on SLAM and odometry systems.

Chapter 3 – Path, Motion and NBV Planning Using Dense LiDAR Reconstruction: An active mapping framework that can autonomously scan an object of interest online, and its deployment in real-world experiments, which revealed limitations in the reconstruction module and motivated the more complex elastic mapping system that we had later developed.

Chapter 4 – Elasticity, Efficiency and Scalability in Large-Scale LiDAR Reconstruction: A dense LiDAR reconstruction core that addresses the limitations presented in the active mapping experiments, namely efficient integration of full-range LiDAR scans at high resolution, the

ability to correct their positions to retain global consistency, the improvement to system scalability via submap fusion, and a formulation for relative uncertainties among SLAM nodes to ensure fusion accuracy.

Chapter 5 – Semantic Analysis in Large-Scale Multi-Sensor Reconstruction: Integration of semantic information into the LiDAR reconstruction system using a multi-camera setup to achieve full-coverage, as well as a probabilistic fusion model across semantic classes to maintain consistency across multiple submaps.

2 Background

This chapter explains necessary background details and information for the remainder of this thesis. This DPhil project focuses on reconstruction for robotic exploration tasks in large-scale environments. Section 2.1 discusses the problems that this thesis is trying to address, motivated by international competitions such as the Defense Advanced Research Projects Agency (DARPA) Subterranean (SubT) Challenge. Section 2.2 provides an overview on the state-of-the-art methods deployed in these challenges to solve these problems. Important concepts on the prerequisite odometry, loop closure and SLAM modules for the proposed reconstruction system (but not included in it) are explained in Section 2.3, Section 2.4 and Section 2.5 respectively. Finally, Section 2.7 describes the hardware used in the experiments throughout this DPhil project, including both robot platforms and perception sensors.

2.1 Problem Statement

In the field of robotics, the problem of exploration can be defined as surveying an environment with a robot to maximise the knowledge of this area. The robotic task that this thesis studies is reconstructing the explored environment of the robot online, using the information gathered during the surveying process,

with an emphasis on assisting other applications such as path planning and navigation. We are interested in exploration in large-scale environments where long survey time would be required to fully scan these areas, such as structured urban scenarios [14, 16], unstructured industrial and disaster sites [5, 10, 11], and organic natural environments [14, 17, 18]. We also investigate confined indoor scenarios, especially narrow corridors and doorways [12].

The DARPA SubT Challenge presents an representative and challenging example for scenarios that align with the focus of this thesis. The different phases of the DARPA SubT Challenge covered a wide range of complex environments, such as human-made tunnel systems, urban underground, and natural cave networks [19]. These environments pose significant challenges to the autonomy, perception, mobility of robots as well as network communications.

This section discusses on a high level the exploration and mapping modules deployed in the DARPA SubT Challenge by teams such as CoSTAR [20], CERBERUS [21], CSIRO Data61 [22] and Explorer [23]. We focus on the problem that these systems are addressing and the challenges that these modules face. In Chapter 3 and Chapter 4 of this thesis, we present similar complications in real-world experiments which are then addressed using the proposed system in this thesis.

2.1.1 Autonomous Exploration

Let the term *known* environment define the space that a robot has observed using sensors, with *unknown* referring to the area that has not been observed by these sensors. In the context of DARPA SubT Challenge, autonomous exploration refers to the system functionality of finding a path for the robot to traverse through the known environment that would most efficiently scan the unknown space and expand the existing reconstruction [21–23]. One of the biggest challenges for path planning in the typical environments of DARPA SubT Challenge [19] is presented by the uneven terrain, irregular wall surfaces

and narrow passages, similar to the real-world challenges presented in Chapter 3 and Chapter 4. Path planning methods have to find a safe route that does not collide with any obstacles, and for wheeled and legged robots specifically, find a traversable path on the ground. This requires the reconstructions to have high resolution so that the terrain, surfaces and especially narrow corridors can be represented as accurately as possible. Chapter 4 describes how this thesis expands a state-of-the-art reconstruction pipeline to realise large-scale LiDAR mapping with high resolution and efficiency.

In addition, the reconstruction has to represent the occupied space at least for obstacle avoidance, and preferably provides explicit known free space representation to ensure the safety of planned path — by only planning in free space and avoiding unknown space [21]. This will be discussed in Section 2.6 and Chapter 4.

The DARPA SubT Challenge environments also present dynamic obstacles such as falling debris, which require robustness and agility in the obstacle avoidance method. Robotic exploration systems commonly employ a local map with a limited size that can be updated quickly for this purpose [20–23]. This will be further discussed in Section 2.2. A similar technique is also employed by the proposed system, which will be explained in detail in Chapter 4.

2.1.2 Robotic Mapping

Individual systems required when mapping in DARPA SubT Challenge involve odometry, localisation, SLAM and reconstruction techniques. Multi-storey structures, inclinations and slopes, sharp turns and slippery terrain challenge the odometry system; sudden changes in lighting condition and environment dampness degrade perception sensing [19]. Section 2.3 discusses these complications in visual and LiDAR odometry.

SLAM loop closure detection and registration methods are required to correct odometry drifts; reconstruction techniques (as well as the path planner in the case of [21]) therefore also need to be connected to the SLAM modules and

maintain the ability to be adjusted for correction, which in this thesis is referred to as elasticity, to achieve global consistency (Chapter 4).

Furthermore, as the environment increases in scale and the exploration grows in its length, more challenges are present for odometry, SLAM and reconstruction systems in aspects such as accuracy, efficiency and scalability. These challenges are discussed further in following sections (Section 2.3 and Section 2.5) as well as in later chapters (Chapter 3 and Chapter 4).

2.2 Exploration and Planning

While this thesis focuses on reconstruction itself, autonomous exploration and mapping is a highly relevant field as both the motivation and application for reconstruction, as demonstrated by many systems in the DARPA SubT Challenge. This section discusses the high-level system design of these autonomous exploration and mapping modules, emphasising the viewpoint decision-making and path planning features. More active mapping methods are covered in Section 3.2 in detail.

2.2.1 Overview

Robotic systems deployed in DARPA SubT Challenge incorporate a wide collections of modules to tackle complicated tasks. The most essential modules include odometry, localisation, SLAM and exploration. For example, on a high-level Agha et al. [20] proposed the concept of Simultaneous Planning, Localisation, And Mapping (SPLAM), considering the localisation and mapping uncertainty in path planning to achieve resilient traversability and risk-awareness.

These systems conventionally have an incremental and iterative structure. As the robot explores more space, its pose is localised against the existing map and then used to update the global reconstruction with new observations. Based on the updated map, the exploration module finds the next viewpoint that will maximise the expansion of the global reconstruction and computes a safe path to

the next best viewpoint — the NBV problem. The robot executes the computed path, and starting the next iteration of localisation, mapping and planning.

The term NBV is commonly used in the field of robotics referring to problems that focus on viewing an object or structure, such as the work of Isler et al. [24] and Delmerico et al. [25]. However, similar techniques can also be employed to map an area of interest [6, 20, 26], which is in line with the autonomous exploration problem described in Section 2.1.1. Hence in this thesis, the term NBV is used both in the context of scanning an object or structure and in that of mapping an area of interest.

The following sections describe the design of specific components within such a system, namely the decision-making modules within a NBV planner which uses Information Gain (Section 2.2.2) and the path planner (Section 2.2.3). Chapter 3 further presents an active perceptual planning prototype that incorporates a similar framework as described above.

2.2.2 Information Gain

To solve a NBV problem, there are two conventional approaches, namely frontier-based and information gain-based. Frontier-based methods refer to focusing on future exploration around the boundary between the known and unknown space. For instance, the method proposed by Williams et al. [27] is incorporated in [22] to guide the robot towards the frontier of known space.

Information gain-based methods on the other hand use certain formulations to measure the potential contribution to the map by a scan, such as the formulation of counting unknown voxels to compute **VolumetricGain** used by team CERBERUS [21].

In addition, Team Explorer [23] focuses on sensor coverage of structures in the environment. Their method emphasises covering surfaces within a distance and angle range, desiring a thorough inspection of areas of interest for search-and-rescue scenarios.

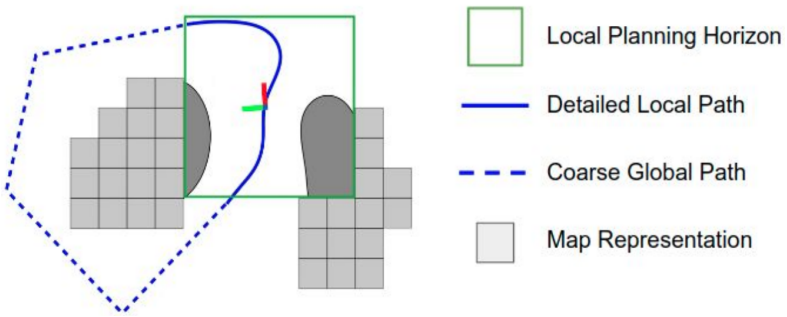


Figure 2.1: The simplified illustration of local and global planning in [23]. The coordinate frame (RGB axis) represents the robot. The global map representation (light grey blocks) has lower resolution for scalability while the local map (dark grey area) has higher resolution for detailed local obstacle avoidance. Hence the global path (dotted blue line) is coarse and the local path (solid blue curve) is smooth and detailed.

2.2.3 Path Planner

Once the NBV decision has been made, the path planning module is responsible for navigating towards the desired goal. A common architecture employed by teams in DARPA SubT Challenge is a combination of local and global planning and mapping [21, 23]. Fig. 2.1 illustrates a simplified example of the planner design by Scherer et al. [23]. The global map is maintained at a low resolution for efficiency and scalability. While this map allows for coarse global planning, it cannot support the detailed obstacle avoidance through complex environments. Hence a local map with high resolution but limited range is also maintained

Team CERBERUS [21] uses a graph-based exploration planner known as GBPlanner [28] for both local and global mapping. GBPlanner is used to build an undirected graph that populates the map and avoids obstacles to derive candidate exploration paths. The desired viewpoint (NBV) is selected from the graph, and Dijkstra's shortest path algorithm [29] is then used to find the most efficient path to reach the desired goal along the graph.

There is a wide range of path planners that have been developed over the years. A complete path planner is a multifaceted system involving map representations, perception sensor characteristics and sometimes robot trajectory history. Some of the most common core algorithms employed by path planners in active mapping systems include Rapidly-exploring Random Tree (RRT) [30]

and RRT* [31], focusing on finding feasible and efficient paths through complex environments. Section 3.2 provides a more complete review on those path planning methods incorporated in state-of-the-art active mapping systems.

2.3 Odometry

Estimating the state (e.g. pose) of the robot correctly is essential for any robotic exploration, navigation and reconstruction tasks. The accuracy of the incrementally created map in these tasks is based on the accuracy of state estimation. *Odometry* focuses on estimating the motion of the robot as it moves through the environment. We often emphasise achieving low drift rates and local smoothness. This section will describe several conventional odometry algorithms. Meanwhile place recognition, global localisation and odometry drift correction will be discussed in Section 2.4 and Section 2.5.

2.3.1 Visual Odometry

Visual Odometry (VO) refers to the technique of estimating the motion of the robot using images streamed from one or more cameras. A typical extension to VO is Visual Inertial Odometry (VIO), integrating high-frequency Inertial Measurement Unit (IMU) sensors into camera-based odometry. In the scenarios where visual feature tracking is hard to achieve between consecutive camera frames, such as during highly dynamic motion where the overlap between camera views is limited, IMU input provides an important source of short-term high-frequency odometry information that does not rely on perception, maintaining the estimation of robot's state. The survey paper by Delmerico and Scaramuzza [25] assessed a variety of VIO benchmarks. Due to the wide availability of cameras and IMUs, there are many commercial VIO systems available [32–35].

Modern cameras are able to capture and stream a large amount of data due to their high resolution, but mobile robots' on-board hardware limits the available computation. In order to achieve real-time visual tracking, many VO and VIO systems first incorporate feature extraction techniques and replace dense camera

images with sparse features that capture characteristics of original images but require significantly less data. For example, ROVIO by Bloesch et al. [33] uses FAST corner detector [36], and OKVIS by Leutenegger et al. [32] uses BRISK feature detector and descriptor [37].

VO and VIO systems also need to estimate the relative transformation among latest camera images as well as other desired camera states. Some of these systems are based on variants of the Extended Kalman Filter (EKF), such as ROVIO by Bloesch et al. [33] and OpenVINS by Geneva et al. [38]. ROVIO [33] uses an iterated EKF framework and OpenVINS [38] extends Multi-State Constraint Kalman Filter (MSCKF) [39].

An alternative approach for VIO is optimisation and smoothing within a sliding window, which improves estimation robustness compared to methods based on EKF variants [25]. Methods such as SVO+GTSAM [40], OKVIS [32], VINS-Mono [34] and Kimera [35] all employ a factor graph to model the constraints among robot poses and camera observations within the latest period of time. These systems then optimise the feature-based errors to estimate the latest poses of the camera using optimisation libraries such as g^{2}o library [41], iSAM2 [42] and GTSAM [8].

However, a major drawback of the modern VO and VIO lies in the limited FoV of conventional cameras. Rapid motions, especially sharp turns, and complicated environments (such as the DARPA SubT Challenge) can lead to limited overlap between consecutive frames and their features, usually resulting in unreliable odometry estimation or even failure. LiDAR odometry provides a solution to this limitation.

2.3.2 LiDAR Odometry

With LiDAR sensors becoming cheaper and more commercially available in recent years, LiDAR-based state estimation becomes a popular alternative and extension to VO and VIO. Compared to RGB cameras, LiDAR directly measured information about the geometry of the environment via depth measurements,

while VO and VIO require an additional step of triangulation which introduces error. In addition, a conventional 3D LiDAR typically has 360° horizontal FoV. It is easier for LiDAR odometry to maintain reliable state estimation in scenarios where VO can fail.

The results of Hilti SLAM challenge [43] show that systems leveraging both vision and LiDAR outperform pure vision-based methods significantly. Because this DPhil project uses LiDAR as the main perception sensor, LiDAR odometry is a key input to the proposed reconstruction system SE-Atlas.

Iterative Closest Point (ICP) and its Variants

The conventional output from a LiDAR sensor is point clouds. ICP is one of the most widely used techniques to compute the relative transformation between a pair of point clouds [44]. ICP typically computes the rotation and translation that would minimise the point-to-point distance between two point clouds, but also allows for replacing point-to-point distance with different formulations, a common alternative of which is point-to-plane distance.

Many active mapping methods employ ICP [45, 46] or its variants [47] for point cloud localisation. Systems that choose surface mesh as map representation, such as those utilising TSDF [48–50], localise the sensor by minimising the difference between the latest scan and a predicted measurement rendered from the current map. Vespa et al. [50] chose a variant of ICP and created an odometry system for depth camera, *supereight*. It was later extended to incorporate LiDAR measurements [51].

Methods that employ surfel representations [52, 53] further leverage the position and orientation of surfels to create improved geometric constraints for pose graph odometry and SLAM systems. One of the most recent example is *ElasticLiDAR++* proposed by Park et al. [53], It is a map-centric SLAM system integrating LiDAR, inertial and visual sensor inputs. The registration of the current scan is conducted based on point-to-plane distance between every pair

of corresponding surfels. While the localisation and mapping results of *ElasticLiDAR++* are impressive, this system requires heavy computation, and could not reach real-time as presented by Park et al. [53].

Overall, ICP and its variants are reliable methods to achieve high alignment accuracy, but it relies on a good pose initialisation. Some of the more basic ICP-based methods are also often time-consuming unless the point clouds are heavily downsampled.

LiDAR Odometry And Mapping (LOAM) and its Variants

To reduce the required computation time of point cloud registration, several state-of-the-art LiDAR odometry systems extract point cloud features instead of using the entire point cloud.

For example, LOAM [54] extracts point features with very high or low curvature, which correspond to edges and planes in a point cloud respectively. The relative transform between two consecutive point clouds is then computed by minimizing point-to-line and point-to-plane distances. The overall system takes on a coarse-to-fine multi-threaded structure. The coarse scan-to-scan matching can run at $10\,\mathrm{Hz}$, which is the typical LiDAR sensor frequency. The refine thread registers the current scan with a accumulated map, running at a lower frequency of $2-5\,\mathrm{Hz}$.

Shan and Englot [55] proposed LeGO-LOAM, a variant of LOAM that is tailored for ground robots. LeGO-LOAM leverages the assumption of a large ground plane that is always present to further reduce the computation weight in segmentation and optimisation. Shan et al. [56] further extended LOAM with a factor graph and proposed LIO-SAM. LIO-SAM integrates IMU readings into the LiDAR odometry as additional factors in the graph structure, and registers the current scan at a local scale instead to a global map to improve real-time performance. The factor graph incorporated by LIO-SAM can also accept loop closure factors, which extends it beyond the typical odometry functionalities (Section 2.4 and Section 2.5).

2.3.3 Leg Odometry

Walking robots can also integrate joint sensing measurements to provide high frequency state estimations for these highly dynamic robot platforms. This state estimation process is usually referred to as the leg odometry. However it relies on an accurate kinematic model of the robot as well as a complex modelling of the interaction between the robot's feet and the ground.

For simplification, some works on leg odometry assumed ideal non-slipping contacts between feet and ground, and that each foot has a point contact with the ground. IMUs are also commonly integrated with raw leg odometry to constrain the robot base motion, such as [57–59]. Bloesch et al. [58] fused joint encoders and an on-board IMU together using Observability Constrained Extended Kalman Filter (OCEKF). This method was improved by Bloesch et al. using Unscented Kalman Filter (UKF) [59]. A later work of Bloesch et al. [60] presented Two State Implicit Filter (TSIF). TSIF avoids an explicit process model to achieve higher modelling flexibility. It is the default leg odometry state estimator deployed on ANYmal quadruped robot [61].

However, the non-slippery assumption is unrealistic, and incorrectly modelling/detecting foot contact is a major contributor to leg odometry drift [62]. In [59], Bloesch et al. employed a simple threshold based on the Mahalanobis distance of the UKF innovation to detect and reject outliers caused by foot slippage. Camurri et al. [63] presented a foot contact detector for robust leg odometry using robot's internal force sensing by estimating the probability of reliable contact. Lin et al. [64] proposed using a deep learning method to estimate foot contact based on kinematic and IMU measurements. This method has been evaluated on the MIT Mini-Cheetah quadruped robot [65].

2.3.4 Multi-sensor Fusion

Multi-sensor fusion techniques refer to combining different sensor measurements and leveraging the advantages of complementary sensors. This usually improves the robustness and accuracy of odometry systems.

As previously mentioned, LiDAR can improve the robustness of visual tracking because LiDAR can create detailed and metrically accurate maps for localisation. On the other hand, combining VIO with LiDAR addresses the relatively low frequency of LiDAR as well as the motion distortion in LiDAR point clouds. Modern 3D LiDAR sensors collect range returns continuously over time. A complete 360° scan is accumulated and then made available to downstream applications. The high dynamics of a legged robot, especially in rotation, can lead to motion distortion of these scans. High frequency odometry sources such as visual and inertial give pose estimation between LiDAR scans so that each LiDAR measurements can be projected into the map correctly.

The fusion of these odometry methods could be a simple architecture where VIO provides the initial estimation for LiDAR odometry to improve scan matching accuracy [66, 67]. LOCUS [68] accepts multiple odometry inputs, including VIO, kinematic and wheeled, as priors for LiDAR odometry. This method has been deployed in the DARPA SubT Challenge.

On the other hand, HERO [69] accepts multiple odometry inputs but only uses the most reliable source. Such a heuristic relies on metrics tailored for experiment scenarios and has poor generalisability towards new environments. Zhao et al. [70] proposed Super Odometry and configured the different odometry estimations to feed relative pose constraints into a factor graph. The reliability of each odometry source is reflected in the covariance of measurement in the factor graph, and unreliable sensor readings (e.g. in the case of sensor failure) will have increased covariances and reduced effect during factor graph optimisation.

These aforementioned methods are referred to as *loosely-coupled* because they handle different odometry sources independently and fuse them together in a later stage. An alternative approach is *tightly-coupled*, which jointly estimates the robot state using all available sensors. Variants of the Kalman Filter, such as MSCKF, are popular techniques for tightly-coupled multi-sensor fusion. For example, Yang et al. [71] and Zuo et al. [72, 73] all relied on a MSCKF framework

to integrate features in point clouds, in vision and IMU measurements tightly together. A more recent work, R^{2}Live proposed by Lin et al. [74], uses iterated Kalman filtering.

Kalman filter variants are also a common technique for fusing leg odometry into other state estimators. Wooden et al. [75] developed a state estimator for Boston Dynamics Big Dog quadruped robot that uses leg odometry to assist VO when VO loses tracking. Ma et al. [76] then extended this system for Boston Dynamics LS3 quadruped, using EKF to integrate leg odometry into VIO. The leg odometry functions as back-up state estimator when visual tracking fails in long-term operation. Pronto, presented by Fallon et al. [77], used an EKF for long-term multi-sensor fusion on the Atlas humanoid robot, incorporating stereo camera and LiDAR. This system was further extended by Nobili et al. [78] and Camurri et al. [79] for the ANYmal [61] and HyQ [80] quadrupeds, respectively.

One of the odometry sources used by the proposed system in this thesis is Visual Inertial Lidar/LEgged Navigation System (VILENS) by Wisth et al. [81]. It incorporates all previously mentioned sensor types in one factor graph and uses iSAM2 [42] for optimisation. In the scenario where visual or LiDAR odometry is unreliable due to there being few features, VILENS does not discard these observations like typical multi-sensor systems, but treats each feature as a factor and maintains them in the graph to account for the limited feature counts. One additional key functionality that VILENS provides for the proposed system is the motion undistortion in LiDAR point cloud. Chapter 4 will present the improvements in LiDAR point cloud and volumetric reconstruction with motion undistortion.

2.4 Loop Closure Detection

State-of-the-art odometry systems can achieve impressively high accuracy. For instance, LOAM consistently achieved an average error $\sim 1\,\%$ in all the experiments by Zhang et al. [54]. However, the accumulation of some degree of odometry error is unavoidable during large-scale exploration tasks.

In the context of SLAM the odometry drift is usually addressed using loop closure detection and correction. Take factor graph SLAM as an example. Upon the detection of a loop closure, a factor graph SLAM system introduces a new constraint between the head and the tail of the loop, optimises the factor graph, and then propagates pose corrections back through the trajectory. These corrections can also be applied to the map representation used within SLAM systems.

Because this DPhil project uses LiDARs as the main perception sensor, this section will focus on a variety of state-of-the-art loop closure detection techniques based on LiDAR point clouds.

2.4.1 Scan Context

Kim and Kim proposed Scan Context, an egocentric spatial descriptor for 3D LiDAR scans [82]. It is designed for global place recognition and loop closure detection. Different from many conventional LiDAR point cloud descriptors that rely on histograms and point count distributions [83–85], Scan Context descriptor encodes a 3D LiDAR point cloud into a matrix by dividing the scan into azimuthal and radial bins in the sensor local frame, and stores the maximum height of points in each bin. Such a design improves the efficiency of encoding and the preservation of point cloud internal structure compared to the conventional methods based on point distribution histogram [82].

The centre of the scan, i.e. the pose of the LiDAR, functions as a global keypoint that holds an egocentric Scan Context descriptor. Kim and Kim further introduced a two-phase search algorithm for loop closure detection based on Scan Context. They first extracted the ring key, a rotation-invariant and less informative descriptor, from Scan Context to enable faster search for potential loop closure candidates. A slower but more accurate pairwise similarity score is then used for finding loop closures from candidate pairs and any localisation refinement such as ICP, avoiding searching and matching across the whole database.

Scan Context has been deployed in several state-of-the-art LiDAR SLAM systems, including the one-year long-term LiDAR localisation by Kim et al. [86] and the multi-agent system DiSCo-SLAM by Huang et al. [87].

2.4.2 SegMatch and SegMap

Different from Scan Context [82], SegMatch [88] proposed by Dubé et al. is a matching algorithm based on segments of LiDAR scans. It was later expanded into a map representation solution by Dubé et al. called SegMap [89].

A necessary prerequisite of the segmentation step in SegMatch is removing the ground. The technique implemented by Dubé et al. was originally proposed by Douillard et al. [90]. After the removal of the ground plane, point cloud segments in SegMatch is first extracted using a fixed-radius cylindrical neighbourhood and then clustered using the "Cluster-All Method" [90]. Dubé et al. then used several different descriptors for segments, and focused on presenting two of them, namely an eigenvalue-based descriptor and an ensemble of shape histograms.

SegMatch employs a learning-based approach to identify matches between segments from the map and the current LiDAR scan. The matching process first generates candidate matches using k-d tree search in feature space, and then uses a random forest in a classifier to make the final matching decision. Having a learning-based segment matching technique means that there is a training phase. In SegMatch, the classifier was trained and tested using sequence 06 of the KITTI dataset [16], and then assessed for loop closure detection using sequence 05 of KITTI. It was further tested using more complex data from the Clausiusstrasse in Zurich, with the segmentation method replaced by the region growth method proposed by Rabbani et al. [91]. This demonstrated that the learnt descriptor of SegMatch can be generalised to a different urban environment when trained using an urban dataset. It also showcased the modular design of the overall pipeline.

SegMatch is then expanded into SegMap by Dubé et al. [89], a unified method for map representation in LiDAR SLAM problems. The segmentation method in SegMap is improved with an incremental Euclidean distance-based region-growing technique [92], then a descriptor is extracted from each segment using a Convolutional Neural Network (CNN) with an autoencoder-like architecture. By accumulating descriptors and centroids of corresponding segments in a world frame, SegMap constructs a global segment map. For localisation and matching segments between new LiDAR scans and the map, SegMap uses the same k-d tree approach as SegMatch. Compared to SegMatch [90], SegMap's descriptor extractor architecture allows the compressed representation of the descriptor to used to reconstruct an approximate map at any time. In addition, SegMap incorporates semantic information in its descriptor as well for easier distinction between static and dynamic objects. While the SegMap descriptor were trained using KITTI [16] in the experiments presented by Dubé et al. [89], it has been further assessed in non-urban scenarios. The multi-robot SLAM experiments covered unstructured disaster environments that resembles industrial facilities. This demonstrated the improved generalisability of SegMap compared to SegMatch.

2.4.3 Efficient Segmentation and Mapping

SegMatch [88] and SegMap [89] focused on urban scenarios, but these methods usually do not generalise well to structure-poor natural environments. Therefore Tinchev et al. proposed Efficient Segmentation and Mapping (ESM)[18, 93] to address LiDAR-based place recognition in vegetated areas such as forests or orchards. Notably, ESM can run without a GPU, making it suitable for platforms such as UAVs.

ESM employs a Euclidean segmentation method similar to SegMatch [88] because region-growth methods such as [91] rely on surface normals, which are unstable when computed in natural environments. An oriented key pose is then extracted from each segment, defining the orientation and position of

the segment. The key pose extraction method proposed by Tinchev et al. was compared with the keypoint extraction method employed in SegMatch, and the result demonstrated that a consistent key pose can mitigate changes in point cloud appearance due to different viewpoints [93]. Instead of relying on a hand-crafted set of features in its descriptor, ESM proposes a novel description method and learns segment representations to handle the variability of the environment. It learns a descriptor space which efficiently represents the similarities between partial observations of the same segment, making it robust to incomplete data.

ESM was designed specifically for global place recognition relative to a prior LiDAR map without any prior information regarding the current sensor pose. In the experiments presented by Tinchev et al. [18, 93], the random forest used for segment matching in ESM was also trained using sequence 06 in KITTI dataset [16]. It was then compared against SegMatch [88] using both KITTI dataset and a forest dataset. The results demonstrated that ESM outperformed SegMatch in both cases [18, 93]. Tinchev et al. further assessed their system using a sequence collected in a foliage-heavy forest in Cornbury Park, Oxfordshire, showing the generalisability of ESM.

2.5 SLAM

As explained in the previous section, SLAM systems correct odometry drift after loop closures are detected. Besides the importance in localisation, SLAM is one of the most essential components of autonomous robotic systems. It has applications in all scenarios where a map is needed but is not available *a priori* [94]. SLAM systems aim to reconstruct a map of the environment with global consistency unlike the local map in odometry systems. This global understanding of the environment is sometimes required for other robotics tasks such as path planning, navigation and exploration. For instance, the simple example of "infinity corridor" in Fig. 2.2, presented by Cadena et al. [94], demonstrates that with the global map constructed after SLAM loop closure, point B and C can be connected with a "shortcut" that is not present in the odometry map.

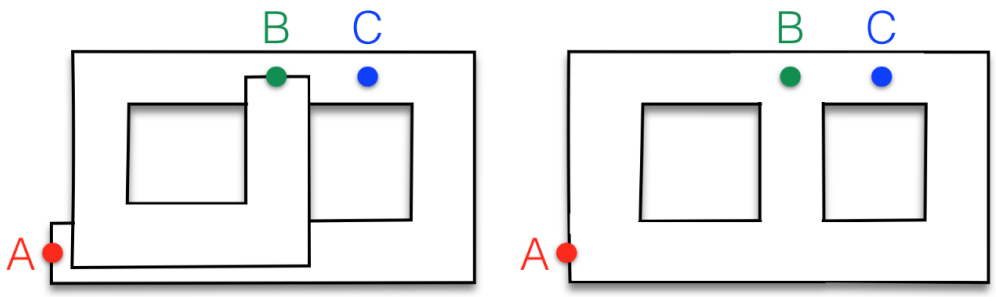


Figure 2.2: The example of loop closure and SLAM provided in [94]. **Left**: the odometry map where the robot started at point A and eventually reached point B and caused two loop closures at point A and B respectively. **Right**: the SLAM map after loop closures and drift correction.

SLAM loop closures reveal the true topology of the environment to the robot for other applications.

The SLAM problem has been studied for over 30 years, yet it still remains a relevant research topic. This section will provide a brief overview of different techniques employed in SLAM systems throughout history.

2.5.1 Classical Methods

The survey papers by Bailey and Durrant-Whyte [95, 96] provided an overview on the classical methods of solving SLAM between 1986 and 2004.

EKF-SLAM

One of the most common solution for SLAM at the time of the survey papers by Bailey and Durrant-Whyte [95, 96] was based on EKF and its variants. Some early methods in navigation for mobile robots [97–99] employed Kalman filter-type algorithms for localisation. Later methods formulate a SLAM problem as a state-space model including the pose of the robot as well as landmarks. Conventionally, these states are modelled to have additive Gaussian noise, leading to the use of EKF.

EKF-SLAM focuses on the latest state; the history of the robot's poses and their connections to the observable landmarks are marginalised into the last state

using the **motion/kinematic model** and the **observation model**. Intuitively, standard EKF-SLAM systems such as [100] have a two-phase structure, incorporating a **Time-update** phase and an **Observation-update** phase to model kinematics and observations, respectively. Many traditional SLAM systems employed Kalman filter-based methods and focused on improving its efficiency and real-time capability. One example of large-scale SLAM based on Kalman filter is the work of Guivant and Nebot [17]. Their system has been tested in the Victoria Park, Sydney, Australia. It addresses the issue of real-time processing, vehicle high speed, uneven terrain and dynamic clutter. The dataset collected in this trial also became a popular benchmark for SLAM systems.

Fast-SLAM

One major drawback of EKF-SLAM is the linearisation of non-linear motion and observation models, which sometimes can lead to significant inconsistency in its solution [101]. Hence, another commonly implemented classical method, Fast-SLAM [102, 103] based on Rao-Blackwellised particle filters, was proposed by Montemerio et al. . This approach focuses on describing the motion of the robot directly with more generalised non-Gaussian probability distributions and non-linear process models. Particle filtering is not computationally feasible for the high dimensional state-space of a SLAM problem, hence Rao-Blackwellisation is applied to reduce the sample space in particle filter. Fast-SLAM 2.0 [103] has been assessed using a real-world large-scale outdoor experiment, the Victoria Park trial [17], and produced an accurate map.

However, a later analysis conducted by Strasdat et al. [104] revealed that, for the same dataset, keyframe-based Bundle Adjustment (BA) and optimisation method (also referred to as Structure from Motion (SfM) in the field of computer vision) was demonstrated to be more accurate per unit of computation time compared to the classical filter-based methods. As a result, graph-based SLAM systems start to become the more prominent formulation [94].

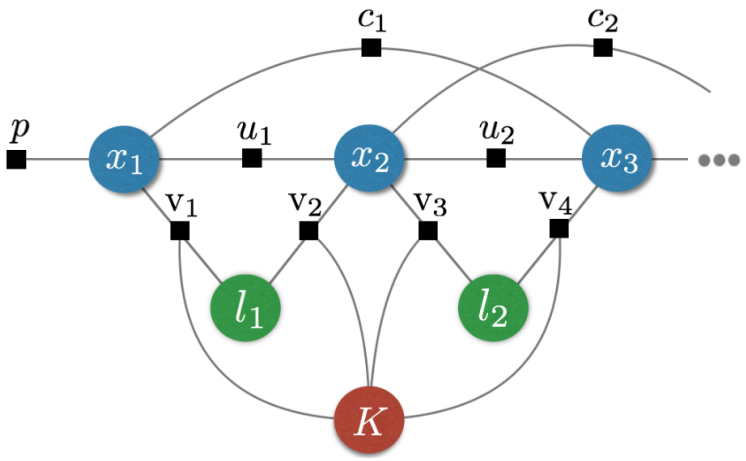


Figure 2.3: The basic example of SLAM formulated as a factor graph provided in [94]. Blue nodes represent robot poses at consecutive time steps $(x_1, x_2, x_3, ...)$, green nodes represent landmarks (l_1, l_2), the red node represents the variable parameters associated with sensor calibration (K). Black squares represent the factor of constraints among nodes): u represents odometry constraints, v represents observations, c represents loop closures, and p represents prior factors.

2.5.2 Graph-based SLAM

Modern SLAM systems conventionally include two main components, namely a *front-end* and a *back-end*. The front-end commonly handles the inputs from one or more sensors and processes the data accordingly. For example, techniques for visual feature extraction (Section 2.3.1) and multi-sensor fusion (Section 2.3.4) are usually included in the front-end. This section focuses on the back-end of a SLAM system that performs localisation and mapping using the abstracted and processed sensor inputs.

Currently, the standard formulation for the back-end of modern SLAM systems is a *maximum a posteriori* estimation using factor graphs [105] to model the interdependence among robot poses and map landmarks. Fig. 2.3 presents an example of SLAM based on factor graph. Instead of marginalising past robot poses to the latest state like classical methods (Section 2.5.1), the estimated state space of a graph-based SLAM system typically maintains the whole trajectory of the robot, so as to recover the maximum a posteriori for the entire history of robot poses and the map [106]. This process is referred to as *smoothing* to differ from the classic *filtering* approach.

For example, in the *square root Smoothing and Mapping (SAM)* system proposed by Dellaert and Kaess [106], the maximum a posteriori optimisation is formulated as a non-linear least square problem. After linearising the problem using Jacobian matrices, it can be summarised as the following linear least square problem,

$$\delta^* = \arg\min_{\delta} ||A\delta - b||_2^2 \tag{2.1}$$

where vector δ represents the state space, δ^* is the desired solution of the least square problem, matrix A contains all the Jacobian matrices of the odometry and observation model represented by the factor graph and vector b contains all the measurements on robot poses and landmarks. Because δ now contains the entire robot trajectory with no marginalisation, matrix A is and will remain sparse with the trajectory growing longer and the scale of the SLAM problem growing larger. The sparsity of A means that it can be factorised more efficiently, making the optimisation problem easier to solve than a filtering problem. While this new method by Dellaert and Kaess [106] is more efficient than classical methods, it was designed as a batch algorithm, making it not as efficient when handling an incremental SLAM problem [107].

Kaess et al. extended the aforementioned square root SAM formulation to be more efficient when applied incrementally and presented incremental Smoothing and Mapping (iSAM) [107]. The matrices involved in the optimisation problem in iSAM are updated directly when new measurements arrive to avoid the unnecessary computation required for reapplying factorisation every time in square root SAM. This method is further improved by Kaess et al. [42] using a novel data structure referred to as Bayes tree. By incorporating the Bayes tree, Kaess et al. presented a new system for sparse non-linear incremental optimisation, iSAM2, with improved efficiency through incremental variable re-ordering and fluid re-linearisation.

A competition to iSAM is the g^{2}o library [41], which is employed in the state-of-the-art SLAM system ORB-SLAM [108–110]. The original ORB-SLAM

by Mur-Artal et al. [108] is a feature-based monocular SLAM system, focusing solely on visual input. It uses ORB features [111] to sparsify visual inputs, and uses DBoW2 [112] to construct a map of ORB features for a bag of words place recogniser. It relies on the g^{2}o library [41] as the optimiser. The overall pipeline is constituted of three threads, namely tracking, local mapping and global mapping. The tracking thread handles feature extraction and aligning the latest scan with the local map. The local mapping thread optimises the consistency within the local map of ORB features. The global mapping thread is only triggered when a loop closure is detected and applies correction to the whole trajectory when odometry drift is corrected. The following works [109, 110] extend ORB-SLAM with more sensor inputs such as stereo, RGB-D and fisheye cameras, as well as additional features such as multi-map SLAM.

The main focus of modern SLAM includes accuracy, efficiency and scalability. Hence the scale of the environment and the duration of the exploration are both interesting topics for investigation. These aforementioned optimisers, g^{2}o (as a component of ORB-SLAM) and iSAM, have both been tested against large-scale outdoor datasets such as KITTI [16] and Victoria Park [17] and presented impressive performance.

The proposed reconstruction system of this thesis relies on a pose graph SLAM [113]. The loop closure detection and correction is essential to the proposed system in order to achieve global consistency as well as scalability in large-scale reconstructions. The overall map created by the proposed system is represented by multiple submaps, inspired by the Atlas framework [114]. These submaps are connected to a SLAM factor graph and can be moved around to maintain consistency upon loop closure correction. This design will be explained more thoroughly in Chapter 4.

2.5.3 Loop Closure Robustness

One potential challenge for conventional SLAM systems is false positive loop closures [113]. In a graph-based SLAM system, a false loop closure can lead to

the addition of incorrect constraints into the factor graph, resulting in incorrect optimisation, inferior trajectory or a complete failure of the SLAM back-end.

One popular solution among existing systems is *switchable constraints* proposed by Sünderhauf and Protzel [115]. This approach adds a switchable variable to re-weight the cost function for each newly added loop closure constraint in a factor graph. All loop closure constraints are treated as true positive initially when added; but when detected as an outlier, i.e. a false positive, the switchable variable of this loop closure will be adjusted to down-weigh the corresponding constraint in the overall cost function. The drawbacks of this method are two-fold. First, the addition of a large number of switchable variables increases the complexity of a SLAM system. Second, there is no guarantee that the outliers are fully down-weighed [113].

Another solution is to employ a robust cost function [116], such as the method proposed by Chebrolu et al. [117]. A standard technique to improve optimisation robustness is M-estimators by Zhang [118], which replaces the standard square error with a robust function that has a lower penalising effect outside the basin close zero error. To improve the generalisability of M-estimator and avoid manual tuning, Agamennoni et al. [119] proposed self-tuning M-estimators by considering the tuning parameter of M-estimator as a variable in the optimisation problem within an iterative two-step Expectation Maximisation procedure. Chebrolu et al. similarly employed an Expectation Maximisation algorithm, but optimised for the shape parameter of Barron's robust function [116] to fit the probability distribution of the residuals.

Yang et al. [120] proposed another approach for SLAM loop closure outlier rejection. This approach leverages the Black-Rangarajan duality [121] between robust estimation and outlier processes, and uses Graduated Non-Convexity (GNC) to compute a robust solution without initialisation. GNC replaces the robust function with a surrogate function that has a control parameter, and gradually recover the original cost function.

Similarly, the AdaptivE RObust least-Squares (AEROS) approach proposed by Ramezani et al. [113] incorporates the Black-Rangarajan duality [121] as well as the generalised robust cost function of Barron [116]. AEROS formulates optimisation as an Iteratively Reweighted Least Squares (IRLS) problem and jointly estimates a hyper-parameter together with sensor poses to achieve an adaptative robust cost function that represent the entire set of M-estimators. AEROS has been compared against other state-of-the-art outlier rejection methods such as GNC and switchable constraints using publicly available synthetic datasets and real LiDAR-SLAM datasets collected using 2D and 3D LiDARs. It demonstrated its competitiveness against state-of-the-art methods as well as its superiority in real-world experiments [113].

2.6 Representation

To this point, we have focused on the estimation of robot trajectory which, while using sensor measurements, does not focus directly on the representation of the map itself. There is a variety of representations for 3D environments such as surface meshes and volumetric maps, and a wide range of data that can be stored in the map such as Signed Distance Function (SDF) and occupancy probability. Conventional techniques for each representation usually provide distinctive features that different robotic tasks will prefer. This section gives an overview on the characteristics of common representations that have been incorporated in 3D reconstruction and active mapping systems, and explain why volumetric occupancy map is chosen as the main method for this thesis. A more detailed analysis on different reconstruction representations will be covered in Section 4.2.

2.6.1 Surface Mesh and Surfels

The active perception methods proposed by Hollinger et al. [122, 123] and Hover et al. [124] based their mapping decisions upon surface meshes that are created from point clouds via Poisson surface reconstruction algorithm [125].

Two important prerequisites to these techniques were applying filtering and downsampling to the accumulated point cloud, and estimating normals for surfaces. These processes are essential for surface mesh accuracy. However, they are computationally expensive to apply on large point clouds that is also continuously growing online at a high frequency. By utilising a streaming surface reconstruction technique [126], Kriegel et al. [47, 127, 128] designed a series of active mapping systems that focus on boundaries of scanned surfaces. Surface mesh representation has the benefit of clearly defined frontiers of the known surface, but it cannot easily provide an understanding on the 3D environment, e.g. whether a certain space is safe for robot exploration.

Representations such as TSDF and Euclidean Signed Distance Function (ESDF) decrease the necessity of the filtering and smoothing steps by voxelising the 3D space and weighting each new observation with confidence. Conventionally, by applying the Marching Cubes method [129], TSDF and ESDF representations can be converted into surface meshes easily. Newcombe et al. [48] used a global, densely-allocated TSDF volume to achieve reconstructions with groundbreaking details. To improve the scalability of TSDF map, a hash-table is a common technique implemented in reconstruction systems [130, 131]. For instance, Niessner et al. [132] employed a TSDF volume for large-scale mapping by exploiting the sparsity of environment via a technique called Hashing Voxel Grid (HVG). Tanner et al. [133] also used HVG for efficient large-scale TSDF reconstruction over kilometers. Their pipeline BOR^{2}G further incorporates multiple types of sensor inputs, including long-range LiDAR.

A limitation of these aforementioned methods is that their scalability is achieved by only storing a truncated volume in front of (and behind) observed surfaces. As explained in Section 2.1.1, explicitly representing known free space is a desired feature in reconstruction algorithms because path planners can leverage it to ensure a safe path that does not traverse into unknown areas. Not mapping the free space between the sensor and each observed surface therefore

limits the usage of aforementioned TSDF reconstruction methods in robotic exploration and planning. Reijgwart et al. [5] proposed a reconstruction pipeline, Voxgraph, which used their own TSDF and ESDF representation [4, 134] and explicitly mapped all the known free space in each observation. However, this constrained their scalability in large-scale reconstruction tasks — in their real-world experiments the sensing range and voxel resolution were limited to $16\,\mathrm{m}$ and $20\,\mathrm{cm}$ respectively. To account for challenging scenarios such as narrow passages and difficult terrain in the DARPA SubT Challenge, reconstructions with higher resolution are needed.

Alternatively, the approach of Whelan et al. [52] uses surfels as its primary surface representation. A surfel is an abbreviated term for a **surf**ace **el**ement. In the field of reconstruction, surfels are 3D surface units, each containing at least a 3D coordinates and the local surface normal. More advanced surfel representation methods also store in each surfel additional information such as resolution/radius or point distribution and uncertainty. Recent work by Park et al. [53] revised the dense surfel model and proposed novel representation and matching methods for dense LiDAR SLAM. However, surfels do not explicitly represent known free space that is safe for robot path planning.

2.6.2 Occupancy Voxel Map

Reconstruction methods that use occupancy probability, on the hand, allow for a clear distinction between observed free and unobserved space [135, 136]. A commonly used for storing occupancy information in dense 3D reconstruction is OctoMap [137], an octree data structure that allows for scaling the voxel resolution for efficient integration and memory usage. Many active mapping systems have OctoMap as their reconstruction core, such as [138, 139]. Isler et al. [24] also used OctoMap and proposed several Information Gain formulations based on occupancy probability for making NBV decisions. These formulations were further assessed in [140].

Because of our focus is on robotic exploration, this thesis mainly employs volumetric occupancy representation for reconstruction. Our initial active mapping framework [10] similarly incorporated the OctoMap library [137] and took on the Information Gain-driven approach as the system of Isler et al. [24, 140]. The real-world experiments (Section 3.6) revealed several limitations of this method, such as poor scalability and efficiency at high resolution. Therefore to address these issues, our following works [11–13] extended *supereight* [50], an efficient multi-resolution reconstruction pipeline that allows for both TSDF and occupancy representation. This will explained in detail in Chapter 4.

2.7 Hardware

This section describes the hardware employed in the development and experiments of the proposed system, which in general can be categorised into sensors (Section 2.7.1) and platforms (Section 2.7.2).

2.7.1 Sensors

Because this thesis focuses on mapping and reconstruction, perception sensors used in this thesis are the main emphasis of this section, about which Fig. 2.4 presents an overview.

We use the Leica BLK 360 laser scanner shown in Fig. 2.4 to collect highly detailed and accurate point cloud of experiment sites as ground truth for reconstruction. This tripod-based laser scanner has a horizontal FoV of 360° and a vertical FoV of 300°, as well as millimetre accuracy at $\sim 60\,\mathrm{m}$. Software tools such as CloudCompare* are used to process these scans into uniformly distributed ground truth data at desired resolution (such as $1\,\mathrm{cm}$). This scanner however takes minutes to collect a single scan, making it feasible mainly for offline data collection. This DPhil project therefore focuses on creating an online reconstruction system.

*https://www.danielgm.net/cc/

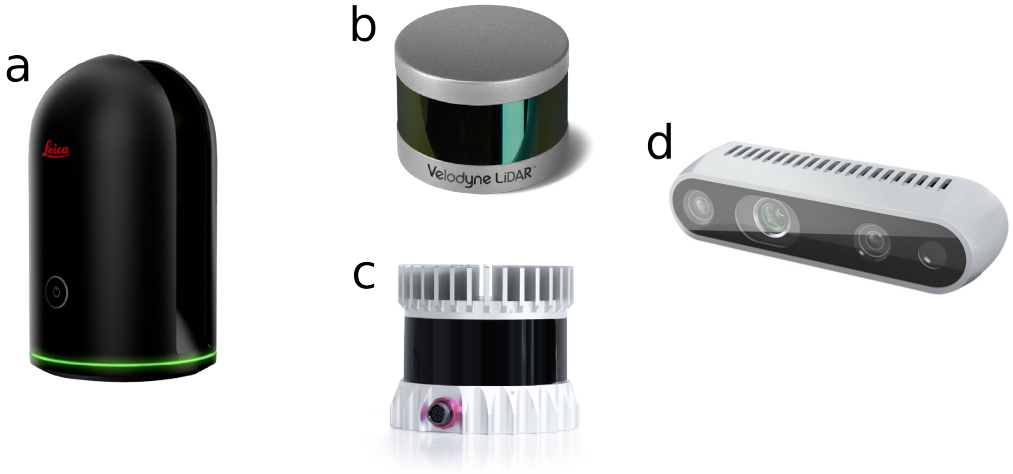


Figure 2.4: The primary sensors used in this thesis. **a**: BLK 360 laser scanner; **b**: Velodyne VLP-16 LiDAR; **c**: Ouster OS1-64 LiDAR; **d**: Intel Realsense D435i camera.

LiDAR 3D LiDAR is the chosen type of sensor in this thesis for scanning and reconstructing large-scale environments. These sensors consist of one or more lasers that continually move (usually rotate) to measure the distance to surfaces around the sensor. Many modern LiDAR sensors can provide accurate measurements over long distance and cover $360°$ horizontally, hence they are the preferred sensor type for the reconstruction and exploration tasks that this thesis focuses on.

For instance, a Velodyne VLP-16 LiDAR was used in the active mapping system [10] (Chapter 3). It has a sensing range of $\sim 100\,\mathrm{m}$, a typical accuracy of $3\,\mathrm{cm}$, $360°$ horizontal FoV and $\pm 15°$ vertical FoV. It has 16 channels of laser beams and produces scans at $5-20\,\mathrm{Hz}$. Another brand of LiDAR used in our later experiments [11–14] is Ouster, including OS1-64 and OS0-64. Both OS1-64 and OS0-64 provide scans of 64×1024 points at a frequency of $10\,\mathrm{Hz}$, covering full $360°$ horizontally. OS1-64 has a vertical FoV of $\pm 16.6°$ and a range of $120\,\mathrm{m}$, while OS0-64 has a wider vertical FoV of $\pm 45°$ but shorter range of $50\,\mathrm{m}$.

As previously mentioned, a complete LiDAR scan is accumulated over time from individual laser measurements and then published as one point cloud. Hence this point cloud can suffer from motion distortion when the sensor is deployed on a highly dynamic robot, and needs to be corrected for when creating 3D reconstructions.

Camera Cameras measure the intensity of light on surfaces that is within particular spectral bands (e.g., colour, infra-red). RGB-D cameras further utilises infra-red to measure the depth of surfaces in front of the sensor besides providing coloured 2D information. Cameras are employed only as auxiliary sensors in the reconstruction aspect of this thesis, for purposes such as terrain mapping (Chapter 3) and semantic segmentation (Chapter 5) instead of directly contributing to the large-scale reconstruction. However, they play a key role in odometry and localisation, as explained in in Section 2.3.

In our experiments, the incorporated cameras are Realsense D435 and D435i. These are stereo RGB-D cameras, consisting of two imagers and a infra-red projector. The advantage of having stereoscopic vision, compared to monocular cameras, is that depths of the scene can be estimated via the process of triangulation directly using disparities between a single pair of images from the sensor. To estimate depths using stereoscopic vision, it is essential to know the accurate intrinsic and extrinsic parameters of both cameras. The distance between the pair of imagers, also referred to as the baseline, is known as part of the hardware. We use the multi-camera calibration functionality in toolbox Kalibr[†] by Eidgenössische Technische Hochschule Zurich (ETH Zurich) to compute these parameters.

The infra-red modules in D435 and D435i cameras further assist the depth estimation in low lighting conditions where RGB images are of poor quality. In outdoor scenarios, on the other hand, infra-red sensing is significantly affected by sunlight, but stereo cameras can still provide a reliable source of depth estimation.

2.7.2 Platforms

In this thesis, the term *platform* refers to the mobile hardware that carries sensors and conducts robotic tasks such as scanning and exploration. The typical ones employed in this thesis include quadruped robots and handheld devices, as

[†]https://github.com/ethz-asl/kalibr

Figure 2.5: An overview of the typical mobile platforms in this thesis. **Left**: ANYbotics ANYmal (Version B) with LiDAR and camera; **middle**: hand-held multi-sensor rig called Frontier with LiDAR and camera; **right**: Boston Dynamics Spot carrying Frontier.

showcased in Fig. 2.5. Different types of mobile platform present distinct types of movement and degree of dynamics. The proposed system takes advantage of the 6 DOF of these platforms during reconstruction, and at the same time allows for addressing challenges brought by the high dynamics, such as motion distortion.

In addition, it is common for these mobile platforms in the field of robotics to carry multiple sensors on-board. While the man perception sensor used for reconstruction in the proposed system is LiDAR, other sensors are also utilised for purposes such as localisation.

ANYbotics ANYmal (Version B)

This robot was deployed as the main robot platform in the active mapping system [10] in Chapter 3. It is a prototype walking robot with several drawbacks such as low traversability over difficult terrain (slopes) and low accuracy in leg odometry. It has a typical walking speed of $0.5\,\mathrm{m\,s^{-1}}$ and a maximum speed of $1\,\mathrm{m\,s^{-1}}$. It carries a Velodyne VLP-16 LiDAR and an Intel Realsense D435 RGB-D camera. The RGB sensor in D435 has a FoV of $69°$ horizontally and $42°$, and a resolution of 1920×1080. The FoV of the depth sensor is $87°$ horizontally and $58°$ vertically. Its resolution is 1280×720 and its ideal range is $0.3{-}3\,\mathrm{m}$. This robot carries three on-board Compulab Fit-PC IPC3 (A) computers for locomotion, navigation and additional systems.

Frontier multi-sensor rig

The Frontier multi-sensor rig carries an Ouster LiDAR (OS1-64 or OS0-64) and an Intel Realsense D435i RGB-D camera. The Realsense D435i camera is identical to D435 with an addition of a built-in IMU. Frontier carries a Intel NUC8i7BEH NUC computer as its processor. The Frontier rig can both be a hand-held device as well as a payload on a robot (Fig. 2.5).

Boston Dynamics Spot

For some of the experiments presented in this thesis, the Boston Dynamics Spot carries a Frontier rig for LiDAR reconstruction. The robot itself carries five additional Intel Realsense D430 stereo cameras, with two facing forwards, one on each side and one facing backwards. Compared to ANYmal (Version B), Spot is a much more agile robot with a maximum walking speed of $1.6\,\mathrm{m\,s^{-1}}$ and a faster turning speed. It also has a significantly more accurate leg odometry and much more robust traversability, enabling reliable stair-climbing. Spot carries an on-board computer known as the Spot Core[‡] for its own locomotion and navigation computation.

[‡]https://www.bostondynamics.com/sites/default/files/inline-files/spot-core.pdf

3

Path, Motion and NBV Planning Using Dense LiDAR Reconstruction

This chapter includes elements of the following publication:

[10] Y. Wang, M. Ramezani, and M. Fallon. "Actively Mapping Industrial Structures with Information Gain-Based Planning on a Quadruped Robot". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 8609–8615

Acknowledgements
I would like to thank Dr Milad Ramezani and Prof Maurice Fallon for conducting the real-world experiments with me to assess the proposed active perceptual planning system in this chapter.

This chapter presents an active mapping system that has been designed and developed as the framework and foundation of this DPhil project. This system focuses on NBV and path planning to ensure safe and autonomous exploration around an object of interest. This was an initial prototype that leveraged state-of-the-art methods when it was developed, but was not further upgraded with the latest developments in exploration and navigation techniques such as those deployed in the DARPA SubT Challenge [20–23]. It was a proof of concept that enabled real-world experiments, which then revealed limitations in conventional reconstruction methods. This in turn motivated the extensions and improvements of the reconstruction core that is employed in the overall pipeline

— the improved reconstruction core will be explained in detail in Chapter 4.

3.1 Introduction

Active perceptual planning, in the context of robotics, refers to autonomous exploration by a mobile robot equipped with sensors so as to conduct a survey of an object or site of interest. The decisions of where to conduct further scanning are made autonomously by algorithms instead of manually by humans. Such a system can be of assistance for the regular inspection and monitoring of remote or dangerous facilities such as offshore platforms. The specific problem that is being addressed with the proposed system is actively mapping an object of interest in complicated industrial settings. The object will be of unknown shape, surrounded by uneven terrain and mobility hazards. The mock-up helicopter (Fig. 3.1) at Fire Service College in Gloucestershire, UK presents such an situation. Although there have been a multitude of research exploring active mapping on varied robot platforms for many applications, such a problem still poses a challenge.

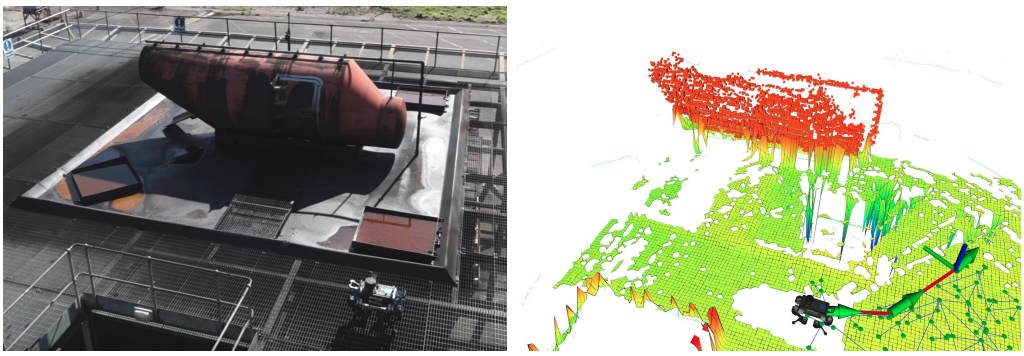


Figure 3.1: Left: The ANYmal robot actively mapping a mock-up helicopter at the Fire Service College in Gloucestershire, UK. The safety stanchions, stairwells and the skirt under the helicopter are mobility hazards. **Right**: the state of the mapping system showing the object reconstruction, the elevation map, a RRT plan and the next walking goal.

UAV [26, 141] is a popular choice of robotic platforms for these kinds of missions for their 6 DOF manoeuvrability. However, it is difficult to operate aerial platforms within confined spaces or on windy offshore platforms, and

their sensing payload is limited. Wheeled and tracked robots [24, 142] have the advantage of better stability and more payload allowance, but are affected by difficult terrain which UAV can fly over. In comparison, quadrupeds have the advantage of other ground robots, can cover the same terrain as wheeled or tracked robots but can also cross mobility hazards and climb stairs. Advances in quadruped mobility and hardware reliability have been significant and the first industrial prototypes are being tested on live industrial facilities [1]. Therefore the proposed active mapping system was designed for a quadruped robot, ANYmal. Fig. 3.2 gives an overview on the hardware setup.

The perception sensors used in this proposed system [10] include a Velodyne VLP-16 LiDAR sensor and a Realsense d435 RGB-D camera. Many active mapping systems use depth camera as their primary mapping sensor. We instead rely on LiDAR for reconstruction. Compared to depth cameras, LiDAR sensors maintain significantly higher accuracy at long range, hence is more suitable for large-scale industrial environments. VLP-16 also provides a 360° horizontal FoV, making the proposed system more efficient in exploring the environment.

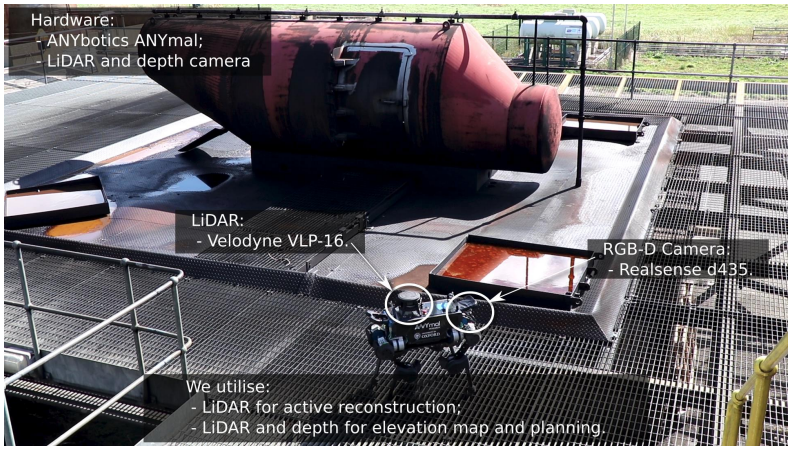


Figure 3.2: An overview of the hardware. The robot platform of the proposed system is an ANYmal quadruped robot and the perception sensors include a LiDAR and an RGB-D camera.

The aim of this work was a realistic validation in an industrial scenario. Many systems proposed by past literature were developed in ideal lab settings. The work of Isler et al. [24] proposed effective Information Gain formulations,

as demonstrated in their following work [25], but their system omitted path planning and collision avoidance, and did not consider the cost of travelling between viewpoints.

This presented active mapping framework adapts this Information Gain approach and formulates it as a NBV problem. Using the LiDAR and the depth camera, my system builds and maintains a 3D model of the object of interest as well as the local environment. This system determines the best pose for the robot to conduct further exploration on the basis of metrics drawn from both the object (Information Gain) and the environment map (cost of mobility). It enables the robot to not only scan an object of interest efficiently, but also traverse an unknown environment. We have deployed my system on the ANYmal quadruped robot in real time and evaluated it in simulated as well as real-world environments.

I acknowledge that the information gain formulations leveraged by this system are no longer the most up-to-date methods used in the state-of-the-art robotic exploration and navigation systems. For instance, in the DARPA SubT Challenge, team CoSTAR [20] implemented an uncertainty-aware global planning technique referred to as Pobabilistic Local and Global Reasoning on Information roadMaps (PLGRIM). PLGRIM maintains an Information RoadMap (IRM) on both the local and global level of planning, capturing information about occupancy, coverage, traversal risk and free space connectivity. It then solves for a Pobabilistic Local and Global Reasoning on Information roadMaps (POMDP) policy in a receding horizon fashion to generate a plan that maximises coverage.

Additionally, in the real-world experiments presented in this work, the presented system relied on point cloud maps available *a priori* for accurate and reliable LiDAR localisation. However, localisation is not within the scope of this research. These prior models are not used in the planning, navigation or reconstruction process, because this work focuses on designing a model-free active mapping system.

The success in these real-world experiments was a proof of concept for the prototype active perceptual planning framework. While this prototype leverages standard components and does not present a particularly novel contribution in its system design, it provides a platform for testing reconstruction techniques. These real-world experiments revealed several limitations in the current reconstruction method, such as the rigidity of the map, the low efficiency and scalability of the reconstruction, and the motion distortion in the point cloud. I addressed these limitations in my following works, which will be elaborated upon in the next chapter (Chapter 4).

3.2 Literature Review

This section presents an overview of the literature relevant to this active mapping framework. As Section 2.6 have provided a brief description of a variety of conventional map representations implemented by reconstruction pipelines, this section focuses on the typical design of planning methods in active mapping systems.

In the context of active mapping systems, the term *prior information* refers to the knowledge of the object or site of interest in advance of the reconstruction operation, such as its 3D model. Depending on the availability of prior information, active mapping systems are typically categorised into *model-based* and *model-free* approaches [143, 144].

Model-based Methods

By leveraging prior information, model-based active mapping methods are able to more efficiently plan and optimise a collection of viewpoints before carrying out operations. The applications range from reconstructing a site [145], inspecting an existing model [26, 146] or targeting specific areas in the prior model to improve model certainty [122]. Model-based systems are typically applied in industrial scenarios because Computer-Aided Design (CAD) models are often

available [143]. These systems are desired for routine survey and inspection due to their ability to optimise operation efficiency in advance.

Blaer and Allen [145] designed a model-based system to map large indoor and outdoor environments. The prior information required by their system was a 2D footprint map of the site. Based on this prior map, their system took two stages to reconstruct a 3D model. In the first stage, their system generated an almost complete initial 3D mesh. It first randomly sampled a set of scan candidates in the prior map, and defined an Art Gallery Problem (AGP) to optimise viewpoint placements and minimise the number of scans required. Then their system planned a trajectory to execute the poses by solving a Travelling Salesman Problem (TSP). However, in practice there were many 3D obstructions not represented in the 2D map. Therefore the initial 3D mesh reconstruction at this stage contained many holes due to occlusion. In the second stage, their system sequentially planned 3D views to cover holes in the initial model. Based on the initial mesh, their system updated the environment model with a volumetric occupancy representation. The volumetric representation allowed for estimating occlusion and computing observable unknown volumes. This system iteratively selected the viewpoint that could observe the most unknown space to conduct further scan, until a desired coverage has been achieved.

Because Blaer and Allen formulated their problem as an AGP and a TSP, it was necessary for their system to have access to a complete and accurate prior map. In one of their experiments, a building was omitted in the 2D footprint. This resulted in a large vacancy in the mesh, as indicated in Fig. 3.3. In this case, the hole in the mesh was later completed in the second stage of their system. However, when the prior map is inaccurate, there is a risk of the AGP and TSP methods selecting impossible viewpoints inside omitted objects, or planning trajectories through obstructions. This is particularly of concern for ground robots, such as the mobile platform of Blaer and Allen (an ATRV-2 AVENUE robot) and our quadruped (ANYmal). Therefore, such a design is unable to handle changes in the object or the environment, or to generalise well among a

wide range of scenarios. Blaer and Allen designed their system focusing on static environments like historical sites. Meanwhile, a complicated industrial facility is a dynamic environment, and requires a different problem formulation.

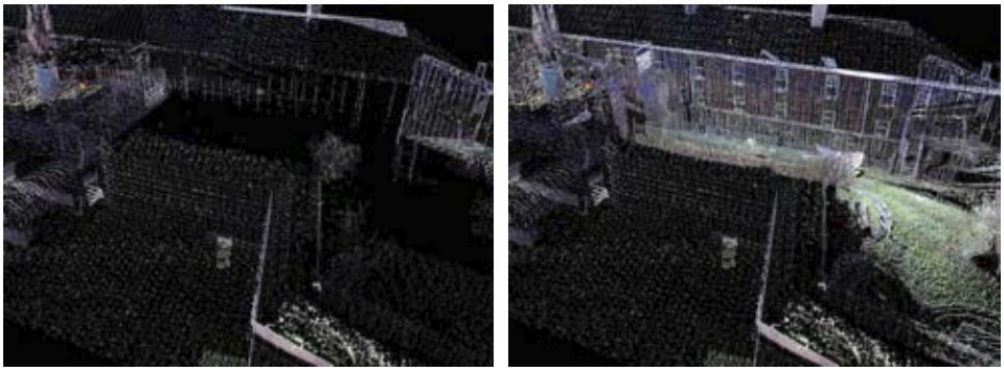
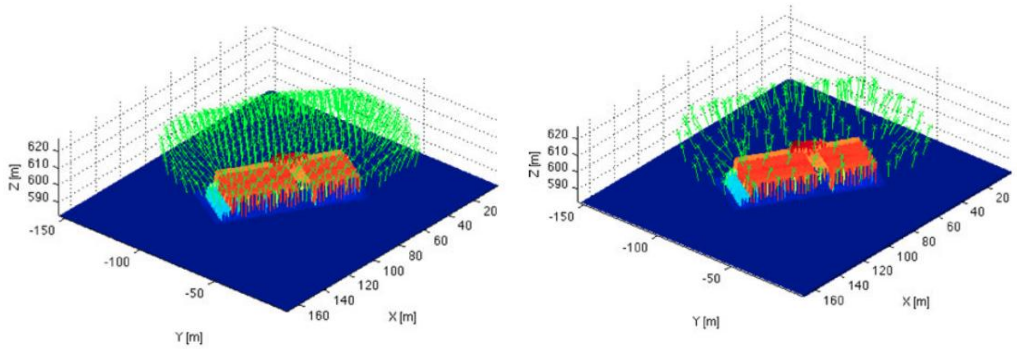


Figure 3.3: Experimental result of a reconstructed point cloud from Blaer and Allen [145]. A hole in the reconstruction after the first stage is due to an initial viewpoint being occluded by an omitted building (left). The hole was filled in the second stage with two more scans (right).

The system of Schmid et al. [146] used a 2.5D Digital Surface Model (DSM) as prior information to actively reconstruct outdoor sites. Their robot platform was a Vertical Take Off and Landing (VTOL) UAV. The system of Schmid et al. first uniformly populated the area of interest with viewpoint candidates. The orientations of these viewpoints were dependent on the surface normal in the DSM below. Their system then selected a subset of viewpoints based on redundancy. The redundancy evaluation considered the distance and angle difference between a new viewpoint and already selected ones. Fig. 3.4 gives an example of their viewpoint selection process. Their system then optimised the trajectory by solving a TSP.

Schmid et al. designed their system for large environments. Using 2.5D DSM and defining the viewpoint search space in 2D reduced the dimension of the problem and simplified the viewpoint planning process. As a result, their system took $25\,\mathrm{min}$ to map a $146\times125\,\mathrm{m}^2$ area and $81\,\mathrm{min}$ for a $274\times326\,\mathrm{m}^2$ area. However, this constrained the orientations of these viewpoints; their system could only plan viewpoints scanning primarily downwards. This made



(a) Uniformly spaced viewpoint candidates **(b)** Viewpoints selected based on redundancy

Figure 3.4: An example of viewpoint generation and selection in the system of Schmid et al. [146].

it hard for the system to scan surfaces other than the top, therefore sacrificing details on the side.

Based on prior information, a model-based process can also choose to focus on the most interesting regions to more efficiently improve the reconstruction quality. The system of Hollinger et al. [122] was designed to decrease the uncertainty of a ship hull surface mesh, assuming the availability of an initial mesh model. Their system first computed the local mesh uncertainty, which represented how likely the reconstruction was incorrect in a local region. It then sampled viewpoints at which the robot could inspect these high uncertainty areas. Once the viewpoints were selected, they optimised the trajectory among these viewpoints by formulating it as a TSP, and used a RRT [30] to avoid collision over the course of the trajectory.

Bircher et al. [26] designed a model-based system for inspection based on their original model-free system[147]. The purpose of this system was to increase the resolution of an existing coarse model. It used a volumetric representation to plan viewpoints around the object of interest. Using the prior model, their system was able to estimate the number of observable surface voxels at any scan candidates. The sensor placements in this system, however, were not planned in advance. Instead, they were decided by sequentially finding the scan

pose that could observe the largest number of surface voxels that had not yet been inspected, as the robot scanned more and more surfaces. Such a system architecture is more similar to model-free active mapping systems, which will be discussed below.

Having access to a prior model allows the model-based active mapping systems to optimise the sensor placement planning in advance, making these systems more efficient in applications such as routine inspection [122, 146] in designated contexts and tasks. However, errors or changes in the prior map pose challenges to these systems [145]. It is of interest to explore future functionalities such as reactive path planning for model-based systems. This would allow autonomous resurvey tasks to be conducted in a environment that is subject to dynamics and changes, such as industrial facilities and disaster sites.

Model-free Methods

Being independent of prior information, on the other hand, allows a mapping system operate in environments that have not yet been explored. Because there is no prior map, model-free systems cannot plan viewpoints in advance like model-based systems; instead, they usually take on iterative system architectures and build up the reconstruction incrementally. In each iteration, a model-free system first generates or computes a collection of scan candidates, which are poses of the sensor or configurations of the robot. Then a decision needs to be made to select the next viewpoint from this collection of candidates. This decision on where to conduct the next scan is referred to as a NBV problem. After the NBV has been determined, the system plans a path to it to continue exploration.

Because of this iterative structure, model-free active perception systems are more versatile and can be applied to a wider variety of objects and sites. For instance, the system of Bircher et al. [147] aimed at the exploration of unknown spaces of different scales. The approach of Kriegel et al. [47, 128] was designed to reconstruct objects of arbitrary shape but confined size. Newer approaches scales to even larger environments [148, 149]. The path planning component of

these systems also allows for collision avoidance in a complicated environment, for example in the systems of Vasquez-Gomez et al. [45].

While these system architectures of model-free active mapping systems appear similar in general, different systems have unique implementations of the core components. First, scan candidates can be computed based on the current reconstruction of the object or generated using random sampling. Different representation approaches, such as surface and volumetric, allow for different candidate generation methods.

The systems of Kriegel et al. [47, 127, 128] and that of Kompis et al. [148] employed surface mesh representation, and the scan candidates were planned on the boundaries of existing surface mesh. Yervilla-Herrera et al. [46] chose volumetric representation, so they did not define boundaries of the current reconstruction. Instead, their system randomly generated the pose of their mobile base and the configuration of their robot arm to produce viewpoint candidates. Leveraging the explicitly represented known free space and unknown space in volumetric occupancy representation, Dai et al. [150] defined the *frontier* of the existing map as the free voxels with unobserved neighbouring voxels, and sampled viewpoint candidates focusing on the frontiers.

RRT is another commonly used method for generating viewpoint candidates, such as the systems of Bircher et al. [147] and Vasquez-Gomez et al. [45]. An RRT grows by randomly sampling nodes and connecting new nodes to the existing tree via collision-free edges. The systems of Bircher et al. [147] and Vasquez-Gomez et al. [45] used the RRT nodes as viewpoint candidates. The advantage of RRT is that all candidates generated this way are already reachable — eliminating the need to re-evaluate the safety of each node again.

Following the generation of candidates, the NBV needs to be determined. The selection method depends on how the object or site is represented in the system. When volumetric representation is employed, it is possible to estimate quantitatively the amount of reconstruction improvement that is expected to be achieved at each candidate pose. This is referred to as the Information Gain

formulation. For instance, Bircher et al. [147] computed the volume of observable unknown space. The systems of Kriegel et al. [47, 128] also considered the entropy in each voxel based on the occupancy probability of each voxel, and so did the design of Dai et al. [150]. Isler et al. [24] and Delmerico et al. [140] proposed and assessed a series of Information Gain formulations based on voxel entropies. On the other hand, in [127] Kriegel et al. chose surface mesh as their representation, and the NBV selection was solely based on the Euclidean distance between the candidate and the current pose of the sensor.

Once the NBV is determined, the path planning component is used to produce a collision-free trajectory for the robot to execute. As mentioned above, when an RRT is used to generate candidate poses, the tree branches are already collision-free, therefore the systems of Bircher et al. [147] and Vasquez-Gomez et al. [45] did not employ an additional path planner. In the system of Yervilla-Herrera et al. [46], because their scan candidates were randomly generated, they needed to assess the reachability of each candidate pose. Yervilla-Herrera et al. employed the RRT* [31] method to plan paths to each candidate; those that were not reachable were rejected.

RRT* planning also provided the system of Yervilla-Herrera et al. [46] with an more optimal path compared to RRT. They conducted comparison experiments in simulation between RRT* and RRT and assessed their performance with a distance-based traversal cost. The path provided by RRT* had a lower cost than that by RRT in all experiments.

Model-based systems can also be adapted to incorporate uncertainties in the prior model and to improve the quality of reconstruction. In a work following up on [122], Hover et al. [124] addressed the potential lack of prior information. They took on a coarse-to-fine multi-stage technique. This system first conducted an initial low-resolution *identification survey* (first stage) by following a predefined path. Fig. 3.5a gives an example of the identification survey. The robot travelled on the surface of a generous bounding box that was placed around the ship hull at a safe distance. Their system then planned paths for a high-resolution

inspection survey (second stage) to improve the quality of the mesh and to find small anomalies on the surface of ship hull such as sea mines. Because there was a initial surface mesh already available, the second stage was similar to a model-based system and planned all the viewpoints in one go. Hover et al. treated the viewpoint generation problem as an AGP. The goal of these viewpoints was to cover the whole mesh as well as to achieve redundancy. At this stage, the path to visit all viewpoints was produced by solving a TSP and collision avoidance was realised via a bi-directional RRT, known as RRT-Connect [151]. An example of a feasible inspection survey is given in Fig. 3.5b.

(a) Identification Survey **(b)** Inspection Survey **(c)** Trajectory Optimisation

Figure 3.5: An example of the multi-stage survey in the work of Hover et al. [124]

Their system also designed a third stage for path length optimisation, as shown in Fig. 3.5c. It used an RRT* to shorten the overall trajectory produced previously in AGP while at the same time preserving the inspection coverage.

There are recent systems combining both surface meshes and Information Gain to make NBV decisions. In one of the most recent works from the Vision For Robotics Lab (V4RL) at ETH Zurich, Kompis et al. [148] employed TSDF and ESDF as the core map representation [152], and sampled viewpoints around the surface frontier voxels - observed voxels that are part of the surface and have at least one unobserved neighbour voxel. They then chose the NBV based on Information Gain, which in their proposed method is derived from the observation distance and how many times a voxel has been observed. Song et al. [149] proposed a different system design. Their system maintains both a surfel representation [52] and a volumetric occupancy map [137]. The volumetric map was maintained at a low resolution and used for global path planning to maximise

global coverage, and the surfel map was used for local inspection planning to improve reconstruction details and decrease surface uncertainty.

These reconstruction methods can be applied in complex real-world applications. For instance, Mascaro et al. [153] presented a LiDAR-based mapping system for construction tasks using irregular on-site objects in novel and extreme environments. Built upon a LiDAR-based map, this system segments object-like instances such as rocks and stones out for further grasping and manipulation tasks. This system is designed for a heavy-duty robotic walking excavator and its autonomous manipulation tasks in complex large-scale environments. In its real-world experiments, this system demonstrated impressive results.

This thesis decided to focus on model-free active mapping due to its versatility in unknown and complicated environments, such as unstructured industrial context. While there have been a range of active mapping systems designed for different environments, a significant number of them were assessed in simulated experiments or ideal laboratory contexts [24, 148, 149]. Complications in real-world experiments, such as the scenarios presented by Mascaro et al. [153], are sometimes overlooked. Therefore we designed an initial active mapping framework, inspired by a range of model-free approaches, and assessed this system with experiments in an industrial setting to study any limitations.

3.3 System Architecture

This section describes the modules and structure of the active mapping system proposed in [10]. Fig. 3.6 illustrates a block diagram of the system architecture. This system is based on an iterative pipeline, commonly incorporated by model-free active mapping systems [26, 45, 148]. At the start of each iteration, the robot executes a scanning action of rolling the robot base while standing (further described in Section 3.6.1) to collect a sensor sweep. These sensor measurements are incorporated into a map, then the route to a new scan location (NBV) is planned, and the robot is requested to walk to the NBV for further

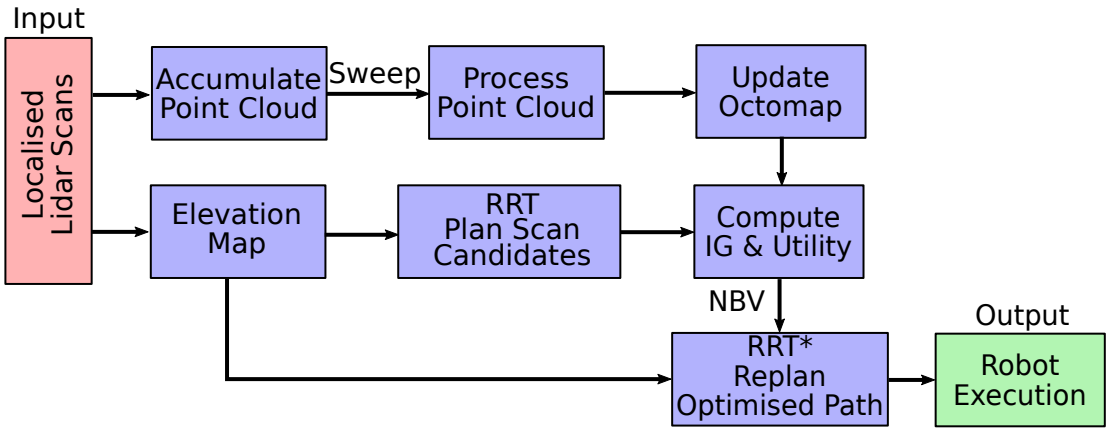


Figure 3.6: Block diagram of the active mapping system architecture.

exploration. This sequence is repeated, until a termination criterion (such as map completion) is met.

The LiDAR sensor produces point clouds C in the LiDAR frame $\{\mathcal{L}\}$, which are then transformed into the base frame $\{\mathcal{B}\}$. During a scanning action, point cloud C is first accumulated in the odom frame $\{\mathcal{O}\}$ using the current odometry estimation of robot pose ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{B}} \in \mathbf{SE}(3)$. The accumulated point cloud is then transformed into the map frame $\{\mathcal{M}\}$ using ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{O}} = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{B}}{}^{\mathcal{O}}\mathbf{T}_{\mathcal{B}}{}^{-1}$ at the end of the scanning action. The accumulated point cloud is denoted *sweep* S in this system. Due to the incremental nature of the odometry module, this 2-step accumulation ensures that the *sweep* point cloud contains smooth surfaces. In addition, the robot runs a localisation system with little drift on the scale of the presented experiments in [10], allowing for the assumption that the robot pose ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{B}}$ is accurate. This is further discussed in Section 3.6.

The *sweep* is then downsampled for uniformity and filtered to remove outliers. Next, the system uses the processed *sweep* as well as the pose of the robot ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{B}}$ to update the occupancy probabilities of voxels in the reconstruction. The reconstruction core of this active mapping system is a single unitary OctoMap [137]. The inability for the OctoMap to support odometry drift and loop closures is addressed in Chapter 4.

I also use the LiDAR measurements to generate an elevation map of the environment surrounding the robot. The elevation map has a useful range of

about $10\,\mathrm{m}$, allowing only local planning. The path planning module evaluates terrain traversability subject to the elevation map and builds an RRT to generate a collection of viewpoint candidates $\mathbf{X}_{\text{candidate}}$. A viewpoint candidate $\mathbf{x} \in \mathbf{X}_{\text{candidate}}$ is a pose where the robot could go to for the next scanning action.

I use a utility function u_x to determine the best scan candidate (NBV), $\mathbf{x}_{\text{best}} \in \mathbf{X}_{\text{candidate}}$:

$$u_x = g_{\mathbf{x}} \times (1 - c_{\text{pos}_{\mathbf{x}}}) \times (1 - c_{\text{tra}_{\mathbf{x}}}). \tag{3.1}$$

This function combines contributions from

- information gain $g_{\mathbf{x}}$: which measures the expected improvement of the model if given a sweep from that pose,
- position cost $c_{\text{pos}_{\mathbf{x}}}$: which penalises poses that have already been visited or are too close to the object,
- traversal cost $c_{\text{tra}_{\mathbf{x}}}$: which models the cost of travelling to a specific scan candidate pose.

This system is only a prototype and a proof of concept, and the purposes of the position cost $c_{\text{pos}_{\mathbf{x}}}$ in this system is to address limitations of the currently incorporated information gain formulations (Section 3.4.1 and Section 3.4.2). In an improved system, the information gain measurement $g_{\mathbf{x}}$ should enforce behaviours such as avoiding visited positions. The position cost should then focus on other factors. These measures are discussed further in following sections.

Finally, this system replans an optimised path using RRT* [31] before the robot takes the next mapping action.

3.4 Next Best View Decision Making

Given a partial model of the object of interest, this system needs to determine the expected improvement in the model should a scan be made from a particular scan candidate pose. The approach is to trace a series of rays from a hypothetical pose and to estimate the expected information gain of observable voxels.

3.4.1 Information Gains

Let $\mathbf{R_x}$ denote the set of rays cast by a viewpoint $\mathbf{x}$. For each ray $\mathbf{r} \in \mathbf{R_x}$, $\mathbf{V_r}$ is the set of voxels that the ray intersects with before reaching its endpoint. The information gain $g_\mathbf{x}$ at candidate $\mathbf{x}$ is the sum of volumetric information, $\mathcal{I}$, in every voxel $\mathbf{v} \in \mathbf{V_r}$ along each ray $\mathbf{r} \in \mathbf{R_x}$:

$$g_x = \sum_{\forall \mathbf{r} \in \mathbf{R_x}} \sum_{\forall \mathbf{v} \in \mathbf{V_r}} \mathcal{I}. \tag{3.2}$$

I incorporated two formulations for $\mathcal{I}$ from Isler et al. [24], namely *Occlusion Aware* $\mathcal{I}_{\text{OA}}$ and *Rear Side Entropy* $\mathcal{I}_{\text{RSE}}$ which are summarised here.

Other formulations proposed by Isler et al. are less relevant due to LiDAR sensor's long range and 360° FoV.

Occlusion Aware

This measure determines how effectively uncertainty will be reduced by scanning at a certain pose considering voxel visibility.

Given the occupancy probability $P_o(v)$ of voxel v, the entropy of the voxel is obtained from:

$$H(v) = \text{ - } P_o(v) \ln P_o(v) \text{ - } (1 \text{ - } P_o(v)) \ln(1 \text{ - } P_o(v)). \tag{3.3}$$

Then the *Occlusion Aware* volumetric information of voxel v, $\mathcal{I}_{\text{OA}}(v)$, is defined as:

$$\mathcal{I}_{\text{OA}}(v) = P_v(v) H(v), \tag{3.4}$$

where $P_v(v)$ is the visibility probability of voxel v, computed as :

$$P_v(v_n) = \prod_{i=0}^{n-1} (1 - P_o(v_i)). \tag{3.5}$$

In Eq. (3.5), v_n is the n-th voxel along the ray r; $v_i, i = 0...n-1$, is a voxel that ray r intersects before reaching v_n.

Rear Side Entropy

This measure is based on the *Occlusion Aware* volumetric information but focuses on voxels at the rear of observed surfaces. *Rear Side Entropy* is formulated as:

$$\mathcal{I}_{\text{RSE}}(v) = \begin{cases} \mathcal{I}_{\text{OA}}(v) & v \text{ is a } \textit{Rear Side Voxel}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

The idea is that a *Rear Side Voxel* is also likely to be occupied by the object. Focusing exploration on these voxels concentrates scans on the object rather than on surrounding free space.

Implementation

While these metrics proposed by Isler et al. [24] are useful, their experimental validation was limited to lab experiments with a stereo camera planning over a fixed set of poses. I was motivated to develop a more realistic field system which operates in a large-scale industrial site.

The work of Isler et al. [24] constructs a bounding sphere around the object of interest, and distributes a set of 48 viewpoints on this sphere to guarantee that they are safe poses for the sensor. This is hard to achieve in a model-free real-world scenario, where areas that are safe for robot traversal cannot be defined as clearly. My system instead plans scan candidates progressively using an RRT which uses the LiDAR elevation map (Section 3.5). Similarly, the work of Vesquez-Gomez et al. [45] generates viewpoints in the robot's configuration space by growing a RRT. Compared to the work of Vesquez-Gomez et al. , I further incorporate RRT* to optimise the trajectory once the NBV has been determined, which will be explained in more details in Section 3.5.

The robot used in our experiments scans the environment using a Velodyne LiDAR, which has a 360° FoV horizontally and long sensing range, making it suitable for mapping large-scale objects or environments. In the experimental section (Section 3.6), I compare *Rear Side Entropy* and *Occlusion Aware* in field experiments.

Limitations

The implementation of information gain computation in this system has limitations that prevent it from enforcing certain preferred behaviours such as avoiding revisited scan poses. These limitations are not inherent issues of the original formulations proposed by Isler et al. [24], but problems in how the model is reconstructed in this system using LiDAR point clouds.

This system has been observed to direct the robot to scan poses that can scan a large amount of unknown open space and repeatedly plans NBV around these areas. The reasons behind such behaviours will be explained in Section 3.4.2. To penalise revisiting scanned areas, I have included a heuristic within the position cost $c_{\text{pos}_\mathbf{x}}$ instead.

3.4.2 Position and Traversal Cost

In my path planning module (Section 3.5), the RRT grows only within the traversable area of the elevation map, therefore the collection of scan candidates $\mathbf{X}_{\text{candidate}}$ does not contain invalid or unreachable poses. As a result, the utility value $u_\mathbf{x}$ of each scan candidate is penalised based on the nature of ANYmal and the configuration of the LiDAR system.

Position Cost

The position cost $c_{\text{pos}_\mathbf{x}}$ is defined as:

$$c_{\text{pos}_\mathbf{x}} = \begin{cases} 1 - d_{\text{thres}}^{-1} \times d_\mathbf{x} & d_{\text{thres}} \geq d_\mathbf{x} \geq 0, \\ 0 & d_\mathbf{x} > d_{\text{thres}}, \end{cases} \tag{3.7}$$

where $d_\mathbf{x}$ is the distance to an already visited scanning pose or the object itself, and d_{thres} is a user defined threshold.

In this system, $c_{\text{pos}_\mathbf{x}}$ is used to avoid rescanning a previously visited region and to maintain a reasonable distance between robot and object. As explained previously, the penality of revisiting is included to address a current limitation in the system.

Using *Occlusion Aware* volumetric information, the system plans NBV in regions where the robot can observe more void space. The main contribution to the information gain g_x is from void rays $\mathbf{r}_{\text{void}}$ — rays that do not hit any surfaces. Voxels $\mathbf{v}_{\text{void}} \in \mathbf{V}_{\mathbf{r}_{\text{void}}}$ are mainly unknown (occupancy probability $P_o(\mathbf{v}_{\text{void}}) = 0.5$). An unknown voxel produces the highest entropy based on Eq. (3.3). Compounded with the long scan range of LiDAR, the *Occlusion Aware* formulation will give significantly higher information gains for void rays $\mathbf{r}_{\text{void}}$ compared to other LiDAR rays, heavily biasing this system towards poses that scan open space.

Furthermore, the current implementation of this system takes in filtered LiDAR scans as inputs for reconstruction. These filtered point clouds do not provide information on void rays $\mathbf{r}_{\text{void}}$. Only when ray $\mathbf{r}$ hits a surface and generates a point in the point cloud, $P_o(\mathbf{v})$ of a voxel $\mathbf{v} \in V_{\mathbf{r}}$ is then updated to be occupied or known free voxels based on OctoMap [137] and the work of Isler et al. [24]. For a void ray $\mathbf{r}_{\text{void}}$, however, $P_o(\mathbf{v}_{\text{void}})$ does not get updated. As a result for *Occlusion Aware*, $\mathcal{I}_{\text{OA}}(\mathbf{v}_{\text{void}})$ will not decrease, causing the robot to stop exploring. I therefore included the penality of revisiting scanned areas within the position cost $c_{\text{pos}_{\mathbf{x}}}$ to avoid this behaviour.

On the other hand, $\mathbf{r}_{\text{void}}$ do not contribute to *Rear Side Entropy* volumetric information. Every scan decreases the entropy of observed voxels.

The proper solution to address this limitation in the system is to use raw LiDAR scans or data packets to correctly distinguish void rays, use void rays to update open space, and reduce the entropy of unknown voxels in open space accordingly. The position cost $c_{\text{pos}_{\mathbf{x}}}$ can instead represent the difficulty for the robot to conduct a scan at the candidate pose, such as challenging terrain.

In addition, if a scan candidate pose $\mathbf{x}$ is farther away from the object of interest, less rays in $\mathbf{R}_{\mathbf{x}}$ are able to observe the object, because Velodyne LiDAR rays have an uniform angular distribution across the sensor's horizontal 360° FoV. In the *Rear Side Entropy* formulation, the information gain of this pose $g_{\mathbf{x}}$ will therefore be lower compared to candidates that are closer to the object of

interest. The robot will be directed closer to the object of interest to ensure a high resolution scan, but scarifying the wide FoV that LiDAR provides. Hence, I include the position cost $c_{\text{pos}_\mathbf{x}}$ that applies to any viewpoint $\mathbf{x}$ too close to any surfaces so this system maintains a desired distance between the robot and the object of interest, to leverage the horizontal $360°$ FoV of Velodyne LiDAR.

Isler et al. [24] predefined a set of scan candidates in their system so that the distance of scan poses to the object surface was fixed. However, in this system, the distance between the robot and the scan surface is dynamic so that the robot can avoid obstacles in the environment. Furthermore, since the ANYmal operates on a 2.5D manifold, it is necessary for the quadruped to adjust the distance to the object surface so as to efficiently scan objects of different sizes. Combining Information Gain with a position cost based on distance to surfaces helps this system achieve a balance between coverage and resolution.

Traversal Cost

The traversal cost $c_{\text{tra}_\mathbf{x}}$ represents the difficulty for the robot to execute a certain path to candidate $\mathbf{x}$ because of the roughness of terrain and the distance.

Currently our approach classifies the elevation map discretely as either safe ($c_{\text{tra}_\mathbf{x}} = 0$) or not traversable ($c_{\text{tra}_\mathbf{x}} = 1$).

In addition, a constant traversal cost penalises scan candidates that are behind the robot, because large turns are more difficult for the robot to execute without causing significant localisation noise. This policy also encourages the robot to explore forward rather than alternating direction. This makes the system more time and energy efficient.

3.5 Path and Motion Planning

The path planning module in this system consists of two phases, as indicated in Fig. 3.6. Both phases rely on an elevation grid map generated from LiDAR measurements of the environment. The proposed system uses the approach of [154] to compute the slope and normal of each cell and in turn a measure

of the traversability of the terrain. The traversability is used to determine which states planned by the RRT and RRT* are valid and reachable.

In the first phase, the RRT grows into the traversable area without a goal until a user-defined number of nodes have been generated. These nodes are scan candidates $\mathbf{X}_{\text{candidate}}$. This system then compute the utility value $u_{\mathbf{x}}$ for each viewpoint candidate $\mathbf{x} \in \mathbf{X}_{\text{candidate}}$ independently and choose the NBV $\mathbf{x}_{\text{best}}$ with the highest individual value. This is similar to the approach of Vasquez-Gomez et al. [45], though Vasquez-Gomez et al. uses RRT to plan paths in their robot's configuration space. RRT helps ensuring that these scan candidate poses are both safe and reachable for the robot. Following that, the second phase of my path planning module uses RRT* to replan the route to NBV, optimising travel distance.

The current framework leverages standard techniques employed by state-of-the-art systems when it was developed. While the path planning module was designed to improve upon the work of Isler et al. [24, 25] for a more realistic and complex setting, this module was just a proof of concept for a legged robot to conduct active mapping operations in a real-world environment.

3.5.1 Termination Condition

In a model-free active mapping system, it is difficult to evaluate the completeness of reconstruction. I terminate operation using a user-defined threshold on the utility value u_{thres} after a planning sequence.

When the utility value of the NBV $u_{\mathbf{x}_{\text{best}}}$ falls below the threshold (Eq. (3.8)), no new scan candidate has satisfactory quality, and the active mapping procedure terminates.

$$u_{\mathbf{x}_{\text{best}}} < u_{\text{thres}} \quad \forall \mathbf{x} \in \mathbf{X}_{\text{candidate}}. \tag{3.8}$$

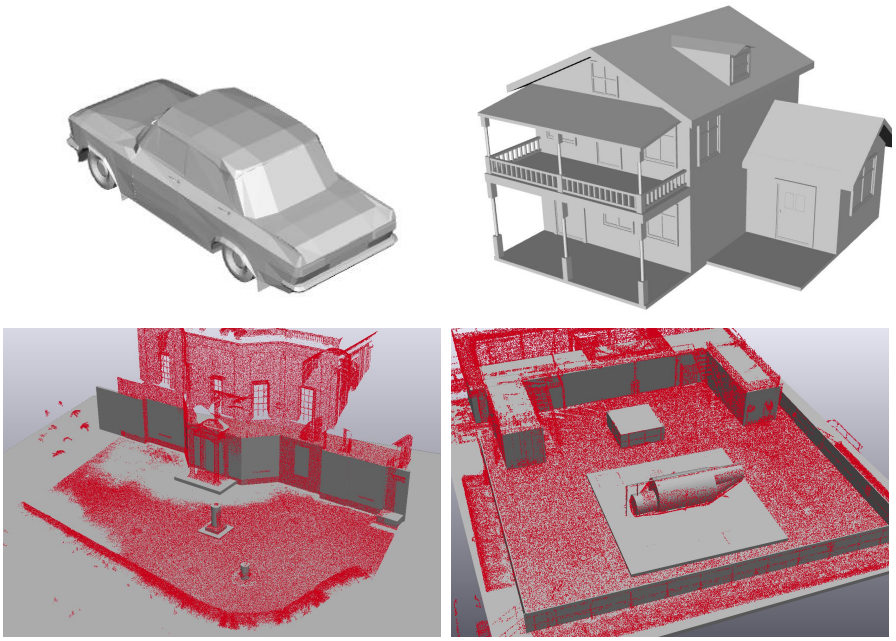


Figure 3.7: 3D models of a car and house (**top**) and of our real-world test sites (**bottom**) are used to evaluate our system. We created the 3D models using reconstructions made using a survey-grade LiDAR (in red).

3.6 Experiments

To demonstrate this system's functionality and to test the volumetric information formulations, we carried out experiments of increasing complexity — with the simple virtual models in Fig. 3.7 and Gazebo reconstructions of our envisaged test locations to assess this system's ability to avoid collisions. Finally, we deployed our system on the real ANYmal robot in these environments. The results are detailed in the following sections.

The real-world experiments involved scanning a building facade at Green Templeton College ($4\times35\,\mathrm{m}^2$) in Oxford (Fig. 3.8) and a mock-up helicopter on the oil rig training site at the Fire Service College ($3\times8\,\mathrm{m}^2$) in Gloucestershire (Fig. 3.1).

In these experiments, we used a LiDAR localisation system running on the robot's navigation computer. The system registered LiDAR clouds against a prior point cloud map using ICP [155] seeded with legged odometry. At the scale of our experiments, a deformable map representation was not needed.

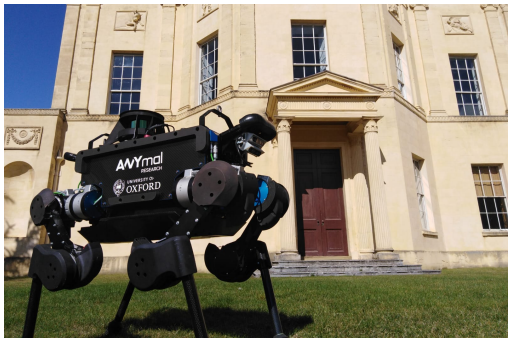


Figure 3.8: One of our experiments in Section 3.6 mapped this building facade at Green Templeton College, Oxford.

3.6.1 Hardware

Platform

The robot platform employed in this work is an ANYbotics ANYmal (version B) [61]. The robot has 12 actuated joints, as well as the 6 DOF floating base link. It is capable of trotting at the maximum speed of $1.0\,\mathrm{m\,s^{-1}}$ and traversing complex terrain, e.g. stairs, kerbs and ramps.

Sensor

The primary sensor of this system is a Velodyne VLP-16 LiDAR which has 16 laser beams spread across a $\pm 15.0°$ vertical FoV and measures ranges with the accuracy of $\pm 3\,\mathrm{cm}$ across full $360°$ horizontally.

Utilising the robot's wide range of motion, I designed a scanning action to roll the robot base and the LiDAR from $40°$ to $-40°$ (while standing). The action improves the vertical FoV of the presented system to $\pm 55°$ and allows mapping objects much taller than the robot. Using this action, the proposed system collects individual LiDAR sweeps.

3.6.2 Simulated Experiments

We conducted experiments in simulation to map models of a car and a house (Fig. 3.7 (top)). We then used a Leica BLK360 laser scanner to create accurate reconstructions of our two test sites, the facade of a building and a helicopter

deck (Fig. 3.7 (bottom)). I modelled the major surfaces of these test sites to create Gazebo simulations of the test sites.

In these experiments, an approximate location and size of the object of interest are known, which aids the extraction of useful measurements from the accumulated *sweep*. This informs our system about where the OctoMap should be constructed and the volumetric information be computed. We chose a $5\,\mathrm{cm}$ resolution for our OctoMap octree based on the angular resolution of the Velodyne LiDAR and expected measuring distance given the scale of our experiment sites.

For path planning and NBV selection, the presented system grows an RRT up to 150 nodes, every iteration, within a $12{\times}12\,\mathrm{m}^2$ elevation map centred around the robot. This allows the robot to plan and conduct mapping actions around the object.

To quantify the mapping results, I employ different criteria including point cloud coverage (c_p), travel distance (d_t) and number of scan actions (n_s). In addition, we compute the overall task time (t_{all}) as well as the average time per-scan spent computing information gains and determining the NBV (t_{nbv}) to evaluate the system's online feasibility in real-world robotic tasks.

To compute point cloud coverage, I aligned the accumulated point cloud with the ground truth and determined the points in the accumulated cloud within $4.3\,\mathrm{cm}$ of the nearest point in the ground truth, approximately the distance between the centre of our OctoMap voxel and its vertex ($\frac{\sqrt{3}}{2}{\times}5\,\mathrm{cm}$). These points are classified as observed.

Point cloud coverage c_p is then defined as:

$$c_p = \frac{N_\mathrm{O}}{N_\mathrm{GT}} \tag{3.9}$$

where N_GT and N_O are the total number of points in the ground truth model and the number of points observed in the model so far, respectively.

As summarised in Table 3.1, the point cloud coverage gained with *Occlusion Aware* Information Gain formulation is slightly higher than that with *Rear Side Entropy*. This can also be seen in Fig. 3.9, which demonstrates the point cloud

Object	IG	Experiments Evaluation c_p (%)	d_t (m)	n_s	t_{all} (mm:ss)	t_{nbv} (sec)
Car	OA	69.69	35.39	8	08:02	2.70
	RSE	69.01	40.18	10	10:21	2.46
House	OA	59.41	42.89	12	10:06	3.78
	RSE	58.98	44.64	12	11:17	3.65
Facade	OA	95.11	33.98	9	07:44	17.82
	RSE	94.24	37.82	9	08:21	19.69
Helicopter	OA	83.76	41.56	12	12:26	5.45
	RSE	83.01	43.61	13	13:06	5.20

Table 3.1: Comparison between two volumetric information measures in simulation environments (OA - *Occlusion Aware*; RSE - *Rear Side Entropy*).

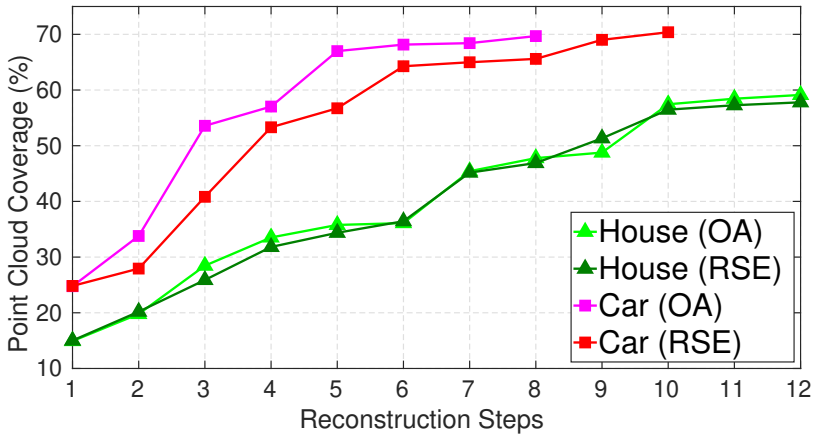


Figure 3.9: Point cloud coverage per step for the car and house models.

coverage per scan. The maximum coverage never reaches 100 % in our system as the top surfaces of objects are higher than the robot and cannot be observed from the ground, as shown in Fig. 3.12.

While there is on average an 8.5 % reduction in travel distance when our system employs *Occlusion Aware* compared to *Rear Side Entropy*, this particular system is also subject to the random scan candidate placement by RRT. Hence the performance difference between two volumetric information formulation in simulation so far is not significant enough for us to make a conclusive decision on which is the better formulation. Both approaches allowed my system to accomplish the mapping task. In both cases, the travel distance, the overall run time and the NBV computation time are all feasible for real experiments.

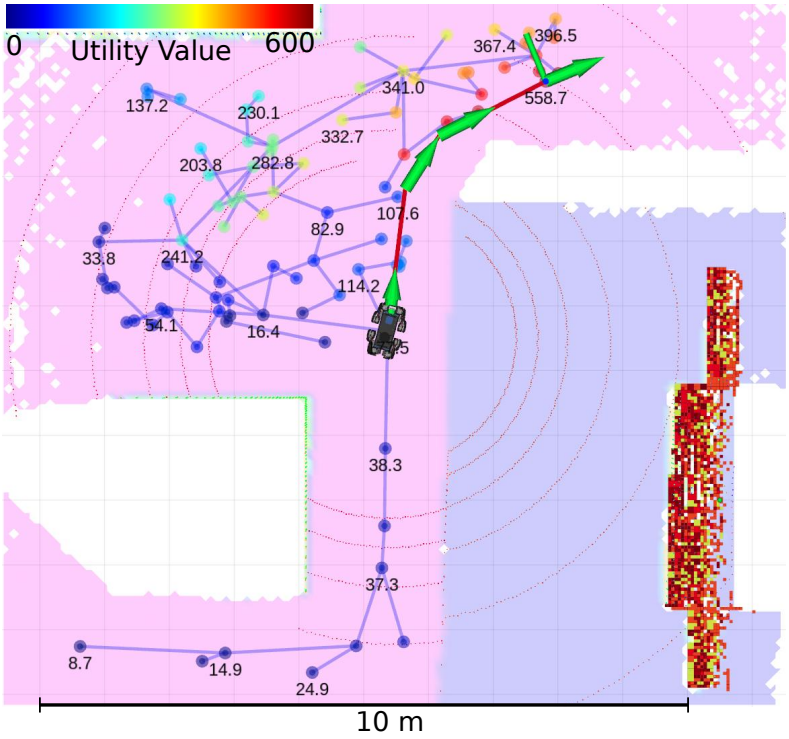


Figure 3.10: Illustration of the presented system mapping the simulated model of the helicopter/oil rig site. The system grows an RRT around the current pose of the robot in the traversable area. The utility metric is computed at the tree nodes. The node with the maximum utility is selected as the NBV. Finally, using the RRT* algorithm, the path from the current pose to the NBV is replanned.

3.6.3 Real-World Experiments

Based on the simulated results in the previous section, we used the *Occlusion Aware* volumetric information gain metric in our real-world experiments.

Table 3.2 summarises the evaluation of the reconstruction results in both experiments. Compared to experiments in simulation where the floors were all perfectly flat and complete, in the two presented real-world experiments the ground were grass and metal grating respectively; therefore elevation maps in the presented real-word experiments contained significantly more unknown cells than those in simulation experiments. In addition, the LiDAR sensor is just $\sim$70 cm from the ground, so we can only plan in a $7\times7\,\mathrm{m}^2$ area around the robot. We therefore decreased the number of RRT viewpoint candidates from 150 to 75, consequently decreasing the computation time of determining the NBV t_{nbv}.

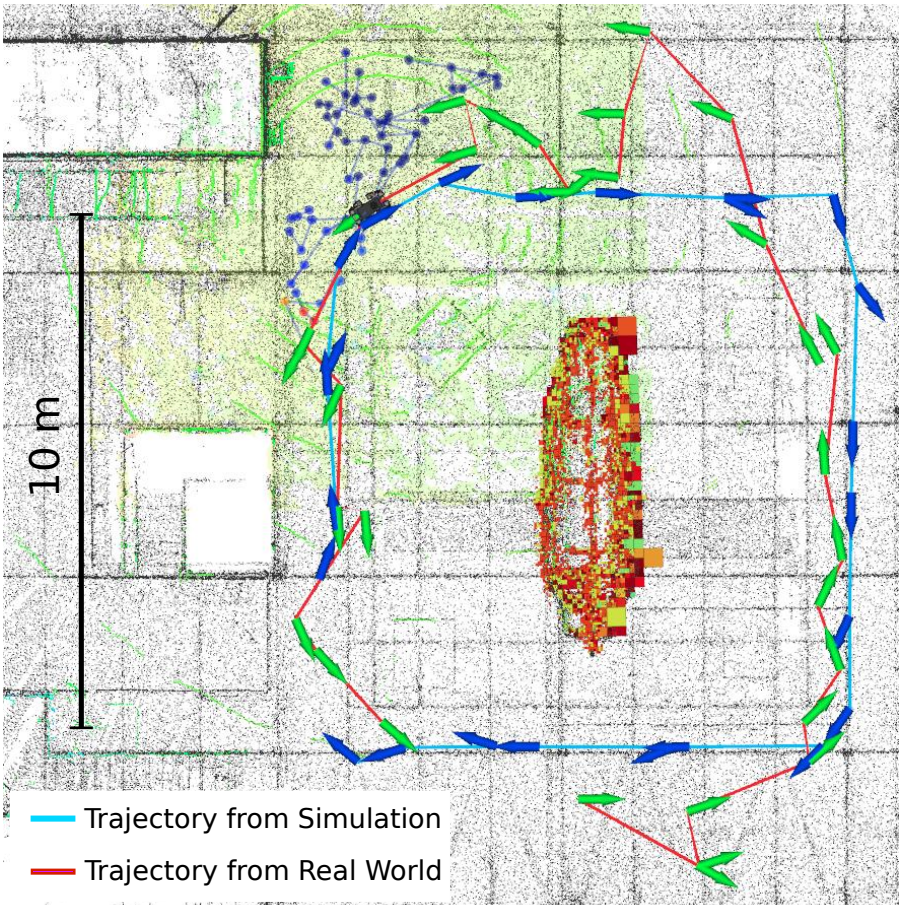


Figure 3.11: Example of our system mapping the real helicopter. The route the robot took in the real experiment is shown in red with the reconstruction of the helicopter (as an octree) shown in the centre. Also illustrated, in blue, is the route taken by our method running in simulated model, as a comparison.

	Experiments evaluation				
Object	c_p (%)	d_t (m)	n_s	t_{all} (mm:ss)	t_{nbv} (sec)
Facade	88.06	37.61	15	20:56	9.82
Helicopter	78.60	49.19	26	35:25	2.00

Table 3.2: Results for our system in the real-world experiments.

Comparing Table 3.2 with Table 3.1, the computation times for the real experiments at facade and helicopter locations are on average half of the time taken in simulation.

The presented approach allows the robot to avoid the mobility hazards for the helicopter experiment in Fig. 3.1: stairwells, open edges on the deck and a skirt around the helicopter.

Fig. 3.11 presents a comparison between the robot trajectories in the real

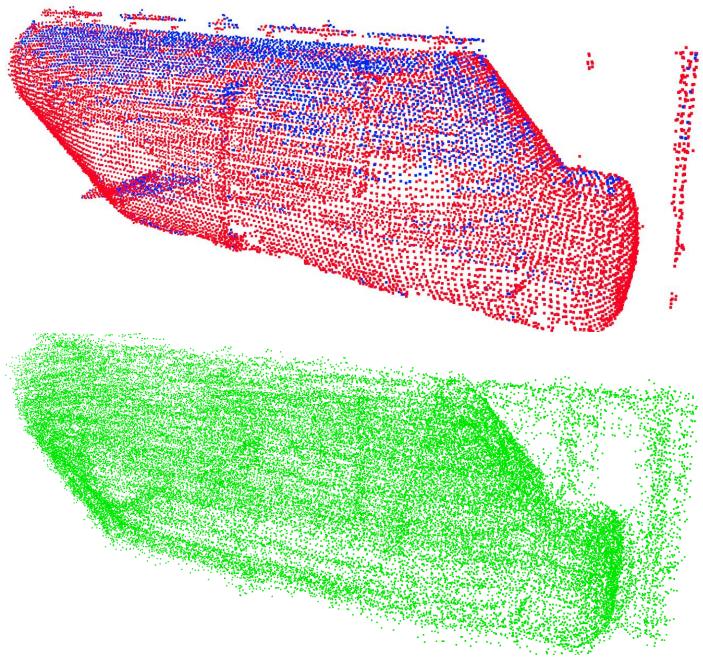


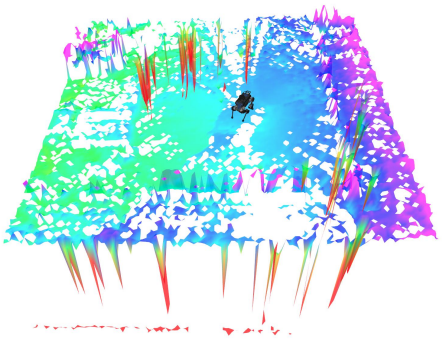
Figure 3.12: Top: Point cloud coverage in the real helicopter experiment. Red cloud indicates the observed area and blue represents the unobserved part of the helicopter. **Bottom**: Our system succeeded in reconstructing the helicopter body.

Fire Service College helicopter experiment (counter-clockwise) and in simulation (clockwise). The paths taken by my system in both scenarios are similar, demonstrating the practicality of the presented system in real scenarios.

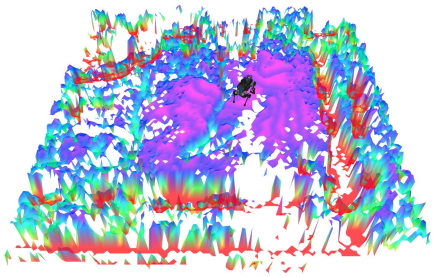
Fig. 3.12 demonstrates the success of the presented system in reconstructing the helicopter body (in green), compared to the ground truth (in red). Due to the limitation in the elevation map and in turn my path planning module, this system planned more scans in real-world experiments than in simulation.

In the helicopter scenario, the number of scans required more than doubled and as a result the run time almost tripled. While a major contributor to the run time was the time spent by the robot operator judging if planned paths were safe, the significant increase in planned scans was the result of the quality of the elevation map. The LiDAR was able to scan through the floor railing (Fig. 3.13), resulting in holes and significantly low readings in the elevation map (Fig. 3.14a). This in turn lead to limited traversable areas, as indicated in Fig. 3.14b. As a result, the path planned was not as efficient as that in the simulation, as presented in Fig. 3.11.

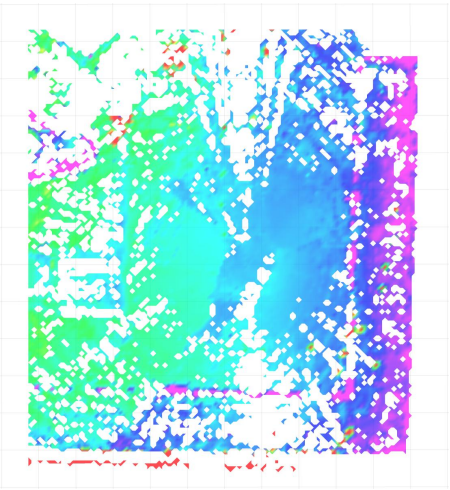
Figure 3.13: An example view of the forward facing camera on ANYmal in the real-world experiment, showing the steel wire mesh flooring.



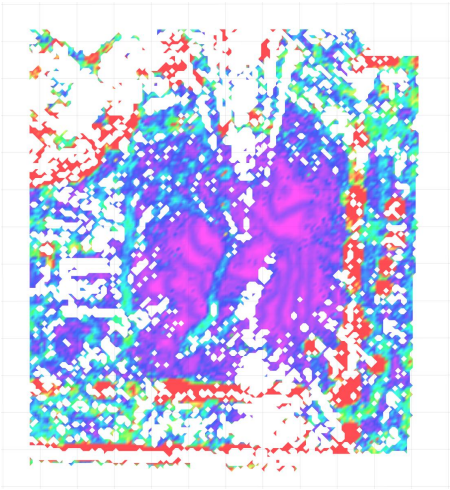
(a) An example of the elevation map during the experiment.



(b) An example of the traversability map during the experiment. The higher area (purple) was traversable.



(c) Top down view of the elevation map



(d) Top-down view of the traversability map

Figure 3.14: Example of the elevation and traversability map.

3.6.4 Limitations

Section 3.4.1 and Section 3.4.2 discussed several limitations in information gain and position cost within the current implementation of this system. This system cannot use information gain alone to enforce the behaviour of avoiding revisiting scanned poses, and specific penalities have been introduced into the position cost to address this. The proper solution to this problem, however, is to use raw LiDAR packets to correctly detect void rays and update free open space accordingly. View planning based on information gain should then be able to prevent the robot from revisiting scanned areas by itself.

In addition, the traversal cost can be improved from the current discrete classification into a continuous function, taking into consideration the quantified traversability along the path as well as other factors such as the path length. Incorporating full traversability estimation will allow the robot to navigate over rough terrain or challenging environments, such as kerbs and ramps, so as to fully utilise the dynamics of a legged robot.

These real-world experiments further revealed several limitations in this iteration of the system on reconstruction accuracy. First, the sweeping motion incorporated in this system caused misalignment in surfaces in each LiDAR point cloud. Modern 3D LiDAR sensors collect laser range returns continuously over time. A complete 360° scan is accumulated and then made available to downstream applications. Therefore the high dynamics of a legged robot, especially in rotation, leads to motion distortion. In the specific case of the system in [10], the raw measurements of LiDAR was further filtered into an unorganised point cloud before being used for reconstruction, and distortion persisted in each LiDAR scan. A highly dynamic motion such as a *sweep* therefore caused such surface misalignment as demonstrated in Fig. 3.15. In the presented experiments of this initial framework, the effect of motion distortion was minor because of the limited scale of the experiment site, and was mitigated via point cloud filtering

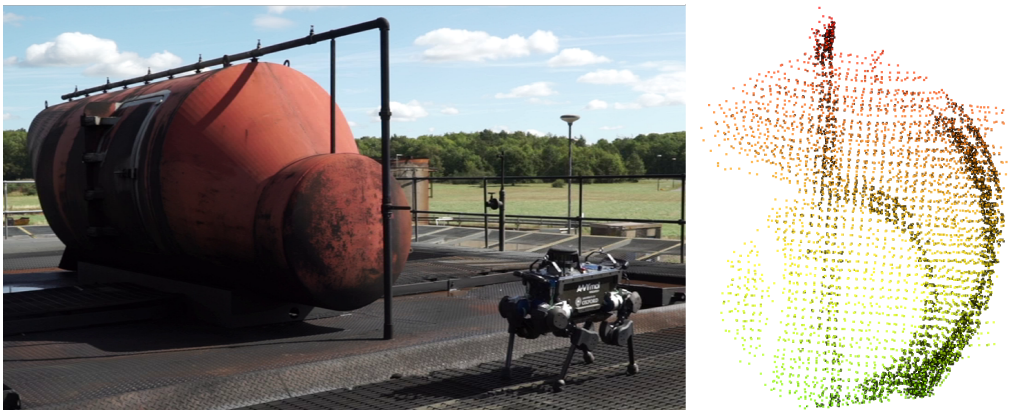


Figure 3.15: An example of distorted point cloud during the real-world Fire Service College experiment. **Left**: the pose of the robot when taking the example *sweep*; **right**: the front view of the *sweep* point cloud, before filtering and smoothing. This *sweep* contained a double-layered surface on the model body due to motion distortion.

and smoothing. However, it will present a more significant challenge in large-scale environment or with a more dynamic robot. Chapter 4 will present this issue in more detail.

In addition, the system presented in [10] was coupled only with an odometry system. In the limited space of the presented experiment settings, this odometry system demonstrated almost no drift, hence there was no need to incorporate a SLAM system. However, for exploration in larger environments, it is necessary to incorporate SLAM and loop closure, such as AEROS by Ramezani et al. [113], to address the unavoidable odometry drift. This in turn requires the dense volumetric reconstruction to be elastic so that the effect of loop closure is not only applied to the point cloud map used in LiDAR SLAM but also to this system's dense volumetric reconstruction. Ho et al. [138] proposed the technique Virtual Occupancy Grid, but it was designed for environments of smaller scales and sensors with shorter range. This will be explained in detail in the follow chapter (Section 4).

Last but not least, OctoMap [137] has a limited integration speed, memory efficiency and scalability. The ray-casting step in point cloud integration was feasible given the scale of the experiment in this presented system, but for much larger environments, it will become significantly more time consuming, espe-

cially if the $5\,\mathrm{cm}$ resolution is maintained. The memory required to store the whole reconstruction is also scaling poorly with OctoMap as the core. An alternative reconstruction technique with adaptative resolution, such as *supereight* proposed by Vespa et al. [50], is therefore very attractive to this thesis.

3.7 Conclusion

This chapter presents an active mapping system using OctoMap [137] that does not rely on a prior model for planning or navigation. This presented system allows a quadruped robot to explore and reconstruct both small and large scale objects, in particular industrial assets, with few assumptions about the test environment and requiring only high level human supervision. It has been tested in fully realistic scenarios and allowed the robot to accomplish mapping missions in a complicated environment, creating accurate reconstructions online.

To this end, this system was a successful proof of concept that allows for deploying and testing reconstruction techniques, as it was able to complete the active mapping task in our real-world experiments. However, it has a few inherent limitations in its implementation. The reconstruction process cannot handle void rays and one of the information gain formulations implemented will bias this system towards repeatedly scanning unknown open space. I incorporated a heuristic within the position cost to address this undesired behaviour, but an improved system should be able to achieve such functionality using information gain alone. Another limitation is the traversal cost, which can better represent the terrain traversability along the planned path than the discrete classification that is implemented in the current system.

These real-world experiments also revealed several key limitations in the reconstruction core. The first is the effect of motion distortion in LiDAR measurements and the impact of it on the quality of reconstructed surfaces. The second is the map rigidity in conventional reconstruction methods unable to incorporate the SLAM loop closure correction. The third is to achieve a balance between the integration and memory efficiency of the system, and the scale and

resolution of the map. To address these limitations, this thesis extended the multi-resolution reconstruction pipeline *supereight* [50] and developed SE-Atlas, which will be presented in detail in the following chapter.

4

Elasticity, Efficiency and Scalability in Large-Scale LiDAR Reconstruction

This chapter includes elements of the following publications:

[11] Y. Wang, N. Funk, M. Ramezani, S. Papatheodorou, M. Popovic, M. Camurri, S. Leutenegger, and M. Fallon. "Elastic and Efficient LiDAR Reconstruction for Large-Scale Exploration Tasks". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2021, pp. 5035–5041

[12] Y. Wang, M. Ramezani, M. Mattamala, and M. Fallon. "Scalable and Elastic LiDAR Reconstruction in Complex Environments Through Spatial Analysis". In: *Proc. of the European Conference on Mobile Robotics (ECMR)*. Aug. 2021

[13] Y. Wang, M. Ramezani, M. Mattamala, S. T. Digumarti, and M. Fallon. "Strategies for Large Scale Elastic and Semantic LiDAR Reconstruction". In: *J. of Robotics and Autonomous Systems (RAS)* [2022]

Acknowledgements
I would like to acknowledge the contributions of Nils Funk, Sotiris Papatheodorou and Prof Stefan Leutenegger in maintaining and developing the original *supereight* tracking and reconstruction pipeline, especially the adaptative resolution feature of *supereight* that the proposed system relies on.
I would also like to thank Matias Mattamala and Dr Milad Ramezani for deriving the proposed relative uncertainty formulation with me.

This chapter describes the core 3D LiDAR reconstruction module, referred to

as *supereight* Atlas (SE-Atlas), which is the heart of a broader larger elastic SLAM pipeline. SE-Atlas started as a collaboration with the Smart Robotics Lab (SRL) at Imperial College London (ICL) [11], and was later expanded to include many new features such as spatial overlap analysis and relative uncertainty analysis [12]. It addresses the challenges of large-scale reconstruction and exploration tasks that have been revealed by real-world experiments in Chapter 3. This chapter will briefly explain the background of SE-Atlas — a multi-resolution reconstruction pipeline *supereight* designed by Vespa et al. [50], and then focus on my contributions to SE-Atlas that expand *supereight* to integrate long-range LiDAR measurements, and improve LiDAR scan integration accuracy and efficiency, long-term exploration map scalability, and reconstruction elasticity upon SLAM loop closure. Additional features of SE-Atlas related to the incorporation of semantic information are explained in the following chapter (Chapter 5).

4.1 Introduction

Dense surface reconstruction is an active research topic. Being able to recover rich geometric information in real time is important for applications such as active mapping [26, 148, 150], obstacle avoidance [4] and industrial inspection [122, 124]. While Building Information Models (BIMs) are commonly available for modern buildings, there are scenarios where these models no longer represent the real situation, e.g. after renovations or disasters.

Large-scale and outdoor environments further pose challenges such as map scalability and scan integration efficiency. Although systems have been developed to reconstruct models of these scenarios offline using point clouds from laser scanners, autonomous exploration and reconstruction in large-scale outdoor environments or multi-storey scenarios is still an open challenge in mobile robotics. This has been the motivation of international competitions such as the DARPA SubT Challenge [19, 156, 157]. Real-world experiments of our preliminary active mapping system [10] demonstrated a series of challenges, as explained in Chapter 3.

A major challenge in reconstruction is global consistency, because the accumulation of some degree of odometry error is unavoidable during large-scale exploration — even for highly accurate LiDAR localisation approaches such as LOAM [54]. This odometry error increases with the distance or duration of exploration by the robot; hence in large-scale environments or long-term exploration tasks, odometry drift poses a significant impact on the accuracy of the reconstruction. As explained in Section 2.4 and Section 2.5, this error is typically corrected using loop closures and pose graph optimisation in the context of SLAM. A factor graph-based SLAM system optimises the whole history of the robot and propagates pose correction back through the trajectory.

Within a SLAM system, these corrections are also applied to the internal map representation. In a LiDAR-based SLAM system, the map representation is typically an accumulation of individual scans, each associated with a corresponding scan pose. In this case the map correction is a simple process. When SLAM corrects the history of poses during optimisation, each scan can be re-projected into the map again using updated scan poses to constitute a globally consistent map. However, a point cloud is not the most suitable map representation for exploration and path planning purposes as it does not have a sense of volume and free space. Conventional dense reconstructions (such as surface mesh or voxel map) for exploration and navigation, on the other hand, are typically rigid when they are constructed on-the-fly by an exploring robot, making it difficult to incorporate the effect of loop closures. Therefore, the first problem that SE-Atlas is designed to address is the rigidity of 3D reconstructions. Instead the goal is to update reconstructions online and improve reconstruction accuracy based on SLAM loop closure corrections.

Another challenge is finding a good trade-off between the resolution/scale of the reconstruction, and the speed/efficiency of the system. A precise representation of occupancy is important for robot path planning, especially when planning paths through tunnels and door ways [19], some examples of which are given in Fig. 1.1 and Fig. 4.1.

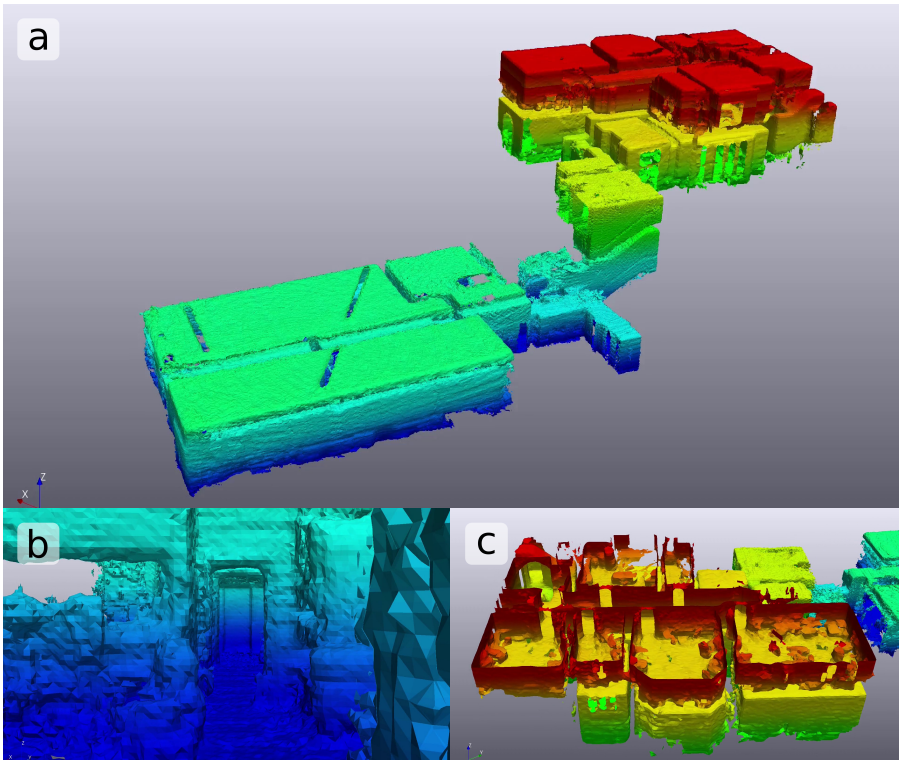


Figure 4.1: The proposed system has been deployed in a multi-storey indoor environment, Oxford Robotics Institute (ORI). **a**: Mesh reconstruction of three floors coloured by height, including the basement, the ground floor and the first floor; **b**: The clean representation of the narrow corridor exiting the basement has been achieved via thanks to the high mesh resolution; **c**: The reconstruction of the first floor with ceiling removed - all doorways have been clearly reconstructed.

The goal is to improve the speed of integrating long-range LiDAR scans and reduce the memory usage of high-resolution 3D reconstruction compared to state-of-the-art techniques such as Voxgraph [5]. Hence we collaborated with the SRL at ICL and leveraged *supereight* [50, 51], a state-of-the-art localisation and reconstruction framework for RGB-D cameras. Because *supereight* does not provide loop closure detection and is tailored for RGB-D cameras, the proposed system uses *supereight* only for reconstruction and relies on external odometry [81] and SLAM [113, 158] modules that are developed by my colleagues in our group and have been proved reliable for LiDAR-based localisation tasks. In this work, I greatly expand *supereight* to incorporate 3D multi-beam LiDAR scans. In Section 4.10.3, experiment results demonstrate that this approach is more efficient than other state-of-the-art pipelines at high resolutions based on metrics such as integration speed and memory usage.

Supereight and SE-Atlas provides the ability to store either a TSDF or occupancy probability. In this work, I experiment with both representations but focus more on the occupancy representation, because the latter explicitly models free space for path planning and navigation as well as for spatial overlap analysis. Overall, the proposed system contains a front-end and a back-end. The front-end relies on accurate multi-sensor odometry to build a local occupancy map around the robot. In the back-end, I implement a novel technique to cluster the SLAM factor graph and reconstruction, inspired by large-scale systems such as the *Atlas* SLAM framework [114]. Instead of building a single global map, the back-end creates submaps and associates them with the corresponding pose from an externally estimated SLAM pose graph. This approach can account for subsequent loop closure corrections and achieve a globally consistent reconstruction.

I further present strategies for spawning and fusing submaps based on geometric understanding of the observed spaces, enabling on-the-fly segmentation of areas that are physically isolated or significantly different from one another, e.g. individual rooms. The proposed system also fuses together submaps with significant overlap to reduce redundant reconstruction. Fig. 4.2 demonstrates the volumetric map produced by the proposed system using data from a 3D LiDAR on a mobile robot, such as a three-storey building. The overarching goal of the work is to achieve scalability of the reconstruction with the size of the environment instead of the exploration length, by controlling the growth of the number of submaps and memory consumption*. In addition, by computing the relative uncertainty between pairs of poses in a SLAM factor graph, my system can avoid incorrectly fusing submaps to improve overall reconstruction accuracy.

Last but not least, the real-world experiments of my active mapping prototype described in Chapter 3 demonstrate that LiDAR reconstruction is subject to motion distortion when the sensor is experiencing highly dynamic movements.

*Sparsification of the underlying SLAM pose graph is a related research topic which we do not explore in this work.

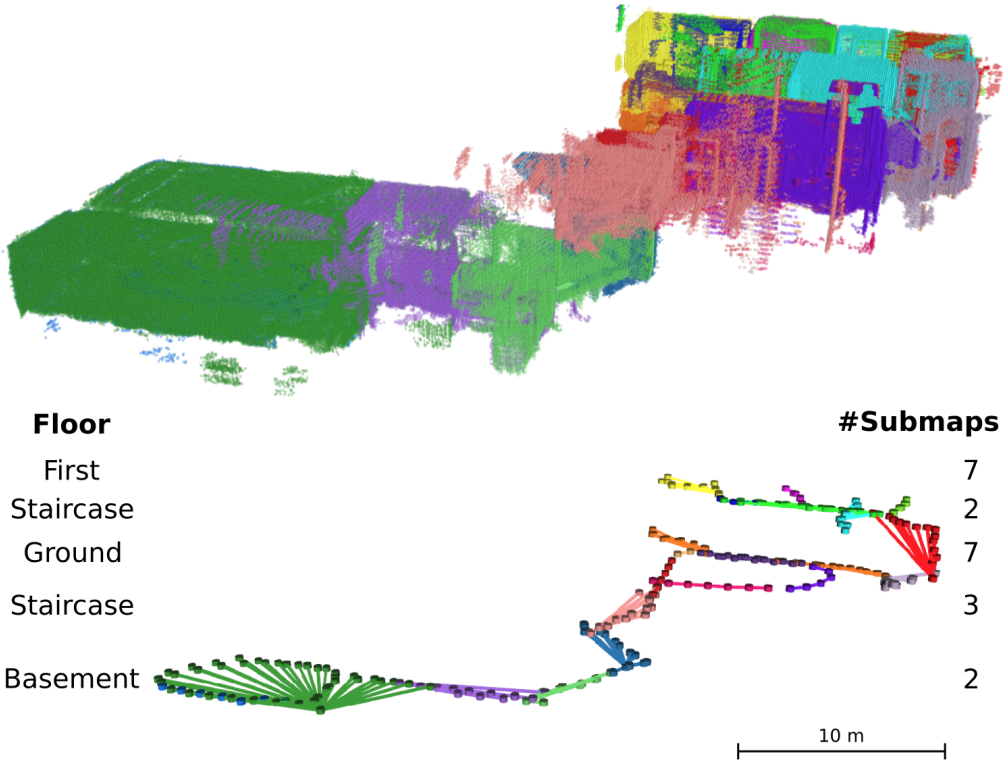


Figure 4.2: Our proposed room segmentation method has been tested in the same multi-floor multi-room indoor environments as Fig. 4.1, ORI. **Top**: The reconstruction result of ORI, in which each room is segmented by a submap indicated by a unique colour; **Bottom**: The clustered pose graph nodes using spatial overlap analysis using the same colours as their corresponding submaps.

Accounting for motion distortion when integrating LiDAR measurements is another problem that I address in SE-Atlas, which will be evaluated using reconstruction accuracy in Section 4.10.

The contributions of the proposed system are the following:

- An elastic 3D reconstruction system that can support corrections to its underlying shape, e.g. due to loop closures.
- A SLAM factor graph clustering and submap fusion strategy using probabilistic and spatial understanding that allows the reconstruction's memory usage to grow proportionally with the size of the environment rather than the duration of exploration.
- Incorporation of LiDAR into a state-of-the-art reconstruction framework, *supereight*, which achieves multi-fps (3 Hz) full range (60 m) LiDAR scan

integration with high resolution ($\sim$5 cm) to enable high precision motion planning and long range autonomy.

- Details of modifications to the *supereight* LiDAR reconstruction pipeline [51] to incorporate motion undistortion.

- A new formulation for relative uncertainty derived from the work of Mangelson et al. [7] and GTSAM [8], and a formal treatment of uncertainty in submap fusion.

Section 4.10 demonstrates the performance of SE-Atlas using both simulation and real-world datasets. I focus on its improvement in specific metrics such as integration speed, memory usage, system scalability, reconstruction accuracy, and the capability of room segmentation.

4.2 Literature Review

This section reviews existing works that are most relevant to the proposed system in this DPhil project. We first provide a detailed discussion on different map representations used in active mapping systems in Section 4.2.1. Approaches that address the challenge of map elasticity and scalability in long-term large-scale reconstruction and exploration tasks are discussed in Section 4.2.2. Finally, techniques that segment reconstructions into individual enclosed spaces are reviewed in Section 4.2.3.

4.2.1 Reconstruction

Section 2.6 briefly reviewed different reconstruction representations used in state-of-the-art active mapping systems. We categorised them into two main categorises, namely surface mesh and volumetric occupancy reconstruction. In this section, we expand the explanations on both categories and further include less common representations such as point cloud.

Surface Representation

Classical surface-based active mapping systems first generate triangular meshes based on sensor observations, e.g. point clouds, to represent the surface of the object. Surface mesh reconstruction provides detailed information regarding the surface geometry of the object of interest, and clearly defines the boundaries of the scanned surface, making them suitable for frontier-driven active mapping methods.

The model-based active mapping system of Hollinger et al. [122] and the model-free system of Hover et al. [124] both utilised Poisson surface reconstruction [125] to reconstruct ship hulls. Hollinger et al. focused on reinspecting regions in a prior mesh that were mostly likely to be poorly reconstructed initially, i.e. high mesh uncertainty. They then planned focused surveys towards high-uncertainty areas to improve the mesh quality. Hover et al. designed their system for a high-resolution mesh reconstruction. They focused on the coverage information given by the mesh, so that their system could conduct a detailed inspection survey and guarantee full coverage over the hull.

In a model-free system, such as the early work of Kriegel et al. [127], a surface mesh representation also has the advantage of providing clear definition on boundaries and holes. The system of Kriegel et al. iteratively sought the boundaries of the current mesh and planned scan candidates along them to expand the reconstruction. The closest candidate was selected as the NBV. To avoid collision, they defined a bounding box surrounding the object of interest. The sensor was planned to stay outside the box. Such a design limited the application of this system because it assumed that there were no obstacles outside the bounding box. The system of Kriegel et al. was designed to scan an object of small scale using a push-broom laser profiler on a manipulator arm in an ideal laboratory environment. In this case, assuming the environment being obstacle-free is justifiable. This assumption cannot be made in many outdoor real-world scenarios such as an industrial setting, because a simple bounding

box around the object of interest in these scenarios cannot always guarantee avoiding all obstacles.

One major limitation of surface representation is the computation time required to generate the mesh, especially in model-free systems where real-time feasibility is required. In real-world experiments, the process of generating a triangulated mesh often requires filtering and smoothing noisy point clouds. Hover et al. [124] mitigated this disadvantage by generating the mesh offline after conducting the identification survey, similar to the post-process mesh generation method proposed by Gopi et al. [159] and Lipman et al. [160]. However, when the point clouds are accumulated incrementally online, the mesh generation needs to be able to process a stream of point clouds. Recomputing the mesh from the whole cloud every iteration is not always feasible in real-time applications. Kriegel et al. [127] relied on a custom surface mesh generation method [126] to address this challenge. The streaming point clouds were converted into surface mesh and inserted into the existing reconstruction incrementally.

A solution to reduce the need for point cloud filtering and smoothing is voxelising the 3D space. Map representations such as TSDF and ESDF use a voxel map to store in each voxel the distance to surface and the corresponding weight/confidence of the distance measurement. The Marching Cubes [129] algorithm can then be used to convert them into a mesh reconstruction. For example, the KinectFusion system by Newcombe et al. [48] employs a dense TSDF volume. This is a mapping and tracking system capable of reconstructing room-sized scenes in real time. It continuously registers the depth data streamed from a Kinect sensor against a global TSDF model using a coarse-to-fine ICP algorithm. KinectFusion, however, requires a GPU to maintain the real-time capability.

For large-scale reconstructions, Niessner et al. [132] expanded a TSDF volume using a spatial hashing scheme. By only storing data densely around surfaces, the system of Neissner compresses space and takes advantage of the

sparsity of large-scale environments to ensure system scalability. In addition, the use of a hash table to store TSDF information allows for real-time data access and update. Similarly, BOR^{2}G by Tanner et al. [133] used a hashing scheme named Hashing Voxel Grid (HVG) for efficient large-scale TSDF reconstruction in experiments of multiple kilometres long. Different from the work of Newcombe et al. [48] and Niessner et al. [132], which focused on short-range Kinect sensors, BOR^{2}G further employed long-range LiDAR in their reconstruction experiments.

Most recently, Schmid et al. [6] proposed an active mapping system named *GLocal*. The reconstruction module of GLocal is a submap-based TSDF and ESDF mapping technique called Voxgraph [5], which is in turn based on Voxblox by Oleynikova et al. [152] and C-blox by Millane et al. [134]. Related to this work is the frontier-driven active mapping system by Kompis et al. [148], which also uses Voxblox. Voxblox incrementally builds a TSDF from streaming point clouds using the voxel hashing approach of Niessner et al. [132]. It then computes ESDF based on existing TSDF to require distance information to the closest obstacles, which is then used for path planning and collision avoidance. In order to explicitly represent known free space for path planning, the SDF reconstructions in Voxblox densely map all the free space between observed surface and the sensor, instead of only focusing on surfaces. Reijgwart et al. [5] have demonstrated that Voxgraph can integrate LiDAR measurements in real time on a single CPU core. However, densely integrating all the free space in LiDAR scans is computationally heavy and time consuming [4, 134]. Millane et al. [134] implemented highly efficient ray-casting methods to improve scan integration speed, but the dense conversion from TSDF to ESDF remains a major contributor to computation time. In our outdoor experiments (Section 4.10.3), the typical parameters for Voxgraph to maintain real-time LiDAR scan integration and SDF conversion were $16\,\mathrm{m}$ range with $20\,\mathrm{cm}$ resolution [11]. With such parameters, it would be challenging to plan robot trajectories at full LiDAR ranges.

Volumetric Occupancy Representation

A volumetric occupancy representation similarly discretises space into uniform voxels and stores occupancy information in each voxel. The occupancy information provides several advantages, such as explicit representations of known free space that is distinct from unknown space. Path planners can leverage such distinction to find safer routes. Additionally, occupancy information allows volumetric active mapping systems to decide NBVs based on a quantitative metric known as Information Gain, which measures the expected improvement in the reconstruction if a scan is conducted at a certain pose.

A simple example of utilising occupancy information is given by the system of Blaer and Allen [145]. Blaer and Allen categorised voxels using discrete labels of **unseen**, **seen-empty** and **seen-occupied**. They then defined Information Gain as the number of **unseen** voxels that each candidate scan was expected to observe, and used this formulation to direct the sensor and the robot. This approach is however unable to address the inaccuracy and uncertainty in sensor measurements. If a point was observed behind an occupied voxel, Blaer and Allen [145] classified both voxels as **seen-occupied**, assuming that the occluding voxel in the front was only partially occupied. Given the large scale of their site of interest ($300 \times 300\,\mathrm{m}^2$) as well as their low voxel resolution ($1\,\mathrm{m}$), making such an assumption would not impact the reconstruction accuracy greatly. Their Information Gain was sufficient as a guidance for sensor placement in their system, but discrete labelling will not suffice in the case where higher accuracy and resolution is desired.

For more complicated tasks, occupancy information needs to represent the probability of a voxel being occupied. For instance, Hornung et al. [137] presented an open-source mapping system, OctoMap, for 3D volumetric representation based on octrees. Octrees are a commonly used data structure for efficiently rendering and compressing of point clouds, both static [161] and streaming [162]. An example of an OctoMap representing a point cloud is demonstrated in Fig. 4.3.

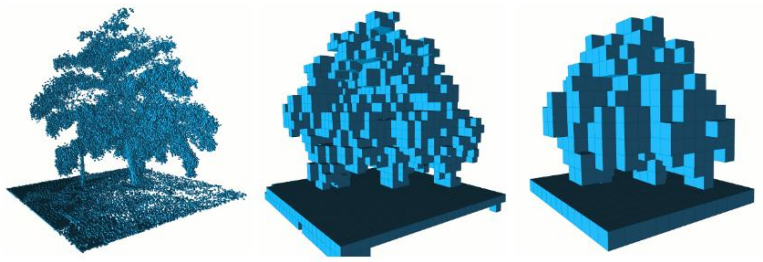


Figure 4.3: Example of OctoMap being used to represent a tree with different resolutions (0.08 m, 0.64 m, 1.28 m) [137]

OctoMap has three main advantages. First, it can store the occupancy probability of observed voxels to address the uncertainty in measurements for a more accurate representation. This also allows for data fusion among multiple sensors and robots. Second, it is capable of representing unobserved areas. This is important for both path planning and exploration. Unobserved areas should be assumed as occupied and avoided in path planning, but is instead the focus of exploration. Last but not least, compared to the single resolution map that voxel hashing methods commonly used [5, 132, 133], Octree data structure provides an intuitive method to maintain multiple resolutions in the same map. Leveraging this feature, OctoMap presents an advantage in memory consumption by storing occupancy information at different resolutions in the reconstruction, which in turn reduces voxel count and map memory usage.

Based on OctoMap, Isler et al. [24] proposed a collection of Volumetric Information formulations for a voxel, namely *Occlusion Aware*, *Unobserved Voxel*, *Rear Side Voxel*, *Rear Side Entropy* and *Proximity Count*. They were designed for Information Gain computation by measuring the entropy/uncertainty in each voxel. By reducing the entropy in the scanning area, these formulations achieved two purposes - exploring unknown spaces in the environment as well as focusing on reconstructing the object of interest. Delmerico et al. [140] then conducted extensive experiments to evaluate the performance of these Volumetric Information formulations for NBV selection, comparing them with the approaches of Kriegel et al. [163] and Vasquez-Gomez et al. [164]. These experiments were conducted in simulation. The authors also demonstrated the

performance of their Volumetric Information formulations in active mapping on a KUKA Youbot with a 5 DOF arm in an office room.

Their experimental setup however was not realistic in real-world scenarios. A set of 48 scan candidate poses were predefined to be uniformly distributed on a bounding space that encased the object of interest. They all faced towards the object, had approximately the same fixed distance to the surface and were guaranteed to be safe. Additionally, the difficulty of traversal between these scan candidates, such as path planning and collision avoidance, was not addressed in their systems. These are all problems that need to be considered in systems that are designed for real-world scenarios.

Bircher et al. [147] designed a completely model-free active mapping system for exploring unknown sites that also utilised OctoMap. They later expanded it into a model-based inspection system [26]. Both of their systems had the iterative structure typical among model-free systems, as explained in Section 3.2. Compared to the approach of Isler et al. [24], scan candidates in the system of Bircher et al. were generated by growing an RRT within known free space. Each node of the tree was considered as a candidate and had its Information Gain computed. Such an approach leveraged the explicit representation of known free space in volumetric occupancy maps, and addressed the path planning and collision avoidance problem necessary in real scenarios. In the model-free system [147], the Information Gain estimated the amount of observable unknown space; in the inspection system [26], it estimated the number of voxels that were on the object surface and observable. Then the system of Bircher et al. found the branch with the highest total gain. This branch defined the path the robot should take.

Another key component of their systems was the receding horizon strategy. When the robot finished exploring the first edge of the previous branch, their system recomputed a new best branch based on the newly explored environment. Such a strategy granted the system good scalability against large-scale and complex active mapping problems.

However, compared to voxel hashing methods, Octree structures are less efficient in data access and update. Vespa et al. [50] therefore proposed *supereight*, a highly efficient multi-resolution reconstruction pipeline. *Supereight* uses an Octree on the high levels of their data structure, and aggregates the last levels of the tree into contiguous blocks of voxels of size 8^3. To achieve efficient tree traversal, something that OctoMap [137] lacks, each voxel in *supereight* is represented by a Morton number via Morton coding, which is similar to the voxel hashing methods implemented by Niessner et al. [132] and Tanner et al. [133]. In addition, *supereight* can store either TSDF or occupancy probability in the voxel map, and can apply the Marching Cubes algorithm [129] to both representations to extract iso-surfaces and create mesh reconstructions. Overall, *supereight* has demonstrated better scan integration speed and memory efficiency than state-of-the-art reconstruction techniques; therefore it has been chosen as the core of the proposed system SE-Atlas. The detailed description of expanding *supereight* to incorporate LiDAR measurements will be explained in Section 4.5.

Hybrid Systems

There are also hybrid systems that utilise both representations. For instance, the two-stage model-based system designed by Blaer and Allen et al. [145] employed a surface mesh representation in the first stage and the volumetric representation in the second one.

The model-free systems of Kriegel et al. [47, 128] also utilised both representations, following their original surface-based system [127]. They used a laser profiler as the sensor and a Kuka KR16 manipulator to allow the laser strip profiler to move with 6 DOF. Their system aimed at high quality surface reconstruction for objects of arbitrary shape, for the purposes of grasping and manipulation as well as small-scale object recognition [163]. Iteratively, their system sought boundaries and holes in the existing mesh, and defined candidate scan paths along estimated surface trend along these boundaries. The volumetric representation was then used to evaluate these candidates. The one that was

expected to reduce the most entropy in the environment was selected as the Next Best Scan. Unlike their system in [127], however, their new systems in [47, 128] relied on growing an RRT in the voxel space to plan a collision-free path between scans.

An important assumption Kriegel et al. made in their systems was that the environment around the object of interest was safe. In [127], as mentioned earlier, their system defined a bounding box encasing the object to avoid collision when planning paths. In [128] and [47], while they used an RRT to realise collision avoidance, the scan paths were planned a fixed distance above the object surface.

Song et al. [149] implemented a similar architecture in their view-path-planning system. The robot platform that this system focuses on is a UAV, and the objects of interest are outdoor large-scale building structures. The system of Song et al. creates both an OctoMap model and a surfel-based map using ElasticFusion [52]. The OctoMap is is used for global collision-free path planning by exploiting free space. The unknown space in the OctoMap is further leveraged to evaluate the utility of an exploration path. The surfel map is used for detailed local inspection path planning to improve the reconstruction of complex regions.

Other Approaches

There also exist some active mapping systems that utilise neither representations. Border et al. [141] designed a system that planned directly on a raw point cloud. This system categorised points into core points, outlier points and frontier points (Fig. 4.4). Frontier points defined the boundaries of the current reconstruction and provided view point candidates. By using a point cloud representation, this system avoided the computation required for mesh generation or ray casting, which can be heavy when the environment is large. However, it was difficult to estimate occlusion using point clouds. Objects with complicated surface structure pose a challenge to this system.

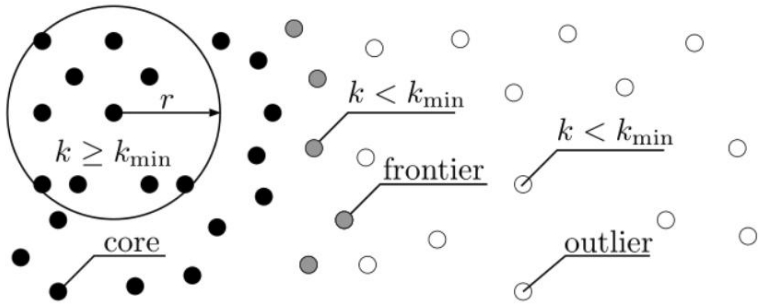


Figure 4.4: A 2D demonstration on how points were categorised in the system of Border et al. [141]. The categorisation was dependent on the number of nearest neighbours within a user-defined radius r. If one point had more neighbours than a threshold k_{min}, this point was classified as a core point (black). A point with insufficient neighbours was an outlier (white). Points with both core and outlier points as its neighbours were frontier points (grey).

4.2.2 Submaps and Elasticity

Online mapping systems need to track the sensor pose at each frame before integrating scans into a reconstruction. Odometry error accumulated through incremental tracking methods, such as frame-to-frame or frame-to-model, can therefore lead to discrepancies in the map.

Dense SLAM systems such as KineticFusion [48] and ElasticFusion [52] use map-centric approaches to tightly couple their reconstructions with the SLAM trajectory. ElasticFusion [52] bends the mapped environment upon loop closure. De Gregorio and Di Stefano [135] proposed occupancy-based methods to erode and re-integrate past scans individually from the existing reconstruction upon pose graph optimisation. These methods, however, suffer from poor scalability in large-scale online operations. For instance, in the system of De Gregorio and Di Stefano [135], the process of removing a scan from the already constructed map (erosion) and that of re-integrating the corrected scan both require the same amount of computation. When the environment has a limited scale, doubling the scan integration time can be feasible, but large-scale outdoor experiments will require a more efficient design.

Another technique to improve global consistency is to represent the full 3D reconstruction using a collection of submaps of limited extent. This technique

originated in SLAM research such as the *Atlas* framework by Bosse et al. [114] and DenseSLAM by Nieto et al. [165]. Both systems maintain an interconnected collection of local submaps instead of a single global map, which sparsifies the environment even in the case of dense reconstruction. Additionally, *Atlas* reuses existing maps instead of spawning a new submap upon loop closure, hence allowing the global map to scale with the size of the explored environment instead of the exploration length.

Submaps can also enable elasticity when a reconstruction requires correction at the event of a loop closure. Ho et al. [138], Sodhi et al. [139] and Reijgwart et al. [5] all exploited submaps to achieve elasticity in dense reconstruction for the purpose of motion planning. The systems of Ho et al. and Sodhi et al. are based on OctoMap [137], and that of Reijgwart et al. is based on Voxblox [4]. Upon loop closure, submaps in all systems can be moved around to keep global consistency. However, these systems both have some limitations. For motion planning, the system of Ho et al. [138] requires ray-casting into every submap for occupancy information, significantly increasing the complexity when there are many submaps. The authors improved their method in [139] where submaps are merged together into a global map for faster voxel query. Global map update is only triggered when a submap's pose changes significantly. In a large-scale environment with a long range-sensor, however, maintaining dense reconstructions of both submaps and a global map in memory is very inefficient.

Our proposed system SE-Atlas is most directly motivated by the work of Reijgwart et al. [5]. Their map-centric dense SLAM pipeline, Voxgraph, constructs TSDF submaps and generates correspondence-free constraints among them for global consistency. It can also be employed for robotic tasks like path planning by computing a ESDF from the TSDF, as explained in 4.2.1. Voxgraph incorporates both LiDAR scans and RGB-D measurements, and was demonstrated to be sufficiently lightweight to run onboard a Micro Aerial Vehicle (MAV).

These approaches spawn new submaps after a temporal interval [5, 139, 166] to bound local odometry drift within each submap. Instead, our proposed sys-

tem, SE-Atlas, creates submaps based on robot travel distance, avoiding redundant submap creations when the robot is stationary [11]. We further propose submap spawning and fusion using a deeper spatial analysis to uniquely represent confined or distinct areas such as rooms. In challenging exploration and navigation tasks such as [167], this feature allows local path planners to require only one or very few submaps when searching for paths in confined space.

4.2.3 Room Segmentation

When a complete map is present, several works have developed methods to segment LiDAR reconstructions or floor plans into individual enclosed spaces. Turner and Zakhor [168] designed a room-segmentation pipeline that partitioned 2D point cloud maps into 2.5D building models via triangulation. To achieve 3D reconstruction, the approach assumed that interior walls were vertical and planar.

More sophisticated methods were developed to parse 3D point clouds into rooms, such as detecting void spaces between walls using point density histograms [169], and extracting planar features before partitioning separate rooms via a multi-label energy minimisation formulation [170, 171]. These methods were limited to single-storey reconstructions.

Ochmann et al. [172] and Nikoohemat et al. [173] proposed methods that handle unstructured 3D point clouds for multi-storey room segmentation. Ochmann et al. introduced a versatile integer linear programming method to incorporate hard constraints, e.g. wall connectivity, to ensure a plausible reconstruction. Nikoohemat et al. employed a mobile LiDAR SLAM system and separated building levels by assuming that sloped trajectory segments represent staircase traversals. They further segmented rooms using an adjacency graph of planer segments.

While these methods achieve the same effect of partitioning confined spaces, the need for prior knowledge of a complete reconstruction is not desired for online exploration tasks. Instead, the system presented in this thesis focuses on

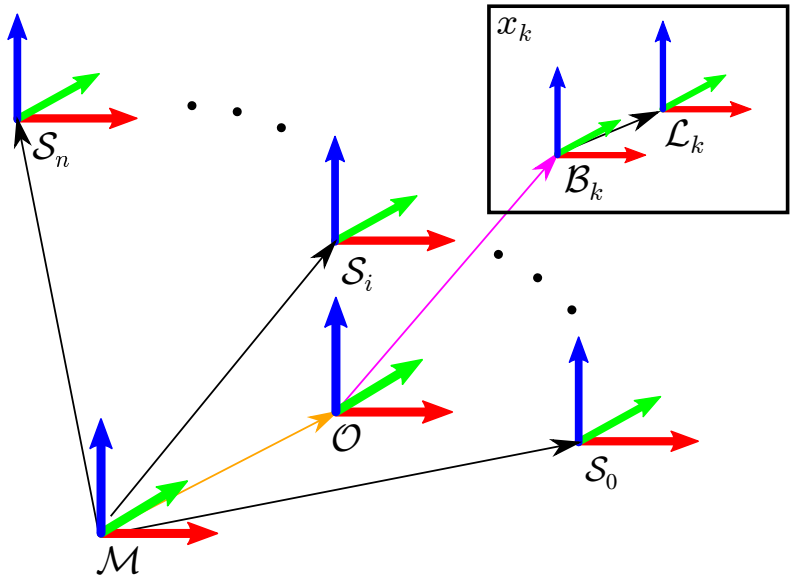


Figure 4.5: The proposed system's frame convention. Orange arrow: transformation provided by SLAM system; purple arrow: transformation provided by odometry system; black arrow: transformation created inside the proposed system.

segmenting 3D reconstructions of confined spaces on the fly, such as individual rooms and staircases, onboard a mobile robot.

4.3 Reference Frames and Notation Definitions

The focus of this thesis is neither SLAM or localisation. However, in this DPhil project, the proposed reconstruction system has relied on a multi-sensor odometry (such as VILENS [81]) and a LiDAR-based factor graph SLAM system (such as AEROS [113]).

This section provides an overview on the definition and frame convention used in the proposed reconstruction system as well as those in the external odometry and SLAM systems. Fig. 4.5 illustrates the frame convention. This thesis uses the conventional robotics terminology of *pose* to refer to the combination of position and rotation.

4.3.1 Odometry and SLAM Notations

The Map frame $\{\mathcal{M}\}$ and the Odometry frame $\{\mathcal{O}\}$ each defines a global fixed frame of reference, provided by the SLAM and odometry module in the pipeline

respectively. The base frame of the robot at time k is defined as $\{\mathcal{B}_k\}$.

SE-Atlas assumes as an input a factor graph $\mathbf{X}$ with q nodes $\mathbf{x}_k, k \in \{0, \ldots, q-1\}$ given by the SLAM system, typical of the state of the art. Each node describes the estimated pose of the robot expressed in the map frame ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{B}_k} \in \mathbf{SE}(3)$.

Each node of the graph is associated with a raw point cloud $\mathbf{C}$ from the LiDAR. The point clouds have a fixed number of points $\mathbf{p} \in \mathbb{R}^3$ expressed in the LiDAR frame $\{\mathcal{L}\}$. Given a node $\mathbf{x}_k$ and its associated point cloud $\mathbf{C}_k$, the pose of the LiDAR ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{L}_k}$ can be computed as follows:

$$ {}^{\mathcal{M}}\mathbf{T}_{\mathcal{L}_k} = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{B}_k}{}^{\mathcal{B}}\mathbf{T}_{\mathcal{L}} \tag{4.1} $$

where ${}^{\mathcal{B}}\mathbf{T}_{\mathcal{L}} \in \mathbf{SE}(3)$ is calibrated and fixed transform between LiDAR frame and base frame.

4.3.2 Reconstruction Notation

The proposed reconstruction core creates n submaps to represent the scanned environment. Each submap $\mathcal{S}_i, i \in \{0, \ldots, n-1\}$ contains the following information:

- a volumetric reconstruction $\mathbf{V}_{\mathcal{S}_i}$ that stores either occupancy or TSDF information in each voxel (an individual voxel is denoted as $\mathbf{v}$),
- the SLAM node indices and LiDAR scans used to construct $\mathcal{S}_i$,
- an accumulated point cloud $\mathbf{C}_{\mathcal{S}_i}$ expressed in map frame $\{\mathcal{M}\}$,
- the root pose of the submap that defines its transformation with respect to the map frame, ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}$,
- an Axis-Aligned Bounding Box (AABB) of the submap reconstruction aligned with the map frame $\{\mathcal{M}\}$.

A further explanation on why the proposed system uses AABB is provided in Section 4.4.2.

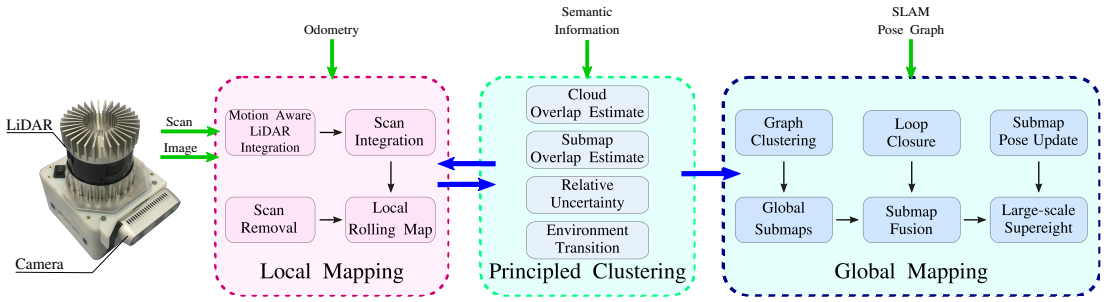


Figure 4.6: An overview of the proposed system that consists of a front-end (*Local Mapping*) and a back-end (*Global Mapping*). *Principled Clustering* contains the proposed novel strategies which estimate cloud and submap overlap, measure relative pose uncertainty in the SLAM graph, and leverage semantic information to detect transitions between indoor and outdoor environments. Section 4.4 provides a more detailed explanation on the key components in this system.

4.4 System Architecture

As mentioned in Section 4.1, the proposed system uses a multi-threaded front-end/back-end structure to maintain both a local rolling map and a collection of global submaps during exploration. Fig. 4.6 provides an overview of the proposed system.

4.4.1 Odometry and SLAM Inputs

While *supereight* does provide its own tracking functionality, it was tailored for RGB-D cameras. Hence I do not use this feature of *supereight* for the LiDAR-based pipeline proposed in this thesis, and instead rely on external odometry [81, 155] and SLAM modules [113, 158]. The odometry module is an input to the front-end of the proposed system, providing the pose of the LiDAR sensor frame $\{\mathcal{L}\}$ in a global odometry frame $\{\mathcal{O}\}$ at time t, denoted as ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{L}_t} \in \mathbf{SE}(3)$. This pose is associated with a raw point cloud $\mathbf{C}_t$ expressed in the LiDAR frame $\{\mathcal{L}_t\}$.

The proposed reconstruction pipeline also requires access to the solution of a graph-based SLAM system, i.e. q nodes $\mathbf{X}_k, k \in \{0, \ldots, q-1\}$, and the Hessian matrix associated with the solution [174]. As explained in Section 4.3, each node describes the estimated pose of the LiDAR frame $\{\mathcal{L}_k\}$ with respect to a fixed map frame $\{\mathcal{M}\}$, denoted as ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{L}_k} \in \mathbf{SE}(3)$.

The SLAM system used is a pose-graph method based on the work of Ramezani et al. [158], where the only inputs are relative odometry measurements and loop closure candidates, both represented as $\mathbf{SE}(3)$ matrices. The covariances associated with these measurements are fixed and tuned using line-search. The solution is determined in an incremental fashion using the iSAM2 algorithm [42] available in the GTSAM library [8].

4.4.2 Reconstruction Outputs

The output of the reconstruction system consists of a local rolling map and a collection of n submaps. *Local Mapping* creates and maintains a local map by integrating the latest point cloud $\mathbf{C}_t$ into a volumetric reconstruction in $\{\mathcal{O}\}$. I crop it around the latest pose ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{L}_t}$ based on the sensing range of the LiDAR. Section 4.5 explains in detail the integration of LiDAR scans into a volumetric reconstruction using *supereight* pipeline, focusing on the extension to *supereight* that is tailored towards LiDAR scans. The component *Motion Aware LiDAR Integration* (Section 4.5.3) is an essential upgrade that I developed to allow SE-Atlas to correctly incorporate motion-undistorted LiDAR scans.

In *Global Mapping*, the information stored in each submap $\mathcal{S}_i, i \in \{0, \ldots, n-1\}$ is explained in Section 4.3.2. *Supereight* provides pipelines that can store either TSDF or occupancy probability in the reconstruction. Though experiments have also been conducted to assess the TSDF pipeline as well, occupancy representation is chosen as the focus in the proposed system because of its explicit representation of free space. Section 4.8.2 will explain how explicitly represented free space is used in the estimation of submap overlap.

I also compute and maintain an AABB for each submap in $\{\mathcal{M}\}$ using the reconstruction (Section 4.3.2). Each submap's AABB is significantly affected by the orientation of $\{\mathcal{M}\}$, so I use AABBs not to accurately estimate scanned space, but as a lightweight method to determine non-overlapping submaps (Section 4.8.2). Therefore I choose to use AABB instead of oriented ones to reduce the time to compute bounding boxes as well as their overlaps.

The *Graph Clustering* module in Fig. 4.6 (Section 4.7.1) first clusters the received SLAM pose graph into individual submaps and determines which submap the *Local Rolling Map* created in the front-end should be integrated into. Upon SLAM loop closure, *Submap Pose Update* module (Section 4.7.4) corrects the pose of submaps to maintain global consistency, and *Submap Fusion* module (Section 4.8.2) decides which submaps should be merged together to improve system scalability.

In the proposed system, I further introduce the following set of strategies for principled clustering which provides additional criteria on when to spawn or fuse submaps:

- *Cloud Overlap Estimate*: to adjust the submap spawning decisions made by Local Mapping – Section 4.7.3.
- *Submap Overlap Estimate*: to propose submap fusion for Global Mapping – Section 4.8.2.
- *Relative Uncertainty*: to reject unreliable submap fusions – Section 4.9.
- *Environment Transition Criterion*: to differentiate between indoor and outdoor environments based on semantic information.

Environment Transition Criterion is based on semantic analysis and on-the-fly adjusts reconstruction parameters. It will be explain in the following chapter (Chapter 5).

4.5 LiDAR *Supereight*

This section describes the key reconstruction foundation of SE-Atlas, *supereight* [50], and the significant improvements I implemented for *supereight* to support long range LiDAR sensing.

Supereight is a volumetric, octree-based SLAM pipeline with adaptive resolution that uses Morton codes to achieve efficient spatial octree traversals [50]. Instead of individual voxels, *supereight* stores blocks which aggregate $8 \times 8 \times 8$

voxels as the finest leaves of the octree structure. This results in fewer memory allocations and improved cache locality during updates, improving performance. It is also capable of integrating data at different octree levels, further increasing efficiency.

In this work, I expand both *supereight*'s multi-resolution TSDF (*MultiresTSDF*) [50] and multi-resolution occupancy (*MultiresOFusion*) [51] pipelines to incorporate LiDAR inputs. The original RGB-D *supereight* uses a pinhole camera model. To incorporate LiDAR data into the framework, organised LiDAR point clouds are converted to depth images, and the projection model is approximated with a spherical camera model by defining a pair of azimuth and elevation angles for each pixel in the depth image based on sensor specification. The new projection model is similarly based on the assumption that rays corresponding to pixels in the depth image are uniformly distributed and fixed.

Compared to the pinhole camera model, the LiDAR model incorporates a longer range and larger FoV, but the distance measurements are sparser. I also create local submaps in this system to replace the single global map in the original pipeline. This is further explained in Section 4.7.

4.5.1 Multi-resolution

SE-Atlas leverages the adaptative resolution feature of *supereight* [50, 51] for efficient LiDAR integration. This section describes the basis of this feature, and the following section (Section 4.5.2) focuses on the specific adjustments for LiDAR.

Long range measurements from LiDAR cover a much larger amount of space than an RGB-D camera. Integrating all of the scanned space at the highest resolution means updating a huge number of voxels and requiring a lot of computation. *Supereight* can update the octree at various levels depending on the effective resolution of the sensor, by updating cubes consisting of several voxels instead of individual voxels. The benefit of this approach is a reduction in the number of octree updates, resulting in reduced integration time [50]. In

addition, the majority of the scanned space is free space with trivial occupancy information. *Supereight* integrate voxels in free space with significantly lower resolution than those on object surfaces, maintaining details on the surfaces while improving integration speed [175]. This performance increase is especially important in the case of LiDAR sensors where a single scan may contain measurements ranging from a few meters to $60\,\mathrm{m}$ away [11, 51].

Due to a larger horizontal FoV as well as longer range, it is more likely for LiDAR rays to hit surfaces at shallow angles than those of RGB-D cameras, which results in aliasing artefacts. In SE-Atlas, *supereight*'s integration level selection method for a particular depth measurement has been adjusted to make it suitable for LiDAR sensors [12, 51]. This update reduces aliasing artefacts, increases speed and decreases memory consumption at large distances.

For each ray $\mathbf{r}$, SE-Atlas considers the minimum angle between two adjacent LiDAR scan rays, which in turn defines a circular cone. It then iterates through voxels within the frustum of the LiDAR sensor and find the cone that each voxel lands in. The depth of the voxel $d_{\mathbf{v}}$ determines the scale of update volume, which is the largest block of voxels that fits inside this cone at $d_{\mathbf{v}}$ up to $8 \times 8 \times 8$ voxels. Thus, measurements can be integrated into volumes at adaptively selected resolutions [12, 51]. It is also important to note that due to the assumption of uniformly distributed scan rays, finding which cone a voxel lands in is a simple arithmetic process, ensuring the efficiency of scan integration.

SE-Atlas use the propagation strategies described in [50] and [51] for MultiresTSDF and MultiresOFusion, respectively, to keep the hierarchy consistent between different integration levels. In MultiresOFusion, the maximum occupancy and observed state at the finest integration level are up-propagated to each parent level to provide fast occupancy queries at different levels. MultiresOFusion also explicitly keeps track of free space at the coarsest possible scale while preserving details about unknown space.

4.5.2 LiDAR Integration

This section explains the process of creating a new reconstruction or updating an existing one based on a new LiDAR scan. A representative LiDAR used in our experiments (Section 4.10) is Ouster OS1-64. It produces organised dense point clouds of 64×1024 points at $10\,\mathrm{Hz}$ (i.e. $\sim$ 655k points/s), with a vertical FoV of $33.2°$ and a horizontal FoV of $360°$. Scans are converted from point clouds to spherical range images to facilitate their inclusion into *supereight*.

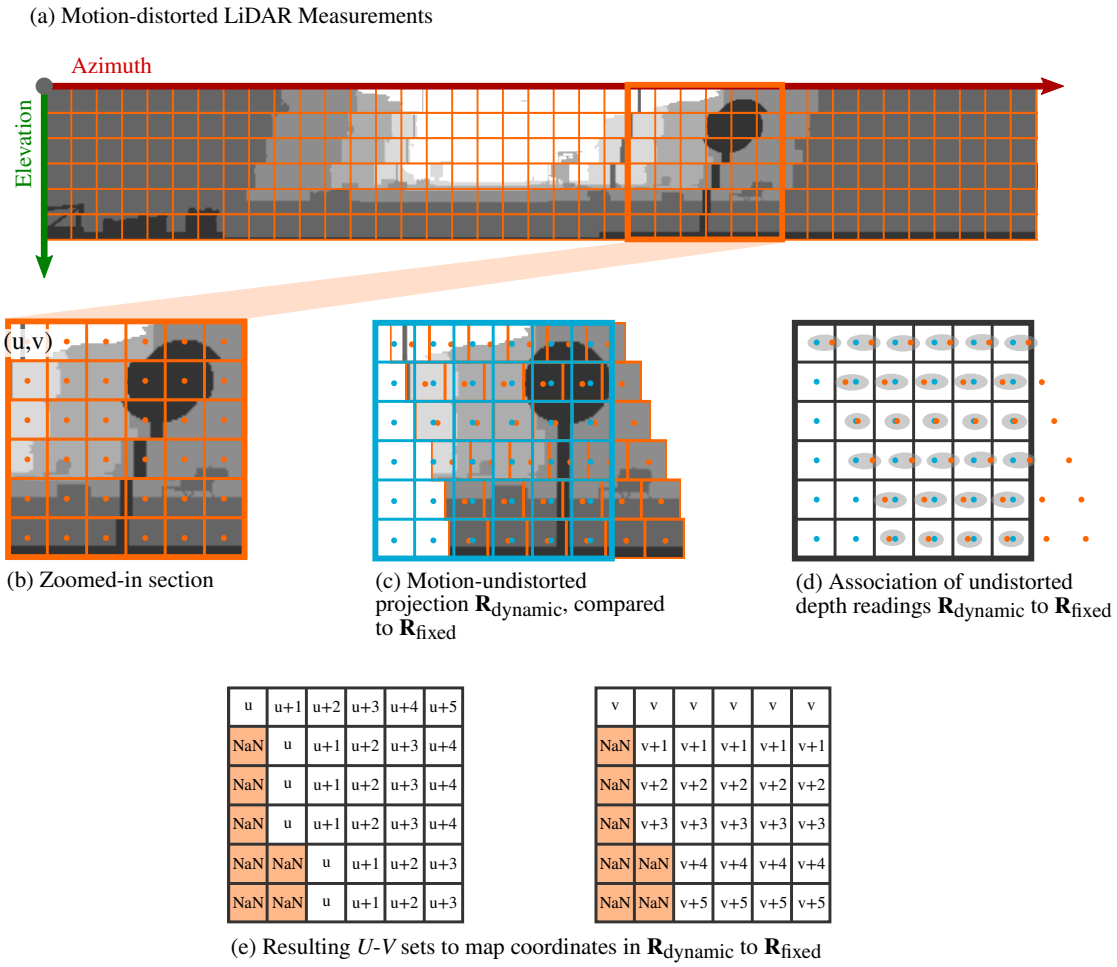
The MultiresOFusion pipeline stores the occupancy probability in log-odds form which results in free, unknown and occupied voxels having negative, zero and positive log-odds values, respectively. Occupancy update follows the convention of adding a new log-odds measurement [137, 175]. The log-odds measurement along a ray is a distance-dependent piecewise linear function explained in detail in [51].

The uncertainty model of LiDAR measurements is first updated because LiDARs are more accurate at long distance compared to RGB-D cameras, the uncertainty of which increases quadratically with range [51]. Given a distance measurement $d_{\mathbf{r}}$ along ray $\mathbf{r}$, SE-Atlas assumes its standard deviation is $\sigma(d_{\mathbf{r}}) = \max(\sigma_{\min}, \lambda_\sigma d_{\mathbf{r}})$ where $\sigma_{\min}$ and λ_σ depend on the sensor characteristics. The log-odds occupancy probability $L(d_{\mathbf{v}})$ in a voxel $\mathbf{v}$ at distance $d_{\mathbf{v}}$ is inspired by [176], but using a piecewise linear function instead [11]:

$$L(d_{\mathbf{v}}) = \begin{cases} L_{\min} & \text{if } d_{\mathbf{v}} \leq 3\sigma(d_{\mathbf{r}}) \\ \frac{-L_{\min}}{3\sigma(d_{\mathbf{r}})} d_{\mathbf{v}} & \text{if } 3\sigma(d_{\mathbf{r}}) < d_{\mathbf{v}} \leq \frac{\lambda_\tau d_{\mathbf{r}}}{2} \\ \frac{-L_{\min}}{3\sigma(d_{\mathbf{r}})} \frac{\lambda_\tau d_{\mathbf{r}}}{2} & \text{if } \frac{\lambda_\tau d_{\mathbf{r}}}{2} < d_{\mathbf{v}} \leq \lambda_\tau d_{\mathbf{r}} \\ \text{no update} & \text{otherwise} \end{cases} \tag{4.2}$$

where $L_{\min}$ denotes the minimum occupancy probability in log-odds and λ_τ is a scaling factor controlling how much occupied space is created behind surface.

The integration process for the MultiresTSDF pipeline is described in [50]. To avoid artefacts when using long-range LiDAR scans, the pipeline is modified so that the TSDF truncation bound adapts to the integration level.



u	u+1	u+2	u+3	u+4	u+5
NaN	u	u+1	u+2	u+3	u+4
NaN	u	u+1	u+2	u+3	u+4
NaN	u	u+1	u+2	u+3	u+4
NaN	NaN	u	u+1	u+2	u+3
NaN	NaN	u	u+1	u+2	u+3

v	v	v	v	v	v
NaN	v+1	v+1	v+1	v+1	v+1
NaN	v+2	v+2	v+2	v+2	v+2
NaN	v+3	v+3	v+3	v+3	v+3
NaN	NaN	v+4	v+4	v+4	v+4
NaN	NaN	v+5	v+5	v+5	v+5

(e) Resulting *U*-*V* sets to map coordinates in $\mathbf{R}_{\text{dynamic}}$ to $\mathbf{R}_{\text{fixed}}$

Figure 4.7: Motion aware LiDAR integration module to remap LiDAR points during dynamic motions. **(a)**: Raw motion-distorted LiDAR scan showing azimuth and elevation. **(b)**: Zoom in of a section. Each orange circle indicates a ray, with its corresponding cell in the grid. **(c)**: After motion undistortion, the undistorted projection $\mathbf{R}_{\text{dynamic}}$ (in orange) does not match the uniform grid (in light blue). **(d)**: We iterate over the undistorted rays to find the closest ray assuming an uniform grid. **(e)**: The output of the association are adjusted coordinates for azimuth (U) and elevation (V) angles of the undistorted projection.

4.5.3 Motion Aware LiDAR Integration

Modern 3D LiDAR sensors collect range returns continuously over time. A complete 360° scan is accumulated by the device's driver and then made available to downstream applications. The high dynamics of a legged robot, especially in rotation, leads to motion distortion of these scans and imprecise maps, if uncorrected (Fig. 4.7 (a) and Fig. 4.7 (b)).

As the input to my proposed reconstruction system, the external odometry

module incorporated a LiDAR motion undistortion component[†]. This component first uses an odometry input with higher frequency, such as from IMU or legged odometry, to interpolate the sensor pose between consecutive LiDAR measurements (at $10\,\mathrm{Hz}$). It then computes the direction and origin of each LiDAR beam on-the-fly and reprojects it as if measured instantaneously, correcting the motion-induced distortion. While the undistorted point cloud is necessary for an accurate reconstruction, the undistortion module is not within the scope of this work and therefore not assessed in the presented experiments.

Instead, the proposed system focuses on preserving the motion undistortion effect in the reconstruction. The motion undistortion procedure in the odometry module results in non-uniform elevation and azimuth angles for each point, violating the uniform grid assumption of the LiDAR projection model that *supereight* [51] relies on (Section 4.5.1), as exampled in Fig. 4.7 (c).

This assumption of LiDAR beams being uniformly distributed in angular space is beneficial for the efficiency of scan integration. In the reconstruction core of the proposed system, a voxel's coordinates in the map frame ${}^{\mathcal{M}}\mathbf{v} \in \mathbb{R}^3$ are projected into the LiDAR frame ${}^{\mathcal{L}}\mathbf{v} = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{L}}^{-1}{}^{\mathcal{M}}\mathbf{v}$ and normalised into a unit vector that represents its direction, before being converted into a pair of column and row indices (u, v) for depth measurement look-up and occupancy update. With the uniform grid assumption, this conversion from ${}^{\mathcal{M}}\mathbf{v}$ to (u, v) is a simple arithmetic process, the complexity of which is $O\left(d_{\text{max}}^3 \cdot r_{\text{voxel}}^{-3}\right)$ — d_{max} represents the LiDAR sensing range and r_{voxel} represents the voxel resolution. However, with motion undistortion dynamically adjusting the elevation and azimuth angles of each LiDAR beam in every scan, to look up a corresponding depth measurement in the direction of a voxel will require a search through neighbouring LiDAR beams to find the closest ray. The complexity therefore becomes $O\left(d_{\text{max}}^3 \cdot r_{\text{voxel}}^{-3} \cdot n\right)$ where n represents the number of neighbours that need to be searched. Given the long LiDAR sensing range and the fine voxel resolution, this approach quickly becomes infeasible.

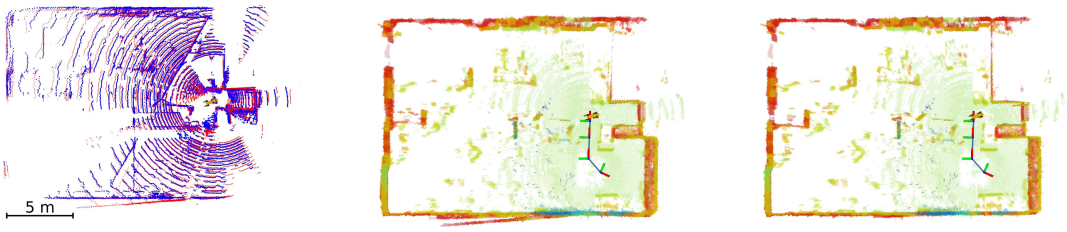
[†]The motion undistortion method was inspired by this package - https://github.com/ethz-asl/lidar_undistortion.

Instead, I designed a method to remap the motion undistorted LiDAR scans, which allows SE-Atlas to preserve not only measurement accuracy, but also the efficiency of the projection model used by *superight* and SE-Atlas. For each LiDAR scan of constant height h and width w (64×1024 in our experiments), let $\mathbf{R}_{\text{fixed}}$ be the default set of ray-casting directions according to the sensor's specification, and $\mathbf{R}_{\text{dynamic}}$ be the unit vectors for all motion-undistorted LiDAR beams. For each ray $\mathbf{r}_{uv} \in \mathbf{R}_{\text{fixed}}$ at row v and column u, I then search in the neighbours of this ray to find the $\hat{\mathbf{r}}_{uv} \in \mathbf{R}_{\text{dynamic}}$ that has the smallest angular difference θ_{diff} from $\mathbf{r}_{uv}$ to establish a match (Fig. 4.7 (d)). The size of the search region is determined by the resolution of the LiDAR and the characteristics of the robot, such as its rotation rate. This search can also be terminated if $\theta_{\text{diff}} < \theta_{\text{thres}}$ to further speed up the procedure; I used $\theta_{\text{thres}} = 0.001\,\text{rad}$, corresponding to $5\,\text{cm}$ voxel resolution at a maximum sensor range of $50\,\text{m}$.

As a result of the remapping, I find corrected coordinates for each ray in the undistorted scan, given by the sets of indices $U = u_{(0,0)} \ldots u_{(h-1,w-1)}$ and $V = v_{(0,0)} \ldots v_{(h-1,w-1)}$, where NaN values are used for the unmatched rays, as shown in Fig. 4.7 (e). The $U - V$ sets are then input into the *supereight* reconstruction pipeline. I further update the projection model of *supereight* to find the corresponding motion-undistorted LiDAR range measurement based on indices in U and V for each ray-casting vector in R_{fixed} for LiDAR integration. The complexity of this remapping process is $O\left(h \cdot w \cdot n\right)$, and is not dependent on the LiDAR range or the voxel resolution and therefore more efficient.

To demonstrate the effect of this particular module, we conducted a brief experiment (Fig. 4.8) where a legged robot was teleoperated around our indoor lab space. At times of high rotation rate obvious motion distortion in the scan can be seen in Fig. 4.8 (a). The distorted scan (red) is misaligned along the lower wall, but the odometry system accounted for the distortion and produced the undistorted scan (blue).

When moving to occupancy mapping, we can see that without properly addressing the dynamic projection model the default *supereight* misaligns surfaces



(a) Motion distorted raw LiDAR point cloud (red) and the undistorted version (blue).

(b) Reconstruction without correcting the dynamic projection model in undistortion, causes misalignment.

(c) Corrected reconstruction via motion undistortion.

Figure 4.8: Proposed motion aware LiDAR integration method (Section 4.5.3) remaps a dynamic projection model, improving the reconstruction — particularly when turning sharply.

in its reconstruction (Fig. 4.8 (b)). By applying the method described above, the reconstruction is corrected as shown in Fig. 4.8 (c), keeping its computational efficiency.

4.6 Local Rolling Map

As explained in Section 4.4, the Local Mapping front-end of the proposed system produces a local reconstruction in odometry frame $\{\mathcal{O}\}$. The scan integration process relies on the latest LiDAR pose in odometry frame ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{L}_t}$ and the corresponding LiDAR scan $\mathbf{C}_t$. Using ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{L}_t}$, every point $\mathbf{p} \in \mathbf{C}_t$ from the LiDAR frame $\{\mathcal{L}_t\}$ is transformed to the odometry frame $\{\mathcal{O}\}$. SE-Atlas then updates the local reconstruction according to Section 4.5.2, incorporating the essential motion-aware adaptation to *supereight* as detailed in Section 4.5.3.

This local reconstruction is centred around the latest robot pose ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{L}_t}$ and moves with the robot as exploration goes on. It is cropped around the robot position based on the LiDAR scan range. This bounds the memory usage of the local reconstruction and limits the odometry drift within this map.

This map is then used as one of the inputs to the Global Mapping back-end of the proposed system to spawn a new global submap or add to an existing one. This will be explained in Section 4.7.

4.7 Elasticity in Large-Scale Long-Term Exploration

This section details the back-end components that provide elasticity to the reconstruction pipeline in Fig. 4.6, namely Graph Clustering, Global Submaps and Submap Pose Update. The principle clustering strategy Cloud Overlap Estimate is also explained in this section because it provides additional submap spawning decisions.

The external pose graph SLAM system [113, 158] produces essential inputs to the Global Mapping back-end. The first input is a pose graph with q poses $\mathbf{X}_k, k \in \{0, \ldots, q-1\}$. This section focuses on how the pose graph is used to determine submap spawning and updating in SE-Atlas. The pose graph SLAM system also produces the Hessian matrix associated with the pose graph solution [174]. This matrix is leveraged by the Relative Uncertainty Estimation strategy that will be explained in Section 4.9.

The external SLAM system [158] globally optimises the odometry output from systems like VILENS [81] or ICP-based methods [155]. Overall, the computed trajectory is locally consistent with drift rates in the order of $1\,\mathrm{m}$ per $100\,\mathrm{m}$ travelled. In this way we collects a sequence of point clouds which are registered to one another locally, as well as a corresponding relative pose estimate for the robot/device. When loop closures occur, the SLAM system form a full pose graph. I would also like to acknowledge that methods such as LOAM [54] and ScanContext [82] could further improve the current pose graph SLAM system.

4.7.1 Graph Clustering

The Graph Clustering module processes the pose graph from the SLAM system and groups graph nodes together into different submaps. The clustered graph further guides scan integration (Section 4.7.2) and submap fusion (Section 4.8).

To perform clustering, I first divide the pose graph edges into odometry and loop closure edges. Odometry edges represent constraints between consecutive

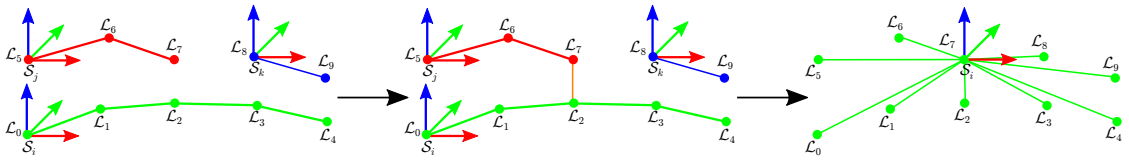


Figure 4.9: An example of graph clustering and submap fusion based around a loop closure. Nodes $\mathcal{L}_{0:4}$ and $\mathcal{L}_{5:9}$ represent the two traversals of a location from a pose graph. $\mathcal{L}_{0:4}$ belong to green submap $\mathcal{S}_i$, $\mathcal{L}_{5:7}$ to red submap $\mathcal{S}_j$ and $\mathcal{L}_{8:9}$ to blue submap $\mathcal{S}_k$. Because these nodes have been grouped together by the graph clustering (Section 4.7.1), these three submaps are all merged into submap $\mathcal{S}_i$.

pairs of nodes, while loop closure edges are the constraints between nodes that are distant in the graph but correspond to similar scans of revisited places.

If there are no loop closures, grouping into submaps is based only on the odometry chain within a distance threshold λ_{odom}. In this case, the first node $\mathbf{x}_{i,0}$ of submap $\mathcal{S}_i$ defines the submap's root pose ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}$ using its corresponding LiDAR pose ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{L}_{i,0}}$.

For the subsequent nodes, I compute the distance travelled along the pose graph from the root pose. If a new node is within the distance threshold λ_{odom}, the latest local map is integrated into $\mathcal{S}_i$ according to Section 4.7.2. When the new node exceeds the distance threshold, a new submap $\mathcal{S}_{i+1}$ is spawned with that node. This is based on the assumption that the odometry drift is proportional to distance travelled.

Upon loop closure, I cluster together nodes that are within a threshold λ_{cluster} around the pair of nodes that form the closure. Fig. 4.9 presents an example of clustering. In this example, $\mathcal{L}_2$ and $\mathcal{L}_7$ are connected by a loop closure edge. I then compute the distances from every surrounding node to this loop closure pair along the pose graph, again assuming that odometry drift is proportional to distance travelled. In the case of $\mathcal{L}_9$, its distance is computed as:

$$\begin{aligned} d_{\mathcal{L}_7,\mathcal{L}_9} &= d_{\mathcal{L}_7,\mathcal{L}_8} + d_{\mathcal{L}_8,\mathcal{L}_9} \\ &= ||{}^{\mathcal{M}}\mathbf{t}_{\mathcal{L}_7} - {}^{\mathcal{M}}\mathbf{t}_{\mathcal{L}_8}|| + ||{}^{\mathcal{M}}\mathbf{t}_{\mathcal{L}_8} - {}^{\mathcal{M}}\mathbf{t}_{\mathcal{L}_9}|| \end{aligned} \tag{4.3}$$

and because $d_{\mathcal{L}_7,\mathcal{L}_9} < \lambda_{\text{cluster}}$, $\mathcal{L}_9$ is included in this cluster. Loop closure clusters guide *submap fusion* in Section 4.8.

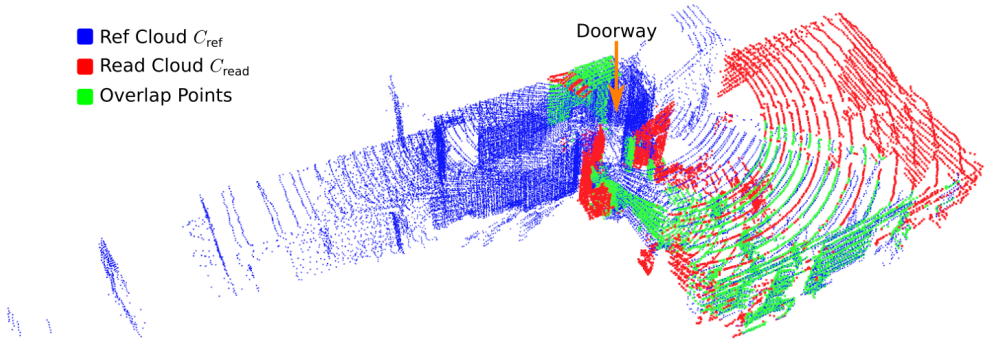


Figure 4.10: An example of Cloud Overlap Estimate showing limited overlap between the reference cloud $\mathbf{C}_{\text{ref}}$ and the read cloud $\mathbf{C}_{\text{read}}$ when entering a room through a narrow doorway.

4.7.2 Global Submap Integration and Spawning

After clustering, the most recent local rolling map at the latest pose $\mathbf{x}_k$ is integrated into submap $\mathcal{S}_i$. I first compute the relative pose between $\mathcal{O}$ and $\mathcal{S}_i$:

$$ {}^{\mathcal{S}_i}\mathbf{T}_{\mathcal{O}} = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}^{-1}\, {}^{\mathcal{M}}\mathbf{T}_{\mathcal{B}_k}\, {}^{\mathcal{O}}\mathbf{T}_{\mathcal{B}_k}^{-1} \tag{4.4} $$

where ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{B}_k}$ and ${}^{\mathcal{O}}\mathbf{T}_{\mathcal{B}_k}$ are provided by the SLAM and odometry modules respectively. When $\mathbf{x}_k$ is the first node of a submap according to the result of Graph Clustering, the corresponding local map forms the initial reconstruction of $\mathcal{S}_i$ and ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i} = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{L}_k}$.

To integrate a local reconstruction into a submap, I first project the coordinate of voxel $\mathbf{v}_{\text{local}}$ from the local map into submap $\mathcal{S}_i$ using ${}^{\mathcal{S}_i}\mathbf{T}_{\mathcal{O}}$, and obtain $\mathbf{v}_{\mathcal{S}_i}$. Then I update the voxel in $\mathcal{S}_i$ at coordinate $\mathbf{v}_{\mathcal{S}_i}$ at the same scale as the voxel in the local map, following the mathematical models proposed in [50, 51].

4.7.3 Cloud Overlap Estimate

Cloud Overlap Estimate adds another trigger to spawn submaps based on point cloud overlap. The tracking performance of the odometry system is affected by major changes in overlap such as when entering a new room [177]. Hence when traversing between two disconnected spaces via a narrow passage, i.e. the scenario presented in Fig. 4.10, it is beneficial to spawn a new submap and create an elastic connection.

Algorithm 1: Cloud Overlap Estimate.

input: New LiDAR cloud $\mathbf{C}_{\text{read}}$ and submap cloud $\mathbf{C}_{\mathcal{S}_{\text{ref}}}$,
output: Cloud overlap ratio $R_{\text{point,read}}$
begin
  Voxel filter $\mathbf{C}_{\text{read}}$ and $\mathbf{C}_{\mathcal{S}_{\text{ref}}}$ to resolution r_{filter}
  for Point $\mathbf{p}_i \subset \mathbf{C}_{\text{read}}$ **do**
    Search for $\mathbf{p}_{\text{neighbour}} \subset \mathbf{C}_{\mathcal{S}_{\text{ref}}}$ that is the closest to $\mathbf{p}_i$
    if $||\mathbf{p}_i, \mathbf{p}_{\text{neighbour}}|| < \sqrt{3} \times r_{\text{filter}}$ **then**
      $N_{\text{point,overlap}} = N_{\text{point,overlap}} + 1$
    end if
  end for
  $R_{\text{point,read}} = N_{\text{point,overlap}} / N_{\text{point,read}}$
  return $R_{\text{point,read}}$
end

When the robot crosses through a doorway, a significant proportion of the scene that the robot perceives will likely change — with LiDAR beams blocked by the doorway. Traversing through any narrow constriction with a LiDAR will cause there to be low overlap between consecutive point cloud scans. In such a scenario, the proposed system will spawn a new submap when going through a doorway and create a new room, as demonstrated in Fig. 4.2.

Alg. 1 presents how I compute the overlap ratio $R_{\text{point,read}}$ between the point cloud of a new scan $\mathbf{C}_{\text{read}}$ and the accumulated submap cloud $\mathbf{C}_{\mathcal{S}_{\text{ref}}}$ of a reference submap $\mathcal{S}_{\text{ref}}$. Both $\mathbf{C}_{\text{read}}$ and $\mathbf{C}_{\mathcal{S}_{\text{ref}}}$ are filtered to the same resolution r_{filter} for uniformity in overlap estimation. Points from $\mathbf{C}_{\text{read}}$ that are within a threshold distance from their closest points in $\mathbf{C}_{\mathcal{S}_{\text{ref}}}$ are considered as overlapping points. The threshold is set to $\sqrt{3} \times r_{\text{filter}}$ — the diagonal of a cube with length r_{filter}.

If $\mathbf{C}_{\text{read}}$ shares sufficient overlap ($R_{\text{point,read}} > 0.6$) with the accumulated submap cloud $\mathbf{C}_{\mathcal{S}_{\text{ref}}}$, the new scan is integrated into a submap $\mathcal{S}_{\text{ref}}$, and the accumulated submap cloud $\mathbf{C}_{\mathcal{S}_{\text{ref}}}$ grows by adding $\mathbf{C}_{\text{read}}$. For instance, in Fig. 4.11 (a) and (b), node $\mathcal{L}_{17}$ is integrated into submap $\mathcal{S}_5$, and LiDAR scan $\mathbf{C}_{17}$ is accumulated into the submap cloud $\mathbf{C}_{\mathcal{S}_5}$. However, I constrain point cloud accumulation of each submap by not combining accumulated submap clouds together during submap fusion. In Fig. 4.11, the volumetric occupancy submaps $\mathcal{S}_0$ and $\mathcal{S}_5$ in (c) are fused together into one submap $\mathcal{S}'_0$ in (d), but

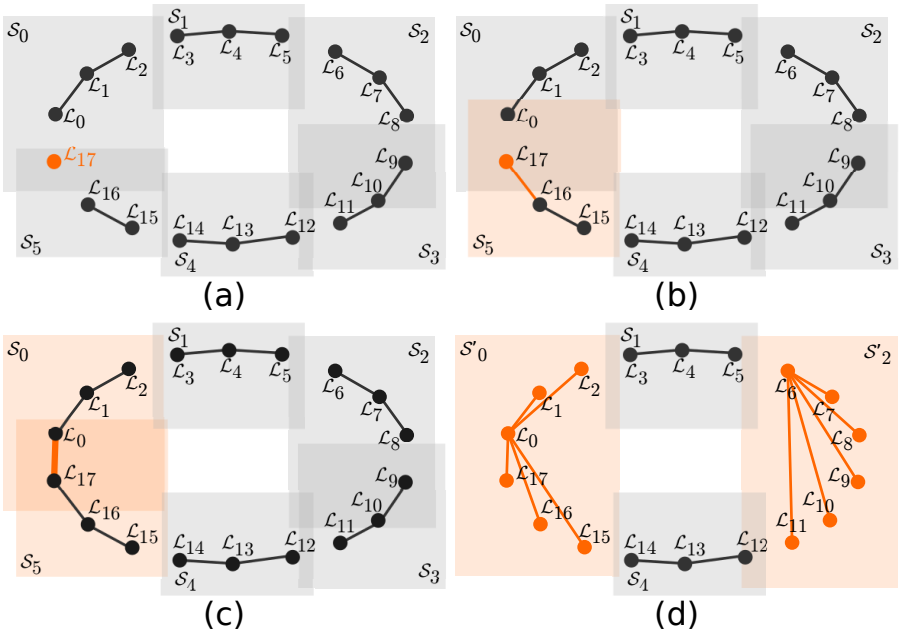


Figure 4.11: An example of scan integration and submap fusion. **(a)** There are 17 LiDAR scans $\mathcal{L}_{0:16}$ in the existing reconstruction. The scans are clustered into 6 submaps $\mathcal{S}_{0:5}$, shown as grey rectangles (■) which represent the AABB of each submap. $\mathcal{L}_{17}$ is the latest new LiDAR scan, and there have been no loop closures. **(b)** *Graph Clustering* allocates $\mathcal{L}_{17}$ to submap $\mathcal{S}_5$, and the new scan passes the *Cloud Overlap Estimate* (Section 4.7.3). Then $\mathcal{L}_{17}$ is integrated into $\mathcal{S}_5$, expanding the AABB (as shown by the orange rectangle ■). **(c)** There is a new loop closure edge (orange line —) given by the SLAM pose graph between $\mathcal{L}_0$ and $\mathcal{L}_{17}$. The head and tail of the loop closure connection define the submap overlap search range from $\mathcal{S}_0$ to $\mathcal{S}_5$. **(d)** *Graph Clustering* proposes the fusion of submaps $\mathcal{S}_5$ and $\mathcal{S}_0$ into $\mathcal{S}'_0$. *Submap Overlap Estimate* (Section 4.8.2) proposes the fusion between submaps $\mathcal{S}_3$ and $\mathcal{S}_2$ in $\mathcal{S}'_2$. If both proposals pass the *Relative Uncertainty* criterion (Section 4.9), fusion is executed and all submap indices are updated accordingly. The AABBs of $\mathcal{S}_0$ and $\mathcal{S}_2$ are therefore expanded (shown by ■), but their accumulated submap clouds are not, as explained in Section 4.7.3.

submap cloud $\mathbf{C}_{\mathcal{S}_5}$ is not combined with $\mathbf{C}_{\mathcal{S}_0}$. This prevents these submap clouds from growing indefinitely as exploration continues.

The criterion ($R_{\text{point,read}} > 0.6$) for spawning a new submap is based on the intuition that $50\,\%$ of the point cloud is inside and outside of a room when the LiDAR is perfectly in the doorway. In our experiments, setting the threshold at $60\,\%$ means a more conservative submap spawning behaviour in which the reconstruction becomes more elastic when the reliability of ICP in odometry and SLAM is low due to limited Cloud Overlap Estimate (e.g. corridors, stairs). Increasing the threshold further leads to a higher memory cost. A threshold

lower than $50\,\%$ could affect the functionality of room segmentation in the proposed system.

4.7.4 Submap Pose Update

The Submap Pose Update module ensures global consistency in the reconstruction. When loop closure occurs, the pose graph and poses of the SLAM system are updated.

The naive approach of updating all the submaps upon loop closure is computationally infeasible for real-time applications, as discussed by Sodhi et al. [139]. Instead, I define a criterion to determine whether a submap $\mathcal{S}_i$ needs to be corrected, such that a large-scale reconstruction can be selectively and efficiently updated. Let ${}^{\mathcal{M}}\hat{\mathbf{T}}_{\mathcal{S}_i}$ denote the updated transformation ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}$ of $\mathcal{S}_i$ with respect to the map frame $\mathcal{M}$. I empirically determined translational and rotational thresholds which trigger a submap correction, respectively $10\,\mathrm{cm}$ and $2.5°$. If the position/rotation change exceeds its threshold, the submap is corrected:

$$||{}^{\mathcal{M}}\hat{\mathbf{t}}_{\mathcal{S}_i} - {}^{\mathcal{M}}\mathbf{t}_{\mathcal{S}_i}|| > d_{\text{update}} \vee ||{}^{\mathcal{M}}\hat{\mathbf{R}}_{\mathcal{S}_i}^{-1}{}^{\mathcal{M}}\mathbf{R}_{\mathcal{S}_i}|| > \theta_{\text{update}} \tag{4.5}$$

Because I do not maintain a global map in the SE-Atlas pipelines, this update only needs to correct the root poses of the submaps, with no additional global map fusion required.

4.8 Scalability via Submap Fusion

The Submap Fusion module in Fig. 4.6 merges the submaps and prevents new submaps from being spawned when the same space is revisited. Updating an existing submap is more memory efficient than creating two overlapping submaps. This has the advantage of making the reconstruction complexity grow proportionally with the amount of space explored rather than the duration of the exploration.

4.8.1 Loop Closure Fusion

The basic submap fusion strategy in the proposed system combines submaps where a loop closure is detected [11]. For each loop closure cluster described in Section 4.7.1, I search through all existing submaps and find those that contain nodes from this cluster. These submaps are then fused together as illustrated in Fig. 4.9.

To fuse the submaps $\mathcal{S}_j$ and $\mathcal{S}_i$, I first transform every voxel of $\mathcal{S}_j$ with coordinates $\mathbf{v}_j \in \mathbb{R}^3$ into the coordinate system of $\mathcal{S}_i$ to obtain $\mathbf{v}_i$:

$$\begin{bmatrix} \mathbf{v}_i \\ 1 \end{bmatrix} = {}^{\mathcal{S}_i}\mathbf{T}_{\mathcal{S}_j} \begin{bmatrix} \mathbf{v}_j \\ 1 \end{bmatrix}, \; {}^{\mathcal{S}_i}\mathbf{T}_{\mathcal{S}_j} = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}^{-1} \, {}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_j} \tag{4.6}$$

If the voxel $\mathbf{v}_i$ falls out of the current scanned space in $\mathcal{S}_i$, it will be newly allocated and assigned as $\mathbf{v}_j$ in $\mathcal{S}_j$. Otherwise, the voxel data in $\mathbf{v}_j$ will be integrated into $\mathbf{v}_i$ following the model in Section 4.5.2.

4.8.2 Submap Overlap Estimate

Submap fusion merges existing submaps, and reduces the memory usage of the overall system by fusing repeated reconstructions of the same physical space together. The loop closure-based strategy explained in Section 4.8.1 [11] triggers submap fusion using the loop closures detected in the SLAM system. However, it only merged submaps that were created when the robot travelled very close to a previous pose, i.e. the submaps at the head and tail ends of the loop closure.

In a large-scale (outdoor) environment, a long range ($\approx$60 m) LiDAR sensor can repeatedly scan the same space from poses that are far away from one another, resulting in significant redundancy between submaps that loop closure fusion alone cannot address. Therefore I introduce an additional strategy for submap fusion based on the overlap of scanned spaces, leveraging the explicit representation of free space in volumetric occupancy maps. This improves the reconstruction scalability when revisiting explored areas, in spite of the SLAM graph growing linearly.

Algorithm 2: Submap Overlap Estimate.

input: Pair of occupancy submaps $\mathcal{S}_{\text{read}}$ and $\mathcal{S}_{\text{ref}}$,
output: Submap overlap ratios $R_{\text{voxel,read}}$ and $R_{\text{voxel,ref}}$
begin
  for Voxel $\mathbf{v}_{\text{read}} \subset \mathcal{S}_{\text{read}}$ **do**
    Find $\mathbf{v}_{\text{ref}} \subset \mathcal{S}_{\text{ref}}$ at the same coordinates as $\mathbf{v}_{\text{read}}$
    if $\mathbf{v}_{\text{ref}}$ *is not unknown* **then**
      if *Both* $\mathbf{v}_{\text{ref}}$ *and* $\mathbf{v}_{\text{read}}$ *are free or occupied* **then**
        $N_{\text{voxel,overlap}} = N_{\text{voxel,overlap}} + 1$
      end if
    end if
  end for
  $R_{\text{voxel,read}} = N_{\text{voxel,overlap}} / N_{\text{voxel,read}}$
  $R_{\text{voxel,ref}} = N_{\text{voxel,overlap}} / N_{\text{voxel,ref}}$
  return $R_{\text{voxel,read}}$ and $R_{\text{voxel,ref}}$
end

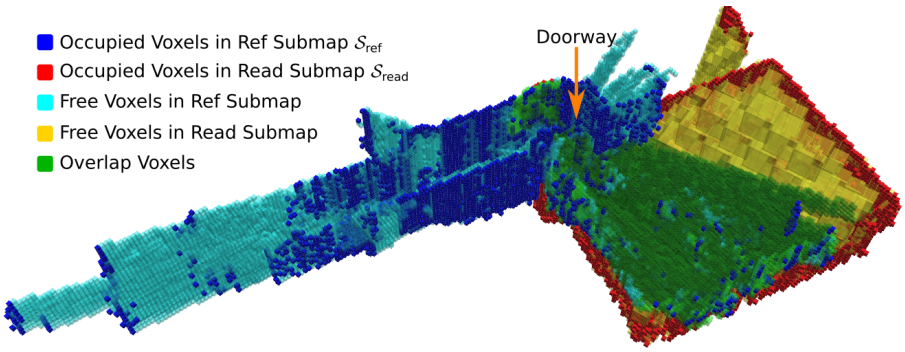


Figure 4.12: An example of Submap Overlap Estimate, highlighting the voxel overlap between the volumetric occupancy reconstructions of the reference submap $\mathcal{S}_{\text{ref}}$ and the read submap $\mathcal{S}_{\text{read}}$.

Submap overlap is computed by comparing the volumetric reconstruction as well as the occupancy information stored in each individual submap. Fig. 4.12 demonstrates such a case. Alg. 2 describes the estimation of voxel overlap between a pair of submaps $\mathcal{S}_{\text{read}}$ and $\mathcal{S}_{\text{ref}}$. This pair of submaps are fused together if either the ratio $R_{\text{voxel,ref}}$ or $R_{\text{voxel,read}}$ exceeds a configurable threshold λ_{fusion}. This threshold is empirically chosen as 0.7. It represents significant redundancy among submaps and does not require further tuning between experiments.

To ensure that the root poses of submaps are corrected by loop closure before fusion, a submap overlap search range is defined using the head and tail of each loop closure. For example, in Fig. 4.11 (c), SLAM loop closure is between $\mathcal{L}_0$

and $\mathcal{L}_{17}$, and they belong to $\mathcal{S}_0$ and $\mathcal{S}_5$, respectively. Hence the search range for Submap Overlap Estimate is $\mathcal{S}_{0:5}$. Candidate submap pairs for fusion are those that have an overlap greater than the threshold, e.g. $\mathcal{S}_2$ and $\mathcal{S}_3$.

Iterating through all voxels is a computationally intense process. Therefore, I add a conservative but efficient preliminary heuristic based on the AABB of each submap before computing submap voxel overlap. Using the AABB of $\mathcal{S}_{\text{ref}}$ and $\mathcal{S}_{\text{read}}$, I compute the volumetric overlap percentages $\{R_{\text{aabb,ref}}, R_{\text{aabb,read}}\}$ and compare them with λ_{fusion}. If both AABB overlaps are smaller than the threshold, such as $\mathcal{S}_1$ and $\mathcal{S}_4$ in Fig. 4.11 (c), the proposed system skips computing voxel overlap. Because this is only a preliminary check to avoid unnecessary submap overlap computations, orientated bounding boxes are not necessary, and using AABB instead helps keeping this step computationally light.

4.9 Relative Uncertainty

To stay memory efficient, SE-Atlas discards the individual submaps after fusing them into a parent. However, since each submap is internally rigid, local consistency is essential within the submaps. To improve submap fusion reliability and, consequently, retain global consistency, a strategy is proposed to keep the errors within each submap bounded.

The proposed approach uses the relative uncertainty between the root poses of the candidate submaps to measure fusion confidence. To compute the relative uncertainty, SE-Atlas uses the Hessian matrix of the SLAM problem to extract a joint covariance of the root poses of submaps. The following formula was derived based on the conventions used in the GTSAM library [8], with reference to the approach of Mangelson et al. [7] to determine a single covariance matrix encoding the relative uncertainty between the poses.

First, the probability distribution of the relative transformation from submap $\mathcal{S}_i$ to $\mathcal{S}_j$ is defined as:

$$^{\mathcal{S}_i}\mathbf{T}_{\mathcal{S}_j} = {}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}^{-1}{}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_j} \tag{4.7}$$

where the poses ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_i}$ and ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_j}$ indicate probability distributions on $\mathbf{SE}(3)$ following a right-hand composition:

$$\mathbf{T} = \bar{\mathbf{T}}\,\mathrm{Exp}(\xi) \tag{4.8}$$

$\bar{\mathbf{T}}$ is the mean transformation of the distribution, and ξ is a perturbation that follows a Gaussian distribution. In Eq. (4.7) we consider that the poses have covariances $\Sigma_{\mathcal{M}\mathcal{S}_i}$ and $\Sigma_{\mathcal{M}\mathcal{S}_j}$, respectively.

In order to derive the expressions for the relative uncertainty, the adjoint action of $\bar{\mathbf{T}}$ on ξ is needed, denoted as $\mathrm{Ad}_{\bar{\mathbf{T}}}(\xi)$, and defined as follows:

$$\begin{aligned}
\mathrm{Ad}_{\bar{\mathbf{T}}}(\xi) := \mathrm{Ad}_{\bar{\mathbf{T}}}\xi &= \mathrm{Log}(\bar{\mathbf{T}}\,\mathrm{Exp}(\xi)\bar{\mathbf{T}}^{-1}) \\
\mathrm{Exp}(\mathrm{Ad}_{\bar{\mathbf{T}}}\xi) &= \bar{\mathbf{T}}\,\mathrm{Exp}(\xi)\bar{\mathbf{T}}^{-1} \\
\bar{\mathbf{T}}^{-1}\mathrm{Exp}(\mathrm{Ad}_{\bar{\mathbf{T}}}\xi) &= \mathrm{Exp}(\xi)\bar{\mathbf{T}}^{-1} \\
\mathrm{Exp}(\xi)\bar{\mathbf{T}} &= \bar{\mathbf{T}}\,\mathrm{Exp}(\mathrm{Ad}_{\bar{\mathbf{T}}^{-1}}\xi)
\end{aligned} \tag{4.9}$$

Expanding Eq. (4.7) using Eq. (4.8) and Eq. (4.9):

$$\begin{aligned}
&{}^{\mathcal{S}_i}\bar{\mathbf{T}}_{\mathcal{S}_j}\mathrm{Exp}({}^{\mathcal{S}_i}\xi_{\mathcal{S}_j}) \\
&= \mathrm{Exp}(-{}^{\mathcal{M}}\xi_{\mathcal{S}_i})\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}^{-1}\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}\mathrm{Exp}({}^{\mathcal{M}}\xi_{\mathcal{S}_j}) \\
&= {}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}^{-1}\mathrm{Exp}(-\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}}{}^{\mathcal{M}}\xi_{\mathcal{S}_i})\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}\mathrm{Exp}({}^{\mathcal{M}}\xi_{\mathcal{S}_j}) \\
&= {}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}^{-1}\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}\mathrm{Exp}(-\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}}{}^{\mathcal{M}}\xi_{\mathcal{S}_i})\mathrm{Exp}({}^{\mathcal{M}}\xi_{\mathcal{S}_j})
\end{aligned} \tag{4.10}$$

Let ${}^{\mathcal{S}_i}\bar{\mathbf{T}}_{\mathcal{S}_j} \triangleq {}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}^{-1}\,{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}$, then the following equivalence can be established:

$$\mathrm{Exp}({}^{\mathcal{S}_i}\xi_{\mathcal{S}_j}) = \mathrm{Exp}(-\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}}{}^{\mathcal{M}}\xi_{\mathcal{S}_i})\mathrm{Exp}({}^{\mathcal{M}}\xi_{\mathcal{S}_j}) \tag{4.11}$$

The covariance of the perturbation on the left should be equal to the one on the right. However, the covariance cannot be computed directly because of the properties of the exponential map. Instead, we define ${}^{\mathcal{S}_i}\xi'_{\mathcal{S}_j} = -\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{{}^{\mathcal{M}}\bar{\mathbf{T}}_{\mathcal{S}_i}}{}^{\mathcal{M}}\xi_{\mathcal{S}_i}$, and use the Baker-Campbell-Hausdorff (BCH) formula [178] up to first order:

$$\begin{aligned}
E[{}^{\mathcal{S}_i}\xi_{\mathcal{S}_j}\,{}^{\mathcal{S}_i}\xi_{\mathcal{S}_j}^T] \approx\; & E[{}^{\mathcal{M}}\xi'_{\mathcal{S}_i}\,{}^{\mathcal{M}}\xi'^T_{\mathcal{S}_i}] + E[{}^{\mathcal{M}}\xi_{\mathcal{S}_j}\,{}^{\mathcal{M}}\xi_{\mathcal{S}_j}^T] \\
& + E[{}^{\mathcal{M}}\xi'_{\mathcal{S}_i}\,{}^{\mathcal{M}}\xi_{\mathcal{S}_j}^T] + E[{}^{\mathcal{M}}\xi_{\mathcal{S}_j}\,{}^{\mathcal{M}}\xi'^T_{\mathcal{S}_i}]
\end{aligned} \tag{4.12}$$

which, after computing the covariance terms, provides an approximation for the covariance of the relative transformation:

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathcal{S}_i\mathcal{S}_j} \approx\ & (\mathrm{Ad}_{\mathcal{M}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{\mathcal{M}\bar{\mathbf{T}}_{\mathcal{S}_i}})\boldsymbol{\Sigma}_{\mathcal{M}\mathcal{S}_i}(\mathrm{Ad}_{\mathcal{M}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{\mathcal{M}\bar{\mathbf{T}}_{\mathcal{S}_i}})^T \\ & + \boldsymbol{\Sigma}_{\mathcal{M}\mathcal{S}_j} \\ & - (\mathrm{Ad}_{\mathcal{M}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{\mathcal{M}\bar{\mathbf{T}}_{\mathcal{S}_i}})\boldsymbol{\Sigma}_{\mathcal{M}\mathcal{S}_i,\mathcal{M}\mathcal{S}_j} \\ & - \boldsymbol{\Sigma}_{\mathcal{M}\mathcal{S}_i,\mathcal{M}\mathcal{S}_j}^T(\mathrm{Ad}_{\mathcal{M}\bar{\mathbf{T}}_{\mathcal{S}_j}^{-1}}\mathrm{Ad}_{\mathcal{M}\bar{\mathbf{T}}_{\mathcal{S}_i}})^T\end{aligned} \tag{4.13}$$

Lastly, we compute the eigenvalues of the relative uncertainty and apply a threshold $\lambda_{\text{uncertainty}}$ on them to finally decide if the fusion is accepted.

As an illustration, Fig. 4.11 presents the case of fusing two pairs of submaps, namely $\mathcal{S}_0$ and $\mathcal{S}_5$, and $\mathcal{S}_2$ and $\mathcal{S}_3$. In the example of fusing $\mathcal{S}_0$ and $\mathcal{S}_5$, SE-Atlas first computes the relative uncertainty between the root poses of $\mathcal{S}_0$ and $\mathcal{S}_5$, which are $\mathcal{L}_0$ and $\mathcal{L}_{15}$.

For $\mathbf{SE}(3)$ transformations ${}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_0}, {}^{\mathcal{M}}\mathbf{T}_{\mathcal{S}_5} \in \mathbb{R}^6$, the relative uncertainty $\boldsymbol{\Sigma}_{\mathcal{S}_0\mathcal{S}_5}$ is a 6×6 matrix, and there are 6 eigenvalues — 3 for translation and 3 for rotation. SE-Atlas compares the 3 translation eigenvalues against the configurable threshold $\lambda_{\text{uncertainty}}$. If one of the eigenvalues exceeds the threshold, the fusion between $\mathcal{S}_0$ and $\mathcal{S}_5$ is rejected. I set $\lambda_{\text{uncertainty}}$ to be $0.2\,\mathrm{m}$ in our experiments based on our SLAM noise model, which is not subject to change between experiments.

4.10 Experimental Results

This section presents the results of our proposed system, SE-Atlas, tested using a wide variety of experiments in both simulation and real world. Features of SE-Atlas explained in this chapter are demonstrated here, namely the integration speed of long-range LiDAR scans at high resolution, the scalability of map memory consumption in large-scale long-term explorations, and the improvement in reconstruction accuracy via loop closure and motion aware LiDAR integration. A further ablation study about the effect of each SE-Atlas component on reconstruction accuracy and coverage is presented in Section 4.11.

Figure 4.13: The Boston Dynamics Spot robot with a Frontier multi-sensor rig mounted on it. Frontier contains an Ouster LiDAR and an Intel Realsense RGB-D camera. The Spot carries 6 Intel Realsense cameras.

4.10.1 Experiment Setup

This section presents the series of datasets that the proposed system has been assessed with. These experiments include:

- Exp 4.10.2 (ARCHE):
  A large-scale outdoor experiment with a MAV presented by Reijgwart et al. [5]. The MAV carries an Ouster LiDAR.

- Exp 4.10.2 (NCD Long):
  A large-scale outdoor experiment with a handheld Frontier device in the Newer College Dataset (NCD) [14]. The Frontier multi-sensor rig contains an Ouster LiDAR and an Intel Realsense D435i RGB-D camera.

- Exp 4.10.6 (ORI):
  A multi-storey multi-room mapping experiment in the Oxford Robotics Institute (ORI) with a Boston Dynamics Spot robot as presented in Fig. 4.13. The Spot robot carries a Frontier multi-sensor rig.

- Exp 4.10.7 (Simulation):
  Looping explorations of a Unmanned Ground Vehicle (UGV) in a small and a large room network using the Gazebo simulator.

Table 4.1 gives details of the different LiDAR sensors used in these experiments. These LiDAR sensors all produce organised point cloud scans of 64×1024

Experiment	Section	LiDAR properties			
		Model	VFoV	HFoV	Max range (m)
ARCHE (MAV)	4.10.2	Ouster OS1-64	33.2°	360°	120
NCD Long (handheld)	4.10.2	Ouster OS1-64	33.2°	360°	120
ORI (Spot)	4.10.6	Ouster OS0-64	90°	360°	50
Indoor-outdoor (Spot)	5.6	Ouster OS0-64	90°	360°	50
Simulation (UGV)	4.10.7	Ouster OS0-64	90°	360°	50

Table 4.1: LiDAR sensors used in the experiments and their properties. VFoV: Vertical Field of View; HFoV: Horizontal Field of View

points at $10\,\mathrm{Hz}$. The SLAM system creates a node in its pose graph every $2\,\mathrm{m}$ travelled when exploring. The proposed system integrated the LiDAR scans using the *MultiresOFusion* mode [11] for volumetric occupancy reconstruction. The voxel resolution used in these experiments was $6.5\,\mathrm{cm}$ and SE-Atlas integrated LiDAR ranges between $0.5\,\mathrm{m}$ and $60\,\mathrm{m}$. These settings give high resolution and long range while retaining $3\,\mathrm{Hz}$ integration.

Surface mesh representations of the reconstruction, for example Fig. 4.17, were created by applying the Marching Cubes algorithm [129] on the zero-crossings of the occupancy map.

4.10.2 Large-scale Outdoor Experiments

In this section, I first evaluate the efficiency of SE-Atlas when integrating LiDAR scans using the NCD [14] and the dataset made available with [5] (ARCHE). NCD consists of two experiments with different durations, and I present in this chapter the $2.2\,\mathrm{km}$ sequence which is over $44\,\mathrm{min}$ long (NCD Long). Both datasets were large-scale outdoor experiments (approximately $135\times225\,\mathrm{m}^2$ for NCD and $70\times160\,\mathrm{m}^2$ for ARCHE) with an Ouster OS1-64 LiDAR and a RealSense camera (D435i for NCD and D415 for ARCHE).

I then demonstrate the map memory efficiency of SE-Atlas using both NCD Long and ARCHE experiments compared against OctoMap [137] and Voxgraph [5]. I further present the improved scalability in NCD Long reconstructions by incorporating Submap Overlap Estimation as explained in Section 4.8. Last but not least, I show the improvement in reconstruction

accuracy via Cloud Overlap Estimation (Section 4.7.3) and Relative Uncertainty Estimation (Section 4.9).

To emphasise the effect of Principled Clustering strategies, in this section I refer to the SE-Atlas pipeline without any spatial or relative uncertainty analysis as the **baseline**, and the full SE-Atlas system as the **proposed**.

4.10.3 *Supereight* Runtime Efficiency

To evaluate the LiDAR scan integration efficiency of SE-Atlas, I compared the performance of SE-Atlas in large-scale outdoor experiments to other state-of-the-art reconstruction methods. The benchmark algorithms I chose for comparison are OctoMap [137] and Voxgraph [5], to assess the respective Occupancy and TSDF pipelines. For these experiments, I fed one point cloud every $2\,\mathrm{m}$ travelled into the reconstruction systems. All reconstruction computations were performed on a laptop with an Intel® Xeon E3-1505M v6 CPU, 16 GB of RAM and 32 GB of swap memory.

I evaluated the computation time using three different sets of maximum scan range and voxel resolutions:

- $20\,\mathrm{m}$ max range with $26\,\mathrm{cm}$ resolution
- $60\,\mathrm{m}$ max range with $26\,\mathrm{cm}$ resolution
- $60\,\mathrm{m}$ max range with $6.5\,\mathrm{cm}$ resolution

Fig. 4.14 shows the integration time at the different range/resolution combinations for NCD Long and ARCHE experiments (top and bottom rows, respectively). We focus on mapping at high resolution ($6.5\,\mathrm{cm}$) with maximum LiDAR range ($60\,\mathrm{m}$), which is presented in the right column of Fig. 4.14. In both experiments, Voxgraph terminated early due to memory limits, as did OctoMap in NCD Long.

Overall, OctoMap is the least efficient of the evaluated methods. With coarse resolutions, Voxgraph exhibits similar performance to *supereight*. However, at

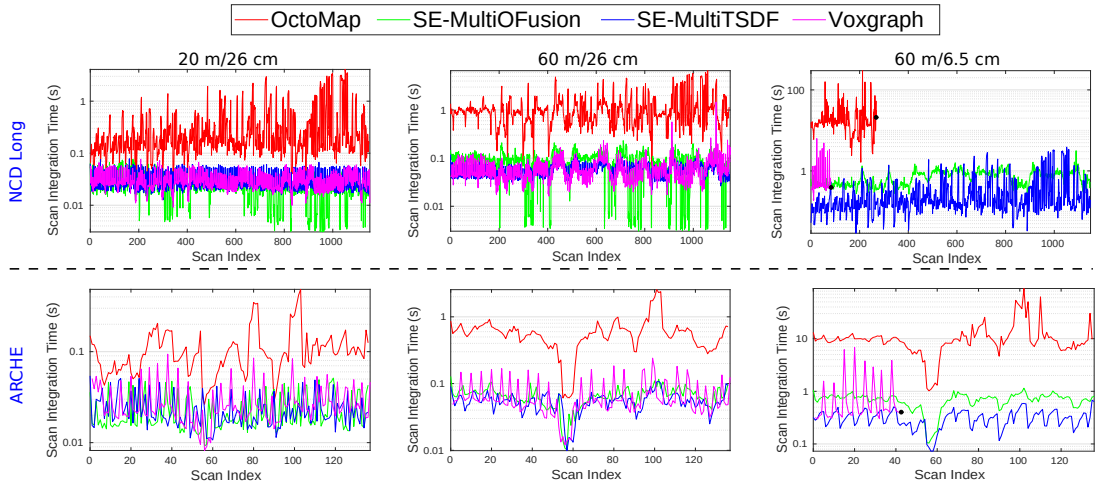


Figure 4.14: Exp 4.10.2 – Integration time per LiDAR scan of different reconstruction systems in large-scale exploration experiments. Our goal is to achieve high resolution at maximum sensor range (right column). Note that the timing plots are in log scale and that the axes are different on each plots

$6.5\,\mathrm{cm}$ resolution, the MultiresTSDF pipeline is faster than Voxgraph, while MultiresOFusion is on a par with Voxgraph.

4.10.4 Reconstruction Memory Scalability

Fig. 4.15 shows the memory consumption of each pipeline, as well as the growth of the number of submaps in the proposed system. The memory usage of OctoMap and Voxgraph increases more quickly than both SE-Atlas pipelines, thus illustrating how *supereight*'s multi-resolution feature improves the memory efficiency of the reconstruction, allowing it to scale to larger environments especially when integrating long-range LiDAR scans.

Please note that in the experiments presented in Fig. 4.15, Submap Overlap Estimation was not leveraged and submap fusion was only based on SLAM loop closures, as explained in Section 4.8.1. Submap fusion prevents new submaps from being spawned when the same space is revisited. Updating an existing submap is more memory efficient than creating two overlapping submaps.

The memory usage of the long-range ($60\,\mathrm{m}$) high-resolution ($6.5\,\mathrm{cm}$) NCD experiments, as presented in Fig. 4.15, demonstrates such benefit. Fig. 4.15 also presents the growth of submaps in both experiments. Up to scan 400 in the

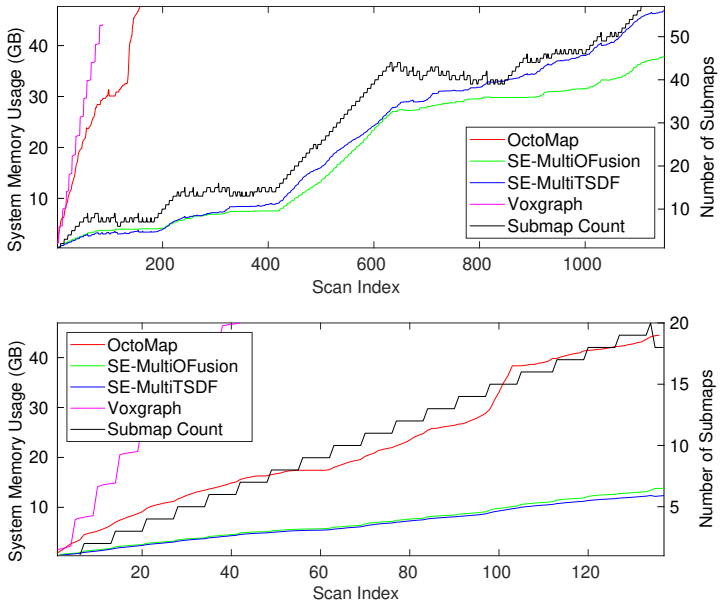


Figure 4.15: Exp 4.10.2 – Memory usage of each pipeline in the NCD Long (top) and ARCHE (bottom) experiments with 60 m range and 6.5 cm resolution. Memory usage of our pipelines had a non-linear profile in NCD Long because of the submap fusion feature.

NCD Long experiment, the number of submaps had limited growth because the experiment stayed within the same area and loops were closed. Thereafter the device explored new open area - with submap growth becoming linear. Ideally submap growth should have fully plateaued when revisiting the same area regularly (after scan 600). This is the reason why Submap Overlap Estimation is incorporated into SE-Atlas to improve system scalability and to enable the submap count to plateau.

Improved Scalability with Spatial Overlap Analysis

By leveraging Submap Overlap Estimation, submaps that have significant overlapping scan volumes are also fused together. This is in addition to submap fusion based on loop closures. This allows the map to properly scale with the size of the environment rather than the length of the exploration. Fig. 4.16 presents the memory usage and submap counter in NCD Long of the proposed SE-Atlas with Submap Overlap Estimation, compared to the baseline SE-Atlas without it. The memory usage is computed by summing the size of allocated

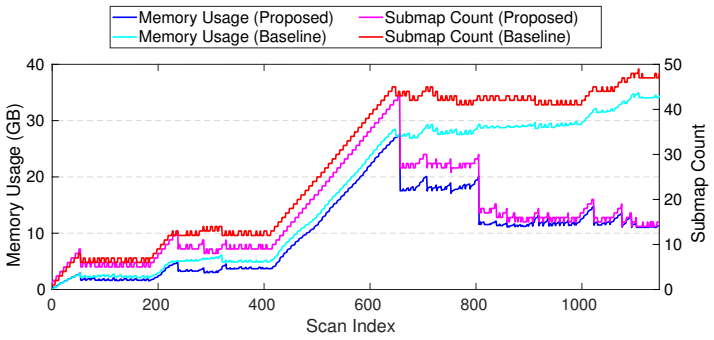


Figure 4.16: Exp 4.10.2 – The memory usage and submap counters of the proposed system with Submap Overlap Estimation and the baseline without it in the NCD Long experiment.

memory for each submap's octree in RAM.

In the baseline method, the submaps can only be merged when loop closures occur which caused memory usage to grow over time. In contrast, Submap Overlap Estimation allows spatially overlapping volumes to be merged, such that the number of submaps can plateau. For the NCD Long experiment, submap count stabilised at about 30 submaps when the entire environment has been explored at scan 650. Memory usage actually decreased after scan 650 to $\sim 18\,\mathrm{GB}$ while maintaining the $6.5\,\mathrm{cm}$ resolution reconstruction. By the end of the experiment, there was a $65\,\%$ reduction in memory usage by leveraging Submap Overlap Estimation compared to the baseline.

4.10.5 Reconstruction Accuracy

In this section, I evaluate the global consistency of the proposed online elastic reconstruction pipeline using MultiresTSDF by comparing the MultiresTSDF map with the ground truth of NCD, and demonstrate the further improvement to accuracy by employing Principled Clustering strategies as well as motion aware LiDAR integration. Section 4.11 further presents an ablation study about the effect of each SE-Atlas module on reconstruction accuracy.

Fig. 4.17 presents the surface mesh and volumetric reconstructions of SE-Atlas. The mesh is achieved by applying the Marching Cubes algorithm to

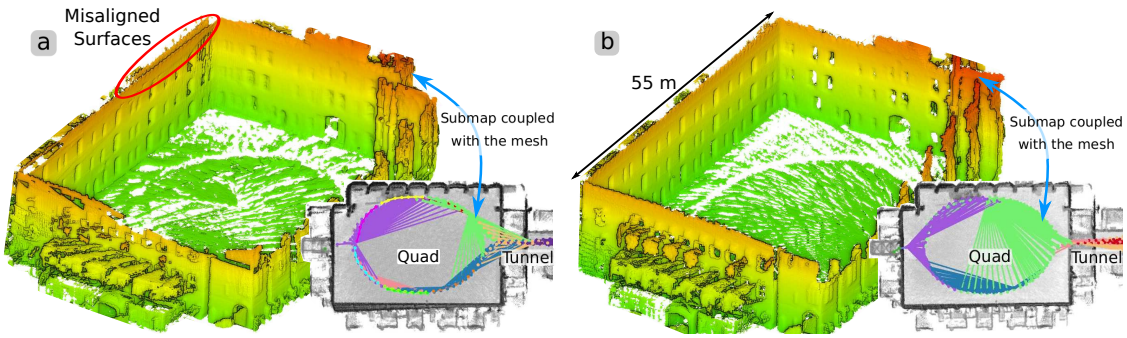


Figure 4.17: Exp 4.10.2 – The proposed spatial overlap analysis improves the global consistency in the reconstruction. Figure **(a)** and **(b)** present the mesh and volumetric representations of NCD Long experiment, created by the baseline (without Principled Clustering) and the full SE-Atlas, respectively. The volumetric representations are shown in grey overlaid with submap clusters. The mesh representations are created from the green submaps. Using spatial understanding of the environment leads to more reliable submap fusion and therefore better alignment. The mesh created without Principled Clustering has misaligned double surfaces while the full SE-Atlas improves the consistency in the mesh.

the MultiresTSDF map. Fig. 4.17 (a) shows the reconstructions created by baseline SE-Atlas without Principled Clustering strategies such as Cloud Overlap Estimation (Section 4.7.3) and Relative Uncertainty Estimation (Section 4.9). In particular the green submap in the Quad area was created as a rigid fusion of several submaps after multiple loop closures were established. As shown in the birds-eye view next to the mesh representation, this submap contains scans taken both in the Quad and the Tunnel. These scans have limited overlap with one another and thus registration between them is unreliable. This led to a duplicate reconstruction of the indicated wall of the Quad.

By applying the proposed Principled Clustering strategies, the reconstruction's consistency is improved as shown in Fig. 4.17 (b). The Cloud Overlap Estimate strategy (Section 4.7.3) makes better decisions when the handheld device travels between the Quad and the Tunnel — maintaining elastic connections between these two spaces in the global reconstruction. The measurement of relative uncertainty (Section 4.9) also rejects unreliable submap fusions. The global volumetric map and the submap mesh both demonstrate improved accuracy in surface alignment.

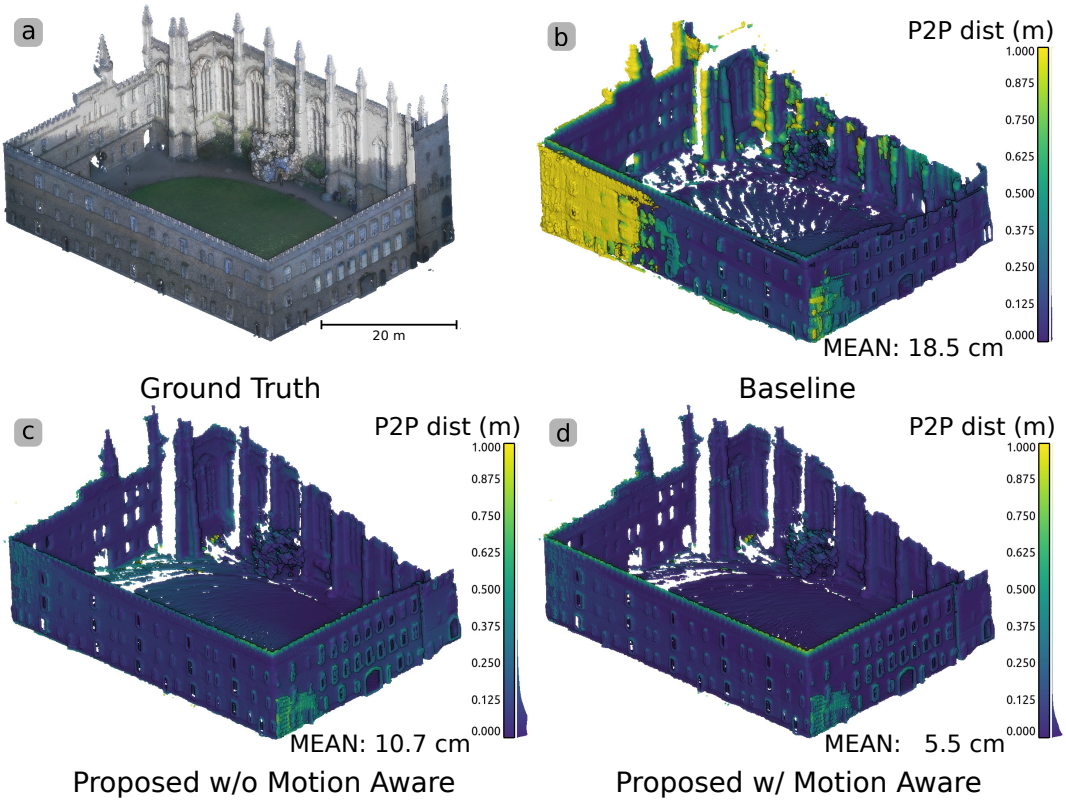


Figure 4.18: Exp 4.10.2 – The comparison between **(a)** the ground truth map of NCD Long experiment and each reconstruction created by **(b)** the baseline, **(c)** the proposed SE-Atlas without motion aware LiDAR integration and **(d)** the proposed SE-Atlas with motion aware LiDAR integration. Colours indicate point-to-point distances (P2P dist) between each reconstruction and the ground truth, and the distributions are also presented beside the colour bar.

The two submap reconstructions presented in Fig. 4.17 were also compared with the ground truth point cloud provided in the dataset [14]. I used CloudCompare[‡] to sample dense point clouds from both meshes, align the reconstructed point clouds with the ground truth, and compute the point-to-point distance error between them. I include both the result with and without the motion aware LiDAR integration module (Section 4.5.3). The results of a quantitative evaluation are presented in Fig. 4.18, together with the average point-to-point error of each reconstruction.

In Fig. 4.18 (b), though for about 90% of the points the error is less than 50 cm, misaligned surfaces created by the baseline method resulted in highly erroneous

[‡]https://www.danielgm.net/cc/

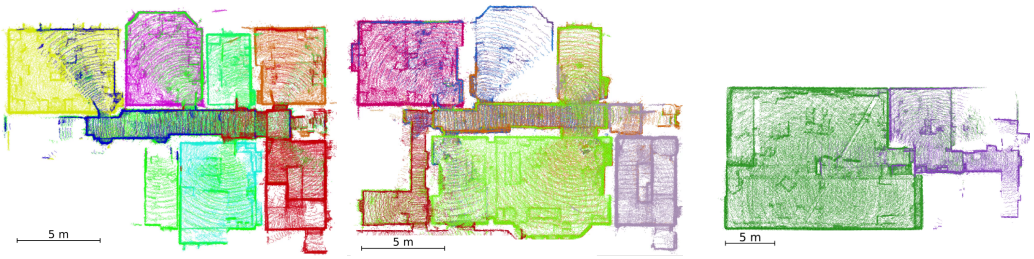


Figure 4.19: Exp 4.10.6 – The volumetric submap reconstructions of each floor of ORI (**Left:** first; **Middle:** ground; **Right:** basement), with each room segmented into unique submaps by the proposed system on the fly during exploration.

(>1 m) points behind existing ones, which are coloured yellow according to the colour map. With Principled Clustering strategies, the proposed system improved surface alignment and in turn the average point-to-point distance error (Fig. 4.18 (c)).

By adding the motion aware LiDAR integration module, the accuracy of the reconstruction is further improved, as presented in Fig. 4.18 (d). The floor is better aligned with the ground truth, and more points have <5 cm error. An ablation study has also been conducted to further assess the effect of each individual Principled Clustering strategy of SE-Atlas on reconstruction accuracy. Results of this ablation study are presented in Section 4.11.

4.10.6 Multi-storey Multi-room Indoor Exploration

In the multi-storey multi-room ORI experiment, a quadruped robot Spot explored three floors of a typical university research lab (Fig. 2.5 and Fig. 4.2). Mounted on the Spot was a copy of the Frontier multi-sensor payload containing an Ouster OS0-64 and an Intel Realsense D435i RGB-D camera. Fig. 4.19 shows the reconstructions of every floor in the building.

New submaps were spawned when the robot entered or exited rooms when the proposed system detected a decrease in cloud overlap. Submap Overlap Estimation then merged overlapping submaps in each room, creating a unique reconstruction for each enclosed space. It further ensured that these submaps remained independent allowing future SLAM loop closures to re-position the room submaps as needed.

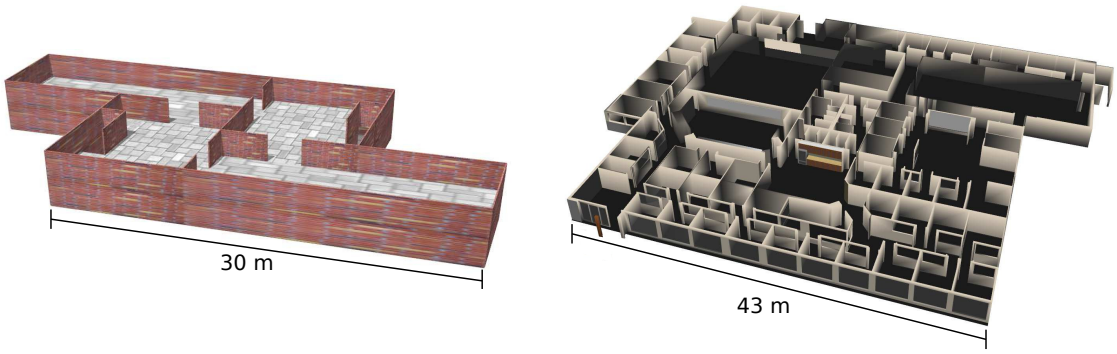


Figure 4.20: Exp 4.10.7 – The Gazebo environments of a small (left) and a large (right) room network for experiments in simulation.

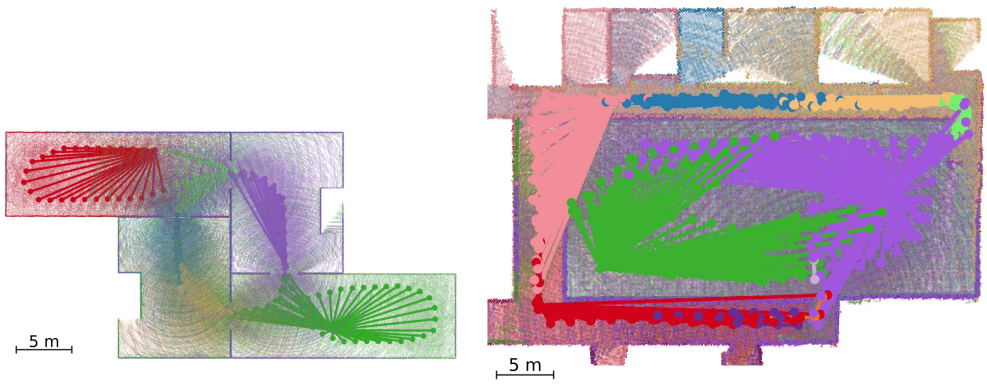
Segmenting rooms on the fly allows real-time applications such as path planning and obstacle avoidance to consider only the submaps local to the robot rather than the entire global reconstruction. The proposed system can thus improve the scalability of other applications.

4.10.7 Room Networks in Simulation

To test the performance for long term operation missions, we carried out two experiments in the Gazebo simulator with a wheel robot carrying a simulated 3D LiDAR in the environments in Fig. 4.20.

In the small room network, the mission looped around the environment three times in both directions. As shown by the clustered poses in Fig. 4.21 (a) there is a clear division between submaps at each doorway. Each room was constructed with either one or two submaps — even after multiple revisits. Fig. 4.21 (b) demonstrated an approximately $700\,\mathrm{m}$ exploration in a section of the large room network. Similarly, the room in the middle only contains two major submaps even after about 10 loops through the room taking different routes. Scans along corridors were also fused together based on overlap and loop closures, creating only 10 submaps in the end.

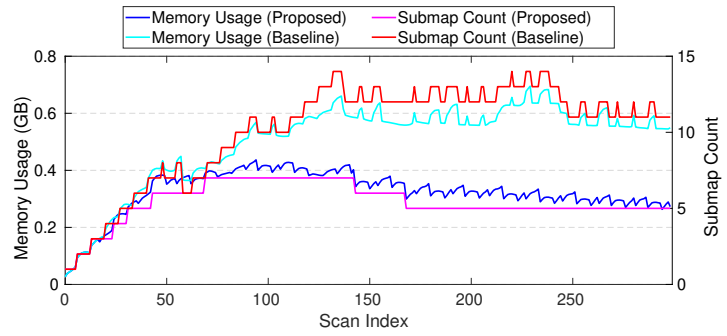
Fig. 4.22 shows that the submap count and memory usage in both simulation experiments plateaued as the environments were repeatedly scanned. The baseline approach in these experiments refers to SE-Atlas without Submap Overlap



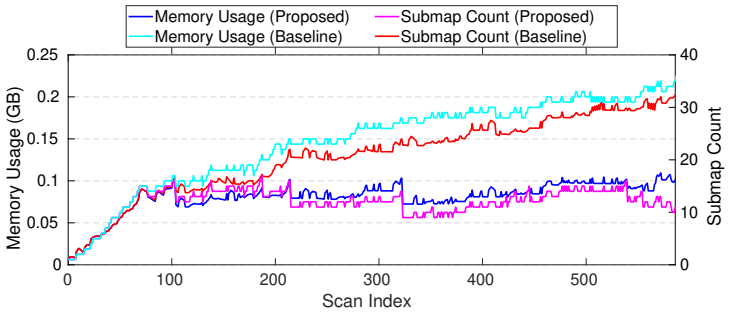
(a) Submap reconstruction in the small room network

(b) Submap reconstruction in the large room network

Figure 4.21: Exp 4.10.7 – The proposed system segments individual rooms and fuses redundant submaps during simulation experiments.



(a) The performance in the small room network (simulated)



(b) The performance in the large room network (simulated)

Figure 4.22: Exp 4.10.7 – The memory usage and submap counters of the proposed system and the baseline system in simulated experiments.

Estimate, which merges submaps based only on SLAM loop closures. Overall, the proposed method decreased the memory usage of both experiments by 50 % compared to the baseline method even after extensive exploration and revisiting, such as the experiment in the large room network (Fig. 4.21 (b)).

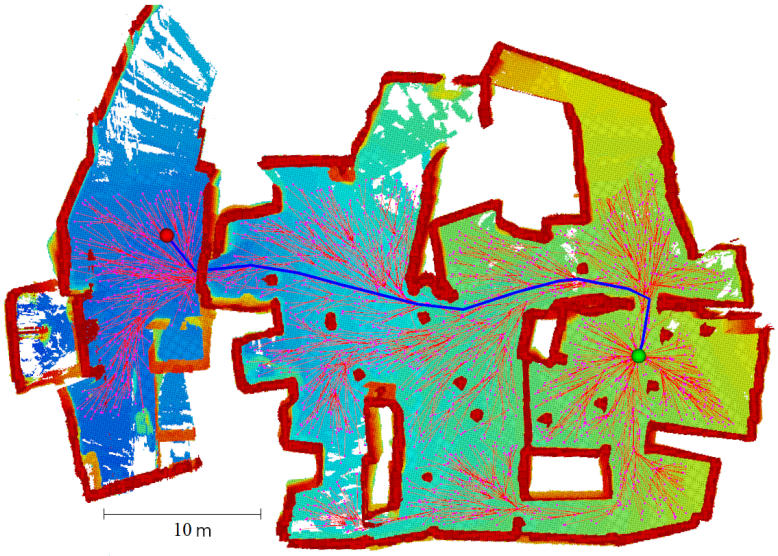


Figure 4.23: Using the reconstruction result for path planning in an underground room network. Green sphere - start; red sphere - end; red tree with magenta nodes - RRT*; blue trajectory - planned path.

4.10.8 Path Planning in Underground Network

To test the MultiresOFusion pipeline of SE-Atlas on a realistic path planning application, we collected a dataset in an underground mine in Corsham Wiltshire, consisting of a room network hewn from the rock. The dataset was collected with the same Frontier multi-sensor payload as in NCD, which contains an Ouster OS1-64, but mounted on a Husky wheeled robot. I ran SE-Atlas with $6.5\,\mathrm{cm}$ resolution and $60\,\mathrm{m}$ range. The resultant occupancy map was then used by an RRT* [31] path planner to compute the shortest collision free path between two locations. The result is presented in Fig. 4.23: the volumetric reconstruction is highly detailed, giving clear definition even in narrow doorways and corridors. This allowed the path planner to find the optimal path to the goal despite obstacles such a support pillars.

4.11 Ablation Study on SE-Atlas Reconstruction Accuracy

Expanding upon the results shown in Section 4.10.5, this section presents the effect of each proposed SE-Atlas Principled Clustering strategy on reconstruction accuracy using the NCD Long experiment [14]. The results of this ablation study are demonstrated in Table 4.2, comparing the reconstructions created by SE-Atlas using the following configurations:

- **Baseline** – No Principled Clustering strategies have been implemented.
- **With Cloud Overlap Estimate** – Only the Cloud Overlap Estimate strategy (Section 4.7.3) has been implemented.
- **With Submap Overlap Estimate** – Only the Submap Overlap Estimate strategy (Section 4.8.2) has been implemented.
- **With Both Overlap Estimate** – Both the Cloud Overlap Estimate and Submap Overlap Estimate strategies have been implemented.
- **With Relative Uncertainty** – All three Principled Clustering strategies, i.e. Cloud Overlap Estimate, Submap Overlap Estimate and Relative Uncertainty (Section 4.9), have been implemented.
- **With Motion Aware Integration** – In addition to all proposed Principled Clustering strategies, motion aware LiDAR integration (Section 4.5.2) has also been implemented.

From each configuration, I take the MultiresTSDF reconstruction, and apply the Marching Cubes algorithm [129] to generate a surface mesh. Then I sample a point cloud with $5\,\mathrm{cm}$ resolution from each mesh using CloudCompare. For reconstruction accuracy evaluation, I conduct the same point-to-point distance computation procedure as explained in Section 4.10.5, comparing each point cloud with the NCD ground truth map shown in Fig. 4.18.

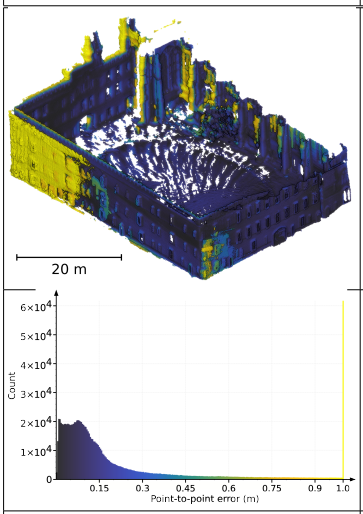
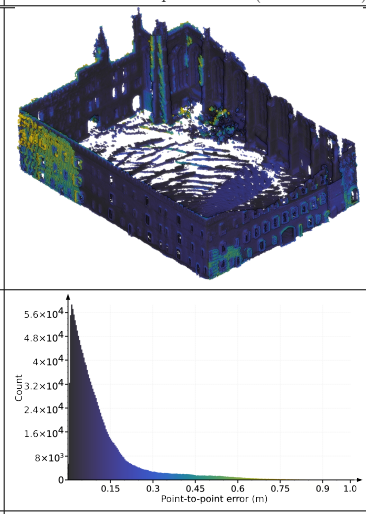
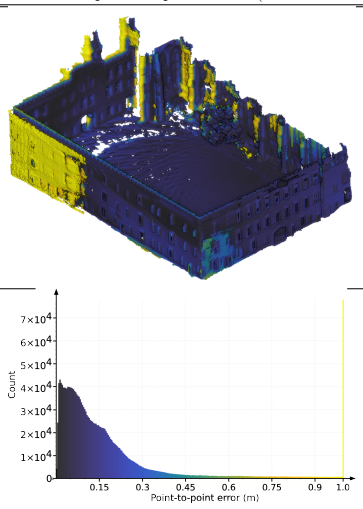
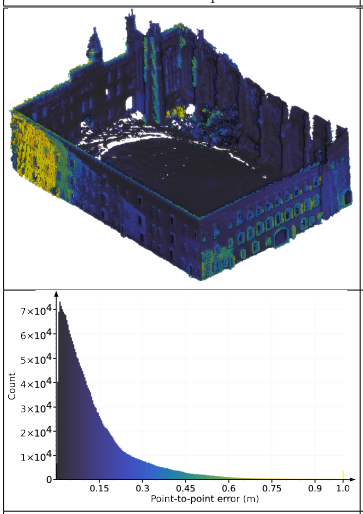
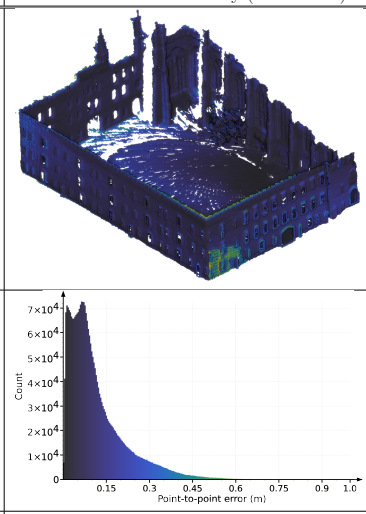
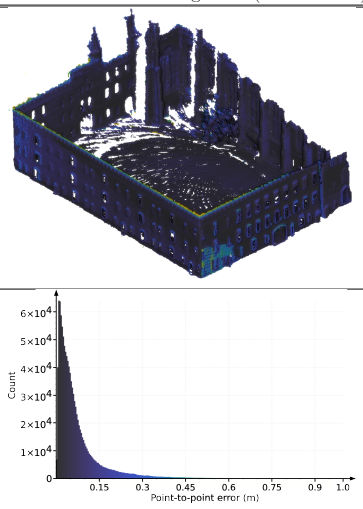
Baseline	With Cloud Overlap Estimate (Section 4.7.3)	With Submap Overlap Estimate (Section 4.8.2)
		
Mean: 18.5 cm Coverage: 35.9 %	Mean: 10.9 cm Coverage: 46.7 %	Mean: 16.4 cm Coverage: 39.3 %
With Both Overlap Estimate	**With Relative Uncertainty (Section 4.9)**	**With Motion Aware Integration (Section 4.5.3)**
		
Mean: 11.2 cm Coverage: 66.6 %	Mean: 10.7 cm Coverage: 54.1 %	Mean: 5.5 cm Coverage: 67.0 %

Table 4.2: Exp 4.10.2 – An ablation study on NCD reconstruction accuracy to demonstrate the effect of each SE-Atlas modules.

Table 4.2 demonstrates the point clouds sampled from aforementioned reconstructions that are coloured based on point-to-point distance errors, as well as the histograms that shows the distributions of these errors. I also compute the mean point-to-point distance error for each point cloud. The Cloud Overlap Estimate strategy (Section 4.7.3) leads to a significant decrease in mean distance error from 18.5 cm to 10.2 cm, because point clouds that have low point cloud overlap and low ICP confidence will not be integrated into the assessed submap. On the other hand, Submap Overlap Estimate strategy (Section 4.8.2) by itself

will not drastically improve reconstruction accuracy, as demonstrated by the reconstruction with double surfaces highlighted in yellow. The mean point-to-point distance decreases by $\sim 2.1\,\mathrm{cm}$ because more points have been fused into the presented submap by the Submap Overlap Estimate strategy. With both Submap Overlap Estimate and Cloud Overlap Estimate strategies implemented, the mean distance error is $11.2\,\mathrm{cm}$. The improvement is mostly the contribution of Cloud Overlap Estimate.

By incorporating submap fusion rejection based on the proposed Relative Uncertainty strategy (Section 4.9), the reconstruction accuracy is further improved, and the mean distance error is decreased to $10.7\,\mathrm{cm}$. There is also no double surface in the reconstruction. The Relative Uncertainty strategy rejects unreliable submap fusions; therefore submaps that have with low confidence and are isolated out by the Cloud Overlap Estimate strategy will remain disconnected from the presented submap. As explained in Section 4.10.5, combining all Principled Clustering strategies with motion aware LiDAR integration leads to the best reconstruction quality for SE-Atlas.

Table 4.2 also shows the effect of proposed Principled Clustering strategies on the map completeness of the assessed submaps. Cloud Overlap Estimate strategy results in a highly incomplete ground plane reconstruction because fewer point clouds have been integrated into the presented reconstruction, while Submap Overlap Estimate strategy significantly improves the completeness of the ground plane because this strategy fuses more submaps together. Similarly, incorporating Relative Uncertainty leads to fewer submap fusions and a less complete ground reconstruction.

In order to quantitatively demonstrate this, I compute the point cloud coverage c_p of each SE-Atlas reconstruction using the same equation as Eq. (3.9):

$$c_p = \frac{N_{\mathrm{O}}}{N_{\mathrm{GT}}}. \tag{4.14}$$

N_{GT} is the total number of points in the ground truth point cloud while N_{O} is the number of observed points in the same ground truth point cloud. Each point

in the ground truth map is considered observed if there exists a point in the reconstructed point cloud within a search radius $\lambda_{\text{neighbour}}$. In this case, $\lambda_{\text{neighbour}}$ is defined as $\sqrt{3} \times r_{\text{sample}} \approx 8.6\,\text{cm}$, the diagonal distance of a $5\,\text{cm}$ voxel. I again used CloudCompare to compute the point-to-point distance, but this time using the reconstruction as reference. Coverage is defined as the percentage of points in the ground truth point cloud that have a distance error smaller than $8.6\,\text{cm}$.

The Baseline reconstruction and the point cloud map with only Submap Overlap Estimate strategy have low coverage percentage because of their low accuracy. When integrated scans are relatively accurate, Cloud Overlap Estimate strategy leads to the lowest coverage percentage of $46.7\,\%$; with both Submap and Cloud Overlap Estimate strategies, the coverage percentage is increased by $\sim 20\,\%$. Relative Uncertainty decreases the map coverage due to submap fusion rejection, but with motion aware LiDAR integration improving the alignment between the SE-Atlas reconstruction and the ground truth, the eventual map coverage reaches the highest percentage.

4.12 Conclusion

To summarise, we designed and developed an elastic and efficient reconstruction pipeline, SE-Atlas, for long-term exploration tasks in large-scale outdoor environments and complex multi-storey indoor scenarios. SE-Atlas addresses the challenges revealed in the real-world experiments in Chapter 3, namely motion distortion in LiDAR scans, loop closure correction towards volumetric map, and the efficiency and scalability of large-scale high-resolution reconstructions [10].

I expanded a state-of-the-art reconstruction framework for RGB-D cameras, *supereight*, to incorporate long-range LiDAR inputs by implementing new projection and noise models based on the characteristics of LiDAR. SE-Atlas therefore leverages the multi-resolution feature of *supereight* to improve the efficiency of LiDAR scan integration and map memory consumption. I further incorporated motion-aware LiDAR integration to ensure undistorted LiDAR scans are correctly integrated into the volumetric reconstruction.

To realise elasticity upon loop closure correction, SE-Atlas exploits submaps that can move around to maintain global consistency, inspired by the Atlas framework by Bosse et al. [114]. Submap creation is guided by SLAM pose graph clustering based on a fixed trajectory distance. Additional submap spawning decisions are made based on cloud overlap evaluation, to create more elastic connections in the reconstruction when the robot explores through narrow passages, because LiDAR odometry usually has low localisation confidence in such a scenario.

SE-Atlas also fuses submaps together based on loop closures. Submaps that are mapping the same physical space are also fused together based on their spatial overlap. This is to improve the memory scalability of the overall reconstruction — SE-Atlas memory consumption scales with the size of scanned space as opposed to the length of exploration time.

SE-Atlas has been tested against experiments in simulation and in real world to verify these functionalities. The elasticity provided by the *Atlas* of submaps allows the correction of the map upon SLAM loop closure, ensuring global consistency. In terms of scan integration speed and map memory efficiency, SE-Atlas outperforms two state-of-the-art reconstruction techniques, OctoMap [137] and Voxgraph [5], in large-scale outdoor experiments such as NCD [14] and ARCHE [5]. The spatial analysis module further increases the scalability of reconstruction, as experiment results demonstrate that the memory consumption of the map no longer grows indefinitely when the whole area of interest has been scanned. In addition, the reconstruction accuracy is improved via cloud overlap analysis and relative uncertainty estimation.

5

Semantic Analysis in Large-Scale Multi-Sensor Reconstruction

This chapter includes elements of the following publication:

[13] Y. Wang, M. Ramezani, M. Mattamala, S. T. Digumarti, and M. Fallon. "Strategies for Large Scale Elastic and Semantic LiDAR Reconstruction". In: *J. of Robotics and Autonomous Systems (RAS)* [2022]

Acknowledgements
I would like to acknowledge the contribution of Dr Sundara Tejaswi Digumarti in maintaining, assessing and adapting the semantic segmentation algorithm originally proposed by Gan et al. [9].

This chapter describes how semantic information was incorporated into the SE-Atlas reconstruction pipeline that has been presented in Chapter 4. An externally developed semantic segmentation module was adapted to introduce semantic classifications into the elastic large-scale LiDAR map explained in the previous chapter. In order to maintain consistency of semantic labels between overlapping submaps and camera frustums, SE-Atlas incorporates a probabilistic formulation inspired by the work of McCormac et al. [179] to fuse multiple semantic distributions together.

The motivation behind integrating semantics into SE-Atlas originates from the limitations of the state-of-the-art semantic reconstruction technique by

Gan et al. [9]. I also developed a detector which determines when the robot transitions between indoor and outdoor environments based on semantic segmentation results [13]. These features extended our European Conference on Mobile Robots (ECMR) 2021 paper [12] and appeared in the Journal of Robotics and Autonomous Systems special issue [13].

The semantic segmentation algorithm used in this pipeline is an adapted version of the original work by Gan et al. [9]. This component is external to SE-Atlas and is briefly explained in Section 5.3. This chapter focuses more on the application and integration of semantic information in the reconstruction, which is my contribution to the proposed system.

5.1 Introduction

Extending upon the online room segmentation methods presented in Chapter 4, we further introduce strategies to improve SE-Atlas reconstruction based on semantic understanding of the observed spaces, and integrate per-voxel semantic labels into the volumetric map.

This project adapts the semantic segmentation module used in the semantic reconstruction framework of Gan et al. [9]. One limitation in the work of Gan et al. is in its reconstruction technique. A rigid global OctoMap [137] was maintained in the system proposed by Gan et al. and the main perception sensor was a short-range RGB-D camera. This resulted in a map that is not suitable for large-scale long-term exploration tasks that require SLAM loop closure to maintain global map consistency.

My proposed system instead integrates per-voxel semantic information into volumetric occupancy submaps (Chapter 4) that are created by long-range 360° LiDAR measurements. I further leverage all cameras on a Boston Dynamics Spot to achieve full coverage of semantics around the robot. The semantic segmentation algorithm of Gan et al. [9] is applied to all cameras on board Spot to cover as much FoV as possible. This is key for improving the navigational capabilities of mobile robots because the awareness of surrounding environments

is significantly expanded compared to a single-camera pipeline. By integrating this multi-sensor framework with SE-Atlas, a submap-focused reconstruction system [11], elasticity and improved scalability are introduced into the semantically annotated map.

To ensure per-voxel semantic consistency in the reconstruction, I also implement a probabilistic formulation inspired by the work of McCormac et al. [179], to merge the semantic labels of the voxels. This addresses potentially conflicting segmentation results across overlapping camera views as well as during fusion among multiple submaps.

Incorporating semantic information into SE-Atlas further enables segmenting areas that are significantly different from one another, such as indoor and outdoor spaces. I use the semantically labelled camera images to detect whether the robot is in an indoor or outdoor space. The motivation behind detecting the transition between indoor and outdoor environments originates from the differences in reconstruction parameters between indoor and outdoor experiments. For instance, in indoor scenarios reconstruction pipelines can usually achieve high integration speed and memory scalability even with very fine resolution because of the size of environment, but such a setup would often be infeasible for large-scale outdoor tasks. Instead of requiring hand-tuning for each experiment, the proposed system can leverage semantic analysis and make such adjustments to parameters on-the-fly as the robot traverses between indoor and outdoor environments.

The features and contributions of the proposed semantic-based module are the following:

- Expansion to a state-of-the-art dense semantic mapping module [9] to introduce elasticity into the reconstruction for large-scale environment.
- On-the-fly adjustment of reconstruction parameters when transiting between indoors and outdoors, based on semantic analysis.

- A probabilistic model to maintain consistency across semantic classes from multiple cameras and upon submap fusion.

5.2 Literature Review

In addition to spatial room segmentation explained in Chapter 4, richer semantic representations of a robot's environment can further enable better decision making and informed navigational planning. Methods used to extract semantic information from images can be broadly classified into two groups: (i) object detection, where objects of interest are identified in a scene and (ii) dense per-pixel label estimation, where class labels are estimated for every pixel of an image. In this section, literature related to the latter category is reviewed, with a focus on approaches using deep neural networks.

Early approaches used Fully Convolutional Network (FCN) with skip connections and a coarse-to-fine hierarchy [180] and dilated convolutions to aggregate multi-scale features [181]. Badrinarayanan et al. [182] introduced SegNet, which has an encoder-decoder architecture with non-linear upsampling in the decoder and max pooled indices in the encoder. Such a design helped improve computational and memory efficiency. A widely used related network structure is the UNet [183], which uses skip connections between the encoder and decoder layers to overcome the vanishing gradients problem.

Later works in semantic segmentation further improved the performance and learning capabilities of these networks, especially when performing semantic segmentation on videos instead of still images. Spatio-temporal coherence is not only a criterion to evaluate the segmentation results, but also an additional information to incorporate into video object segmentation tasks [184]. For instance, Wang et al. [185] proposed a technique based on geodesic distance to avoid over-fitting and learn a more reliable and temporally consistent saliency of superpixels in video object segmentation. Using the learnt saliency as a prior, their formulation of pixel-wise semantic labelling is an energy minimisation problem on a function that considers dynamic location

models, global foreground and background models, and label smoothness potentials. The work of Pathak et al. [186] took inspiration from the key role that motion plays in human vision, and leveraged low-level grouping cues based on motion estimation to improve the accuracy of segmenting objects from frames in a video. They first applied motion-based segmentation technique to videos to extract segments. These segments were then used to train a CNN for video object segmentation. Recent work has also looked at novel network architectures [187, 188] to improve efficiency and robustness. A detailed overview of the state-of-the-art of semantic segmentation is presented by Garg et al. [189].

Given per-pixel semantic labels, a subsequent challenge is fusing them into a consistent map over time. This subfield of *semantic mapping* has been explored in [179, 190]. More recently, Gan et al. [9] presented a probabilistic framework to fuse multiple semantic labels to generate a continuous 3D semantic occupancy map. While the results presented in these works are convincing, they are limited to a single camera mounted on a mobile robot and did not consider scalability or issues of revisiting already explored space.

There are also 3D semantic segmentation techniques designed for dense point clouds, such as PointNet [191] and Kernel Point Convolution (KPConv) [192]. Qi et al. [191] avoided voxelising 3D point clouds to regularise the format, and designed a novel neural network, PointNet, that takes irregular point clouds as inputs, respecting the permutation invariance of points. Each point only holds its 3D coordinates and is processed identically and independently. The key technique is a single symmetrical max pooling operation. The system learns to select the most informative or interesting collection of points and aggregate these learnt values into a global descriptor. PointNet can provide a unified architecture for object classification, segmentation and scene parsing based on point clouds. Similarly, Hugues et al. [192] presented KPConv that operates directly on point clouds without intermediate representations such as voxel maps or meshes. The convolution

Strategy	Network Initialisation	Mean IoU
Original [9]	-	0.4
Greyscale	From scratch	0.35
3-channel greyscale	From scratch	0.31
3-channel greyscale	ImageNet [194]	**0.43**

Table 5.1: Different strategies of re-training the semantic segmentation network and their performance compared with the original method by Gan et al. [9].

weights of KPConv are in Euclidean space in the form of kernel points. The locations of these kernel points are continuous because there are no fixed grid convolutions, and KPConv can even be extended to deformable convolutions that learn to adapt kernel points to local geometry.

In this project, we use 2D semantic segmentation applied to camera inputs instead of 3D point cloud segmentation; however performing object segmentation on noisy non-uniform point clouds and creating semantic reconstructions directly is an interesting direction for future work.

5.3 Semantic Segmentation and Transition Detection

The neural network in the external semantic segmentation module follows the architecture presented in [9], but adapts the UNet with a MobileNet backbone [193], pre-trained on the ImageNet [194] dataset and fine tuned on the Extended NCLT dataset [195]. Representative results of the semantic segmentation network are presented in Fig. 5.1. The network was trained only on images of outdoor environments but estimates meaningful labels for indoor environments - walls are classified as buildings and floor as either sidewalk or road.

The network of Gan et al. [9] was trained on colour images, but the cameras available on Boston Dynamics Spot produce greyscale images. As a result, directly deploying the network on these images yielded poor results. Several strategies have been employed to re-train the network and evaluated their performance, the result of which is presented in Table 5.1. The first strategy

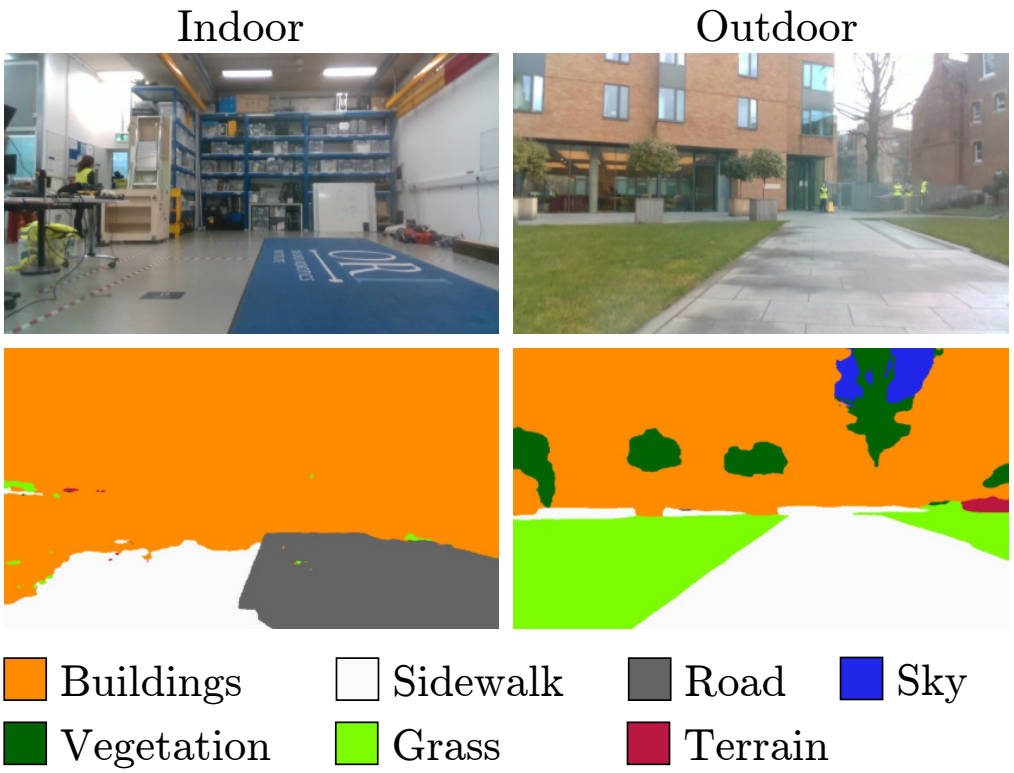


Figure 5.1: Representative images of the environment in which the robot was operated (top row) and the corresponding results of the semantic segmentation network (bottom row).

was to re-train the network from scratch on greyscale version of the images of the UMich dataset [195]. The second was to create 3-channel images like RGB cameras, and populating all channels with the same greyscale image. This strategy again trains the network from scratch. These two approaches did not yield good results, likely because pre-trained ImageNet [194] weights were not leveraged. The third strategy was to create the same 3-channel image as in the second strategy, but use ImageNet pre-training to warm start the training of the network. This improved the performance over the original work of Gan et al. [9].

In SE-Atlas, I use the semantically-labelled images to detect transitions of the mobile robot between indoor and outdoor environments. A high-level description of the algorithm is presented in Alg. 3. I first categorise all semantic labels into whether they are typically and distinctly *outdoor* or not. Class labels such as vegetation, grass, water, terrain etc. belong to the former category. I analyse the distribution of these predicted outdoor labels in the images. For each

Algorithm 3: Indoor/outdoor State Detection.

input:
The semantic information in the latest image $\mathbf{I}_{\text{new}}$,
The latest LiDAR point cloud $\mathbf{C}_{\text{new}}$,
The number of consecutive frames that have been classified as outdoor N_{outdoor},
The number of consecutive frames that have been classified as indoor N_{indoor},
Previous indoor/outdoor state of the robot *is_outdoor*,
output:
Indoor/outdoor state of the robot *is_outdoor*,
parameters:
The percentage threshold of pixels with outdoor labels $\lambda_{\text{semantic}}$,
The distance threshold for LiDAR measurements to be large-scale $d_{\text{large-scale}}$,
The percentage threshold of LiDAR measurements being large-scale λ_{range},
The threshold for the number of consecutive frames λ_{frames}
begin
  Find $R_{\text{outdoor-semantics}}$, the percentage of pixels in $\mathbf{I}_{\text{new}}$ with outdoor labels
  $is\_outdoor\_semantics = \{R_{\text{outdoor-semantics}} > \lambda_{\text{semantic}}\}$
  Find $R_{\text{large-scale}}$, the percentage of range measurements $d_{\text{lidar}} > d_{\text{large-scale}}$ based on $\mathbf{C}_{\text{new}}$
  $is\_outdoor\_range = \{R_{\text{large-scale}} > \lambda_{\text{range}}\}$
  if *is_outdoor_semantics* **AND** *is_outdoor_range* **then**
    $N_{\text{outdoor}}++$
    $N_{\text{indoor}} = 0$
  else
    $N_{\text{outdoor}} = 0$
    $N_{\text{indoor}}++$
  end if
  if *is_outdoor* = FALSE **AND** $N_{outdoor} > \lambda_{frame}$ **then**
    is_outdoor = TRUE
  else if *is_outdoor* = TRUE **AND** $N_{indoor} > \lambda_{frame}$ **then**
    is_outdoor = FALSE
  else
    Keep *is_outdoor* unchanged
  end if
  return *is_outdoor*
end

image, I compute the percentage of pixels with outdoor labels $R_{\text{outdoor-semantics}}$ amongst the whole image to detect whether the robot platform is indoor or outdoor. The proposed system further considers LiDAR range measurements when making this decision. If the percentage of outdoor labels is consistently larger than an empirically determined threshold ($\lambda_{\text{semantic}} = 20\,\%$) and there are long range LiDAR returns ($d_{\text{lidar}} > d_{\text{large-scale}}, d_{\text{large-scale}} = 30\,\text{m}$) more than a certain

percentage ($\lambda_{\text{range}} = 5\,\%$) for a set of successive frames ($\lambda_{\text{frames}} = 5$), the robot is considered to be outdoors. Leveraging LiDAR data makes the system robust to occasional incorrect predictions by the segmentation network because the semantic segmentation algorithm has only been trained using outdoor scenes but no indoor scenes. An ablation study on how semantics and LiDAR range measurements individually affect the detection of indoor/outdoor state is further presented in Section 5.7.

5.4 360° Horizontal Coverage of Semantic Annotation Using a Multi-Camera Setup

The large-scale LiDAR reconstruction of SE-Atlas, as explained in Chapter 4, leverages the 360° horizontal FoV of our LiDAR sensors, and is capable of creating a relatively complete map of an environment just from one single LiDAR scan. Compared to LiDARs, RGB-D cameras have a significantly smaller horizontal FoV. I therefore incorporate all 6 cameras on the robot platform in presented experiments (Boston Dynamics Spot) in addition to the single front-facing camera on the Frontier multi-sensor rig (Section 2.7.2), covering the full 360° horizontally. Fig. 5.2 provides a visualisation of multi-camera setup on the robot and the frustums of all cameras. Dense per-pixel semantic labels are predicted for each camera image using a deep neural network explained in Section 5.3.

When the robot is exploring outdoor environments, the proposed system determines which submap each camera image belongs to based on image timestamp, and integrates semantic information from each frame into its corresponding submap. To integrate semantic labels into the dense 3D reconstruction created from LiDAR, the proposed system iterates through voxels in each submap, and by using the tracked camera pose of j^{th} view ${}^{\mathcal{M}}\mathbf{T}_{\mathbf{x}_j}$, transforms the coordinate of a voxel ${}^{\mathcal{M}}\mathbf{p} \in \mathbb{R}^3$ into the camera's frame to compute a pixel coordinate $\rho_{j,\mathbf{p}} = \pi\left({}^{\mathcal{M}}\mathbf{T}_{\mathbf{x}_j}^{-1}{}^{\mathcal{M}}\mathbf{p}\right)$, where π represents the camera's projection operation using

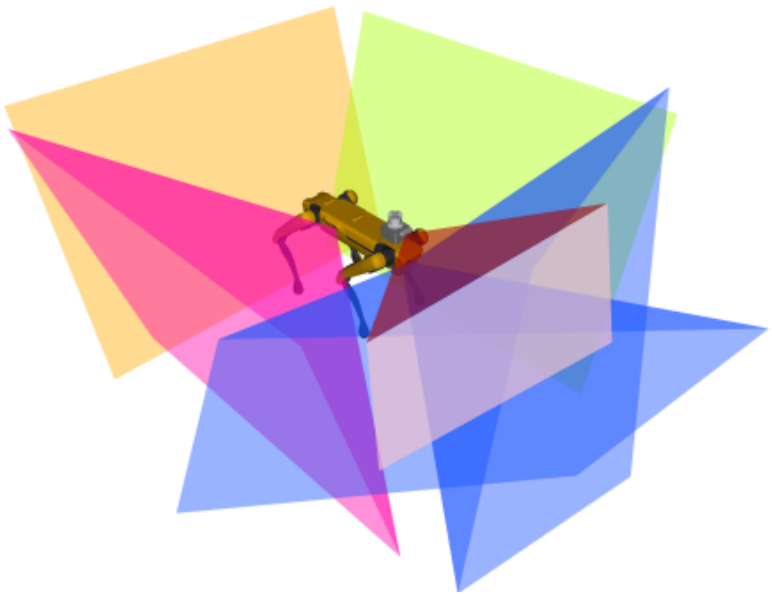


Figure 5.2: The frustums of all the cameras used in our semantic experiments, with one front-facing camera belonging to the Frontier multi-sensor rig and 6 cameras on-board the Boston Dynamics Spot (Section 2.7.2). In the front of the robot, there are two forward-facing cameras inclined towards to the ground, and one camera on Frontier (Section 2.7.2). The left, right and back side of the robot each has one camera as well.

its intrinsic matrix. The pixel coordinate $\rho_{j,\mathbf{p}}$ will then result in a categorical distribution representing the probability of this pixel belonging to a semantic class. This distribution is then stored in the voxel if this voxel has not yet been labelled by any semantic classes, or used to update the existing distribution in the voxel.

Submaps that are created when the robot is indoors have their voxels all given the indoor class label.

5.5 Probabilistic Fusion of Semantic Labels

The formulation to update per-voxel semantic probability distributions is as follows. Let $\mathbf{I}_j$ denote all the semantic information in the image of j^{th} view, each pixel ρ contains a probability distribution across all M class labels $P(Z_\rho = l_m|\mathbf{I}_j)$. I use the recursive Bayesian model presented in [179] to update the probabilistic distribution within a voxel at coordinate $\mathbf{p}$:

$$P(l_m|\mathbf{I}_{0,\dots,j}) = \frac{1}{\alpha} P(l_m|\mathbf{I}_{0,\dots,j-1}) P(Z_{\rho_{j,\mathbf{p}}} = l_m|\mathbf{I}_j), \tag{5.1}$$

where α is a normalising factor.

For each new semantically segmented image input into the proposed system, each voxel will receive at most one update to the probability distribution stored in it, and multiple voxels can be labelled with the same image pixel. However, Fig. 5.2 demonstrates that the cameras used in semantic segmentation have overlapping frustums. As the robot continues its mapping operation, images streaming from even a single camera also overlap each other. Eq. (5.1) provides a mathematically sound model to continuously update the probability distribution in each voxel, and to address any conflicting labelling from different observations. In addition, upon loop closure and submap fusion, different distributions in overlapping voxels from multiple submaps are resolved using the same model. The computational complexity of this operation is $O\left(M \cdot N_{\text{view}} \cdot N_{\text{occupied}}\right)$ for a submap associated with N_{view} camera views that contains N_{occupied} occupied voxels.

5.6 Experiments

To demonstrate semantically annotated mapping, an experiment was conducted with the robot traversing from indoors to an outdoor environment through multiple storeys and rooms. The robotic platform used in this experiment is a Boston Dynamics Spot carrying the Frontier multi-sensor rig (Fig. 4.13). The LiDAR sensor on the Frontier multi-sensor rig is an Ouster OS0-64 LiDAR, with 90° vertical FoV and 360° horizontal FoV. Its maximum sensing range is $50\,\text{m}$.

Due to the limitations in the original work of Gan et al. [9], i.e. rigidity in the reconstruction and small scale of the map, this experiment focused on demonstrating the contributions of incorporating semantics into SE-Atlas, an efficient and elastic reconstruction pipeline. Therefore, the goals of this experiment were two-fold:

- semantically labelling the large-scale high-resolution LiDAR reconstruction of SE-Atlas that corrects itself upon loop closure,

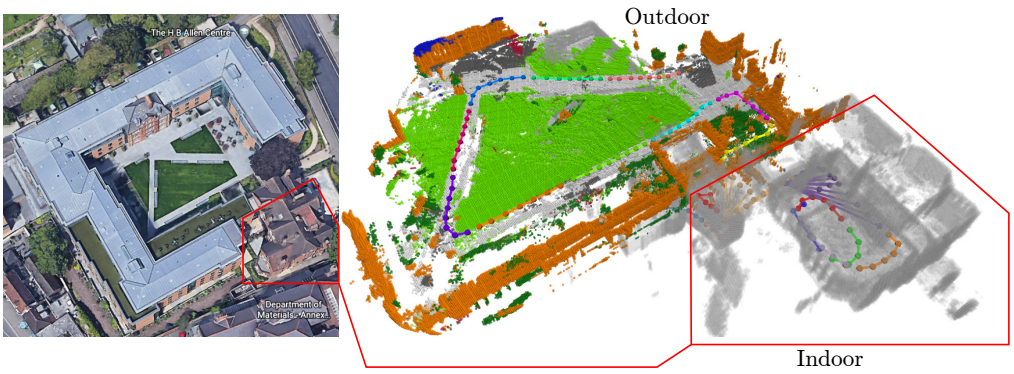


Figure 5.3: Exp 5.6 – Environment Classification: Parts of the constructed map that are classified as indoors (bottom right) are coloured in grey and parts of the map that are outdoors are coloured by their semantic labels. Lengths of the submaps (number of nodes) are also different between the two regions.

- using the predicted semantic labels to determine the indoor/outdoor state and adjusting reconstruction parameters on-the-fly.

Semantic classes such as vegetation, grass, water and terrain are characteristic of outdoor environments. Using the distribution of predicted labels in each image, I computed the percentage of labels belonging to such *outdoor* classes. This is further combined with LiDAR range measurements to determine if the robot was indoors or outdoors at a given time. Alg. 3 provides a more detailed explanation on the indoor/outdoor detection process. Relying on LiDAR data makes the system robust to occasional incorrect predictions by the segmentation network. While this could have been done in a more involved way, for instance by retraining the semantic segmentation network using both indoor and outdoor scenes, I found that this simple approach worked well in our experiments. An ablation study on the effect of semantics and LiDAR range is presented in Section 5.7.

Results of this experiment are shown in Fig. 5.3, where parts of the map that are determined to be indoors are coloured in grey and parts of the map that are outdoors are coloured using the semantic labels.

Fig. 5.4 demonstrates the elasticity granted by submaps. Fig. 5.4 (a) shows the semantically annotated LiDAR reconstruction before SLAM loop closure. The enlarged view highlights a misalignment of the facade of a building, labelled

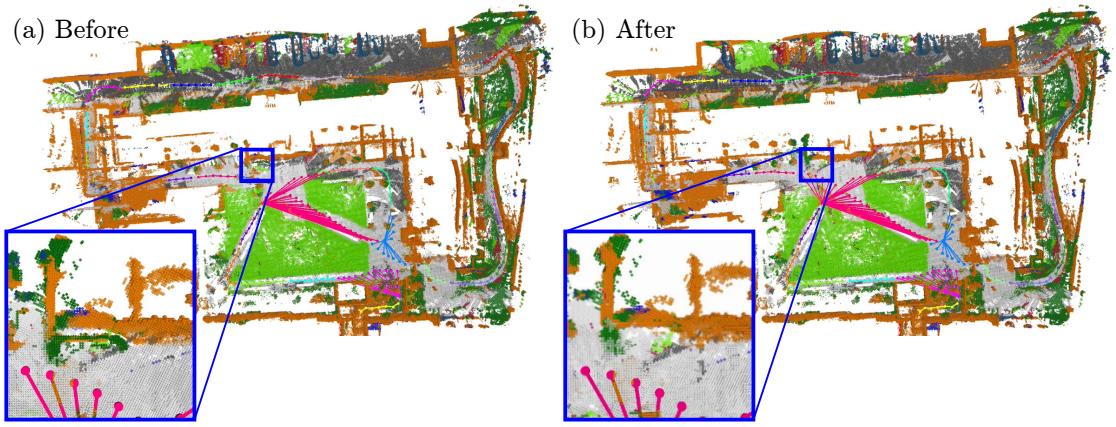


Figure 5.4: Exp 5.6 – Drift correction: **(a)** – the semantically annotated LiDAR reconstruction with building misalignment (orange) before loop closure; **(b)** – the elastic reconstruction after loop closure correcting the SLAM pose graph.

as orange voxels. In Fig. 5.4 (b), SLAM loop closure corrected the pose graph, which in turn realigned the walls in the zoomed-in view.

In addition, SE-Atlas used the semantic understanding of the environment to alter the odometry trajectory distance threshold λ_{odom} used in pose graph clustering to spawn submaps — $5\,\text{m}$ for indoors and $8\,\text{m}$ for outdoors. This helps reduce the memory required in larger environments.

Fig. 5.5 presents the quantitative evaluation of reconstruction accuracy using the same method as the one described in Section 4.10.5. The average point-to-point distance error is $5.4\,\text{cm}$, with the most accurate points around the ground and the lower floors of buildings. For the robot to scan at these upper floors of the building, it must do so from further away, which causes there to be slightly higher error — around $10\,\text{cm}$. The highest errors ($>50\,\text{cm}$, coloured yellow) are the dynamic objects in the scene, e.g. the robot operator.

5.7 Ablation Study on Indoor-Outdoor Detection

This section demonstrates the effect of semantics and LiDAR range measurements separately on the detection of indoor/outdoor state. I isolated the decisions made by each component, and based the transition detection module

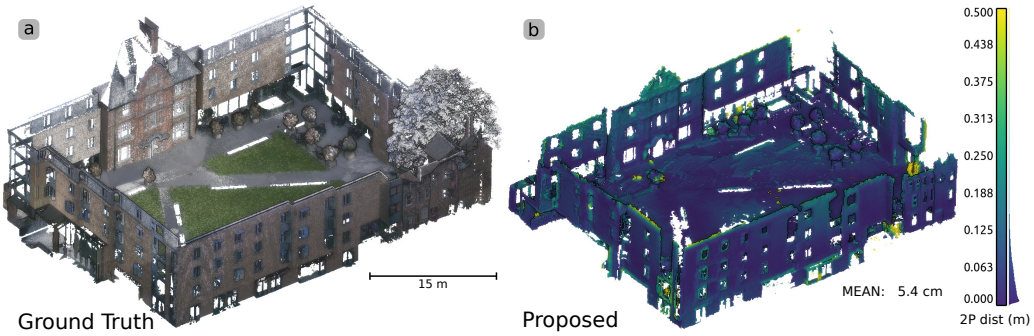


Figure 5.5: Exp 5.6 – Reconstruction of the proposed system **(b)** compared with the ground truth **(a)**.

(Alg. 3) on either criterion individually. The qualitative experiment results are presented in Fig. 5.6.

The ablation study results suggest that neither the current employed semantic segmentation algorithm [9] nor LiDAR range measurements can provide the desired solution for detecting the indoor/outdoor state. On one hand, LiDAR range measurements only reflect whether the robot is in a small-scale environment without any semantic understanding; therefore when the robot is traversing through narrow passages, basing indoor/outdoor state detection on LiDAR range measurements will most likely categorise the environment as indoor. Fig. 5.6 (b) highlights this behaviour using blue rectangles. On the other hand, solely relying the detection of outdoor environments on the current semantic segmentation results is also unreliable, as demonstrated in Fig. 5.6 (a). This is because the current semantic segmentation module is not trained to correctly identify indoor objects. If a significant proportion of the camera view is constantly labelled as outdoor for a few consecutive frames (as explained in Section 5.3), SE-Atlas will classify the environment as outdoor. Therefore in the proposed system, I combined both criteria to achieve a more reliable indoor/outdoor state detector.

With a tailored semantic segmentation algorithm, it is possible to rely purely on semantic information to determine whether the robot is indoor or outdoor. While LiDAR range measurements are relevant to submap spawning parameters such as the odometry trajectory distance threshold λ_{odom} used in pose graph

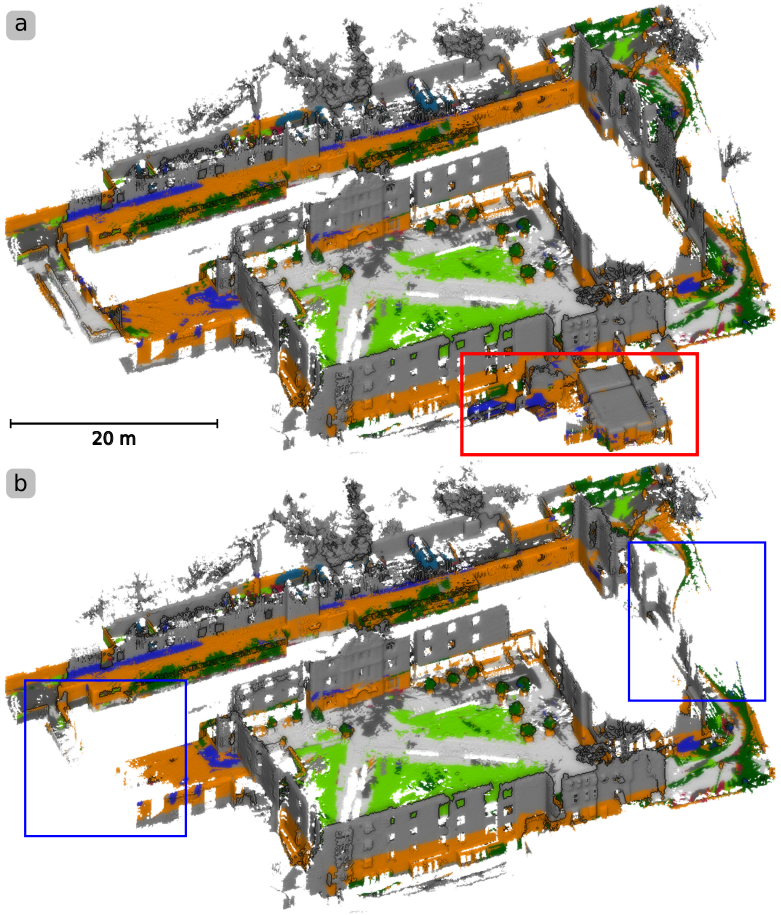


Figure 5.6: Exp 5.6 – Comparison between the reconstruction classified as outdoor using only semantics **(a)** and that using only LiDAR range measurements **(b)**. Only using semantics leads to indoor environments being miss-classified as outdoor (highlighted using red rectangle), while only using LiDAR range miss-classifies narrow outdoor paths as indoor (highlighted using blue rectangles).

clustering, they are not necessarily representing nature of the surroundings. In an improved pipeline, these two criteria should be separated from each other.

5.8 Conclusion

This chapter explains the features in SE-Atlas that leverage semantic information, namely the detection of transition between indoor and outdoor environments and the integration of semantic classes into elastic large-scale LiDAR reconstruction.

The semantic segmentation system is external to SE-Atlas and is inspired by the work of Gan et al. [9]. The original semantic reconstruction framework of

Gan et al. cannot correctly process SLAM loop closures, or efficiently map large-scale environments. These are the limitations of their reconstruction technique. I therefore integrate semantic information from the work of Gan et al. [9] into the elastic large-scale LiDAR reconstruction to address these limitations.

I then use the recursive Bayesian model presented by McCormac et al. [179] to ensure probabilistic consistency across semantic classes when multiple sources of semantic information are fused together, such as from multiple cameras or multiple submaps.

I further evaluate the ratio of outdoor labels in camera views based on semantic segmentation results to determine the type of environment that the robot is in, and adjust reconstruction parameters accordingly on the fly.

These SE-Atlas features have been demonstrated in real-world experiments. Semantic information is integrated into the large-scale LiDAR reconstruction of SE-Atlas to create an elastic dense semantic map. The recursive Bayesian model fuses distributions of semantics across multiple sources, such as overlapping submaps and cameras, to ensure consistency of semantic classes in the overall reconstruction. The odometry trajectory threshold for submap spawning is adjusted online as the robot transits between indoor and outdoor environments.

One of the limitations of the proposed SE-Atlas system is that the indoor/outdoor detection process relies on LiDAR range measurements to be robust against incorrect semantic segmentation results. This can be addressed by retraining the semantic segmentation network with both indoor and outdoor scenes and improving its reliability. Using a tailored semantic segmentation algorithm will allow these two criteria to be separated from each other, and LiDAR range measurements can be treated as a light-weight representation for the scale of the environment instead.

6 Conclusions

To summarise, I designed and developed *supereight* Atlas (SE-Atlas), a multi-resolution elastic reconstruction pipeline for long-range LiDAR scans in large-scale environments and long-term explorations. The motivations behind this DPhil project are the challenges posed by international competitions such as the DARPA SubT Challenge [19], such as navigation through complex and unstructured scenarios as well as efficient and accurate large-scale reconstruction.

I first built an active mapping framework and assessed it within a realistic unstructured industrial setting. real-world experiments revealed several limitations of conventional reconstruction methods such as OctoMap [137]. For instance, long-range LiDAR scans require a highly efficient pipeline to be integrated at high resolution, and large-scale environments and long-term exploration tasks demand improved memory scalability. In addition, a global volumetric or surface mesh reconstruction using conventional techniques often cannot account for SLAM loop closures and corrections. Inaccurate reconstruction due to motion distortion was also observed in the active mapping results of our initial system.

SE-Atlas is therefore proposed after a collaboration between the Dynamic Robot Systems (DRS) at University of Oxford and Smart Robotics Lab (SRL) at Imperial College London (ICL). The proposed system is an elastic and efficient

LiDAR reconstruction pipeline for large-scale environments and long-term exploration tasks. At its core, SE-Atlas uses *supereight* [50], a multi-resolution reconstruction pipeline for RGB-D cameras. I adapted *supereight* to integrate LiDAR measurements by introducing new projection and noise models more suitable for LiDAR. SE-Atlas outperforms state-of-the-art reconstruction techniques, such as Voxgraph [5], in scan integration speed and memory efficiency during long-range high-resolution mapping operations using real-world outdoor datasets [5, 14].

The proposed system is connected with external odometry and SLAM modules. By clustering the nodes of a SLAM pose graph based on travel distance, SE-Atlas integrates LiDAR scans into local submaps instead of maintaining a global reconstruction. Each submap is associated with a node in the SLAM pose graph. Upon loop closure, the SLAM system corrects the pose graph, and the proposed system corrects the poses of submaps accordingly as well. Furthermore, submaps around the head and tail of the closed loop is further fused together to reduce the spatial overlap among submaps and improve system scalability.

I then introduced additional submap spawning and fusion criterion, which are referred to as "Principled Clustering" strategies, based on spatial overlap analysis. SE-Atlas creates a new submap when the overlap between point clouds is too low, because this means low confidence in ICP localisation results. This also functions as an online room segmentation module, creating individual submaps when the robot traverses through a narrow doorway. The spatial overlap between submaps further improves the memory scalability of the proposed system, as SE-Atlas fuses submaps that share significant spatial overlap between each other after loop closure corrections. We also derived a formulation to measure relative uncertainties between poses in a SLAM pose graph, based on the work of Mangelson et al. [7] and using the notation of GTSAM [8]. The estimated relative uncertainty is used to reject unreliable submap fusions till the confidence is high enough. When assessed using Newer College Dataset (NCD) experiments, these principled clustering strategies were

demonstrated to improve the accuracy and scalability of the overall reconstruction.

I further applied an external semantic segmentation algorithm [9] to multiple RGB-D cameras on the robot to detect the transition between indoor and outdoor environments. I integrated semantic information into voxels in the volumetric LiDAR map of SE-Atlas to achieve large-scale high-resolution elastic reconstructions, addressing the shortcomings of the original semantic reconstruction system proposed by Gan et al. [9]. To ensure consistency between semantic classes in each voxel, I relied on a recursive Bayesian model inspired by the work of McCormac et al. [179]. This formulation addresses potential conflicts among semantic classes within each voxel among overlapping camera frustums and upon submap fusions.

The efficiency of scan integration in large-scale environments, the scalability of map memory consumption in long-term explorations, and the accuracy of reconstruction are assessed using both simulated and real-world datasets. We also demonstrated the results of annotating the LiDAR reconstruction with semantic information in a real-world experiment. Last but not least, I tested the capability of using SE-Atlas occupancy map for path planning purposes with a sequence collected in an underground mineshaft. The high-resolution reconstruction allowed the path planner to navigate through narrow doorways and corridors that pose challenges in such scenarios similar to the DARPA SubT Challenge [19].

6.1 Future Works

This section discusses several directions of further explorations and improvements for SE-Atlas.

6.1.1 Active Mapping and Planning

A continuation of this DPhil work is building an active mapping and planning application around the submap-based multi-resolution reconstructions which

can take on a similar iterative and incremental structure as the system presented in Chapter 3.

This active mapping and planning system can leverage the MultiresOFusion pipeline of SE-Atlas and solve the NBV problem based on Information Gain and voxel entropy. The adaptative voxel resolution in a volumetric map created by SE-Atlas can be further exploited for efficiently checking the validity of robot poses. Instead of densely modelling the robot with the highest voxel resolution, the multi-resolution map allows for downsampling voxels inside the robot, decreasing the number of per-voxel occupancy queries when checking the safety of any robot poses. We can therefore model the robot using a more accurate representation than a sphere. For example, the Boston Dynamics Spot has a long and slim base, and modelling it accordingly instead of assuming a spherical shape can better leverage the robot's shape and heading when planning paths through narrow doorways in complex environments.

On the other hand, having a collection of submaps in place of a global reconstruction can present some complications for any occupancy probability lookup, including both Information Gain computation and path planning. Multiple queries across overlapping submaps will be required for checking the occupancy at only one position. A similar problem is discussed by Ho et al. [138] and Sodhi et al. [139]. Instead of merging submaps together for query like the work of Ho et al. as such a process will be extremely time consuming in large-scale environments, we can use bounding boxes of each submap to choose which submap to look up voxel occupancy in and effectively decrease the number of queries needed. In addition, Submap Overlap Estimation and fusion in SE-Atlas usually lead to one submap per isolated space, which should also make the occupancy query more feasible.

6.1.2 Improvement in Real-time Feasibility

To make SE-Atlas more feasible for any real-time robotic applications, we can further improve the speed of SE-Atlas, such as via multi-threading different com-

ponents. For instance, the process of Submap Overlap Estimation and fusion is relatively time-consuming compared to scan integration, especially when the detected SLAM loop closure is large. In the current SE-Atlas, this is alleviated via multi-threaded scan integration so submap fusion does not lead to skipping new scans. Launching the fusion process in a separate thread can further mitigate its impact.

The motion-aware LiDAR integration is a module that can be easily parallelised, as each search consists only of arithmetic processes and does not alter any already stored data. This can further improve the speed of scan integration for any outdoor large-scale explorations.

6.1.3 Improvement in Semantic Segmentation

As part of the future work, we plan to improve the accuracy of the semantic annotation in the LiDAR map, especially around the boundary of objects. Calibration and synchronisation between LiDAR and multiple cameras can benefit the direct projection of measurements between them. Frame-to-frame alignment across camera measurements can also reduce the impact of robot's motion to semantic segmentation.

As discusses in Section 5.2, 3D semantic segmentation based on point clouds [191, 192] is an alternative to the vision-based algorithm [9] employed in the current pipeline. Using point cloud-based segmentation algorithms can directly integrate semantic labels into the volumetric map, avoiding some aforementioned problems.

Furthermore, we would like to expand the capability of our semantic segmentation module. In particular, indoor semantic labelling can assist robot's exploration and planning, such as autonomously inferring floor plans from a multi-storey reconstruction and enabling high-level path planning between floors. We can further leverage semantic information to detect and isolate objects of interest from background and obstacles for active mapping and planning systems in complex environments.

References

[1] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter. "ANYmal in the Field : Solving Industrial Inspection of an Offshore HVDC Platform with a Quadrupedal Robot". In: *Field and Service Robotics*. Tokyo, Japan, 2019.

[2] *ANYbotics | Autonomous Legged Robots for Inspection and More*. en-US. https://www.anybotics.com/.

[3] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous Robots* 34.3 [2013], pp. 189–206.

[4] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. "Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2017, pp. 1366–1373.

[5] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto. "Voxgraph: Globally Consistent, Volumetric Mapping Using Signed Distance Function Submaps". In: *IEEE Robotics and Automation Letters* 5.1 [2020], pp. 227–234.

[6] L. Schmid, V. Reijgwart, L. Ott, J. Nieto, R. Siegwart, and C. Cadena. "A Unified Approach for Autonomous Volumetric Exploration of Large Scale Environments under Severe Odometry Drift". In: *IEEE Robotics and Automation Letters* 6.3 [July 2021], pp. 4504–4511.

[7] J. G. Mangelson, M. Ghaffari, R. Vasudevan, and R. M. Eustice. "Characterizing the uncertainty of jointly distributed poses in the Lie algebra". In: *IEEE Trans. Robotics* 36.5 [2020], pp. 1371–1388.

[8] F. Dellaert and C. Beall. "GTSAM 4.0". In: *URL: https://bitbucket. org/gtborg/gtsam* [2017].

[9] L. Gan, R. Zhang, J. W. Grizzle, R. M. Eustice, and M. Ghaffari. "Bayesian spatial kernel smoothing for scalable dense semantic mapping". In: *IEEE Robotics and Automation Letters* 5.2 [2020], pp. 790–797.

[10] Y. Wang, M. Ramezani, and M. Fallon. "Actively Mapping Industrial Structures with Information Gain-Based Planning on a Quadruped Robot". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 8609–8615.

[11] Y. Wang, N. Funk, M. Ramezani, S. Papatheodorou, M. Popovic, M. Camurri, S. Leutenegger, and M. Fallon. "Elastic and Efficient LiDAR Reconstruction for Large-Scale Exploration Tasks". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2021, pp. 5035–5041.

[12] Y. Wang, M. Ramezani, M. Mattamala, and M. Fallon. "Scalable and Elastic LiDAR Reconstruction in Complex Environments Through Spatial Analysis". In: *Proc. of the European Conference on Mobile Robotics (ECMR)*. Aug. 2021.

[13] Y. Wang, M. Ramezani, M. Mattamala, S. T. Digumarti, and M. Fallon. "Strategies for Large Scale Elastic and Semantic LiDAR Reconstruction". In: *J. of Robotics and Autonomous Systems (RAS)* [2022].

[14] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon. "The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2020, pp. 4353–4360.

[15] Y. Tao, M. Popović, Y. Wang, S. T. Digumarti, N. Chebrolu, and M. Fallon. "3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2022.

[16] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2012.

[17] J. Guivant and E. Nebot. "Optimization of the simultaneous localization and map-building algorithm for real-time implementation". In: *IEEE Trans. Robotics and Automation* 17.3 [2001], pp. 242–257.

[18] G. Tinchev, A. Penate-Sanchez, and M. Fallon. "Learning to See the Wood for the Trees: Deep Laser Localization in Urban and Natural Environments on a CPU". In: *IEEE Robotics and Automation Letters* 4.2 [2019], pp. 1327–1334.

[19] V. L. Orekhov and T. H. Chung. "The DARPA Subterranean Challenge: A Synopsis of the Circuits Stage". In: *Field Robotics* 2 [2022].

[20] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund, M. F. Ginting, K. Ebadi, M. Anderson, T. Pailevanian, E. Terry, M. Wolf, A. Tagliabue, T. S. Vaquero, M. Palieri,

S. Tepsuporn, Y. Chang, A. Kalantari, F. Chavez, B. Lopez, N. Funabiki, G. Miles, T. Touma, A. Buscicchio, J. Tordesillas, N. Alatur, J. Nash, W. Walsh, S. Jung, H. Lee, C. Kanellakis, J. Mayo, S. Harper, M. Kaufmann, A. Dixit, G. Correa, C. Lee, J. Gao, G. Merewether, J. Maldonado-Contreras, G. Salhotra, M. S. Da Silva, B. Ramtoula, Y. Kubo, S. Fakoorian, A. Hatteland, T. Kim, T. Bartlett, A. Stephens, L. Kim, C. Bergh, E. Heiden, T. Lew, A. Cauligi, T. Heywood, A. Kramer, H. A. Leopold, C. Choi, S. Daftry, O. Toupet, I. Wee, A. Thakur, M. Feras, G. Beltrame, G. Nikolakopoulos, D. Shim, L. Carlone, and J. Burdick. "NeBula: Quest for Robotic Autonomy in Challenging Environments; TEAM CoSTAR at the DARPA Subterranean Challenge". In: *arXiv preprint* [2021]. eprint: arXiv:2103.11470.

[21] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth, S. Zimmermann, H. Nguyen, M. Fehr, L. Solanka, R. Buchanan, M. Bjelonic, N. Khedekar, M. Valceschini, F. Jenelten, M. Dharmadhikari, T. Homberger, P. D. Petris, L. Wellhausen, M. Kulkarni, T. Miki, S. Hirsch, M. Montenegro, C. Papachristos, F. Tresoldi, J. Carius, G. Valsecchi, J. Lee, K. Meyer, X. Wu, J. I. Nieto, A. P. Smith, M. Hutter, R. Y. Siegwart, M. W. Mueller, M. F. Fallon, and K. Alexis. "CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge". In: *Field Robotics* 2 [2022].

[22] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. L. Surdo, T. Molnar, R. Steindl, M. Wildie, I. Sa, N. Kottege, K. Stepanas, E. Hernández, G. Catt, W. Docherty, B. Tidd, B. Tam, S. Murrell, M. S. Bessell, L. Hanson, L. Tychsen-Smith, H. Suzuki, L. Overs, F. Kendoul, G. Wagner, D. Palmer, P. Milani, M. J. O'Brien, S. Jiang, S. Chen, and R. C. Arkin. "Heterogeneous Ground and Air Platforms, Homogeneous Sensing: Team CSIRO Data61's Approach to the DARPA Subterranean Challenge". In: *Field Robotics* 2 [2022].

[23] S. Scherer, V. Agrawal, G. Best, C. Cao, K. Cujic, R. Darnley, R. DeBortoli, E. Dexheimer, B. Drozd, R. Garg, I. Higgins, J. Keller, D. Kohanbash, L. Nogueira, R. Pradhan, M. Tatum, V. K. Viswanathan, S. Willits, S. Zhao, H. Zhu, D. Abad, T. Angert, G. Armstrong, R. Boirum, A. Dongare, M. Dworman, S. Hu, J. Jaekel, R. Ji, A. Lai, Y. H. Lee, A. Luong, J. Mangelson, J. Maier, J. Picard, K. Pluckter, A. Saba, M. Saroya, E. Scheide, N. Shoemaker-Trejo, J. Spisak, J. Teza, F. Yang, A. Wilson, H. Zhang, H. Choset, M. Kaess, A. Rowe, S. Singh, J. Zhang, G. A. Hollinger, and M. Travers. "Resilient and Modular Subterranean Exploration with a Team of Roving and Flying Robots". In: *Field Robotics* 2 [2022].

[24] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza. "An Information Gain Formulation for Active Volumetric 3D Reconstruction". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2016, pp. 3477–3484.

[25] J. Delmerico and D. Scaramuzza. "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2018, pp. 2502–2509.

[26] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart. "Receding horizon path planning for 3D exploration and surface inspection". In: *Autonomous Robots* 42.2 [2018], pp. 291–306.

[27] J. Williams, S. Jiang, M. OBrien, G. Wagner, E. Hernandez, M. Cox, A. Pitt, R. Arkin, and N. Hudson. "Online 3D Frontier-Based UGV and UAV Exploration Using Direct Point Cloud Visibility". In: *Proc. of the IEEE Intl. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. 2020, pp. 263–270.

[28] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter. "Graph-based subterranean exploration path planning using aerial and legged robots". In: *J. of Field Robotics* 37.8 [2020], pp. 1363–1388.

[29] E. W. Dijkstra. "A Note on Two Problems in Connexion with Graphs". In: *Numer. Math.* 1.1 [1959], 269–271.

[30] S. M. LaValle. "Rapidly-Exploring Random Trees: A New Tool for Path Planning". In: *Technical Report* [1998]. eprint: arXiv:1011.1669v3.

[31] S. Karaman and E. Frazzoli. "Sampling-based Algorithms for Optimal Motion Planning". In: *Intl. J. of Robotics Research* 30.7 [2011], pp. 846–894.

[32] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. "Keyframe-based visual–inertial odometry using nonlinear optimization". In: *Intl. J. of Robotics Research* 34.3 [2015], pp. 314–334.

[33] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart. "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback". In: *Intl. J. of Robotics Research* 36.10 [2017], pp. 1053–1072.

[34] T. Qin, P. Li, and S. Shen. "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator". In: *IEEE Trans. Robotics* 34.4 [2018], pp. 1004–1020.

[35] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone. "Kimera: From SLAM to spatial perception with 3D dynamic scene graphs". In: *Intl. J. of Robotics Research* 40.12-14 [2021], pp. 1510–1546.

[36] E. Rosten and T. Drummond. "Machine Learning for High-Speed Corner Detection". In: *Proc. of the Eur. Conf. on Computer Vision (ECCV)*. Springer Berlin Heidelberg, 2006, pp. 430–443.

[37] S. Leutenegger, M. Chli, and R. Y. Siegwart. "BRISK: Binary Robust invariant scalable keypoints". In: *Proc. of the Intl. Conf. on Computer Vision (ICCV)*. 2011, pp. 2548–2555.

[38] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. "OpenVINS: A Research Platform for Visual-Inertial Estimation". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020, pp. 4666–4672.

[39] A. I. Mourikis and S. I. Roumeliotis. "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2007, pp. 3565–3572.

[40] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. "On-Manifold Preintegration for Real-Time Visual–Inertial Odometry". In: *IEEE Trans. Robotics* 33.1 [2017], 1–21.

[41] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. "G2o: A general framework for graph optimization". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2011, pp. 3607–3613.

[42] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. "iSAM2: Incremental smoothing and mapping using the Bayes tree". In: *Intl. J. of Robotics Research* [2012].

[43] M. Helmberger, K. Morin, B. Berner, N. Kumar, D. Wang, Y. Yue, G. Cioffi, and D. Scaramuzza. *The Hilti SLAM Challenge Dataset*. 2021. eprint: arXiv:2109.11316.

[44] F. Pomerleau, F. Colas, and R. Siegwart. "A Review of Point Cloud Registration Algorithms for Mobile Robotics". In: *Foundations and Trends in Robotics* 4 [May 2015], pp. 1–104.

[45] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-cid. "Tree-based search of the next best view/state for three-dimensional object reconstruction". In: *Intl. Journal of Advanced Robotic Systems* 15.1 [2018].

[46] H. Yervilla-Herrera, J. I. Vasquez-Gomez, R. Murrieta-Cid, I. Becerra, and L. E. Sucar. "Optimal motion planning and stopping test for 3-D object reconstruction". In: *Intelligent Service Robotics* [2018].

[47] S. Kriegel, C. Rink, T. Bodenmu, and M. Suppa. "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects". In: *J. of Real-Time Image Processing* 10 [4 2015], pp. 611–631.

[48] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. "KinectFusion: Real-time dense surface mapping and tracking". In: *IEEE Intl. Symp. on Mixed and Augmented Reality*. 2011, pp. 127–136.

[49] T Whelan, J McDonald, M Kaess, M Fallon, and H Johannsson. "Kintinuous: Spatially extended KinectFusion". In: *Workshop on RGB-D at Robotics: Science and Systems (RSS)*. 2012.

[50] E. Vespa, N. Funk, P. H. J. Kelly, and S. Leutenegger. "Adaptive-Resolution Octree-Based Volumetric SLAM". In: *Intl. Conf. on 3D Vision*. 2019, pp. 654–662.

[51] N. Funk, J. Tarrio, S. Papatheodorou, M. Popović, P. F. Alcantarilla, and S. Leutenegger. "Multi-Resolution 3D Mapping With Explicit Free Space Representation for Fast and Accurate Mobile Robot Motion Planning". In: *IEEE Robotics and Automation Letters* 6.2 [2021], pp. 3553–3560.

[52] T. Whelan, S. Leutenegger, R Salas-Moreno, B. Glocker, and A. Davison. "ElasticFusion: Dense SLAM without a pose graph". In: *Proc. of the Robotics: Science and Systems (RSS)*. 2015.

[53] C. Park, P. Moghadam, J. L. Williams, S. Kim, S. Sridharan, and C. Fookes. "Elasticity Meets Continuous-Time: Map-Centric Dense 3D LiDAR SLAM". In: *IEEE Trans. Robotics* 38.2 [2022], pp. 978–997.

[54] J. Zhang and S. Singh. "LOAM: Lidar Odometry and Mapping in Real-time." In: *Proc. of the Robotics: Science and Systems (RSS)*. Vol. 2. 9. 2014.

[55] T. Shan and B. Englot. "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4758–4765.

[56] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela. "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 5135–5142.

[57] P.-C. Lin, H. Komsuoglu, and D. Koditschek. "Sensor Data Fusion for Body State Estimation in a Hexapod Robot with Dynamical Gaits". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2005, pp. 4733–4738.

[58] M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C Remy, and R. Siegwart. "State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU". In: *Proc. of the Robotics: Science and Systems (RSS)*. July 2012.

[59] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart. "State estimation for legged robots on unstable and slippery terrain". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2013, pp. 6058–6064.

[60] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter. "The Two-State Implicit Filter Recursive Estimation for Mobile Robots". In: *IEEE Robotics and Automation Letters* 3.1 [2018], pp. 573–580.

[61] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, et al. "Anymal-a highly mobile and dynamic quadrupedal robot". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 38–44.

[62] S. Fahmi, G. Fink, and C. Semini. "On State Estimation for Legged Locomotion Over Soft Terrain". In: *IEEE Sensors Letters* PP [Jan. 2021], pp. 1–1.

[63] M. Camurri, M. Fallon, S. Bazeille, A. Radulescu, V. Barasuol, D. G. Caldwell, and C. Semini. "Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots". In: *IEEE Robotics and Automation Letters* 2.2 [2017], pp. 1023–1030.

[64] T.-Y. Lin, R. Zhang, J. Yu, and M. Ghaffari. "Legged Robot State Estimation using Invariant Kalman Filtering and Learned Contact Events". In: *5th Annual Conf. on Robot Learning*. 2021.

[65] B. Katz, J. D. Carlo, and S. Kim. "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2019, pp. 6295–6301.

[66] J. Zhang and S. Singh. "Visual-lidar odometry and mapping: low-drift, robust, and fast". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2015, pp. 2174–2181.

[67] Z. Wang, J. Zhang, S. Chen, C. Yuan, J. Zhang, and J. Zhang. "Robust High Accuracy Visual-Inertial-Laser SLAM System". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2019, pp. 6636–6641.

[68] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A.-a. Agha-mohammadi. "LOCUS: A

Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time". In: *IEEE Robotics and Automation Letters* 6.2 [2021], pp. 421–428.

[69] A. Santamaria-Navarro, R. Thakker, D. D. Fan, B. Morrell, and A.-a. Agha-mohammadi. "Towards Resilient Autonomous Navigation of Drones". In: *Robotics Research*. Springer International Publishing, 2022, pp. 922–937.

[70] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer. "Super Odometry: IMU-centric LiDAR-Visual-Inertial Estimator for Challenging Environments". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2021, pp. 8729–8736.

[71] Y. Yang, P. Geneva, X. Zuo, K. Eckenhoff, Y. Liu, and G. Huang. "Tightly-Coupled Aided Inertial Navigation with Point and Plane Features". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2019, pp. 6094–6100.

[72] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang. "LIC-Fusion: LiDAR-Inertial-Camera Odometry". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2019, pp. 5848–5854.

[73] X. Zuo, Y. Yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys. "LIC-Fusion 2.0: LiDAR-Inertial-Camera Odometry with Sliding-Window Plane-Feature Tracking". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2020, pp. 5112–5119.

[74] J. Lin, C. Zheng, W. Xu, and F. Zhang. "R2LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping". In: *IEEE Robotics and Automation Letters* 6.4 [2021], pp. 7469–7476.

[75] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert. "Autonomous navigation for BigDog". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2010, pp. 4736–4741.

[76] J. Ma, M. Bajracharya, S. Susca, L. Matthies, and M. Malchano. "Real-time pose estimation of a dynamic quadruped in GPS-denied environments for 24-hour operation". In: *Intl. J. of Robotics Research* 35.6 [2016], pp. 631–653.

[77] M. F. Fallón, M. Antone, N. Roy, and S. Teller. "Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing". In: *Intl. J. of Humanoid Robotics*. 2014, pp. 112–119.

[78] S. Nobili, R. Scona, M. Caravagna, and M. Fallon. "Overlap-based ICP tuning for robust localization of a humanoid robot". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2017, pp. 4721–4728.

[79] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon. "Pronto: A Multi-Sensor State Estimator for Legged Robots in Real-World Scenarios". In: *Frontiers in Robotics and AI* 7 [2020].

[80] C Semini, N. G. Tsagarakis, E Guglielmino, M Focchi, F Cannella, and D. G. Caldwell. "Design of HyQ – a hydraulically and electrically actuated quadruped robot". In: *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 225.6 [2011], pp. 831–849.

[81] D. Wisth, M. Camurri, S. Das, and M. Fallon. "Unified Multi-Modal Landmark Tracking for Tightly Coupled Lidar-Visual-Inertial Odometry". In: *IEEE Robotics and Automation Letters* 6.2 [2021], pp. 1004–1011.

[82] G. Kim and A. Kim. "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018, pp. 4802–4809.

[83] B. Steder, M. Ruhnke, S. Grzonka, and W. Burgard. "Place recognition in 3D scans using a combination of bag of words and point feature based relative pose estimation". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2011, pp. 1249–1255.

[84] L. He, X. Wang, and H. Zhang. "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2016, pp. 231–237.

[85] W. Wohlkinger and M. Vincze. "Ensemble of shape functions for 3D object classification". In: *IEEE Intl. Conf. on Robotics and Biomimetics (ROBIO)*. 2011, pp. 2987–2992.

[86] G. Kim, B. Park, and A. Kim. "1-Day Learning, 1-Year Localization: Long-Term LiDAR Localization Using Scan Context Image". In: *IEEE Robotics and Automation Letters* 4.2 [2019], pp. 1948–1955.

[87] Y. Huang, T. Shan, F. Chen, and B. Englot. "DiSCo-SLAM: Distributed Scan Context-Enabled Multi-Robot LiDAR SLAM With Two-Stage Global-Local Graph Optimization". In: *IEEE Robotics and Automation Letters* 7.2 [2022], pp. 1150–1157.

[88] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena. "SegMatch: Segment based place recognition in 3D point clouds". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 5266–5272.

[89] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena. "SegMap: Segment-based mapping and localization using data-driven descriptors". In: *Intl. J. of Robotics Research* 39.2-3 [2020], pp. 339–355.

[90] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. "On the segmentation of 3D LIDAR point clouds". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2011, pp. 2798–2805.

[91] T Rabbani, F. Heuvel, and G. Vosselman. "Segmentation of point clouds using smoothness constraint". In: *Intl. Archives of Phot., Remote Sens. and Spatial Inf. Sciences* 36 [Jan. 2006].

[92] R. Dubé, M. G. Gollub, H. Sommer, I. Gilitschenski, R. Siegwart, C. Cadena, and J. Nieto. "Incremental-Segment-Based Localization in 3-D Point Clouds". In: *IEEE Robotics and Automation Letters* 3.3 [2018], pp. 1832–1839.

[93] G. Tinchev, S. Nobili, and M. Fallon. "Seeing the Wood for the Trees: Reliable Localization in Urban and Natural Environments". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018, pp. 8239–8246.

[94] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6 [2016], pp. 1309–1332.

[95] T. Bailey and H. F. Durrant-Whyte. "Simultaneous Localization and Mapping: Part I". In: *IEEE Robotics & Automation Magazine* 13.3 [2006], pp. 108–117.

[96] T. Bailey and H. F. Durrant-Whyte. "Simultaneous Localization and Mapping: Part II". In: *J. of Robotics and Autonomous Systems (RAS)* 13.2 [2006], pp. 99–110.

[97] R. Chatila and J. Laumond. "Position referencing and consistent world modeling for mobile robots". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Vol. 2. 1985, pp. 138–145.

[98] N. Ayache and O. D. Faugeras. "Building, Registrating, and Fusing Noisy Visual Maps". In: *Intl. J. of Robotics Research* 7.6 [1988], pp. 45–65.

[99] J. Crowley. "World modeling and position estimation for a mobile robot using ultrasonic ranging". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Vol. 2. 1989, pp. 674–680.

[100] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. "A solution to the simultaneous localization and map building (SLAM) problem". In: *IEEE Trans. Robotics and Automation* 17.3 [2001], pp. 229–241.

[101] S. Julier and J. Uhlmann. "A counter example to the theory of simultaneous localization and map building". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Vol. 4. 2001, pp. 4238–4243.

[102] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem". In: *Proc. of AAAI Conf. on Artificial Intelligence*. Nov. 2002.

[103] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges". In: *Intl. Joint Conf. on Artificial Intelligence* [June 2003].

[104] H. Strasdat, J. M. M. Montiel, and A. J. Davison. "Real-time monocular SLAM: Why filter?" In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2010, pp. 2657–2664.

[105] F. Kschischang, B. Frey, and H.-A. Loeliger. "Factor graphs and the sum-product algorithm". In: *IEEE Trans. on Information Theory* 47.2 [2001], pp. 498–519.

[106] F. Dellaert and M. Kaess. "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing". In: *Intl. J. of Robotics Research* 25.12 [2006], pp. 1181–1203.

[107] M. Kaess, A. Ranganathan, and F. Dellaert. "iSAM: Incremental Smoothing and Mapping". In: *IEEE Trans. Robotics* 24.6 [2008], pp. 1365–1378.

[108] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Trans. Robotics* 31.5 [2015], pp. 1147–1163.

[109] R. Mur-Artal and J. D. Tardós. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Trans. Robotics* 33.5 [2017], pp. 1255–1262.

[110] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM". In: *IEEE Trans. Robotics* 37.6 [2021], pp. 1874–1890.

[111] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *Proc. of the Intl. Conf. on Computer Vision (ICCV)*. 2011, pp. 2564–2571.

[112] D. Galvez-López and J. D. Tardos. "Bags of Binary Words for Fast Place Recognition in Image Sequences". In: *IEEE Trans. Robotics* 28.5 [2012], pp. 1188–1197.

[113] M. Ramezani, M. Mattamala, and M. Fallon. "AEROS: AdaptivE RObust Least-Squares for Graph-Based SLAM". In: *Frontiers in Robotics and AI* 9 [Apr. 2022], p. 789444.

[114] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. "An Atlas framework for scalable mapping". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Vol. 2. 2003, 1899–1906 vol.2.

[115] N. Sünderhauf and P. Protzel. "Towards a robust back-end for pose graph SLAM". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2012, pp. 1254–1261.

[116] J. T. Barron. "A General and Adaptive Robust Loss Function". In: *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019.

[117] N. Chebrolu, T. Läbe, O. Vysotska, J. Behley, and C. Stachniss. "Adaptive Robust Kernels for Non-Linear Least Squares Problems". In: *IEEE Robotics and Automation Letters* 6.2 [2021], pp. 2240–2247.

[118] Z. Zhang. "Parameter estimation techniques: a tutorial with application to conic fitting". In: *Image and Vision Computing* 15.1 [1997], pp. 59–76.

[119] G. Agamennoni, P. Furgale, and R. Siegwart. "Self-tuning M-estimators". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2015, pp. 4628–4635.

[120] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone. "Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection". In: *IEEE Robotics and Automation Letters* 5.2 [2020], pp. 1127–1134.

[121] M. Black and A. Rangarajan. "On the Unification Line Processes, Outlier Rejection, and Robust Statistics with Applications in Early Vision". In: *Intl. J. of Computer Vision* 19 [July 1996], pp. 57–91.

[122] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme. "Active planning for underwater inspection and the benefit of adaptivity". In: *Intl. J. of Robotics Research* 32.1 [2013], pp. 3–18.

[123] G. A. Hollinger and G. S. Sukhatme. "Sampling-based robotic information gathering algorithms". In: *Intl. J. of Robotics Research* 33.9 [2014], pp. 1271–1287.

[124] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard. "Advanced perception, navigation and planning for autonomous in-water ship hull inspection". In: *Intl. J. of Robotics Research* 31 [12 2016], pp. 1445–1464.

[125] M. Kazhdan, M. Bolitho, and H. Hoppe. "Poisson Surface Reconstruction". In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP '06. Cagliari, Sardinia, Italy: Eurographics Association, 2006, pp. 61–70.

[126] T. Bodenmüeller. *Streaming Surface Reconstruction from Real Time 3D Measurements*. 2009.

[127] S. Kriegel, T. Bodenmüller, M. Suppa, and G. Hirzinger. "A surface-based Next-Best-View approach for automated 3D model completion of unknown objects". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2011, pp. 4869–4874.

[128] S. Kriegel, C. Rink, T. Bodenmüller, A. Narr, M. Suppa, and G. Hirzinger. "Next-best-scan planning for autonomous 3D modeling". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Oct. 2012, pp. 2850–2856.

[129] W. E. Lorensen and H. E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm". In: *SIGGRAPH Comput. Graph.* 21.4 [Aug. 1987], 163–169.

[130] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao. "Chisel: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device using Spatially Hashed Signed Distance Fields". In: *Proc. of the Robotics: Science and Systems (RSS)*. Vol. 4. 2015, p. 1.

[131] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. "Bundlefusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration". In: *ACM Transactions on Graphics* 36.4 [2017], p. 1.

[132] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. "Real-Time 3D Reconstruction at Scale Using Voxel Hashing". In: *ACM Transactions on Graphics* 32.6 [2013].

[133] M. Tanner, P. Piniés, L. M. Paz, Ştefan Săftescu, A. Bewley, E. Jonasson, and P. Newman. "Large-scale outdoor scene reconstruction and correction with vision". In: *Intl. J. of Robotics Research* [2018].

[134] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena. "C-blox: A Scalable and Consistent TSDF-based Dense Mapping Approach". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018.

[135] D. De Gregorio and L. Di Stefano. "SkiMap: An efficient mapping framework for robot navigation". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2017, pp. 2569–2576.

[136] D. Duberg and P. Jensfelt. "UFOMap: An Efficient Probabilistic 3D Mapping Framework That Embraces the Unknown". In: *IEEE Robotics and Automation Letters* 5.4 [2020], pp. 6411–6418.

[137] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous Robots* 34.3 [2013], pp. 189–206.

[138] B.-j. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess. "Virtual Occupancy Grid Map for Submap-based Pose Graph SLAM and Planning in 3D Environments". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* 1.2 [2018], pp. 2175–2182.

[139] P. Sodhi, B.-J. Ho, and M. Kaess. "Online and Consistent Occupancy Grid Mapping for Planning in Unknown Environments". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2019, pp. 7879–7886.

[140] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza. "A comparison of volumetric information gain metrics for active 3D object reconstruction". In: *Autonomous Robots* 42.2 [2018], pp. 197–208.

[141] R. Border, J. D. Gammell, and P. Newman. "Surface Edge Explorer (see): Planning Next Best Views Directly from 3D Observations". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2018, pp. 1–8.

[142] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid. "View planning for 3D object reconstruction with a mobile manipulator robot". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* Iros [2014], pp. 4227–4233.

[143] S. Chen, Y. Li, and N. M. Kwok. "Active vision in robotic systems: A survey of recent developments". In: *Intl. J. of Robotics Research* 30.11 [2011], pp. 1343–1377.

[144] M. Karaszewski, M. Adamczyk, and R. Sitnik. "Assessment of next-best-view algorithms performance with various 3D scanners and manipulator". In: *ISPRS J. of Phot. and Remote Sens.* 119 [2016], pp. 320–333.

[145] P. S. Blaer and P. K. Allen. "Data Acquisition and View Planning for 3-D Modeling Tasks". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2007, pp. 417–422.

[146] K. Schmid, H. Hirschmüller, A. Dömel, I. Grixa, M. Suppa, and G. Hirzinger. "View planning for multi-view stereo 3D reconstruction using an autonomous multicopter". In: *Journal of Intelligent & Robotic Systems* 65.1-4 [2012], pp. 309–323.

[147] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart. "Receding horizon next-best-view planner for 3D exploration". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2016, pp. 1462–1468.

[148] Y. Kompis, L. Bartolomei, R. Mascaro, L. Teixeira, and M. Chli. "Informed Sampling Exploration Path Planner for 3D Reconstruction of Large Scenes". In: *IEEE Robotics and Automation Letters* 6.4 [2021], pp. 7893–7900.

[149] S. Song, D. Kim, and S. Choi. "View Path Planning via Online Multiview Stereo for 3-D Modeling of Large-Scale Structures". In: *IEEE Trans. Robotics* 38.1 [2022], pp. 372–390.

[150] A. Dai, S. Papatheodorou, N. Funk, D. Tzoumanikas, and S. Leutenegger. "Fast Frontier-based Information-driven Autonomous Exploration with an MAV". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020, pp. 9570–9576.

[151] J. J. Kuffner and S. M. LaValle. "RRT-connect: An efficient approach to single-query path planning". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Vol. 2. 2000, 995–1001 vol.2.

[152] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. "Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Vol. 2017-Septe. 2017, pp. 1366–1373.

[153] R. Mascaro, M. Wermelinger, M. Hutter, and M. Chli. "Towards automating construction tasks: Large-scale object mapping, segmentation, and manipulation". In: *J. of Field Robotics* 38.5 [2021], pp. 684–699.

[154] P. Fankhauser, M. Bloesch, and M. Hutter. "Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization". In: *IEEE Robotics and Automation Letters (RA-L)* 3.4 [2018], pp. 3019–3026.

[155] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. "Comparing ICP Variants on Real-World Data Sets". In: *Autonomous Robots* 34.3 [Feb. 2013], pp. 133–148.

[156] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, et al. "Autonomous Spot: Long-range autonomous exploration of extreme environments with legged locomotion". In: *arXiv preprint arXiv:2010.09259* [2020].

[157] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood, et al. "LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 80–86.

[158] M. Ramezani, G. Tinchev, E. Iuganov, and M. Fallon. "Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020, pp. 4158–4164.

[159] M. Gopi, S. Krishnan, and C. Silva. "Surface Reconstruction Based on Lower Dimensional Localized Delaunay Triangulation." In: *Computer Graphics Forum* 19 [Sept. 2000], pp. 467 –478.

[160] Y. Lipman, D. Cohen-Or, and D. Levin. "Data-dependent MLS for Faithful Surface Approximation". In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. SGP '07. Barcelona, Spain: Eurographics Association, 2007, pp. 59–67.

[161] R. Schnabel and R. Klein. "Octree-based Point-cloud Compression". In: *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics*. SPBG'06. Boston, Massachusetts: Eurographics Association, 2006, pp. 111–121.

[162] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach. "Real-time Compression of Point Cloud Streams". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Minnesota, USA, May 2012.

[163] S. Kriegel, M. Brucker, Z. C. Marton, T. Bodenmuller, and M. Suppa. "Combining object modeling and recognition for active scene exploration". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2013, pp. 2384–2391.

[164] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, and E. Lopez-Damian. "Volumetric next-best-view planning for 3D object reconstruction with positioning error". In: *Intl. J. of Advanced Robotic Systems* 11 [2014].

[165] J. Nieto, J. Guivant, and E. Nebot. "DenseSLAM: Simultaneous Localization and Dense Mapping". In: *Intl. J. of Robotics Research* 25.8 [2006], pp. 711–744.

[166] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess. "Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 2175–2182.

[167] D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Guenther, M. Tranzatto, P. Fankhauser, and M. Hutter. "Advances in real-world applications for legged robots". In: *J. of Field Robotics* 35 [2018], pp. 1311–1326.

[168] E. Turner and A. Zakhor. "Floor plan generation and room labeling of indoor environments from laser range data". In: *Proc. of the IEEE Intl. Conf. on Computer Graphics Theory and Applications (GRAPP)*. 2014, pp. 1–12.

[169] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. "3d semantic parsing of large-scale indoor spaces". In: *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1534–1543.

[170] C. Mura, O. Mattausch, A. J. Villanueva, E. Gobbetti, and R. Pajarola. "Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts". In: *Computers & Graphics* 44 [2014], pp. 20–32.

[171] C. Mura, O. Mattausch, and R. Pajarola. "Piecewise-planar reconstruction of multi-room interiors with arbitrary wall arrangements". In: *Computer Graphics Forum*. Vol. 35. 7. Wiley Online Library. 2016, pp. 179–188.

[172] S. Ochmann, R. Vock, and R. Klein. "Automatic reconstruction of fully volumetric 3D building models from oriented point clouds". In: *ISPRS J. of Phot. and Remote Sens.* 151 [2019], pp. 251–262.

[173] S. Nikoohemat, A. A. Diakité, S. Zlatanova, and G. Vosselman. "Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management". In: *Automation in Construction* 113 [2020], p. 103109.

[174] F. Dellaert and M. Kaess. "Factor Graphs for Robot Perception". In: *Foundations and Trends® in Robotics* 6.1-2 [2017], pp. 1–139.

[175] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P. H. J. Kelly, and S. Leutenegger. "Efficient Octree-Based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping". In: *IEEE Robotics and Automation Letters* 3.2 [Apr. 2018], pp. 1144–1151.

[176] C. Loop, Q. Cai, S. Orts-Escolano, and P. A. Chou. "A closed-form Bayesian fusion equation using occupancy probabilities". In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE. 2016, pp. 380–388.

[177] S. Nobili, G. Tinchev, and M. Fallon. "Predicting Alignment Risk to Prevent Localization Failure". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2018, pp. 1003–1010.

[178] S Klarsfeld and J. Oteo. "The Baker-Campbell-Hausdorff formula and the convergence of the Magnus expansion". In: *J. of Physics A: Mathematical and General* 22.21 [1989], p. 4565.

[179] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 4628–4635.

[180] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation". In: *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440.

[181] F. Yu and V. Koltun. "Multi-scale context aggregation by dilated convolutions". In: *arXiv preprint arXiv:1511.07122* [2015].

[182] V. Badrinarayanan, A. Kendall, and R. Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 39.12 [2017], pp. 2481–2495.

[183] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Proc. of the Intl. Conf. on Medical Image Computing and Computer-assisted Intervention*. Springer. 2015, pp. 234–241.

[184] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. "A benchmark dataset and evaluation methodology for video object segmentation". In: *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 724–732.

[185] W. Wang, J. Shen, R. Yang, and F. Porikli. "Saliency-aware video object segmentation". In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 40.1 [2017], pp. 20–33.

[186] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. "Learning features by watching objects move". In: *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2701–2710.

[187] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proc. of the Eur. Conf. on Computer Vision (ECCV)*. 2018, pp. 801–818.

[188] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler. "Gated-scnn: Gated shape cnns for semantic segmentation". In: *Proc. of the Intl. Conf. on Computer Vision (ICCV)*. 2019, pp. 5229–5238.

[189] S. Garg, N. Sünderhauf, F. Dayoub, D. Morrison, A. Cosgun, G. Carneiro, Q. Wu, T.-J. Chin, I. Reid, S. Gould, et al. "Semantics for Robotic Mapping, Perception and Interaction: A Survey". In: *Found. and Trends in Robotics* 8 [2020], pp. 1–224.

[190] S. Yang, Y. Huang, and S. Scherer. "Semantic 3D occupancy mapping through efficient high order CRFs". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 590–597.

[191] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 77–85.

[192] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. "KPConv: Flexible and Deformable Convolution for Point Clouds". In: *Proc. of the Intl. Conf. on Computer Vision (ICCV)*. 2019, pp. 6410–6419.

[193] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, and M. Jagersand. "Rtseg: Real-time semantic segmentation comparative study". In: *Proc. of the IEEE Intl. Conf. on Image Processing (ICIP)*. 2018, pp. 1603–1607.

[194] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *Proc. of the IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255.

[195] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. "University of Michigan North Campus long-term vision and lidar dataset". In: *Intl. J. of Robotics Research* 35.9 [2016], pp. 1023–1035.