UNIVERSITY OF CALIFORNIA SAN DIEGO

Distributed Multi-Robot Active OcTree Mapping

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics, and Control)

by

Arash Asgharivaskasi

Committee in charge:

Professor Nikolay Atanasov, Chair
Professor Tara Javidi
Professor Sonia Martínez
Professor Michael Yip

2024

The Dissertation of Arash Asgharivaskasi is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This dissertation would not have been possible without my advisor Nikolay Atanasov. I started my Ph.D. journey right after finishing my Bachelor's and hence I did not have much insight about how to do graduate-level research, let alone robotics in particular. Nikolay's patience and characteristic love for mentoring were the main reasons why I was able to experiment with different topics, find a deep interest in doing research, and eventually find my niche in robotics. Moreover, his trust in me and the fact that he gave me a lot of independence throughout my Ph.D. years helped me to develop much needed leadership skills. His guidance and support have been instrumental in my professional development and my identity as a roboticist. It has been truly a privilege to have him as my advisor.

I would also like to thank my Ph.D. committee members, Tara Javidi, Michael Yip, and Sonia Martinez, for their interest in learning about my research, providing valuable suggestions, and dedicating their time to attend my proposal and defense. The feedback I received from my committee during the candidacy exam significantly influenced the direction of my research, especially regarding extension to multi-robot systems.

During my Ph.D. research I was fortunate to collaborate with many brilliant scholars. I want to thank my coauthors, Shumon Koga, George Pappas, Mariliza Tzes, Panagiotis Tsiotras, Daniel Larsson, Jaein Lim, Thai Duong, Zhirui Dai, Pengzhi Yang, Fritz Girke, Shusen Lin, and Yuhan Liu. Additionally I want to acknowledge and thank my fellow ERL members, Vikas Dhiman, Tianyu Wang, Parth Paritosh, Qiaojun Feng, Ehsan Zobeidi, Zhichao Li, Baoqian Wang, Kehan Long, Abdullah Altawaitan, Hanwen Cao, Yinzhuang Yi, Nikola Raicevic, Yulun Tian, Brian Lee, Mo Shan, Dwait Bhatt, Hojoon Shin, Alexander Khoury, Jason Stanley, Tianji Tang, and other colleagues I miss to mention. Finally, I want to thank my friends and family, for their enduring support in every part of my journey.

Chapter 3, in part, is a reprint of the material as it appears in A. Asgharivaskasi and N. Atanasov, "Active Bayesian multi-class mapping from range and semantic segmentation observations," IEEE International Conference on Robotics and Automation (ICRA), pp. 1-7,

2021, and in A. Asgharivaskasi and N. Atanasov, "Semantic octree mapping and Shannon mutual information computation for robot exploration," IEEE Transactions on Robotics (T-RO), vol. 39, no.3, pp. 1910-1928, 2023. The dissertation author was the primary investigator and author of these papers.

Chapter 4, in part, is a reprint of the material as it appears in A. Asgharivaskasi and N. Atanasov, "Active Bayesian multi-class mapping from range and semantic segmentation observations," IEEE International Conference on Robotics and Automation (ICRA), pp. 1-7, 2021, and in A. Asgharivaskasi and N. Atanasov, "Semantic octree mapping and Shannon mutual information computation for robot exploration," IEEE Transactions on Robotics (T-RO), vol. 39, no.3, pp. 1910-1928, 2023. The dissertation author was the primary investigator and author of these papers.

Chapter 5, in part, is a reprint of the material as it appears in A. Asgharivaskasi, S. Koga, and N. Atanasov, "Active mapping via gradient ascent optimization of Shannon mutual information over continuous *SE(3)* trajectories," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 12994-13001, 2022. The dissertation author was the primary investigator and author of these papers.

Chapter 6, in part, is a reprint of the material as it appears in A. Asgharivaskasi and N. Atanasov, "Distributed optimization with consensus constraint for multi-robot semantic octree mapping," IEEE International Conference on Robotics and Automation (ICRA) workshop on Collaborative Perception and Learning (CoPerception), 2023. Chapter 6, in part, has been submitted for publication of the material as it may appear in A. Asgharivaskasi, F. Girke, and N. Atanasov, "Riemannian optimization for active mapping with robot teams," IEEE Transactions on Robotics (T-RO), 2024. The dissertation author was the primary investigator and author of these papers.

VITA

2018	Bachelor of Science in Electrical Engineering (Communication Systems) with a Minor in Economics, Sharif University of Technology
2021	Master of Science in Electrical Engineering (Intelligent Systems, Robotics, and Control), University of California San Diego
2024	Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics, and Control), University of California San Diego

PUBLICATIONS

A. Asgharivaskasi, F. Girke, and N. Atanasov, "Riemannian Optimization for Active Mapping with Robot Teams", under review.

A. Asgharivaskasi, N. Atanasov, "Semantic OcTree Mapping and Shannon Mutual Information Computation for Robot Exploration", IEEE Transactions on Robotics (T-RO), 2023.

Z. Dai, **A. Asgharivaskasi**, T. Duong, S. Lin, M. Tzes, G. Pappas, and N. Atanasov, "Optimal Scene Graph Planning with Large Language Model Guidance", IEEE International Conference on Robotics and Automation (ICRA), 2024.

P. Yang, S. Koga, **A. Asgharivaskasi**, and N. Atanasov, "Policy Learning for Active Target Tracking over Continuous *SE(3)* Trajectories", in Learning for Dynamics & Control Conference (L4DC), 2023.

D. T. Larsson, **A. Asgharivaskasi**, J. Lim, N. Atanasov, and P. Tsiotras, "Information-theoretic Abstraction of Semantic Octree Models for Integrated Perception and Planning", IEEE International Conference on Robotics and Automation (ICRA), 2023.

P. Yang, Y. Liu, S. Koga, **A. Asgharivaskasi**, and N. Atanasov, "Learning Continuous Control Policies for Information-Theoretic Active Perception", IEEE International Conference on Robotics and Automation (ICRA), 2023.

A. Asgharivaskasi, S. Koga, and N. Atanasov, "Active Mapping via Gradient Ascent Optimization of Shannon Mutual Information over Continuous *SE(3)* Trajectories", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022.

S. Koga, **A. Asgharivaskasi**, and N. Atanasov, "Active SLAM over Continuous Trajectory and Control: A Covariance-Feedback Approach", American Control Conference (ACC), 2022.

S. Koga, **A. Asgharivaskasi**, and N. Atanasov, "Active Exploration and Mapping via Iterative

Covariance Regulation over Continuous *SE(3)* Trajectories", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021.

A. Asgharivaskasi and N. Atanasov, "Active Bayesian Multi-class Mapping from Range and Semantic Segmentation Observations", in IEEE International Conference on Robotics and Automation (ICRA), 2021.

A. Asgharivaskasi and N. Atanasov, "Distributed Optimization with Consensus Constraint for Multi-Robot Semantic Octree Mapping", Workshop on Collaborative Perception and Learning (CoPerception) at ICRA, 2023.

ABSTRACT OF THE DISSERTATION

Distributed Multi-Robot Active OcTree Mapping

by

Arash Asgharivaskasi

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics, and Control)

University of California San Diego, 2024

Professor Nikolay Atanasov, Chair

Many real-world mobile robot applications, such as disaster response, military reconnaissance, and environmental monitoring, require operating in unknown and unstructured environments. This calls for algorithms that empower robots with active information gathering capabilities in order to autonomously and incrementally build a model of an environment. In this dissertation, we present a novel 3-D multi-class online mapping approach using a stream of range and semantic segmentation observations. Moreover, we derive a closed-form expression for the Semantic Shannon Mutual Information (SSMI) between our proposed map representation and a sequence of future sensor observations. Using an octree data structure, we reduce the memory footprint of the map storage for large-scale environments, while simultaneously

accelerating the computation of mutual information. This allows real-time integration of new sensor measurements into the map, and rapid evaluation of candidate future sensor poses for exploration. Additionally, we introduce a differentiable approximation of the Shannon mutual information between grid maps and ray-based measurements, enabling gradient-based occlusion and collision-aware active mapping. The gradient-based active mapping in the continuous space of sensor poses reduces the optimization complexity from exponential in the number of robots to linear, paving the way for extension from a single agent to a team of robots. We formulate multi-robot exploration as a combination of multi-robot mapping and multi-robot planning, where both sub-problems are specified as an instance of multi-agent Riemannian optimization. We propose a general distributed Riemannian optimization algorithm that solves both mapping and planning in fully decentralized manner. Our method, named Riemannian Optimization for Active Mapping (ROAM), enables distributed collaborative multi-robot exploration, with only point-to-point communication and no central estimation and control unit. Lastly, we deploy our active mapping method on a team of ground wheeled robots in both simulation and real-world environments, and compare its performance with other autonomous exploration approaches.

Chapter 1

Introduction

1.1 Motivation

As cyber-physical systems are delegated with real-world tasks requiring unprecedented levels of autonomy, the necessity of enabling mobility as a core capability of said systems become inevitable. In order to achieve reliable mobility, the autonomous system (i.e., robot) needs to understand its environment both in terms of global knowledge of objects and regions of interest, as well as local situational awareness of obstacles. This is, however, fundamentally different from possessing the ability for completing a specific task (e.g., sorting a shelf), in the sense that the robot should learn from unforeseen configurations and dynamics of the environment, and react accordingly, all on the fly. Such observations motivate us to design methods for active information gathering for mobile robots, with the goal of ensuring the survival of the robot and the success of given tasks.

In particular, this thesis presents an autonomous exploration and environment mapping method, by which a robot equipped with range and vision sensing can incrementally estimate a probabilistic multi-class map of the environment, and actively choose trajectories that yield maximally informative future sensor observations. The probabilistic multi-class map representation allows task planning while considering uncertain semantic information, leading to more robust and safer robot operation. Moreover, incorporating semantics into active multi-class mapping can itself be beneficial to the efficiency of exploration, in that objects with higher

detection error rate can be prioritized for more fine-grained sensor measurements. Having said that, large-scale autonomous exploration and mapping on resource-constrained robot platforms demands effective utilization of memory and compute assets in order to enable edge deployment. The work proposed in this thesis aims to address these challenges by using certain data structures and derivation of a closed-form differentiable measure of observation informativeness.

Furthermore, employing a team of collaborating robots may substantially accelerate the pace at which a given task can be accomplished, while the overall system can potentially become more robust due to redundancy. Autonomous exploration of unknown environments using a team of mobile robots demands distributed perception and planning strategies to enable efficient and scalable performance. Ideally, each robot should update its map and plan its motion not only relying on its own observations, but also considering the observations of its peers. Centralized solutions to multi-robot coordination are susceptible to central node failure and require a sophisticated communication infrastructure for reliable operation. Current decentralized active mapping methods consider simplistic robot models with linear-Gaussian observations and Euclidean robot states. We extend our proposed autonomous exploration and mapping method to a team of robots, relying only on one-hop communication and without the need of a centralized estimation and planning node. The presented work can empower the next generation of collaborative robotics, with applications in search and rescue, security and surveillance, and environmental monitoring.

1.2 Problem Statement

The main goal throughout this thesis is to design a distributed multi-robot exploration and multi-class mapping method using streaming range and vision input. The first objective is devising a Bayesian framework to incorporate new sensor data into a probabilistic environment representation. In particular, we are interested to design a mapping method that receives a stream of 3-D point clouds with semantic annotations, and incrementally estimates a probabilistic

multi-class and multi-resolution voxel map. Secondly, the probabilistic map, alongside the geometric and noise specifications of the sensor, are going to be utilized for extracting maximally informative sensor poses that lead to a minimum uncertainty map estimation. Additionally, we require such information-theoretic exploration method to be computationally efficient in a way that it can be deployed on board of a size, weight, and power-constrained robot. This process should take the most recent estimate of the map and robot pose, and compute a collision-free sensing trajectory in the continuous space of 3-D poses, with the goal of minimizing map uncertainty. Lastly, the proposed exploration and mapping method should be generalizable to a team of collaborative robots, necessitating desirable computational complexity with respect to the number of robots, as well as memory efficiency of the map representation to be shared over a point-to-point communication network. This multi-robot exploration method should be provided with the connectivity information of the robot network, where each robot locally computes a multi-class map and an informative sensing trajectory. The objective of multi-robot active mapping is to reach an agreement on the exploration path of the team, as well as a globally consistent map of the environment.

1.3 Related Work

The problem of autonomous exploration and mapping is one of the most extensively studied topics within the robotics community. In this section, we only provide a high-level overview of the methods that are closely related to the overall proposition of this thesis. For a more detailed review of the related works, please refer to the respective chapters in the thesis.

Active mapping for mobile robots dates back to 1990s [161], where a laser-limited sonar sensor was used to maintain a 2-dimensional (2-D) *evidence grid*, later known as *occupancy grid*. An occupancy grid is an array of uniform resolution cells, where each cell represents a rectangular physical region of the environment, and stores information regarding the occupancy of its corresponding space. Since then, advancements in algorithms, sensing, data storage,

and computation hardware have pushed the boundaries in various aspects such as on-board 3-dimensional (3-D) range sensing [58, 94, 122, 130], low memory-footprint large-scale 3-D mapping [70], and uncertainty-aware path planning for exploration [35, 76, 170]. The mapping method presented in this thesis builds upon the previous works in that 3-D sensor measurements are used to incrementally build a probabilistic multi-class extension of the octree map proposed in [70]. Moreover, we extend the concept of uncertainty-aware path planning for exploration to multi-class maps, such that the exploration algorithm takes into account the per-class uncertainty, leading to more effective active mapping.

The problem of active mapping is a specific instance of active information acquisition, which in its simplest form, can be formulated as a multi-armed bandit problem [2, 104, 105]. Despite the vast literature for the bandit problem with global optimality guarantees [101, 102], the kinematic, dynamic, and safety constraints of mobile robots limit the usability of said methods in the context of active mapping with limited sensor field of view. The line of works in [12, 16, 17, 80, 81] have addressed the challenges brought up by the mobile robot constraints via explicitly incorporating the motion and observation models of the robot. The common feature among these works is the linear-Gaussian assumption for the robot observation model, which results in computationally efficient open-loop active perception policies with the possibility of extension to multi-robot scenarios [13–15]. However, this assumption is too simplistic for active occupancy grid mapping using range sensing, limiting the usability of such methods in the cases where the sensor field of view can be occluded. In this work, we go beyond the limitations of linear-Gaussian perception models by presenting a range-category observation model that simultaneously incorporates ray-based range sensing and vision-based semantic segmentation. Our proposed model allows designing a closed-loop, collision and occlusion-aware exploration policy for active mapping of unknown environments using a team of collaborating robots.

The topic of distributed multi-robot exploration lies in the intersection of collaborative information acquisition, such as multi-agent linear bandits [103], and decentralized optimization, e.g., asynchronous distributed convex optimization [108]. The work by Akbari et al. [4] extends

the alternating direction method of multipliers (ADMM) to distributed systems, and obtains a regret bound for individual agents. Sundaram and Ghaserifard [139] investigate the vulnerabilities of distributed optimization algorithms in the face of non-cooperative agents, and present a solution robust to adversarial attacks. Moreover, Ghaserifard and Smith [61] generalize distributed optimization to submodular objective functions. Continuous-time distributed optimization with time-varying directed networks is studied in [147], where the authors formulate distributed optimization as dynamical system. Related to distributed coordination in non-Euclidean spaces, Sepulchre et al. [131] uses Hilbert distance as a measure of consensus for non-commutative spaces, which is specially useful in the context of quantum information theory. The work in [127] formulates coordination of rigid body attitudes as a distributed optimization problem, where the cost function is the sum of chordal distances between each pair of agents. The same problem, for the more general context of Lie groups, have been studied in [126] and [84]. In this dissertation, we propose a distributed multi-agent optimization algorithm where the state of each agent is defined within a general Riemannian manifold. The proposed algorithm possesses consensus and optimality guarantees, which enables cooperative decentralized multi-robot exploration with only point-to-point communication.

1.4 Contributions

The overarching goal of this thesis is to develop a distributed multi-robot exploration and multi-class mapping method that can be deployed on a team of size, weight, and power-constrained robots without the need for a central estimation and control unit. Fig. 1.1 highlights our contributions in this dissertation. We delineate our approach to achieve our goal in the following chapters, each addressing one aspect of the solution.

Chapter 2

We start by briefly reviewing the background mathematical concepts that are key to understand the contributions of this dissertation. We discuss Bayesian inference and its applica-

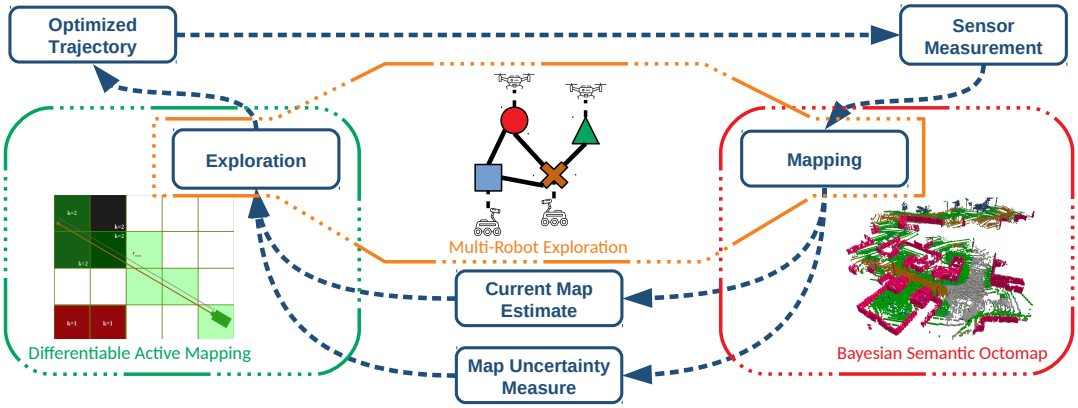


Figure 1.1. Overview of active mapping and the contributions. The robot receives a sequence of sensor measurements, and incrementally builds a map of the environment. In our work, we present a Bayesian semantic octomap method that estimates a 3-D multi-class map from range and semantic segmentation observations (red section). The map estimate is then used to extract a set of future sensor poses that maximize the map accuracy and explored area. We introduce an information-theoretic and gradient-based 3-D sensor pose optimization method that finds the most informative sensing trajectory (green section). The robot receives the computed sensing trajectory, calculates the corresponding motion commands, and repeats the information gathering cycle. We also devise a collaborative multi-robot active mapping algorithm that enables coordination in exploration as well as consistency among each robot's map estimate (orange section).

tion in robot mapping. Additionally, we provide intuitive descriptions for information theoretic quantities such as entropy and the Shannon mutual information. Lastly, a short overview of matrix Lie groups and Riemannian manifolds is presented.

Chapter 3

The first step towards information-theoretic exploration is devising a Bayesian framework for incorporating sensor measurements into a probabilistic map representation. This chapter presents a novel Bayesian mapping method that incrementally constructs a probabilistic multi-class voxel map of an environment using a stream of range and semantic segmentation observations. Additionally, in order to reduce the memory requirements imposed by mapping very large environments, we employ an octree data compression algorithm with controllable data recovery accuracy. We deploy the proposed multi-class octree mapping method on real-world robot platforms and show the mapping performance in an assortment of experiments.

Chapter 4

Once we are able to build an online probabilistic multi-class map, we switch our focus to discuss how future range-category observations can lead to reduction in map estimation uncertainty. Specifically, we use the Shannon mutual information as a quantitative measure of uncertainty reduction. The main contribution of this chapter is a semantic mutual information formula that takes into account per-class uncertainties. This leads to a more efficient exploration strategy compared to traditional occupancy-based methods. Also, we show that exploiting the octree structure of our map representation leads to significant reduction in computation complexity when it comes to calculating the mutual information. We finish the chapter by evaluating the performance of the exploration method in comparison with other baseline active mapping approaches, both in simulation and in the real world.

Chapter 5

The main focus of this chapter is to generalize the idea of mutual information maximization for active mapping from sampling-based methods to gradient-based optimization. Due to the discrete nature of voxel maps, the Shannon mutual information between an octree map and a sensing ray is non-differentiable with respect to sensor pose; hence only admitting sampling-based optimization methods. We propose an approximation of the Shannon mutual information based on spatial averaging of the information values around a given sensor pose. The averaging kernel is differentiable, therefore it allows computing gradients of information with respect to a sensor pose with 6-degrees of freedom. As a result, a collision- and occlusion-aware active mapping objective function is presented that is optimized via gradient-ascent in the continuous space of sensor poses. Unlike the sampling-based approaches, our gradient-based optimization has linear, instead of exponential, complexity with respect to the number of exploring robots. This makes our method suitable for extension to a team of robots, which is the focus of Chapter 6.

Chapter 6

This chapter delivers on the ultimate objective of the thesis, namely distributed multi-robot active multi-class mapping. The goal is design a decentralized exploration method that only includes point-to-point communication between the robots in a team, while requiring global consistency in the map estimation and coordination in exploration. We identify distributed mapping and distributed planning as two main components of distributed exploration. In particular, mapping can be formulated as maximization of sensor observation log-likelihood in the space of multi-class map probabilities, while planning deals with maximizing the collision- and occlusion-aware active mapping objective function of Chapter 5 in the space of robot poses. Both optimizations are instances of multi-agent Riemannian optimization. Hence, our first contribution in this chapter is a general distributed Riemannian optimization algorithm with consensus and optimality guarantees. Next, we apply our proposed distributed Riemannian optimization to both multi-robot mapping and multi-robot planning. Our complete solution enables distributed multi-robot active multi-class mapping with consensus in both map estimates and planned exploration paths, leading to a globally consistent map and coordination among all robots in the team. Lastly, we demonstrate simulated and real-world deployments of our distributed multi-robot exploration method.

Chapter 7

Finally we conclude this dissertation by providing a summary of contributions, and exploring possible extensions for future research endeavours.

Chapter 2

Background

The problem of active 3-D mapping of unknown environments involves concepts from a variety of topics. In its core, probabilistic mapping is an instance of Bayesian filtering, where new sensor measurements need to be systematically used to update a map estimate. Moreover, computing an informative sensing trajectories requires knowledge of how to quantify measurement information, which has been studies in the literature of information theory. Lastly, since we are interested in 3-D mapping using a general 6-degrees of freedom sensor pose model, we are required to use calculus and differentiation in matrix Lie groups. Hence, in order to better understand the relevance of the proposed work within the context of robot active mapping, this chapter reviews several theoretical concepts used in this dissertation.

2.1 Bayesian Inference

We begin our discussion of Bayesian inference by providing a short review of probability theory. Let Ω be the space possible experiment outcomes, formally known as the sample space. The set $\mathscr{F}$ of subsets of Ω that are closed under complementation and (countable) union is called a σ-algebra. A measure on the tuple $(\Omega, \mathscr{F})$ is a function $\mu : \mathscr{F} \to \mathbb{R}$ that satisfies:

1. **Non-negativity**: $\mu(A) \geq 0, \qquad \forall A \in F$ and $\mu() = 0$
2. **Countable additivity**: $\mu(\bigcup_i A_i) = \sum_i \mu(A_i)$ for countable numbers of sets $A_i \in \mathscr{F}$ such that $A_i \cap A_j, \ \forall i, j$

A probability measure $\mathbb{P} : \mathscr{F} \to [0,1]$ is a measure on $(\Omega, \mathscr{F})$ that satisfies $\mathbb{P}(\Omega) = 1$. The probability of an event $A \subset \Omega$ is then defined as $P(A)$. An important concept in probabilistic inference is *conditional probability*, expressed as:

$$\mathbb{P}(A \cap B) = \mathbb{P}(A|B)\mathbb{P}(B), \qquad A, B \in \mathscr{F}.$$

Moreover, the *law of total probability* relates conditional probabilities to the marginal probability of an event. Let $\{A_i\}_{i=1}^{n}$ be a partition of Ω, then the law of total probability states:

$$\mathbb{P}(B) = \sum_{i=1}^{n} \mathbb{P}(B \cap A_i).$$

Another important idea in probability theory is notion of *independent events*. The random events $A, B \subset \Omega$ are considered independent if observing one does not reveal any information about the other. Formally:

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B).$$

It is worth mentioning that independent events should not be confused with disjoint events, where observing one indicates that the other has not occurred.

A fundamental concept in probability is the notion of *random variable*. A random variable is a function $X : \Omega \to \mathscr{R}$ that maps an experiment outcome $\omega \in \Omega$ to a realization r in the realization space $\mathscr{R}$. For discrete random variables $X : \Omega \to \mathbb{Z}$, a *probability mass function* (PMF) $p : \mathbb{Z} \to [0,1]$ can be defined, which represents the probability measure of an outcome $x \in \mathbb{Z}$:

$$p(x) = \mathbb{P}(X = x),$$
$$p(x) \geq 0, \quad \sum_{x \in \mathbb{Z}} p(x) = 1.$$

Similarly, one can define a *probability density function* (PDF) $p : \mathscr{R} \to \mathbb{R}_{\geq 0}$ for continuous

random variables such that:

$$\lim_{\varepsilon \to 0} \int_{x}^{x+\varepsilon\delta x} p(s)ds = \mathbb{P}\{X = x\},$$

$$p(x) \geq 0, \quad \int_{x \in \mathbb{Z}} p(x)dx = 1.$$

The conditional probability and the law of total probability apply to PMFs and PDFs as well. Now we have the preliminary tools to discuss Bayesian inference.

Bayesian inference is a statistical methodology to update the probability of a hypothesis based on new evidence. Centered around the Bayes rule, this approach provides a framework for incorporating prior knowledge and refining predictions as new data becomes available. Let m and z be two random variables. The Bayes rule is expressed as:

$$p(m|z) = \frac{p(z|m)p(m)}{p(z)},$$

where each term is defined as follows.

- The prior $p(m)$ represents the initial belief about the hypothesis before any data is observed. This can be based on historical data, expert knowledge, or previous studies.

- The likelihood $p(z|m)$ quantifies how probable the observed data z is, given that hypothesis m holds true. This requires a model that describes the relationship between the hypothesis and the data.

- The marginal probability $p(z)$ is the total probability of observing data z under all possible hypotheses. This is more clearly stated using the law of total probability:

$$p(z) = \int p(z|m)p(m)dm.$$

- The posterior probability distribution $p(m|z)$ reflects the revised belief about the hypothesis m after considering the new data z.

The posterior distribution provides a probabilistic framework for making inferences about the hypothesis. Predictions and decisions can be made based on this updated distribution, accounting for both prior knowledge and new evidence. Bayesian inference provides a flexible process for incorporating various sources of prior knowledge and adapt to new data, while the probabilistic representation allows quantifying uncertainty of any predictions. Bayesian inference is widely used in diverse fields such as machine learning, epidemiology, finance, and engineering. It underpins algorithms for classification, regression, and probabilistic modeling, among other applications. In machine learning, for instance, Bayesian methods can be used to build robust models that update predictions as more data becomes available. Overall, Bayesian inference offers a powerful and principled approach to statistical reasoning, allowing for dynamic updating of beliefs in light of new evidence.

2.2 Log-Odds Mapping

Log-odds occupancy mapping is a technique used in robotics and computer vision to estimate the probability of occupancy in a given area of a map or environment. Introduced by Moravec and Elfes [95, 106], this method is particularly useful in applications such as simultaneous localization and mapping (SLAM) and robot navigation. The log-odds representation provides a computationally efficient and numerically stable way to manage and update spatial probability distributions. Here, we provide a short introduction to the log-odds occupancy mapping algorithm of [143, Ch. 9].

In order to perform occupancy mapping, the environment is discretized into a grid where each cell represents a small area of the map. Each cell in the grid is associated with an occupancy probability, indicating the likelihood that the cell is occupied by an obstacle or belongs to the free space. The occupancy probability of each cell m is updated based on sensor observations z

and robot position x, following the Bayes rule:

$$p(m|z,x) = \frac{p(z|m,x)p(m)}{p(z|x)},$$

where the likelihood $p(z|m,x)$ is called the observation model. However, directly using probabilities can lead to computational inefficiencies and numerical instabilities due to the need for deriving the normalizing term $p(z|x)$. Since each map cell m can only take one of the two states of *free* or *occupied*, one can divide the two sides of the Bayes rule in order to remove the normalizing term $p(z|x)$:

$$\frac{p(m = occupied|z,x)}{p(m = free|z,x)} = \frac{p(z|m = occupied,x)p(m = occupied)}{p(z|m = free,x)p(m = free)},$$

which can be re-written as:

$$\frac{p(m = occupied|z,x)}{1 - p(m = occupied|z,x)} = \frac{p(z|m = occupied,x)}{1 - p(z|m = occupied,x)} \frac{p(m = occupied)}{1 - p(m = occupied)}. \tag{2.1}$$

The probability p of a cell being occupied is transformed into a log-odds value using the $\text{logit}(p)$ function:

$$\text{logit}(p) = \log \frac{p}{1-p}.$$

Taking the logarithm of the two sides in (2.1) leads to:

$$\text{logit}\Big(p(m = occupied|z,x)\Big) = \text{logit}\Big(p(z|m = occupied,x)\Big) + \text{logit}\Big(p(m = occupied)\Big). \tag{2.2}$$

Based on (2.2), if a sensor detects the presence of an obstacle in a specific cell, the log-odds value for that cell is incremented. Conversely, if the cell is observed as free space, the log-odds value is decremented. Hence, this transformation simplifies the calculation of updates and fusion of information, and allows for straightforward integration of multiple observations. Additionally,

the log-odds representation helps to avoid issues with very small or very large probability values, which can cause computational instability. Once updates are applied, the log-odds value can be converted back to a probability value for interpretation or further processing using the sigmoid function:

$$p = \frac{1}{1 + \exp\left(-\text{logit}(p)\right)}.$$

2.3 Information Theory

Before delving into the main problem of this chapter, we would like to provide a review of information theory, as it is foundational to the contribution of our work.

Information theory, founded by Shannon in 1948 [133], is a branch of applied statistics that focuses on quantifying information. At its core, it provides a framework for understanding the transmission, processing, and storage of information. A fundamental concept in information theory is *entropy*, that quantifies the uncertainty or unpredictability of a random variable. Intuitively, it measures the average amount of "surprise" or "newness" associated with a set of outcomes. Formally, if m is a discrete random variable with a PMF $p(m)$, the entropy $H(m)$ is defined as:

$$H(m) = -\sum_{m \in M} p(m) \log p(m),$$

where M is the set of possible outcomes of m. Entropy is expressed in bits (or nats if using base e logarithms) and reflects the average information content of the random variable m. Furthermore, *conditional entropy* measures the amount of uncertainty remaining about a random variable m given that another random variable z is known. The conditional entropy $H(m|z)$ is expressed as:

$$H(m|z) = -\sum_{m \in M} \sum_{z \in Z} p(m,z) \log p(m|z),$$

where $p(m|z)$ is the conditional probability of m given z, and $p(m,z)$ is the joint probability of m and z. Conditional entropy provides insight into the residual uncertainty about m after

accounting for the information provided by z. Most related to this chapter, *the Shannon mutual information* quantifies the amount of information one random variable m contains about another random variable z. It measures the reduction in uncertainty about m due to knowledge of z, by representing the extent to which m and z share information. Based on its definition, mutual information is derived as follows:

$$I(m;z) = H(m) - H(m|z) = H(z) - H(z|m).$$

This measure is symmetric, meaning $I(m;z) = I(z;m)$, and it is always non-negative. Mutual information provides a measure of the dependency between m and z, with higher values indicating stronger dependencies. In summary, information theory provides key tools for understanding and quantifying information: entropy measures uncertainty, conditional entropy measures uncertainty given another variable, and mutual information measures the shared information between variables.

2.4 Robot Pose Representation in 3-D Spaces

A pose in the 3-D space can be represented by the combination of a position and a rotation. A position is simply a vector in the $\mathbb{R}^3$ vector space. Rotation corresponds to the orientation of an object. However, unlike position, rotations do not belong to a vector space. Specifically, two consecutive rotation operations are not commutative. As a result, rotations form a *non-commutative group*. A group is a set G with an associated composition operator $\odot$ that satisfies the following four properties:

1. **Closure**: $a \odot b \in G, \quad \forall a, b \in G$
2. **Identity element**: There exists a unique $e \in G$ such that $e \odot a = a \odot e = a$
3. **Inverse element**: For any $a \in G$, there exists a unique $a^{-1} \in G$ such that $a \odot b = b \odot a = e$
4. **Associativity**: $(a \odot b) \odot c = a \odot (b \odot c), \quad \forall a, b, c \in G$

Rotations and poses in the 3-D space are special cases of matrix Lie groups, called the *special Orthogonal group* (*SO(3)*) and the *special Euclidean group* (*SE(3)*), respectively. In the following, we provide an overview of these two matrix Lie groups. For proofs and a more in-depth analysis, we refer the interested reader to [18, Chapter 7].

The Lie group *SO(3)* expresses 3-D rotations as matrices:

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} \;\middle|\; \mathbf{R}\mathbf{R}^\top = \mathbf{I},\ \det(\mathbf{R}) = 1 \right\},$$

where $\mathbf{I}$ is the identity matrix in $\mathbb{R}^{3\times 3}$. The orthogonality condition $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$ removes 6 degrees of freedom from the 9-parameter $\mathbf{R}$ matrix, while the positive unit determinant constraint assures maintaining the right-hand 3-D coordinate system, i.e. no reflection. The four Lie group properties of *SO(3)* are listed below:

1. **Closure**: If $\mathbf{R}_1, \mathbf{R}_2 \in SO(3)$ then $\mathbf{R}_1\mathbf{R}_2 \in SO(3)$
2. **Identity element**: $\mathbf{I}_{3\times 3} \in SO(3)$
3. **Inverse element**: $\mathbf{R}^{-1} = \mathbf{R}^\top \in SO(3)$
4. **Associativity**: $\mathbf{R}_1(\mathbf{R}_2\mathbf{R}_3) = (\mathbf{R}_1\mathbf{R}_2)\mathbf{R}_3 = \mathbf{R}_1\mathbf{R}_2\mathbf{R}_3$

Similarly, the space of 3-D poses is expressed via the Lie group *SE(3)*:

$$SE(3) = \left\{ \mathbf{X} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \;\middle|\; \mathbf{R} \in SO(3),\ \mathbf{p} \in \mathbb{R}^3 \right\}.$$

The four Lie group properties of *SE(3)* are shown as follows:

1. **Closure**: If $\mathbf{X}_1, \mathbf{X}_2 \in SE(3)$ then $\mathbf{X}_1\mathbf{X}_2 \in SE(3)$
2. **Identity element**: $\mathbf{I}_{4\times 4} \in SE(3)$
3. **Inverse element**: $\mathbf{X}^{-1} \in SE(3)$

4. **Associativity**: $\mathbf{X}_1(\mathbf{X}_2\mathbf{X}_3) = (\mathbf{X}_1\mathbf{X}_2)\mathbf{X}_3 = \mathbf{X}_1\mathbf{X}_2\mathbf{X}_3$

Each Lie group has an associated Lie algebra, which is a vector space corresponding to the tangent space of its Lie group at the identity element. A Lie algebra captures the local structure of a Lie group, and is instrumental in defining perturbations around an element in the group. The Lie algebra of *SO(3)* is given by:

$$\mathfrak{so}(3) = \left\{ \hat{\theta} \in \mathbb{R}^{3\times 3} \;\middle|\; \theta \in \mathbb{R}^3 \right\},$$

where:

$$\hat{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}.$$

Similarly, the Lie algebra for the *SE(3)* group is defined as:

$$\mathfrak{se}(3) = \left\{ \hat{\xi} \in \mathbb{R}^{4\times 4} \;\middle|\; \xi \in \mathbb{R}^6 \right\},$$

where:

$$\hat{\xi} = \begin{bmatrix} \rho \\ \theta \end{bmatrix}^\wedge = \begin{bmatrix} \hat{\theta} & \rho \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad \rho, \theta \in \mathbb{R}^3.$$

Note that the vector ξ is an element of $\mathbb{R}^6$, characterizing the 6 degrees of freedom for a pose in the 3-D space.

In general, matrix Lie groups can be related to their associated Lie algebra via the *exponential mapping* for matrices, expressed as:

$$\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \frac{\mathbf{A}^n}{n!},$$

where $\mathbf{A} \in \mathbf{R}^{l\times l}$ is a square matrix. The matrix logarithm is the (non-unique) inverse of the

matrix exponential:

$$\log(\mathbf{A}) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}(\mathbf{A}-\mathbf{I})^n}{n}.$$

For the cases of *SO(3)* and *SE(3)*, the above definitions have closed form expressions. A rotation $\mathbf{R} \in SO(3)$ is related to the vector $\theta \in \mathbb{R}^3$ as follows:

$$\mathbf{R} = \exp(\hat{\theta}) = \mathbf{I} + \left(\frac{\sin(\|\theta\|)}{\|\theta\|}\right)\hat{\theta} + \left(\frac{1-\cos(\|\theta\|)}{\|\theta\|^2}\right)\hat{\theta}^2,$$
$$\hat{\theta} = \log(\mathbf{R}) = \frac{\|\theta\|}{2\sin(\|\theta\|)}\left(\mathbf{R}-\mathbf{R}^\top\right), \qquad \|\theta\| = \arccos\left(\frac{\mathrm{tr}(\mathbf{R})-1}{2}\right).$$

Similarly, the exponential and logarithm mappings for *SE(3)* are defined as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \exp(\hat{\xi}) = \mathbf{I} + \hat{\xi} + \left(\frac{1-\cos(\|\theta\|)}{\|\theta\|^2}\right)\hat{\xi}^2 + \left(\frac{\|\theta\|-\sin(\|\theta\|)}{\|\theta\|^3}\right)\hat{\xi}^3,$$
$$\xi = \begin{bmatrix} \rho \\ \theta \end{bmatrix} = \log(\mathbf{X})^\vee = \begin{cases} \theta = \log(\mathbf{R}),\ \rho = J_L^{-1}(\theta)\mathbf{p}, & \text{if } \mathbf{R} \neq \mathbf{I}, \\ \theta = \mathbf{0},\ \rho = \mathbf{p}, & \text{if } \mathbf{R} = \mathbf{I}, \end{cases},$$

where $(\cdot)^\vee$ is the inverse of $(\cdot)^\wedge$ mapping, and the matrix $J_L(\theta)$ is the (left) Jacobian matrix discussed below.

The Jacobian matrix plays a key role in converting the translation part of an $\mathfrak{se}(3)$ element to its corresponding *SE(3)* pose. As we will see in Chapters 5 and 6, this quantity also plays an important role when it comes to computing derivatives in the *SE(3)* Lie group. The Jacobians for *SO(3)* are defined as:

$$J_L(\theta) = \mathbf{I} + \left(\frac{1-\cos(\|\theta\|)}{\|\theta\|^2}\right)\hat{\theta} + \left(\frac{\|\theta\|-\sin(\|\theta\|)}{\|\theta\|^3}\right)\hat{\theta}^2, \qquad J_R(\theta) = J_L(-\theta),$$

where $J_L(\theta)$ and $J_R(\theta)$ are called the left and right Jacobians, respectively. The Jacobians for

the *SE(3)* Lie group can be derived based on their *SO(3)* counterparts:

$$\mathscr{J}_L(\xi) = \begin{bmatrix} J_L(\theta) & Q_L(\xi) \\ \mathbf{0} & J_L(\theta) \end{bmatrix}, \qquad \mathscr{J}_R(\xi) = \begin{bmatrix} J_R(\theta) & Q_R(\xi) \\ \mathbf{0} & J_R(\theta) \end{bmatrix},$$

with $Q_L(\xi)$ and $Q_R(\xi)$ defined as:

$$Q_L(\xi) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \frac{\hat{\theta}^n \hat{\rho} \hat{\theta}^m}{(n+m+2)!}, \qquad Q_R(\xi) = Q_L(-\xi).$$

2.5 Riemannian Manifolds

In the previous section, we discussed the concept of a matrix Lie group as an example of a geometric space where vector space operations such as addition and multiplication are either not admissible, or not commutative. It turns out the matrix Lie groups belong to a more general set of geometries called *Riemannian manifolds*. This section offers the key insights related to Riemannian manifolds that are widely used in Chapter 6. See [22, Ch.3] for a more detailed study.

A smooth manifold $\mathscr{M}$ is a curved topological space that locally resembles the Euclidean space, such that one can apply differentiation and calculus in the vicinity of each element. A simple example of such spaces is the planet Earth; it looks like a flat plane from the perspective of an individual living on its surface, while in reality it is close to a sphere. The associated vector space at point $x \in \mathscr{M}$ is called the *tangent space* $T_x\mathscr{M}$ of $\mathscr{M}$ at x, in the sense that vectors in $T_x\mathscr{M}$ are tangent to $\mathscr{M}$ at x. A Riemannian manifold $\mathscr{M}$ is an smooth manifold that is equipped with a smooth, symmetric, bi-linear, and positive-definite inner product, also known as Riemannian metric. Let $u, v \in T_x\mathscr{M}$ be two tangent vectors at x, the Riemannian metric is then denoted as $\langle v, u \rangle_x \in \mathbb{R}$, which unlike the Euclidean case, depends on the choice of x. The

Riemannian metric induces a norm $\|\cdot\|_x : T_x\mathscr{M} \to \mathbb{R}$ defined as:

$$\|v\|_x = \sqrt{\langle v, v\rangle_x}, \quad v \in T_x\mathscr{M}.$$

The above definition allows generalizing the concept of Euclidean distance to curved spaces. Formally, let $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{R}^n$ be two points in the Euclidean space, and $\gamma : [a,b] \to \mathbb{R}^n$ be a curve connecting the two points, i.e. $\gamma(a) = \mathbf{p}_1$ and $\gamma(b) = \mathbf{p}_2$. The length of such curve is expressed as follows:

$$L(\gamma) = \int_a^b \|\dot{\gamma(t)}\| dt.$$

In Euclidean spaces, a straight line is the minimizer of the above quantity, and its length is defined as the distance between $\mathbf{p}_1$ and $\mathbf{p}_2$. In Riemannian manifolds, the distance is similarly defined as the length of the shortest curve between two elements:

$$d : \mathscr{M} \times \mathscr{M} \to \mathbb{R} \text{ such that } d(x,y) = \inf_{\gamma \in \Gamma} \int_a^b \|\dot{\gamma(t)}\|_{\gamma(t)} dt,$$

where Γ is the set of all curves connecting x and y, i.e. $\gamma(a) = x$ and $\gamma(b) = y$. The curve that satisfies the above minimization is called a *geodesic*. Furthermore, if the geodesic is unique for any pair of x and y in $\mathscr{M}$, the corresponding Riemannian manifold is called *geodesically convex*. Note that geodesics are not necessarily straight lines. Going back to our planet Earth example, the shortest path between two points on the globe is an arch within the semi-circle that contains the two points.

Similar to matrix Lie groups, Riemannian manifolds admit an exponential map that relates vectors in the tangent space to element on the manifold. The exponential $\mathrm{Exp}_x(\cdot) : T_x\mathscr{M} \to \mathscr{M}$ of a tangent vector $v \in T_x\mathscr{M}$ is defined to be $\gamma(1;x,v)$, where $\gamma(\cdot;x,v)$ is a geodesic that satisfies:

$$\gamma(0;x,v) = x, \qquad \dot{\gamma}(0;x,v) = v. \tag{2.3}$$

Intuitively, the exponential map $\mathrm{Exp}_x(v)$ starts from x, and traverses the manifold for 1 second along the geodesic with a velocity magnitude of $\|v\|_x\ m/s$, assuming the standard metric units. The inverse of the exponential map is denoted as $\mathrm{Exp}_x^{-1}(\cdot) : \mathscr{M} \to T_x\mathscr{M}$, which essentially obtains the tangent vector $v \in T_x\mathscr{M}$ corresponding to the geodesic between x and $\mathrm{Exp}x(v)$. The exponential map and its inverse can be thought as analogues of the addition and subtraction operations in the Euclidean space, respectively; although these mappings do not possess the same properties such as commutativity. More so, the exponential mapping in Riemannian manifolds is similar to the matrix exponential discussed in the previous section. In fact, the two exponential mappings are related by the following identity:

$$\mathrm{Exp}_{\mathbf{X}}(\xi) = \mathbf{X}\exp(\hat{\xi}), \qquad \mathbf{X} \in \mathit{SE(3)},\ \xi \in \mathbb{R}^6.$$

Relevant to this dissertation, we also need to discuss computing gradients in Riemannian manifolds. Let $f : \mathscr{M} \to \mathbb{R}$ be a smooth function. The Riemannian gradient of $f(x)$ is defined as the unique tangent vector $v \in T_x\mathscr{M}$ that satisfies:

$$Df(x)[v] = \langle \mathrm{grad}\, f(x), v \rangle_x,$$

where $Df(x) : T_x\mathscr{M} \to \mathbb{R}$ is the differential of f at x defined as the linear map:

$$Df(x)[v] = \lim_{t \to 0} \frac{f(\gamma(t;x,v)) - f(x)}{t},$$

with $\gamma(t;x,v)$ described in (2.3). The notion of Riemannian gradients allow us to perform gradient-based optimization on a Riemannian manifold.

Chapter 3

Bayesian Metric-Semantic Voxel Mapping

Accurate modeling, real-time understanding, and efficient storage of a robot's environment are key capabilities for autonomous operation. Occupancy grid mapping [54] is a widely used, simple, yet effective, technique for distinguishing between traversable and occupied space surrounding a mobile robot. However, as we delegate increasingly sophisticated tasks to autonomous robots, it is required to augment traditional geometric models with semantic information about the context and object-level structure of the environment.

This chapter considers the problem of metric-semantic 3-D mapping using streaming distance and object category observations. See Fig. 3.1 as an illustration of the sensor data available for mapping. Our approach presented in this chapter is based on the papers [8] and [6]. The main **contribution** is the introduction of a Bayesian multi-class octree map estimation procedure which maintains a probability distribution over semantic categories and updates it via a probabilistic range-category perception model. An octree map [70] is an extension of an occupancy grid, introducing adaptive resolution to improve the memory usage and computational cost of generating 3-D maps of large environments. Our proposed method enables real-time large-scale 3-D mapping via efficient octree ray-tracing and map compression. Furthermore, we demonstrate the accuracy and scalability of our mapping method in a variety of simulated and real-world experiments.

Figure 3.1. A robot autonomously explores an unknown environment using an RGBD sensor and a semantic segmentation algorithm.

3.1 Related Works

Dense volumetric environment representations such as occupancy grid maps [54, 142], multi-resolution hierarchical models [70, 144], and signed distance fields (SDF) [113, 128], provide valuable information to both human and autonomous robots, as evidenced by their utility in search and rescue [69], safe navigation [111], and terrain modeling [56]. Moreover, the inclusion of semantic information, such as in the metric-semantic SLAM methods of [123, 174, 175], allows robots to build more sophisticated world models by affording autonomous systems the ability to not only discern occupied from free space, but also to distinguish between the types of objects in their surroundings [136]. As evidence of their usefulness, recent frameworks have leveraged the power of Bayesian statistics to develop algorithms that build semantic environment representations that encode categorical (semantic) information using probabilistic methods which

naturally capture the uncertainty robots hold regarding their world [57, 157].

The main novelty of our method is the ability to deploy large-scale probabilistic metric-semantic maps on size, weight, and power-constrained robot platforms in real-time. This is primarily due to the fact that our mapping does not require any model training or optimization during robot operation, and the memory footprint is minimal thanks to the octree data structure. Our dense voxel mapping allows directly utilizing our method in safety-critical mobile robot applications such as navigation, while the Bayesian filtering framework for map estimation allows quantifying the uncertainty caused by sensing noise, classification domain gap, and localization error. Additionally, incorporating semantics in our map representation enables more sophisticated tasks such as traversability-aware navigation, object-level reasoning, and geometrical grounding of natural language.

3.2 Problem: Bayesian Metric-Semantic Voxel Mapping from RGBD Sensing

Consider a robot with pose $\mathbf{X}_t \in SE(3)$ at time t:

$$\mathbf{X}_t := \begin{bmatrix} \mathbf{R}_t & \mathbf{p}_t \\ \mathbf{0}^\top & 1 \end{bmatrix},$$

where $\mathbf{R}_t \in SO(3)$ is the robot orientation, $\mathbf{p}_t \in \mathbb{R}^3$ is the robot position, and τ is the time step. The robot is navigating in an environment consisting of a collection of disjoint sets $\mathscr{E}_k \subset \mathbb{R}^3$, each associated with a semantic category $k \in \mathscr{K} := \{0, 1, \ldots, K\}$. Let $\mathscr{E}_0$ denote free space, while each $\mathscr{E}_k$ for $k > 0$ represents a different category, such as building, vegetation, terrain (see Fig. 3.1).

We assume that the robot is equipped with a sensor that provides information about the distance to and semantic categories of surrounding objects along a set of rays $\{\eta_b\}_b$, where b is the ray index, $\eta_b \in \mathbb{R}^3$ with $\|\eta_b\|_2 = r_{max}$, and $r_{max} > 0$ is the maximum sensing range.

Definition 1. A *sensor observation* at time t from robot pose $\mathbf{X}_t$ is a collection $\mathscr{Z}_t := \{\mathbf{z}_{t,b}\}_b$ of range and category measurements $\mathbf{z}_{t,b} := (r_{t,b}, y_{t,b}) \in \mathbb{R}_{\geq 0} \times \mathscr{K}$, acquired along the sensor rays $\mathbf{R}_t \eta_b$ with $\eta_b \in \{\eta_b\}_b$ at robot position $\mathbf{p}_t$.

Such information may be obtained by processing the observations of an RGBD camera or a LiDAR with a semantic segmentation algorithm [97]. Fig. 3.1 shows an example where each pixel in the RGB image corresponds to one sensor ray η_b, while its corresponding values in the semantic segmentation and the depth images encode category $y_{t,b}$ and range $r_{t,b}$, respectively. Let $\mathbf{m}$ be a map of the environment modeled as a grid of cells $i \in \mathscr{I} := \{1, \ldots, N\}$, each labeled with a category $m_i \in \mathscr{K}$. In order to model noisy sensor observations, we consider a PDF $p(\mathscr{Z}_t \mid \mathbf{m}, \mathbf{X}_t)$. This observation model allows integrating the measurements into a probabilistic map representation using Bayesian updates. The goal is to construct a multi-class map $\mathbf{m}$ based on the labeled range measurements.

Problem 1. Let $p_t(\mathbf{m}) := p(\mathbf{m} \mid \mathscr{Z}_{1:t}, \mathbf{X}_{1:t})$ be the PMF of the map $\mathbf{m}$ given the robot trajectory $\mathbf{X}_{1:t}$ and observations $\mathscr{Z}_{1:t}$ up to time t. Given a new observation $\mathscr{Z}_{t+1}$ obtained from robot pose $\mathbf{X}_{t+1}$, find the Bayesian update to the map PMF:

$$p_{t+1}(\mathbf{m}) \propto p(\mathscr{Z}_{t+1} | \mathbf{m}, \mathbf{X}_{t+1}) p_t(\mathbf{m}). \tag{3.1}$$

We assume that the robot pose is known and omit the dependence of the map distribution and the observation model on it for brevity. In the next section, we generalize the log-odds occupancy mapping method discussed in Chapter 2 to multi-class voxel maps. In Sec. 3.4, we present a multi-class extension of the octomap [70] algorithm that enables real-time mapping of large 3-D environments. Finally, in Sec. 3.5, we demonstrate the performance of our approach in simulated and real-world experiments.

3.3 Bayesian Multi-Class Voxel Mapping

This section derives the Bayesian update in (3.1), using a *multinomial logit model* to represent the map PMF $p_t(\mathbf{m})$ where each cell m_i of the map stores the probability of object classes in $\mathscr{K}$. To ensure that the number of parameters in the model scales linearly with the map size N, we maintain a factorized PMF over the cells:

$$p_t(\mathbf{m}) = \prod_{i=1}^{N} p_t(m_i). \tag{3.2}$$

We represent the individual cell PMFs $p_t(m_i)$ over the semantic categories $\mathscr{K}$ using a vector of log odds:

$$\mathbf{h}_{t,i} := \begin{bmatrix} \log \frac{p_t(m_i=0)}{p_t(m_i=0)} & \cdots & \log \frac{p_t(m_i=K)}{p_t(m_i=0)} \end{bmatrix}^\top \in \mathbb{R}^{K+1}, \tag{3.3}$$

where the free-class likelihood $p_t(m_i = 0)$ is used as a pivot. Given the log-odds vector $\mathbf{h}_{t,i}$, the PMF of cell m_i may be recovered using the softmax function $\sigma : \mathbb{R}^{K+1} \mapsto \mathbb{R}^{K+1}$:

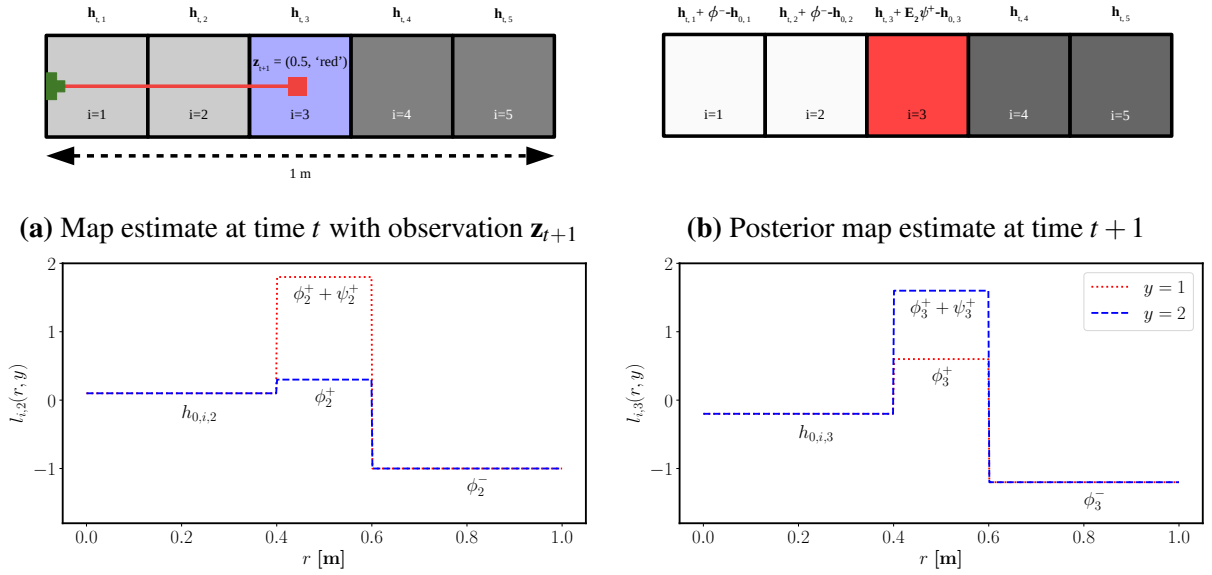
$$p_t(m_i = k) = \sigma_{k+1}(\mathbf{h}_{t,i}) := \frac{\mathbf{e}_{k+1}^\top \exp(\mathbf{h}_{t,i})}{\mathbf{1}^\top \exp(\mathbf{h}_{t,i})}, \tag{3.4}$$

where $\mathbf{e}_k$ is the standard basis vector with k-th element equal to 1 and 0 elsewhere, $\mathbf{1}$ is the vector with all elements equal to 1, and $\exp(\cdot)$ is applied element-wise to the vector $\mathbf{h}_{t,i}$. To derive Bayes rule for the log-odds $\mathbf{h}_{t,i}$, we need to specify an observation model for the range and category measurements.

Definition 2. The *inverse observation model* of a range-category measurement $\mathbf{z}$ obtained from robot pose $\mathbf{X}$ along sensor ray η with respect to map cell m_i is a PMF $p(m_i|\mathbf{z}; \mathbf{X}, \eta)$.

The Bayesian update in (3.1) for $\mathbf{h}_{t,i}$ can be obtained in terms of the range-category inverse observation model, evaluated at a new measurement set $\mathscr{Z}_{t+1}$.

Proposition 1. *Let $\mathbf{h}_{t,i}$ be the log odds of cell m_i at time t. Given sensor observation $\mathscr{Z}_{t+1}$, the*



(a) Map estimate at time t with observation $\mathbf{z}_{t+1}$

(b) Posterior map estimate at time $t+1$

(c) Inverse observation log-odds vector for $i = 3$

Figure 3.2. Illustration of the Bayesian multi-class mapping given a range-category observation $\mathbf{z}_{t+1} = (r_{t+1}, y_{t+1})$ for an environment with object classes of *white* (free class), *red*, and *blue*, encoded as $y = 0$, $y = 1$, and $y = 2$, respectively: (a) Portion of the map $\mathbf{m}$ along the observation $\mathbf{z}_{t+1}$. Each cell has a multi-class log-odds vector $\mathbf{h}_{t,i} \in \mathbb{R}^3$ for $i \in \{1, \ldots, 5\}$ at time t. The cell brightness encodes the occupancy probability, while the cell color represents the most likely category; (b) Map estimate at time $t+1$ after update with $\mathbf{z}_{t+1}$. Note how each multi-class log-odds vector changes based on the inverse observation model $\mathbf{l}_i(\mathbf{z}_{t+1})$; (c) Second and third elements of the inverse observation log-odds vector $\mathbf{l}_i(\mathbf{z}_{t+1})$ for $i = 3$ as a function of range r and category y in observation $\mathbf{z}_{t+1}$. Note that the first element of $\mathbf{l}_i(\mathbf{z}_{t+1})$ is always zero.

posterior log-odds are:

$$\mathbf{h}_{t+1,i} = \mathbf{h}_{t,i} + \sum_{\mathbf{z} \in \mathscr{Z}_{t+1}} (\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i}) \tag{3.5}$$

where $\mathbf{l}_i(\mathbf{z})$ *is the inverse observation model log odds:*

$$\mathbf{l}_i(\mathbf{z}) := \begin{bmatrix} \log \frac{p(m_i=0|\mathbf{z})}{p(m_i=0|\mathbf{z})} & \cdots & \log \frac{p(m_i=K|\mathbf{z})}{p(m_i=0|\mathbf{z})} \end{bmatrix}^\top . \tag{3.6}$$

Proof. See Appendix A. □

To complete the Bayesian multi-class mapping algorithm suggested by (3.5) we need a particular inverse observation model. When a sensor measurement is generated, the sensor ray

continues to travel until it hits an obstacle of category $\mathscr{K} \setminus \{0\}$ or reaches the maximum sensing range r_{max}. The labeled range measurement $\mathbf{z} = (r, y)$ obtained from position $\mathbf{p}$ with orientation $\mathbf{R}$ indicates that map cell m_i is occupied if the measurement end point $\mathbf{p} + \frac{r}{r_{max}}\mathbf{R}\eta$ lies in the cell. If m_i lies along the sensor ray but does not contain the end point, it is observed as free. Finally, if m_i is not intersected by the sensor ray, no information is provided about its occupancy. The map cells along the sensor ray can be determined by a rasterization algorithm, such as Bresenham's line algorithm [24]. We parameterize the inverse observation model log-odds vector in (3.6) as:

$$\mathbf{l}_i((r,y)) := \begin{cases} \phi^+ + \mathbf{E}_{y+1}\psi^+, & r \text{ indicates } m_i \text{ is occupied,} \\ \phi^-, & r \text{ indicates } m_i \text{ is free,} \\ \mathbf{h}_{0,i}, & \text{otherwise,} \end{cases} \tag{3.7}$$

where $\mathbf{E}_k := \mathbf{e}_k\mathbf{e}_k^\top$ and $\psi^+, \phi^-, \phi^+ \in \mathbb{R}^{K+1}$ are parameter vectors, whose first element is 0 to ensure that $\mathbf{l}_i(\mathbf{z})$ is a valid log-odds vector. This parameterization leads to an inverse observation model $p(m_i = k|\mathbf{z}) = \sigma_{k+1}(\mathbf{l}_i(\mathbf{z}))$, which is piece-wise constant along the sensor ray. Fig. 3.2 illustrates our Bayesian multi-class mapping method.

This section described how an observation affects the map PMF $p_t(\mathbf{m})$. Now, we switch our focus to proposing a multi-class version of the octomap technique, where map cells with equal multi-class probabilities can be compressed into a larger voxel.

3.4 Octree Compression

Utilizing a regular-grid discretization to represent a 3-D environment has prohibitive storage and computation requirements. Large continuous portions of many real environments are unoccupied, suggesting that adaptive discretization is significantly more efficient. Octomap [70] is a probabilistic 3-D mapping technique that utilizes an octree data structure to obtain adaptive resolution, e.g., combining many small cells associated with free space into few large cells. An

octree is a hierarchical data structure containing nodes that represent a section of the physical environment. Each node has either 0 or 8 children, where the latter corresponds to the 8 octants of the Euclidean 3-D coordinate system. Thus, the children of a parent node form an eight-way octant partition of the space associated with the parent node. Fig. 3.3 shows an example of a multi-class octree data structure.

We implement a *SemanticOcTreeNode* class as a building block of the multi-class octree structure. A *SemanticOcTreeNode* instance stores occupancy, color, and semantic information of its corresponding physical space, as shown in Fig. 3.3c. The most important data members of the *SemanticOcTreeNode* class are:

- *children*: an array of pointers to *SemanticOcTreeNode* storing the memory addresses of the 8 child nodes,
- *value*: a *float* variable storing the log-odds occupancy probability of the node,
- *color*: a *ColorRGB* object storing the RGB color of the node,
- *semantics*: a *SemanticLogOdds* object maintaining a categorical probability distribution over the semantic labels in the form of a log-odds ratio.

For performance reasons, the *SemanticLogOdds* class only stores the multi-class log-odds for the 3 most likely class labels, with each label represented by a unique RGB color. In this case, the log-odds associated with the rest of the labels lump into a single *others* variable. This relives the multi-class octree implementation from dependence on the number of labels that the object classifier can detect. Moreover, it significantly improves the speed of the mapping algorithm in cases with many semantic categories. See Sec. 3.5.3 for an analysis of mapping time versus the number of stored classes.

The implementation of the multi-class octree is completed by defining a *SemanticOcTree* class, which is derived from the *OccupancyOcTreeBase* class of the octomap library [70] and uses a *SemanticOcTreeNode* as its node type. Fig. 3.3b illustrates the derivation of the *SemanticOcTree*

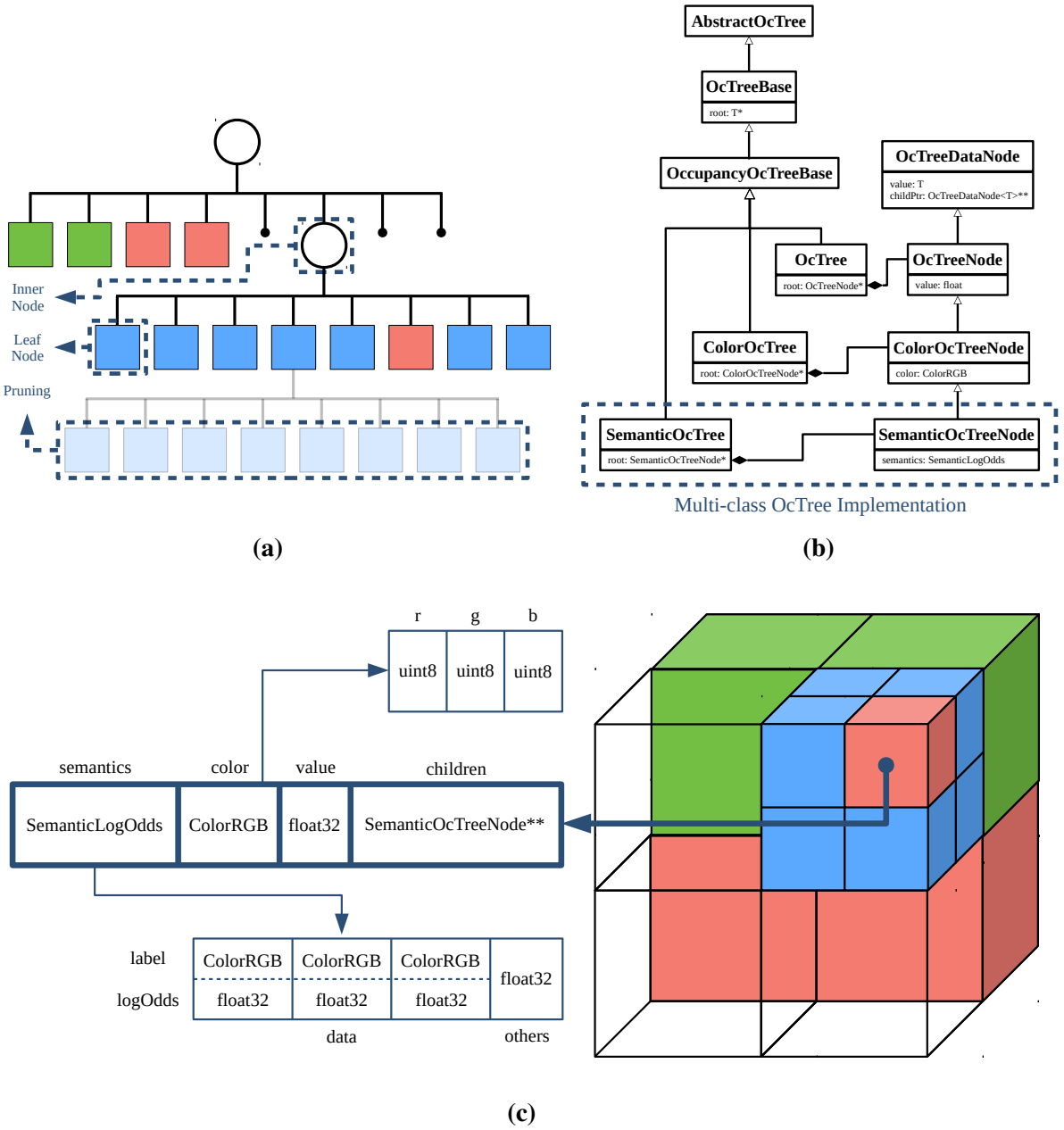


Figure 3.3. Semantic octree data structure: (a) A white circle represents an *inner* node such that its children collectively cover the same physical space as the inner node itself. A colored square represents a partition of the 3-D space where all downstream nodes contain identical semantic and occupancy values; therefore, they can be *pruned* into a *leaf* node. Lastly, black dots represent *unexplored* spaces of the environment. (b) UML diagram showing the class inheritance used for the implementation of a semantic octree. (c) geometric representation of the same octree as above with an overview of the *SemanticOcTreeNode* class.

Algorithm 1. Multi-Class Octree Update of Node $\mathbf{n}_i$

Input: octree node $\mathbf{n}_i$, observation $\mathbf{z} = (r, y)$, mixing coefficient α

$s = \mathbf{n}_i.semantics, \quad s.d = \mathbf{n}_i.semantics.data, \quad s.o = \mathbf{n}_i.semantics.others$
if $\mathbf{z}$ indicates free **then** Update s with ϕ^-
else if $\mathbf{z}$ indicates class y **then**
  if class y is among the 3 most likely classes in s **then** Update s with $\phi^+ + \mathbf{E}_{y+1}\psi^+$
  else
    $h_{aux} = s.o + \log\alpha$ ▷ Derive h_{aux} as a portion α of *others* class
    $s.o \mathrel{+}= \phi^+_{others} + \log(1-\alpha), \quad s_c = \text{CONCAT}(s.d, (y, h_{aux}))$
    Update s_c with $\phi^+ + \mathbf{E}_{y+1}\psi^+$
    Perform descending sort on s_c with respect to log-odds values
    $s.d = s_c[0:2]$ ▷ Pick 3 most likely classes
    $s.o = \log\Big(\exp(s_c[3]) + \exp(s.o)\Big)$ ▷ Combine the least likely class with *others* class
$s_f \leftarrow \min\left\{\max\left\{s_f, \underline{s}\right\}, \overline{s}\right\}$ ▷ Apply thresholds $\underline{s}$ and $\overline{s}$ for log-odds values
$\mathbf{n}_i.semantics = s$
return $\mathbf{n}_i$

and *SemanticOcTreeNode* classes as a UML diagram.

In order to register a new observation to a multi-class octree, we follow the standard ray-casting procedure over an octree, as in [70], to find the observed leaf nodes. Then, for each observed leaf node, if the observation demands an update, the leaf node is recursively expanded to the smallest resolution and the multi-class log-odds of the downstream nodes are updated using (3.5). At the ray's end point, which indicates an occupied cell, we also update the *color* variable by averaging the observed color with the current stored color of the corresponding node. Alg. 1 details the Bayesian update procedure for the multi-class octree.

To obtain a compressed octomap, it is necessary to define a rule for information fusion from child nodes towards parent nodes. Depending on the application, different information fusion strategies may be implemented. For example, a conservative strategy would assign the multi-class log-odds of the child node with the highest occupancy probability to the parent node. In our work, we simply assign the average log-odds vector of the child nodes to their parent node as shown in Alg. 2. The benefit of an octomap representation is the ability to combine similar cells (leaf nodes) into a large cell (inner node). This is called *pruning* the octree. Every time after

Algorithm 2. Semantic Fusion of Two Child Nodes

Input: octree nodes $\mathbf{n}_i$ and $\mathbf{n}_j$

$s_i = \mathbf{n}_i.semantics, \quad s_j = \mathbf{n}_j.semantics$

$\mathscr{K}_f = \text{UNIQUECLASS}(s_i, s_j)$ ▷ Non-repeating list of classes in s_i and s_j

$s_f = SemanticLogOdds()$ ▷ Object instantiation for the fused semantics

▷ Slice $s_i.o$ into smaller probabilities

$o_i = s_i.o - \log(1 + \mathscr{K}_f.size - s_i.d.size), \quad o_j = s_j.o - \log(1 + \mathscr{K}_f.size - s_j.d.size)$

for $y \in \mathscr{K}_f$ **do**

  if $y \notin s_i.d.label \wedge y \in s_j.d.label$ **then** $s_f.d.\text{APPEND}(y, \frac{o_i + s_j.d[y].logOdds}{2})$

  else if $y \in s_i.d.label \wedge y \notin s_j.d.label$ **then** $s_f.d.\text{APPEND}(y, \frac{s_i.d[y].logOdds + o_j}{2})$

  else $s_f.d.\text{APPEND}(y, \frac{s_i.d[y].logOdds + s_j.d[y].logOdds}{2})$

Perform descending sort on $s_f.d$ with respect to log-odds values

$expOthers = \exp(\frac{o_i + o_j}{2})$

for $i > 3$ **do** $expOthers \mathrel{+}= \exp(s_f.d[i].logOdds)$

$s_f.d[3 : end].\text{REMOVE}(), \quad s_f.o = \log(expOthers), \quad s_f \leftarrow \min\left\{\max\left\{s_f, \underline{s}\right\}, \overline{s}\right\}$

return s_f

an observation is integrated to the map, starting from the deepest inner node, we check for each inner node if 1) the node has 8 children, 2) its children do not have any children of their own, and 3) its children all have equal multi-class log-odds. If an inner node satisfies all of these three criteria, its children are pruned and the inner node is converted into a leaf node with the same multi-class log-odds as its children. This helps to compress the majority of the free cells into a few large cells, while the occupied cells usually do not undergo pruning since only their surfaces are observed by the sensor and their inside remains an unexplored region. Due to sensor noise, it is unlikely that cells belonging to the same class (e.g., free or occupied by the same obstacle) attain identical multi-class log-odds. Maximum and minimum limits for the elements of the multi-class log-odds are used so that each cell arrives at a stable state as its multi-class log-odds entries reach the limits. Stable cells are more likely to share the same multi-class probability distribution, consequently increasing the chance of octree pruning. However, thresholding causes loss of information near $p_t(m_i = k) = 1, k \in \mathscr{K}$ which can be controlled by the maximum and minimum limits.

3.5 Experiments

In this section, we utilize our proposed semantic octree mapping technique in simulated and real-world scenarios in order to demonstrate large-scale environment modeling. In Sec. 3.5.1 we use a 3-D simulation environment in order to generate sensor observations, while in Sec. 3.5.2 we test our mapping in a real environment using a ground wheeled robot. Finally, in Sec. 3.5.3, we investigate the influence of the number of stored semantic classes on mapping performance. An open-source implementation of semantic octree mapping is available on GitHub[1].

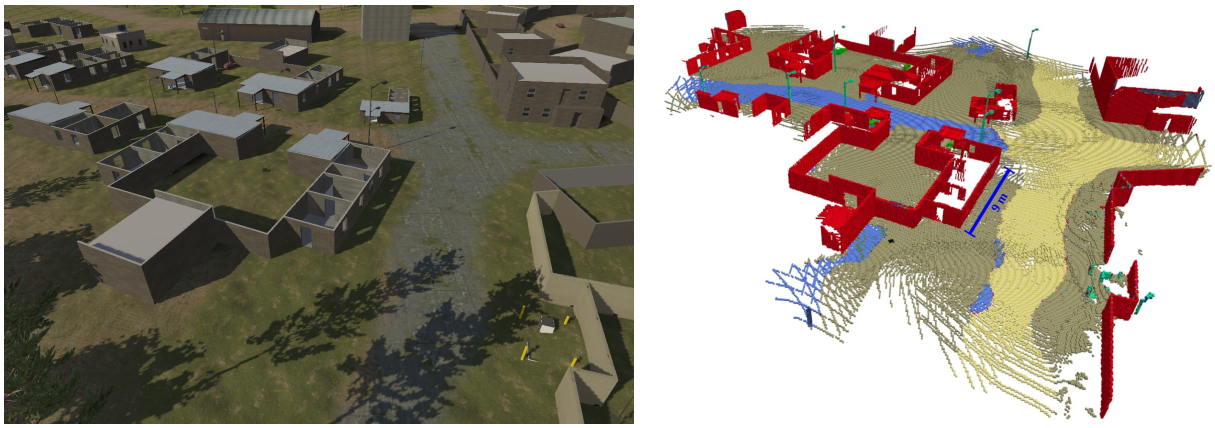
3.5.1 3-D Mapping in Unity Simulation

We evaluate semantic octree mapping in a photo-realistic 3-D Unity simulation, shown in Fig. 3.4a. We use a Husky robot equipped with an RGBD camera and run a semantic segmentation algorithm over the RGB images. We choose a voxel size of $0.5m$ in all experiments. The range measurements have an additive Gaussian noise of $\mathcal{N}(0, 0.1)$ and the semantic segmentation algorithm detects the true class with a probability of 0.95 while the misclassification happens uniformly in the pixel space. The robot autonomously navigates within the simulation environment using an exploration algorithm detailed in Chapter 4. Fig. 3.4b shows the estimated semantic octree map.

3.5.2 3-D Mapping in a Real-World Outdoor Environment

We deployed our semantic OcTree mapping approach on a Husky robot equipped with an Ouster OS1-32 LiDAR and an Intel RealSense D455 RGBD camera. Our software stack is implemented using the *Robot Operating System* (ROS) [120]. The LiDAR is used for localization via iterative closest point (ICP) scan matching [31]. A neural network based on a *FCHarDNet* architecture [32] and trained on the RUGD dataset [158] was used for semantic segmentation. The RGBD camera produces color and depth images with size 640×480 at 30 frames per second. The semantic segmentation algorithm takes a 2-D color image and outputs a semantic label for

[1] https://github.com/ExistentialRobotics/SSMI.



(a) Photo-realistic Unity simulation environment **(b)** Estimated multi-class occupancy map

Figure 3.4. Multi-class mapping in a simulated Unity environment. The robot is equipped with an RGBD sensor and runs semantic segmentation. Different colors represent different semantic categories (grass, dirt road, building, etc.).

each pixel in the image, at an average frame rate of 28.7 frames per second. By aligning the semantic image and the depth map, we derive a semantic 3-D point cloud which is utilized for Bayesian multi-class mapping. Our implementation was able to update the semantic octomap every 0.12 s, on average, while all of the computations where performed on the mobile robot. The experiment was carried out in an approximately 6 acre forested area shown in Fig. 3.5a. The environment contained various terrain features, including asphalt road, gravel, grass, densely forested areas, and hills. Additionally, a number of buildings and other structures such as bleachers, tents, and cars add to the diversity of the type of object categories within the locale. The robot was manually controlled via joystick, and traveled the path shown in Fig. 3.5a (left) while incrementally building the semantic octomap. Fig. 3.5b shows the semantic mapping result overlaying the satellite image obtained via 2-D projection of the semantic octomap. We computed the memory size of the semantic octomap, and compared it with the corresponding regular voxel grid representation, where each voxel contains the same amount of data as an OcTree leaf node at the lowest depth. Fig. 3.5c shows an almost five-fold saving in memory when using OcTree data structure. The importance of the memory savings of the octomap representation becomes more apparent when communication is considered. Our semantic octomap implementation resulted

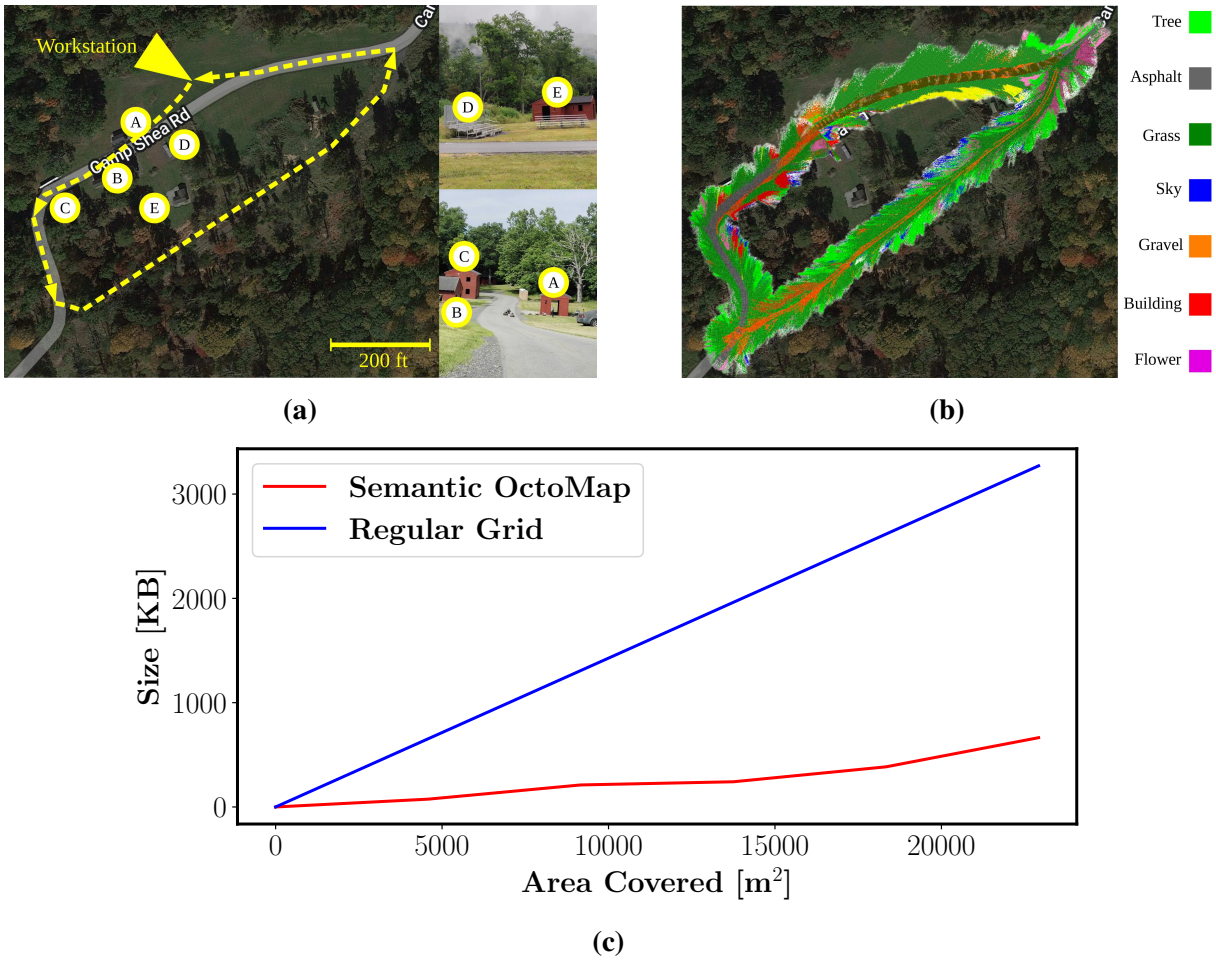


Figure 3.5. Outdoor semantic octree mapping experiment: (a) Satellite image of the experiment locale with robot trajectory shown in yellow and corresponding locations from the ground level point of view. (b) Semantic mapping output overlaying the satellite image. The map is obtained via 2-D projection of the 3-D map. (c) Memory use of regular grid vs. semantic octomap.

in a network bandwidth requirement of 238 KB/s for octomap, whereas a regular grid required 1173 KB/s for map communication.

3.5.3 Mapping Time vs. Number of Stored Classes

We analyse the influence of the number of stored classes in the semantic octomap on the mapping time. Let K_s denote the number of stored semantic classes. Alg. 1 has $O(K_s)$ memory and $O(K_s \log K_s)$ computational complexity (due to sorting in line 9). Furthermore, let p_{miss} be the misclassification probability assumed to be uniformly distributed among all incorrect classes.

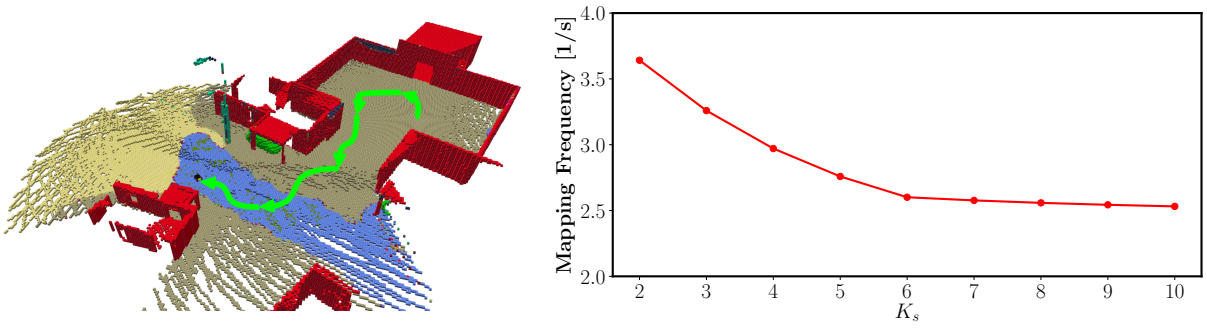


Figure 3.6. Mapping time vs. number of stored classes. Left: Robot trajectory (in green) used for all mapping frequency evaluations. Right: Average mapping frequency as a function of the number of stored semantic classes K_s.

Regarding accuracy, for a classifier with $p_{\text{miss}} < \frac{K-1}{K}$, where K is the number of all object classes, the true class will be always asymptotically recoverable as long as $K_s \geq 2$, thanks to the auxiliary *others* class that stores the accumulated probability of the $K - K_s$ least likely classes (see line 11 of Alg. 1). In general, K_s controls how fast the true class will be detected with the cost of additional memory use and computation. In order to quantitatively evaluate the effect of K_s on mapping time, we consider the same simulated Husky robot as Sec. 3.5.1 with a fixed trajectory, shown in Fig. 3.6 (left), and measure the mapping frequency as a function of K_s. Fig. 3.6 (right) shows the decrease in average mapping frequency as K_s increases. It is important to mention that the trajectory along which the data is collected only visits 6 object classes, which explains the change in slope for $K_s > 6$.

3.6 Summary

This chapter developed techniques for Bayesian multi-class mapping of 3-D environments using range and semantic segmentation observations. By utilizing the octree data structure, we implement a real-time, memory efficient, and yet accurate mapping technique which is capable of representing very large scenes. As demonstrated in simulated and real-world experiments, our probabilistic approach for semantic map estimation provides robustness against the domain shift caused by the difference between the classification training data and the testing environment.

Acknowledgments

Chapter 3, in part, is a reprint of the material as it appears in A. Asgharivaskasi and N. Atanasov, "Active Bayesian multi-class mapping from range and semantic segmentation observations," IEEE International Conference on Robotics and Automation (ICRA), pp. 1-7, 2021, and in A. Asgharivaskasi and N. Atanasov, "Semantic octree mapping and Shannon mutual information computation for robot exploration," IEEE Transactions on Robotics (T-RO), vol. 39, no.3, pp. 1910-1928, 2023. The dissertation author was the primary investigator and author of these papers.

Chapter 4

Information-Theoretic Exploration

Robots are increasingly expected to operate in unknown environments, with little to no prior information, in applications such as disaster response, environmental monitoring, and reconnaissance. Autonomous robot operation in unstructured and unknown environments requires efficient techniques for mapping and exploration using streaming range and visual observations. Information-based exploration techniques, such as Cauchy-Schwarz quadratic mutual information (CSQMI) [35] and fast Shannon mutual information (FSMI) [170], have successfully achieved active binary occupancy mapping with range measurements. However, as we envision robots performing complex tasks specified with semantically meaningful concepts, it is necessary to capture semantics in the measurements, map representation, and exploration objective. This chapter presents Semantic Shannon Mutual Information (SSMI) computation for robot exploration, based on the papers [8] and [6]. Our main **contribution** is derivation of a closed-form efficiently-computable lower bound of the Shannon mutual information between a multi-class octomap and a set of range-category measurements using semantic run-length encoding of the sensor rays. The bound allows rapid evaluation of many potential robot trajectories for autonomous exploration, while taking collision and visibility constraints into account. Unlike traditional class-agnostic exploration methods such as [35, 76, 170], our multi-class information measure captures the uncertainty of different semantic classes, leading to faster and more accurate exploration. Furthermore, the proposed approach relies on general range and class

measurements and general pose kinematics, making it suitable for either ground or aerial robots, equipped with either camera or LiDAR sensors, exploring either indoor or outdoor environments.

4.1 Related Works

Frontier-based exploration [161] is a seminal work that highlighted the utility of autonomous exploration and active mapping. It inspired methods [26, 63] that rely on geometric features, such as the boundaries between free and unknown space (frontiers) and the volume that would be revealed by new sensor observations. Due to their intuitive formulation and low computational requirements, geometry-based exploration methods continue to be widely employed in active perception. Recent works include semantics-assisted indoor exploration [62], hex-decomposition-based coverage planning [77], and Laplace potential fields for safe outdoor exploration [93]. More related to this chapter, receding-horizon "next-best-view" planning [20] presents an active octree occupancy mapping method which executes trajectories built from a random tree whose quality is determined by the amount of unmapped space that can be explored. Similarly, the graph-based exploration methods of [45] and [173] use local random trees in free space to sample candidate viewpoints for exploration, while a global graph maintains the connections among the frontiers in the map. Cao *et al.* [28, 77] introduced hierarchical active 3D coverage and reconstruction which computes a coarse coverage path at the global scale followed by a local planner that ensures collision avoidance via a high-resolution path. As shown by Corah and Michael [42], coverage-based exploration strategies can be formulated as mutual-information maximization policies in the absence of sensor noise. However, in many real-world circumstances sensor measurements are corrupted by non-trivial noise, reducing the effectiveness of geometric exploration methods that do not capture probabilistic uncertainty. For example, due to the domain shift between the training data and the test environment, utilizing a pre-trained semantic segmentation model in the mapping process requires accounting for measurement uncertainty in the exploration policy.

The work by Elfes [55] is among the first to propose an information-based utility function for measuring and minimizing map uncertainty. Information-based exploration strategies have been devised for uncertainty minimization in robot localization or environment mapping [23, 100, 153]. Information-theoretic objectives, however, require integration over the potential sensor measurements, limiting the use of direct numerical approximations to short planning horizons. Kollar and Roy [83] formulated active mapping using an extended Kalman filter and proposed a local-global optimization, leading to significant gains in efficiency for uncertainty computation and long-horizon planning. Unlike geometry-based approaches, information-theoretic exploration can be directly formulated for active simultaneous localization and mapping (SLAM) [13, 29, 30, 154], aiming to determine a sensing trajectory that balances robot state uncertainty and visitation of unexplored map regions. Stachniss *et al.* [138] approximate information gain for a Rao-blackwellized particle filter over the joint state of robot pose and map occupancy. Julian *et al.* [76] prove that, for range measurements and known robot position, the Shannon mutual information is maximized over trajectories that visit unexplored areas. However, without imposing further structure over the observation model, computing the mutual information objective requires numerical integration. The need for efficient mutual information computation becomes evident in 3-D environments. Cauchy-Schwarz quadratic mutual information (CSQMI) [35] and fast Shannon mutual information (FSMI) [170] offer efficiently computable closed-form objectives for active occupancy mapping with range measurements. Henderson *et al.* [68] propose an even faster computation based on a recursive expression for Shannon mutual information in continuous maps.

The recent success of machine learning methods of perception has motivated learning autonomous exploration policies. Chen *et al.* [36] attempt to bridge the sim2sim and sim2real gaps via graph neural networks and deep reinforcement learning. This enables decision-making over graphs containing relevant exploration information which is provided by human experts in order to predict a robot's optimal sensing action in belief space. Lodel *et al.* [90] introduce a deep reinforcement learning policy which recommends next best view that maximizes information

gain via defining mutual information as the training reward. Zwecher *et al.* [177] employ deep reinforcement learning to find an exploration policy that plans collision-free coverage paths, while another neural network provides a predicted full map given the partially observed environment. Zhang *et al.* [166] propose a multi-agent reinforcement learning exploration method where regions of interest, free space, and robots are represented as graph nodes, and hierarchical-hops graph neural networks (H2GNN) are used to identify key information in the environment. Related to multi-robot exploration, the authors in [71] utilize an actor-critic strategy to map an unknown environment, where Voronoi partitioning divides the exploration regions among the robots. As this chapter demonstrates, incorporating semantic uncertainty in addition to geometric information in the exploration process can be beneficial. Additionally, using Shannon mutual information as an objective function may help train more generalizable exploration policies because it mitigates the need for training sensor-specific models. Hence, the techniques proposed in our work are complementary to learning approaches and can provide robustness to measurement uncertainty and domain shift caused by sensor and operational condition variations.

Active semantic mapping has recently attracted much attention due to the proliferation of fast object detection and semantic segmentation algorithms implemented on mobile robot platforms. The authors in [152] use a two-layer architecture, where the knowledge representation layer provides a belief over the environment state to the action layer, which subsequently chooses an action to gather information or execute a task. The work in [140] presents a semantic exploration policy which takes an occluded semantic point cloud of an object, finds a match in a database to estimate the full object dimensions, and then generates candidate next observation poses to reconstruct the object. The next best view is computed via a volumetric information gain metric that computes visible entropy from a candidate pose. The semantic map used in this paper is a collection of bounding boxes around objects. Active semantic mapping has also been employed to develop sample-efficient deep learning methods. Blum *et al.* [21] propose an active learning method for training semantic segmentation networks where the novelty (epistemic uncertainty) of the input images is estimated as the distance from the training data in the

embedding space, while a path planning method maximizes novelty of future input images along the planned trajectory, assuming novel images are spatially correlated. Georgakis *et al.* [59] actively train a hierarchical semantic map generation model that predicts occupancy and semantics given occluded input. The authors use an ensemble of map generation models in order to predict epistemic uncertainty of the predicted map. The uncertainty is then used to choose trajectories for actively training the model with new images that differ the most with the training data of the current model. The work presented in this chapter distinguishes itself from the aforementioned works by introducing a dense Bayesian multi-class mapping with a closed-form uncertainty measure, as opposed to sampling-based uncertainty estimation. Moreover, our information-theoretic objective function,named SSMI, directly models sensor noise specifications, unlike volumetric information gain.

Our method distinguishes itself from the aforementioned works by introducing a closed-form uncertainty measure over a dense Bayesian multi-class map, as opposed to sampling-based uncertainty estimation. Moreover, our information-theoretic objective function directly models sensor noise specifications, unlike volumetric information gain. The work presented in this chapter is most related to CSQMI [35] and FSMI [170] in that it develops a closed-form expression for mutual information. However, instead of a binary map and range-only measurements, our formulation considers the multi-class octree map introduced in Chapter 3, built from range and category observations. Since the same occupancy map can be derived from many different multi-class maps, the information associated with various object classes will fail to be captured if we solely rely on occupancy information, as the case in CSQMI and FSMI. Therefore, we expect to perform exploration more efficiently by using the multi-class perception model, and consequently, expanding the notion of uncertainty to multiple classes.

4.2 Problem: Shannon Mutual Information between a Multi-Class Map and Range-Category Observations

Consider a robot with pose $\mathbf{X}_t \in SE(3)$ at time t and deterministic discrete-time kinematics:

$$\mathbf{X}_t := \begin{bmatrix} \mathbf{R}_t & \mathbf{p}_t \\ \mathbf{0}^\top & 1 \end{bmatrix}, \qquad \mathbf{X}_{t+1} = \mathbf{X}_t \exp(\tau \hat{\mathbf{u}}_t), \tag{4.1}$$

where $\mathbf{u}_t := [\mathbf{v}_t^\top, \boldsymbol{\omega}_t^\top]^\top \in \mathscr{U} \subset \mathbb{R}^6$ is the control input, consisting of linear velocity $\mathbf{v}_t \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$. Moreover, consider the range-category observation model and Bayesian mapping defined in Sec. 3.2 of Chapter 3, used to estimate a multi-class voxel map via the update rule of (3.1). We consider the following problem.

Problem 2. Given a prior map PMF $p_t(\mathbf{m})$ at time t and a finite planning horizon T, maximize the ratio:

$$\max_{\mathbf{u}_{t:t+T-1}} \frac{I(\mathbf{m}; \mathscr{Z}_{t+1:t+T} \mid \mathscr{Z}_{1:t})}{J(\mathbf{X}_{t:t+T-1}, \mathbf{u}_{t:t+T-1})} \text{ subject to } (4.1), (3.1), \tag{4.2}$$

of the mutual information $I(\mathbf{m}; \mathscr{Z}_{t+1:t+T} \mid \mathscr{Z}_{1:t})$ between the map $\mathbf{m}$ and future sensor observations $\mathscr{Z}_{t+1:t+T}$ to the motion cost $J(\mathbf{X}_{t:t+T-1}, \mathbf{u}_{t:t+T-1})$ of the control sequence $\mathbf{u}_{t:t+T-1}$.

The definitions of the mutual information and motion cost terms in (4.2) are:

$$\begin{aligned} I(\mathbf{m}; \mathscr{Z}_{t+1:t+T} | \mathscr{Z}_{1:t}) &:= \sum_{\mathbf{m} \in \mathscr{K}^N} \int \cdots \int p(\mathbf{m}, \mathscr{Z}_{t+1:t+T} | \mathscr{Z}_{1:t}) \\ &\qquad \times \log \frac{p(\mathbf{m}, \mathscr{Z}_{t+1:t+T} \mid \mathscr{Z}_{1:t})}{p(\mathbf{m} | \mathscr{Z}_{1:t}) p(\mathscr{Z}_{t+1:t+T} | \mathscr{Z}_{1:t})} \prod_{\tau=1}^{T} \prod_{b} d\mathbf{z}_{t+\tau,b}, \\ J(\mathbf{X}_{t:t+T-1}, \mathbf{u}_{t:t+T-1}) &:= q(\mathbf{X}_{t+T}) + \sum_{\tau=0}^{T-1} c(\mathbf{X}_{t+\tau}, \mathbf{u}_{t+\tau}), \end{aligned} \tag{4.3}$$

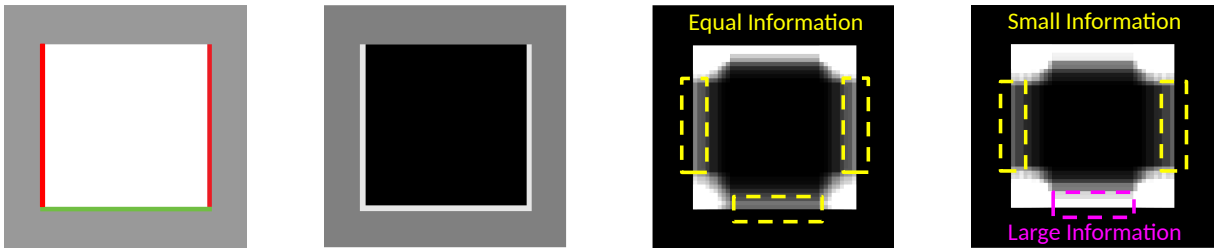
where the integration in (4.3) is over all possible values of all sensor beams over all times $\mathbf{z}_{t+\tau,b}$, and the strictly positive terms $q(\mathbf{X})$ and $c(\mathbf{X}, \mathbf{u})$ model terminal and stage motion costs (e.g., distance traveled, elapsed time), respectively.

We derive an efficient approximation to the mutual information term in Sec. 4.3. In Sec. 4.4, we present a fast computation of mutual information over a semantic octree map using run-length encoding. This allows autonomous exploration of large 3-D environments by rapidly evaluating potential robot trajectories online and (re-)selecting the one that maximizes the objective in (4.2). Finally, in Sec. 4.5, we demonstrate the performance of our approach, and compare with state-of-the-art techniques in a comprehensive set of 2-D and 3-D experiments.

4.3 Semantic Shannon Mutual Information

Proposition 1 in Chapter 3 allows a multi-class formulation of occupancy grid mapping, where the uncertainty of a map cell depends on the probability of each class $p_t(m_i = k)$, instead of only the binary occupancy probability $1 - p_t(m_i = 0)$. Moreover, the inverse observation model in (3.7) may contain different likelihoods for the different classes which can be used to prioritize the information gathering for specific classes. Fig. 4.1a shows an example where the estimated map of an environment with 3 classes, *free*, $class_1$, $class_2$, contains two regions with similar occupancy probability but different semantic uncertainty. In particular, the red and green walls have the same occupancy probability of 0.9, as shown in Fig. 4.1b, but the red region more certainly belongs to $class_1$ and the green region has high uncertainty between the two classes. As can be seen in Fig. 4.1c, the mutual information associated with a binary occupancy map cannot distinguish between the red and green regions since they both have the same occupancy probability. In contrast, the multi-class map takes into account the semantic uncertainty among different categories, as can be seen in Fig. 4.1d where the uncertain green region has larger mutual information than the certain red region.

These observations suggest that more accurate uncertainty quantification may be achieved using a multi-class instead of a binary perception model, potentially enabling a more efficient exploration strategy. However, computing the mutual information term in (4.3) is challenging because it involves integration over all possible values of the observation sequence $\mathscr{Z}_{t+1:t+T}$.



(a) **(b)** **(c)** **(d)**

Figure 4.1. Comparison between the information surfaces of binary and multi-class map representations. (a) Environment with three classes *free*, $class_1$, and $class_2$ where the white and gray regions represent free and unknown space, respectively, with $\mathbf{p}^{white}(m_i) = [1,0,0]$ and $\mathbf{p}^{gray}(m_i) = [0.3,0.3,0.3]$. The red and green regions have the same occupancy probability of $p(m_i = \text{occupied}) = 0.9$ but different class uncertainty, i.e., $\mathbf{p}^{red}(m_i) = [0.1,0.8,0.1]$ and $\mathbf{p}^{green}(m_i) = [0.1,0.45,0.45]$. (b) Binary occupancy map, where the intensity of each pixel is proportional to its occupancy probability, regardless of object class. (c) Occupancy mutual information surface. (d) Semantic mutual information surface. Each pixel in the information surfaces shows the value of mutual information between the map and a set of range-category observations, uniformly sampled from a 360° field of view at each pixel location.

Our main result in this section is an efficiently-computable lower bound on $I(\mathbf{m}; \mathscr{Z}_{t+1:t+T} | \mathscr{Z}_{1:t})$ for range-category observations $\mathscr{Z}_{t+1:t+T}$ and a multi-class occupancy map $\mathbf{m}$.

To compute the mutual information between an observation sequence $\mathscr{Z}_{t+1:t+T}$ and the map $\mathbf{m}$, we need the PDF of a range-category measurement $\mathbf{z}_{\tau,b} \in \mathscr{Z}_{t+1:t+T}$ conditioned on $\mathscr{Z}_{1:t}$. Let $\mathscr{R}_{\tau,b}(r) \subset \mathscr{I}$ denote the set of map cell indices along the ray $\mathbf{R}_\tau \eta_b$ from robot position $\mathbf{p}_\tau$ with length r. Let $\gamma_{\tau,b}(i)$ denote the distance traveled by the ray $\mathbf{R}_\tau \eta_b$ within cell m_i and $i^*_{\tau,b} \in \mathscr{R}_{\tau,b}(r)$ denote the index of the cell hit by $\mathbf{z}_{\tau,b}$. We define the PDF of $\mathbf{z}_{\tau,b} = (r,y)$ conditioned on $\mathscr{Z}_{1:t}$ as:

$$p(\mathbf{z}_{\tau,b} | \mathscr{Z}_{1:t}) = \frac{p_t(m_{i^*_{\tau,b}} = y)}{\gamma_{\tau,b}(i^*_{\tau,b})} \prod_{i \in \mathscr{R}_{\tau,b}(r) \setminus \{i^*_{\tau,b}\}} p_t(m_i = 0). \tag{4.4}$$

This definition states that the likelihood of $\mathbf{z}_{\tau,b} = (r,y)$ at time t depends on the likelihood that the cells m_i along the ray $\mathbf{R}_\tau \eta_b$ of length r are empty and the likelihood that the hit cell $m_{i^*_{\tau,b}}$ has class y. A similar model for binary observations has been used in [35, 76, 170]. Moreover, we select a subset $\underline{\mathscr{Z}}_{t+1:t+T} = \{\mathbf{z}_{\tau,b}\}_{\tau=t+1,b=1}^{t+T,B}$ of the observations $\mathscr{Z}_{t+1:t+T}$ in which the sensor

rays are non-overlapping. Precisely, any pair of measurements $\mathbf{z}_{\tau,b}, \mathbf{z}_{\tau',b'} \in \underline{\mathscr{Z}}_{t+1:t+T}$ satisfies:

$$\mathscr{R}_{\tau,b}(r_{max}) \cap \mathscr{R}_{\tau',b'}(r_{max}) = \emptyset. \tag{4.5}$$

In practice, constructing $\underline{\mathscr{Z}}_{t+1:t+T}$ requires removing intersecting rays from $\mathscr{Z}_{t+1:t+T}$ to ensure that the remaining observations are mutually independent. The mutual information between $\mathbf{m}$ and $\underline{\mathscr{Z}}_{t+1:t+T}$ can be obtained as a sum of mutual information terms between single rays $\mathbf{z}_{\tau,b} \in \underline{\mathscr{Z}}_{t+1:t+T}$ and map cells m_i observed by $\mathbf{z}_{\tau,b}$. This idea is inspired by CSQMI [35] but we generalize it to multi-class observations and map.

Proposition 2. *Given a sequence of labeled range observations $\mathscr{Z}_{t+1:t+T}$, let $\underline{\mathscr{Z}}_{t+1:t+T} = \left\{\mathbf{z}_{\tau,b}\right\}_{\tau=t+1,b=1}^{t+T,B}$ be a subset of non-overlapping measurements that satisfy* (4.5)*. Then, the Shannon mutual information between $\mathscr{Z}_{t+1:t+T}$ and a multi-class occupancy map $\mathbf{m}$ can be lower bounded as:*

$$I(\mathbf{m}; \mathscr{Z}_{t+1:t+T} | \mathscr{Z}_{1:t}) \geq I(\mathbf{m}; \underline{\mathscr{Z}}_{t+1:t+T} | \mathscr{Z}_{1:t}) = \sum_{\tau=t+1}^{t+T} \sum_{b=1}^{B} \sum_{k=1}^{K} \sum_{n=1}^{N_{\tau,b}} p_{\tau,b}(n,k) C_{\tau,b}(n,k), \tag{4.6}$$

where $N_{\tau,b} := |\mathscr{R}_{\tau,b}(r_{max})|$,

$$\begin{aligned}
p_{\tau,b}(n,k) &:= p_t(m_{i^*_{\tau,b}} = k) \prod_{i \in \tilde{\mathscr{R}}_{\tau,b}(n) \setminus \{i^*_{\tau,b}\}} p_t(m_i = 0), \\
C_{\tau,b}(n,k) &:= f(\phi^+ + \mathbf{E}_{k+1}\psi^+ - \mathbf{h}_{0,i^*_{\tau,b}}, \mathbf{h}_{t,i^*_{\tau,b}}) + \sum_{i \in \tilde{\mathscr{R}}_{\tau,b}(n) \setminus \{i^*_{\tau,b}\}} f(\phi^- - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}), \\
f(\phi, \mathbf{h}) &:= \log\left(\frac{\mathbf{1}^\top \exp(\mathbf{h})}{\mathbf{1}^\top \exp(\phi + \mathbf{h})}\right) + \phi^\top \sigma(\phi + \mathbf{h}),
\end{aligned}$$

and $\tilde{\mathscr{R}}_{\tau,b}(n) \subseteq \mathscr{R}_{\tau,b}(r_{max})$ is the set of the first n map cell indices along the ray $\mathbf{R}_\tau \eta_b$, i.e., $\tilde{\mathscr{R}}_{\tau,b}(n) := \{i \mid i \in \mathscr{R}_{\tau,b}(r), |\mathscr{R}_{\tau,b}(r)| = n, r \leq r_{max}\}$.

Proof. See Appendix B. □

In (4.6), $p_{\tau,b}(n,k)$ represents the probability that the n-th map cell along the ray $\mathbf{R}_\tau \eta_b$

belongs to object category k while all of the previous cells are free. The function $f(\phi, \mathbf{h})$ denotes the log-ratio of the map PMF $\sigma(\mathbf{h})$ and its posterior $\sigma(\phi + \mathbf{h})$, averaged over object categories in $\mathscr{K}$ (see (B.3) in Appendix B for more details). As a result, $C_{\tau,b}(n,k)$ is the sum of log-ratios for the first n cells along the ray $\mathbf{R}_\tau \eta_b$ under the same event as the one $p_{\tau,b}(n,k)$ is associated with. Therefore, the lower bound $I(\mathbf{m}; \underline{\mathscr{Z}}_{t+1:t+T} | \mathscr{Z}_{1:t})$ is equivalent to the expectation of summed log-ratios $C_{\tau,b}(n,k)$ over all possible instantiations of the observations in $\underline{\mathscr{Z}}_{t+1:t+T}$.

Proposition 2 allows evaluating the informativeness according to (4.2) of any potential robot trajectory $\mathbf{X}_{t:t+T}$, $\mathbf{u}_{t:t+T-1}$. In order to perform informative planning, first, we identify the boundary between the explored and unexplored regions of the map, similar to [161]. This can be done efficiently using edge detection, for example. Then, we cluster the corresponding map cells by detecting the connected components of the boundary. Each cluster is called a *frontier*. A motion planning algorithm is used to obtain a set of pose trajectories to the map frontiers, determined from the current map PMF $p_t(\mathbf{m})$. Alg. 3 summarizes the procedure for determining a trajectory $\mathbf{X}^*_{t:t+T}$, $\mathbf{u}^*_{t:t+T-1}$ that maximizes the objective in (4.2), where $J(\mathbf{X}_{t:t+T-1}, \mathbf{u}_{t:t+T-1})$ is the length of the corresponding path. This kinematically feasible trajectory can be tracked by a low-level controller that takes the robot dynamics into account.

Evaluation of the mutual information lower bound in Proposition 2 can be accelerated without loss in accuracy for map cells along the observation rays that contain equal PMFs. In the next section, we investigate this property of the proposed lower bound within the context of the semantic octree representation introduced in the previous chapter. In particular, a fast semantic mutual information formula is presented based on compression of range-category ray-casts over octree representations.

Algorithm 3. Information-Theoretic Path Planning

Input: robot pose $\mathbf{X}_t$, map estimate $p_t(\mathbf{m})$

1: $\mathscr{F} = \text{FINDFRONTIERS}(p_t(\mathbf{m}))$
2: **for** $f \in \mathscr{F}$ **do**
3: $\quad \mathbf{X}_{t+1:t+T}, \mathbf{u}_{t:t+T-1} = \text{PLANPATH}(\mathbf{X}_t, p_t(\mathbf{m}), f)$
4: $\quad$ Compute (4.2) over $\mathbf{X}_{t:t+T}, \mathbf{u}_{t:t+T-1}$ via (4.6)
5: **return** $\mathbf{X}^*_{t:t+T}$, $\mathbf{u}^*_{t:t+T-1}$ with the highest value

4.4 Semantic Ray-Length Encoding for Octree Representations

A ray cast through an octree representation may visit several large cells within which the class probabilities are homogeneous. We exploit this property to obtain the mutual information between a multi-class octomap and a single ray as a summation over a subset of octree leaf nodes instead of individual map cells. This simplification provides a significant performance gain with no loss of accuracy. The following formulation can be considered a multi-class generalization of the run-length encoding technique introduced by [170], using the mutual information lower bound in (4.6) and the multi-class OcTree defined in Chapter 3.

Suppose that the map cells along a single beam $\mathbf{R}_\tau \eta_b$ have piecewise-constant multi-class probabilities such that the set $\{m_i \mid i \in \mathscr{R}_{\tau,b}(r_{max})\}$ can be partitioned into $Q_{\tau,b}$ groups of consecutive cells indexed by $\mathscr{R}^q_{\tau,b}(r_{max})$, $q = 1, \ldots, Q_{\tau,b}$, where:

$$p_t(m_i = k) = p_t(m_j = k), \quad \forall i, j \in \mathscr{R}^q_{\tau,b}(r_{max}), \quad \forall k \in \mathscr{K}\,. \tag{4.7}$$

In this case, the log-odds probabilities encountered by a ray cast can be compressed using semantic run-length encoding, defined as below.

Definition 3. A *semantic run-length encoding* (SRLE) of a ray $\mathbf{R}_\tau \eta_b$ cast through a multi-class octree is an ordered list of tuples of the form $[(\omega_{\tau,b,q}, \chi_{t,q})]_{q=1}^{Q_{\tau,b}}$, where $\omega_{\tau,b,q}$ and $\chi_{t,q}$ respectively represent the width and the log-odds vector of the intersection between the ray and the cells in $\mathscr{R}^q_{\tau,b}(r_{max})$. The width $\omega_{\tau,b,q}$ is the number of octree *elements* along the ray intersection, where

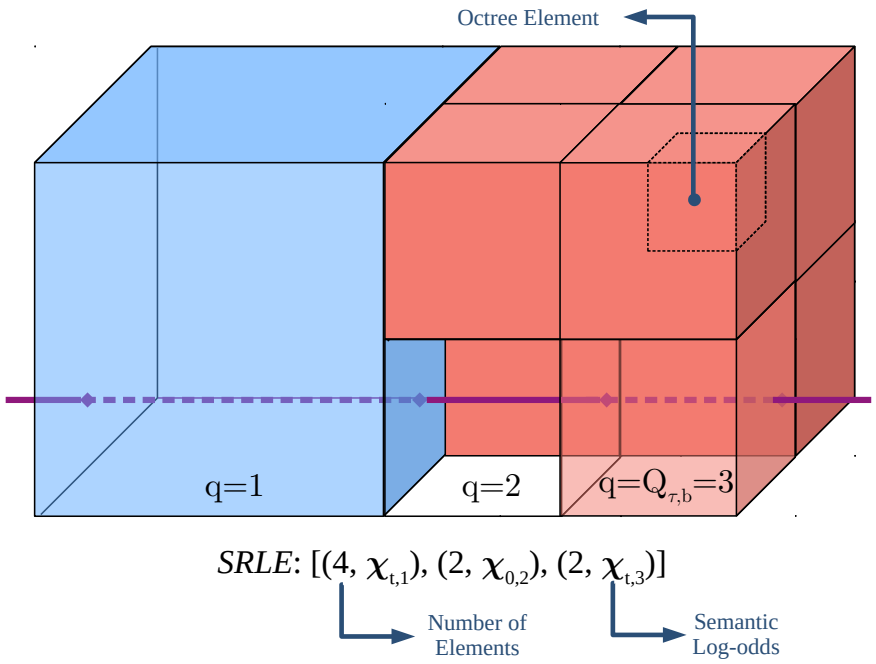


Figure 4.2. Ray cast representation as semantic run-length encoding (SRLE). The multi-class log-odds $\chi_{t,q}$ are uniform within each cube. The voxel corresponding to $q = 2$ is unexplored, hence its multi-class log-odds are denoted as $\chi_{0,2}$.

an octree element is a cell with the smallest physical dimensions.

Fig. 4.2 shows an example of SRLE over a semantic octomap. While SRLE can be used in a uniform-resolution grid map, it is particularly effective of a multi-class octree, which inherently contains large regions with homogeneous multi-class log-odds. Additionally, the octree data structure allows faster ray casting since it can be done over octree leaf nodes [3, 125], instead of a uniform-resolution grid as in [24].

SRLE ray casting delivers substantial gains in efficiency for mutual information computation since the contribution of each group $\{m_i \mid i \in \mathscr{R}^q_{\tau,b}(r_{max})\}$ in the innermost summation of (4.6) can be obtained in closed form.

Proposition 3. *The Shannon mutual information between a single range-category measurement*

$\mathbf{z}_{\tau,b}$ *and a semantic octomap* $\mathbf{m}$ *can be computed as:*

$$I(\mathbf{m};\mathbf{z}_{\tau,b}|\mathscr{Z}_{1:t}) = \sum_{k=1}^{K}\sum_{q=1}^{Q_{\tau,b}} \rho_{\tau,b}(q,k)\Theta_{\tau,b}(q,k) \tag{4.8}$$

where $Q_{\tau,b}$ *is the number of partitions along the ray* $\mathbf{R}_\tau\eta_b$ *that have identical multi-class log-odds and the multi-class probabilities for each partition are denoted as:*

$$\begin{cases} \pi_t(q,k) = p_t(m_i = k) \\ \chi_{t,q} = \mathbf{h}_{t,i} \end{cases} \qquad \forall i \in \mathscr{R}^q_{\tau,b}(r_{max}).$$

Furthermore, defining $\omega_{\tau,b,q} = |\mathscr{R}^q_{\tau,b}(r_{max})|$ *as the width of q-th partition, we have:*

$$\begin{aligned}
\rho_{\tau,b}(q,k) &:= \pi_t(q,k)\prod_{j=1}^{q-1}\pi_t^{\omega_{\tau,b,j}}(j,0), \\
\Theta_{\tau,b}(q,k) &:= \beta_{\tau,b}(q,k)\frac{1-\pi_t^{\omega_{\tau,b,q}}(q,0)}{1-\pi_t(q,0)} + \frac{f(\phi^- - \chi_{0,q},\chi_{t,q})}{(1-\pi_t(q,0))^2}\Big[(\omega_{\tau,b,q}-1)\pi_t^{\omega_{\tau,b,q}+1}(q,0) \\
&\qquad\qquad - \omega_{\tau,b,q}\pi_t^{\omega_{\tau,b,q}}(q,0) + \pi_t(q,0)\Big], \\
\beta_{\tau,b}(q,k) &:= f(\phi^+ + \mathbf{E}_{k+1}\psi^+ - \chi_{0,q},\chi_{t,q}) + \sum_{j=1}^{q-1}\omega_{\tau,b,j}f(\phi^- - \chi_{0,j},\chi_{t,j}).
\end{aligned}$$

Proof. See Appendix C. □

In (4.8), $\rho_{\tau,b}(q,k)$ relates to the event that the partition $\mathscr{R}^q_{\tau,b}(r_{max})$ belongs to category k while all of the previous partitions along the ray $\mathbf{R}_\tau\eta_b$ are free. Analogous to the definition of $C_{\tau,b}(n,k)$ in Proposition 2, $\beta_{\tau,b}(q,k)$ is the weighted sum of log-ratios $f(\phi,\chi)$ for the first q partitions along the ray $\mathbf{R}_\tau\eta_b$ under the same event as the one $\rho_{\tau,b}(q,k)$ is associated with. Accumulating the multi-class probabilities within the partition $\mathscr{R}^q_{\tau,b}(r_{max})$ yields $\Theta_{\tau,b}(q,k)$, see (C.5) for more details. Therefore, the mutual information in (4.8) is equivalent to the expectation of accumulated log-ratios $\Theta_{\tau,b}(q,k)$ over all possible instantiations of $\mathbf{z}_{\tau,b}$.

Proposition 3 allows an extension of the mutual-information lower bound in Proposition 2

to semantic octomap representations, summarized in the corollary below. The proof follows directly from the additive property of mutual information between a semantic octomap and a sequence of independent observations.

Corollary 1. *Given a sequence of range-category observations $\mathscr{Z}_{t+1:t+T}$, the Shannon mutual information between $\mathscr{Z}_{t+1:t+T}$ and a semantic octomap* $\mathbf{m}$ *can be lower bounded as:*

$$I(\mathbf{m}; \mathscr{Z}_{t+1:t+T} | \mathscr{Z}_{1:t}) \geq I(\mathbf{m}; \underline{\mathscr{Z}}_{t+1:t+T} | \mathscr{Z}_{1:t}) = \sum_{\tau=t+1}^{t+T} \sum_{b=1}^{B} \sum_{k=1}^{K} \sum_{q=1}^{Q_{\tau,b}} \rho_{\tau,b}(q,k) \Theta_{\tau,b}(q,k), \quad (4.9)$$

where $\underline{\mathscr{Z}}_{t+1:t+T}$ is a subset of non-overlapping measurements that satisfy (4.5)*, and $\rho_{\tau,b}(q,k)$ and $\Theta_{\tau,b}(q,k)$ are defined in Proposition 3.*

The same approach as in Alg. 3 is used for autonomous exploration over a semantic octomap. However, we employ the information computation formula of (4.9) to quantify the informativeness of candidate robot trajectories. The active mapping method in Alg. 3 provides a greedy exploration strategy, which does not change subsequent control inputs based on the updated map distribution. Greedy exploration may be sub-optimal and manifests itself as back and forth travel between map frontiers. We alleviate this behavior by (a) computing the information along the whole trajectory as opposed only at the frontiers or next best view, and (b) re-plan frequently to account for the updated map distribution. Discounted by distance traveled as the cost of a trajectory, this leads to a more accurate calculation of information gain along a candidate path which rules out most of the back and forth visiting behaviour. It is also important to mention that the main scope of this chapter is introduction of the mutual information between a multi-class semantic octree representation and range-category observations. Our method enables fast and accurate evaluation of information for any set of candidate trajectories, likes of which can be generated by random tree methods [128, 170] or hierarchical planning strategies [28] or, in the simplest form, a greedy approach that computes paths to each frontier. We believe utilizing our proposed information measure to score candidate viewpoints would

be complementary, rather than an alternative, to the state-of-the-art exploration methods that use sophisticated optimization strategies [28, 45]. In the next two chapters, we show how our proposed semantic mutual information formula can be employed alongside optimization-based path planning approaches.

4.4.1 Computational Complexity

Note that the mutual information computations in both (4.6) and (4.9) can be performed recursively. For (4.6), we have:

$$\begin{aligned} p_{\tau,b}(n+1,k) &= p_{\tau,b}(n,k)\frac{p_t(m_{j^*_{\tau,b}}=k)p_t(m_{i^*_{\tau,b}}=0)}{p_t(m_{i^*_{\tau,b}}=k)}, \\ C_{\tau,b}(n+1,k) &= C_{\tau,b}(n,k) - f(\phi^+ + \mathbf{E}_{k+1}\psi^+ - \mathbf{h}_{0,i^*_{\tau,b}}, \mathbf{h}_{t,i^*_{\tau,b}}) \\ &\quad + f(\phi^+ + \mathbf{E}_{k+1}\psi^+ - \mathbf{h}_{0,j^*_{\tau,b}}, \mathbf{h}_{t,j^*_{\tau,b}}) \\ &\quad + f(\phi^- - \mathbf{h}_{0,i^*_{\tau,b}}, \mathbf{h}_{t,i^*_{\tau,b}}), \end{aligned} \tag{4.10}$$

where $j^*_{\tau,b}$ and $i^*_{\tau,b}$ correspond to the index of farthest map cell in $\tilde{\mathscr{R}}_{\tau,b}(n+1)$ and $\tilde{\mathscr{R}}_{\tau,b}(n)$, respectively. A similar recursive pattern can be found in (4.9):

$$\begin{aligned} \rho_{\tau,b}(q+1,k) &= \rho_{\tau,b}(q,k)\frac{\pi_t(q+1,k)\pi_t^{\omega_{\tau,b,q}}(q,0)}{\pi_t(q,k)}, \\ \beta_{\tau,b}(q+1,k) &= \beta_{\tau,b}(q,k) - f(\phi^+ + \mathbf{E}_{k+1}\psi^+ - \chi_{0,q}, \chi_{t,q}) \\ &\quad + f(\phi^+ + \mathbf{E}_{k+1}\psi^+ - \chi_{0,q+1}, \chi_{t,q+1}) \\ &\quad + \omega_{\tau,b,q} f(\phi^- - \chi_{0,q}, \chi_{t,q}). \end{aligned} \tag{4.11}$$

This implies that the innermost summations of (4.6) and (4.9) can be obtained in $O(N_{\tau,b})$ and $O(Q_{\tau,b})$, respectively, where $N_{\tau,b}$ is the number of map cells along a single ray $\mathbf{R}_\tau \eta_b$ up to its maximum range, and $Q_{\tau,b}$ is the number of groups of consecutive cells that possess the same multi-class probabilities. In an environment containing K object classes, evaluating the

informativeness of a trajectory composed of T observations, where each observation contains B beams, has a complexity of $O(TBKN_{\tau,b})$ for a regular-grid multi-class representation and a complexity of $O(TBKQ_{\tau,b})$ for a multi-class octree representation.

As we demonstrate in Sec. 4.5.3, for a ray $\mathbf{R}_\tau \eta_b$ we often observe that $Q_{\tau,b}$ is significantly smaller than $N_{\tau,b}$ thanks to the octree pruning mechanism. Since $N_{\tau,b}$ scales linearly with the map resolution, the complexity of information computation over a semantic octomap grows sub-linearly with respect to the inverse of the octree element dimensions, which is a parameter analogous to the map resolution.

4.5 Experiments

In this section, we evaluate the performance of SSMI in simulated and real-world experiments. We compare SSMI with two baseline exploration strategies, i.e., frontier-based exploration [161] and FSMI [170], in a 2-D active binary mapping scenario in Sec. 4.5.1 and a 2-D active multi-class mapping scenario in Sec. 4.5.2. All three methods use our range-category sensor model in (3.7) and our Bayesian multi-class mapping in (3.5) but select informative robot trajectories $\mathbf{X}_{t+1:t+T}(\mathbf{u}_{t:t+T-1})$ based on their own criteria. In Sec. 4.5.3, we evaluate the improvement in ray tracing resulting from SRLE through an experiment in a 3-D simulated Unity environment. Additionally, in Sec. 4.5.4, we use a similar 3-D simulation environment to apply SSMI alongside Frontier, FSMI, and hierarchical coverage maximization method TARE [28]. In this section we use our octree-based multi-class information computation introduced in Sec. 4.4 in order to demonstrate large-scale realistic active multi-class mapping. Finally, Sec. 4.5.5, we test SSMI exploration in a real environment using a ground wheeled robot. An open-source implementation of SSMI is available on GitHub[1].

In each planning step of 2-D exploration, we identify frontiers by applying edge detection on the most likely map at time t (the mode of $p_t(\mathbf{m})$). Then, we cluster the edge cells by detecting the connected components of the boundaries between explored and unexplored space. We plan a

[1] https://github.com/ExistentialRobotics/SSMI.

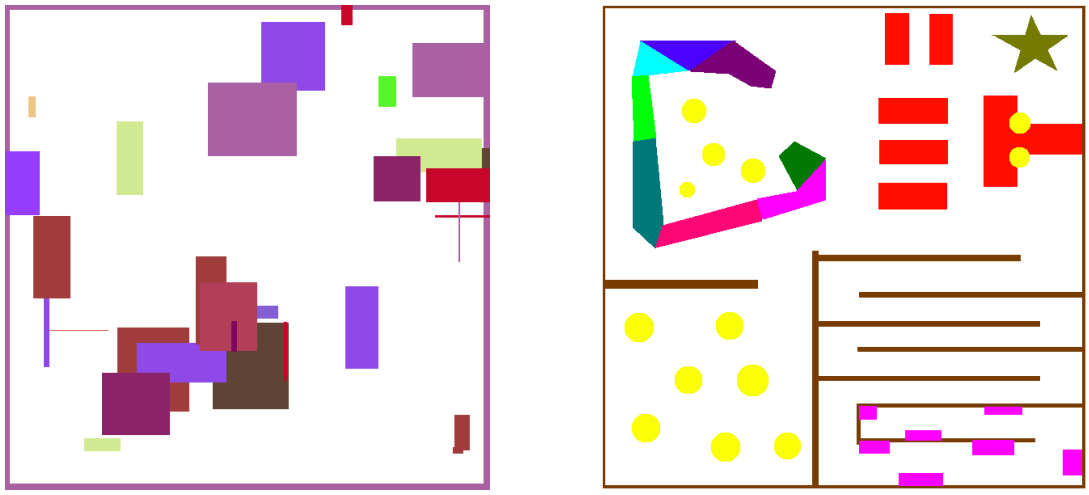
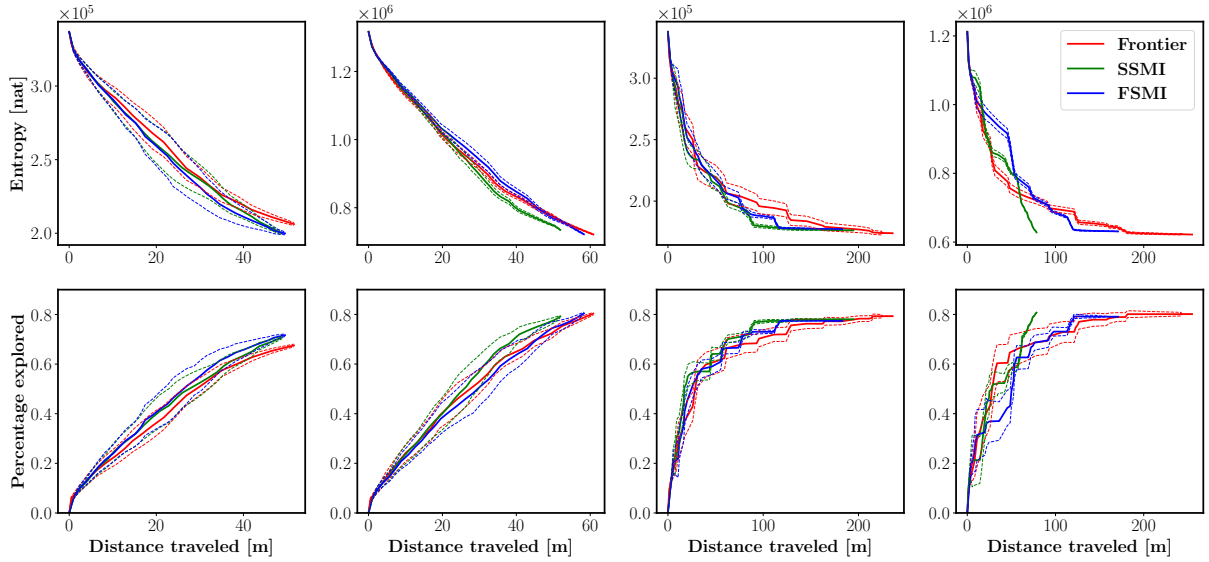


Figure 4.3. Synthetic environments used for comparisons among frontier-based exploration [161], FSMI [170], and SSMI. Different semantic categories are represented by distinct colors. Left: An instance of procedurally generated random environment with 10 object classes. Right: Hand-designed environment with corridor and block structures with 12 object classes.

path from the robot pose $\mathbf{X}_t$ to the center of each frontier using A^* graph search and provide the path to a low-level controller to generate $\mathbf{u}_{t:t+T-1}$. For 3-D exploration, we first derive a 2-D occupancy map by projecting the most likely semantic octomap at time t onto the $z = 0$ surface and proceed with similar steps as in 2-D path planning.

4.5.1 2-D Binary Exploration

We consider active binary occupancy mapping first. We compare SSMI against Frontier and FSMI in 1 structured and 10 procedurally generated 2-D environments, shown in Fig. 4.3. A 2-D LiDAR sensor is simulated with additive Gaussian noise $\mathcal{N}(0, 0.1)$. Fig. 4.4a and Fig. 4.4c compare the exploration performance in terms of map entropy reduction and percentage of the map explored per distance traveled among the three methods. SSMI performs similarly to FSMI in that both achieve low map entropy by traversing significantly less distance compared to Frontier.



(a) Random map / Binary exploration

(b) Random map / Semantic exploration

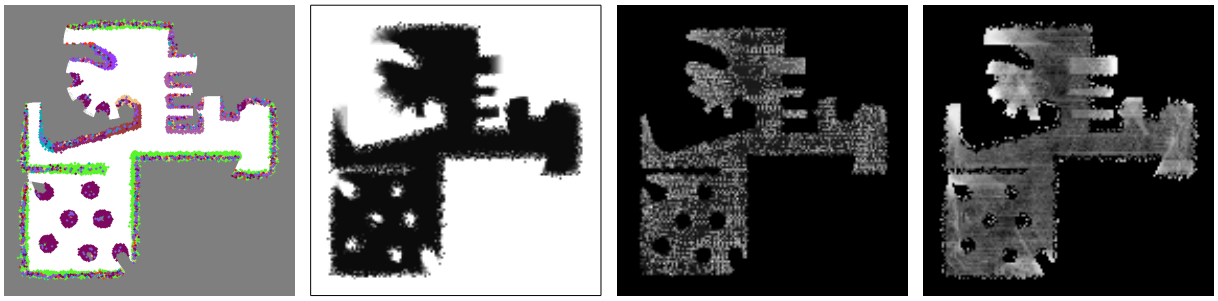
(c) Structured map / Binary exploration

(d) Structured map / Semantic exploration

Figure 4.4. Simulation results for active mapping on the environments in Fig. 4.3, for 20 exploration iterations. Solid and dotted lines represent mean and 1 standard deviation from the mean, respectively. (a), (b): Exploration performance averaged over 10 random environments with 3 random starting positions for each instance. Exploration in the random environments sometimes did not terminate before the maximum number of iterations, and therefore the corresponding curves do not flatten. This can be attributed to the fact that random maps, with the same size as the structured map, contain more frontiers that need to be explored in each iteration since each scan has a higher probability of being occluded in multiple angles due to the lack of certain patterns such as corridors. (c), (d): Exploration performance on the structured environment averaged over 3 random starting positions. For the structured map, the exploration terminates before reaching the maximum number of iterations, which explains the flat curves at the end of the corresponding plots.

4.5.2 2-D Multi-Class Exploration

Next, we use the same 2-D environments in Fig. 4.3 but introduce range-category measurements. Range measurements are subject to additive Gaussian noise $\mathcal{N}(0, 0.1)$, while category measurements have a uniform misclassification probability of 0.35. Fig. 4.4b and Fig. 4.4d compare the semantic exploration performance for all three strategies. SSMI reaches the same level of map entropy as FSMI and Frontier but traverses a noticeably shorter distance. This can be attributed to the fact that only SSMI distinguishes map cells whose occupancy



(a) Partially explored semantic map

(b) Entropy surface

(c) Occupancy mutual information surface

(d) Multi-class mutual information surface

Figure 4.5. Comparison between different mutual information formulations used for exploration. (a), (b): A snapshot of 2-D exploration showing the map estimate and the corresponding uncertainty, where the entropy for each pixel i is computed as (4.12). (c), (d): The mutual information used to find the informative trajectory by FSMI and SSMI, respectively. Brighter pixels indicate larger values.

probabilities are the same but their per-class probabilities differ from each other. To further illustrate this, we visualize the entropy and information surfaces used by FSMI and SSMI. Fig. 4.5a shows a snapshot of semantic exploration while Fig. 4.5b visualizes the entropy of each pixel i computed as:

$$H(m_i|\mathscr{Z}_{1:t}) = -\sum_{k=0}^{K} p_t(m_i = k)\log p_t(m_i = k), \tag{4.12}$$

where $\mathscr{Z}_{1:t}$ denote realized observations until time t. The task of exploration can be regarded as minimizing the conditional entropy summed over all pixels, i.e., map entropy. However, since the observations are not known in advance, we resort to estimate the reduction in uncertainty by computing the expectation over the observations. Accounting for the prior uncertainty in map, we arrive at maximizing mutual information as our objective, which is related to entropy as follows:

$$H(m_i) - \mathbb{E}_{\mathscr{Z}_{1:t}}\{H(m_i|\mathscr{Z}_{1:t})\} = I(m_i;\mathscr{Z}_{1:t}). \tag{4.13}$$

Therefore, the exploration performance is highly dependent upon the mutual information formulation, since it directly dictates how the uncertainty is quantified. As shown in Fig. 4.5d and

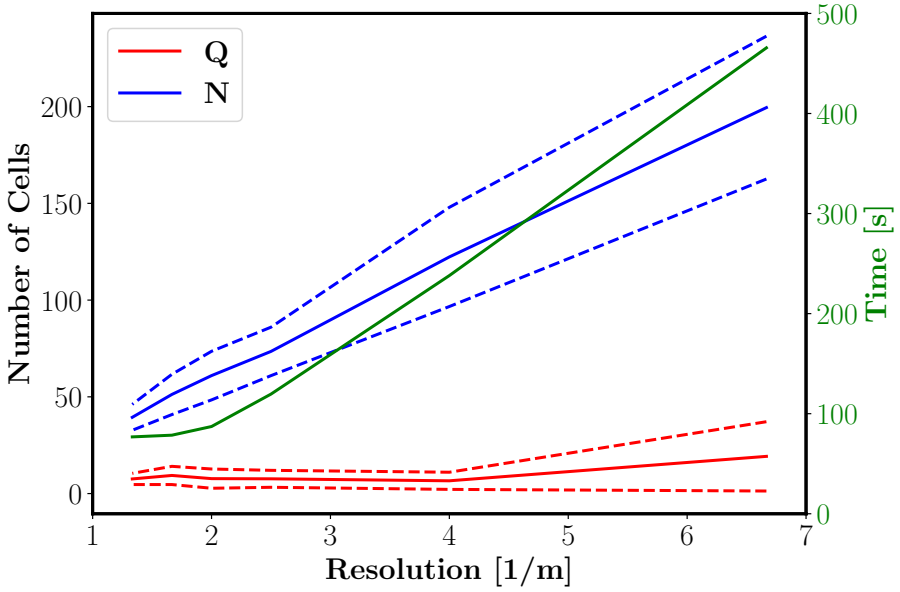


Figure 4.6. Variation of the visited octomap cells and octree elements denoted as Q and N, respectively, with respect to the map resolution. Solid blue and red lines represent the average values for Q and N over all ray castings, while the dashed lines show one standard deviation from the average. The green curve shows the total exploration time for each map resolution. All measurements are accumulated in the course of 5 exploration iterations.

resulted from capturing per-class uncertainties, semantic mutual information of SSMI, computed in (4.6) provides a smoother and more accurate estimation of information-rich regions compared to the binary mutual information formula used by FSMI (equation (18) in [170]) shown in Fig. 4.5c.

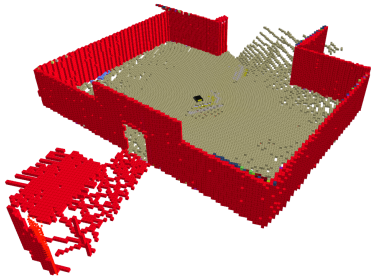
4.5.3 SRLE Compression for 3-D Ray Tracing

In this subsection, we evaluate the ray-tracing compression resulting from SRLE through an experiment in the same photo-realistic 3-D Unity simulation environment used in Chapter 3. In order to remove irrelevant randomness, the sensors and the semantic segmentation are defined as error-free. We define *map resolution* as the inverse of the dimensions of an octree element. For resolutions ranging from $1.3m^{-1}$ to $6.6m^{-1}$, we run 5 exploration iterations using the semantic octomap and information computation of Sec. 4.4 and store all ray traces in SRLE format. Fig. 4.6 shows the change in distribution for the number of OctoMap cells Q and OcTree

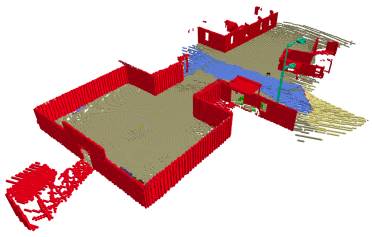
elements N visited during each ray trace, as well as the time required to execute each exploration episode as a function of map resolution. In other words, N represents the number of cells to be processed during mapping and information computation as if the environment was represented as a regular 3-D grid, while Q represents the actual number of processed semantic octomap cells. The pruning mechanism of the octree representation results in a substantial gain in terms of the number of cells visited for each ray tracing. As opposed to the almost linear growth of N, the distribution for Q is effectively independent of the map resolution, except for very fine resolutions where void areas between observations rays prevent efficient pruning. However, for map resolutions larger than $2m^{-1}$, the exploration time tends to grow larger with the increase of map resolution. This is attributed to the recursive ray insertion method of OctoMap in which it is required to re-compute log odds for each OcTree element along an observation ray whenever an observation ray does not carry the same (free or object class) state as the visited cell. In the subsequent 3-D experiments, we choose map resolution of $2m^{-1}$ in order to balance between performance and map accuracy.

4.5.4 3-D Exploration in a Unity Simulation

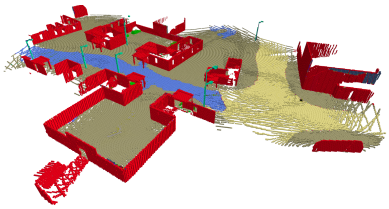
We evaluate SSMI in the same 3-D simulation environment as the previous experiments, however, this time the range measurements have an additive Gaussian noise of $\mathcal{N}(0, 0.1)$ and the semantic segmentation algorithm detects the true class with a probability of 0.95 while the misclassification happens uniformly in the pixel space. Fig. 4.7 shows several iterations of the exploration process. For comparison, we implemented a 3-D version of FSMI [170] that utilizes run-length encoding to accelerate the information computation for a binary octomap. Moreover, we deploy the state-of-the-art hierarchical exploration method of TARE [28] in our 3-D Unity simulation environment. Fig. 4.8a shows the change in map entropy versus distance traveled and total elapsed time for all exploration strategies. We observe that SSMI is the most efficient in terms of solving the trade-off between path length and information gathered along the path. SSMI achieves the lowest entropy in the multi-class octomap. Similar to the discussion



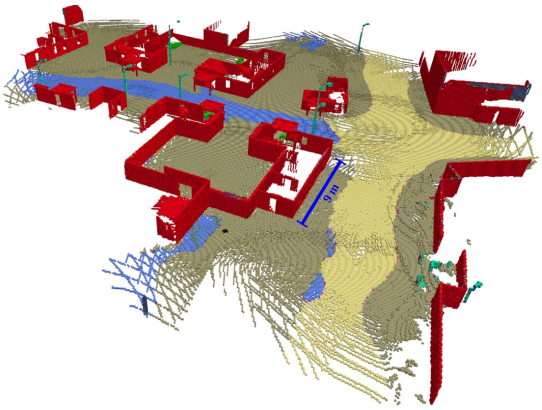
(a) The robot begins exploration.



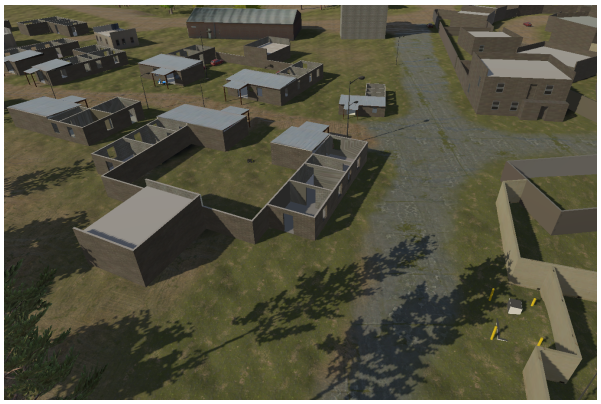
(b) After 16 iterations, the robot starts to refine previously explored areas to fill partially observed objects.



(c) The robot explores unknown regions located on the boundaries of the explored area at iteration 40.



(d) Multi-class occupancy map after 60 exploration iterations



(e) Photo-realistic Unity simulation environment

Figure 4.7. Time lapse of autonomous exploration and multi-class mapping in a simulated Unity environment. The robot is equipped with an RGBD sensor and runs semantic segmentation. Different colors represent different semantic categories (grass, dirt road, building, etc.).

in Sec. 4.5.2, this observation can be ascribed to the fact that, among the compared methods, the only objective function which captures the uncertainty in both semantic classes and occupancy of the environment is the one used by SSMI. On the other hand, SSMI and FSMI require evaluation of mutual information along each candidate trajectory, which has the same cardinality as the number of all frontiers in the current map estimate $p_t(\mathbf{m})$, whereas the hierarchical planning method employed by TARE only requires local trajectory computation with a global coverage path obtained at a coarse level. As a result, TARE exploration can be performed over a relatively shorter time period compared to SSMI and FSMI in scenarios where the number of frontiers is large, e.g. outdoor areas. Parallel computation of mutual information for each candidate

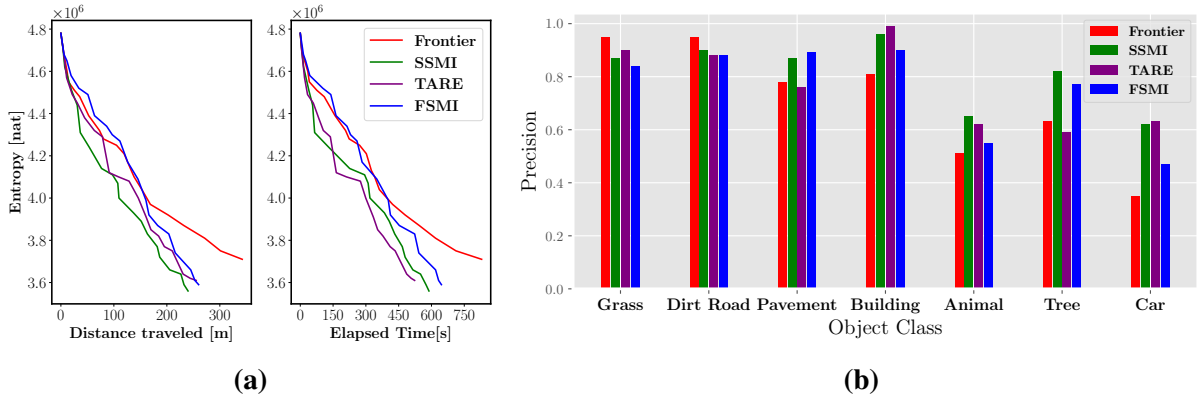
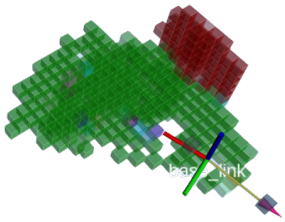


Figure 4.8. Simulation results for exploration in Unity 3-D environment. (a) Exploration performance compared between SSMI, Frontier, TARE, and FSMI. (b) Mapping precision for observed semantic classes.

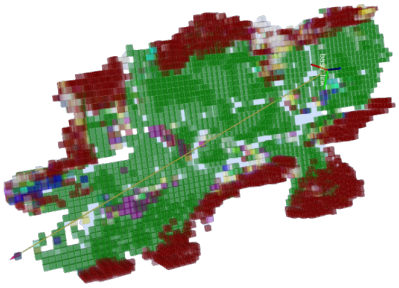
trajectory or using heuristics such as frontier size in order to sort candidate solutions would improve the computation time of SSMI; however we believe these are outside of the scope of this paper. Fig. 4.8b compares the mapping precision of various object classes for the tested methods. SSMI exhibits higher precision for object categories that appear rarely, such as the *Animal* or *Tree* classes while Frontier slightly outperforms SSMI when it comes to mapping the *Grass* and *Dirt Road* categories. This can be explained by the tendency of SSMI towards achieving high overall classification precision even if it requires slight reduction of precision for certain object categories. Furthermore, TARE achieves the highest precision for the *Building* class, which can be justified by the observation that the computed global coverage path tends to traverse near building walls.

4.5.5 3-D Exploration in a Real-World Office Environment

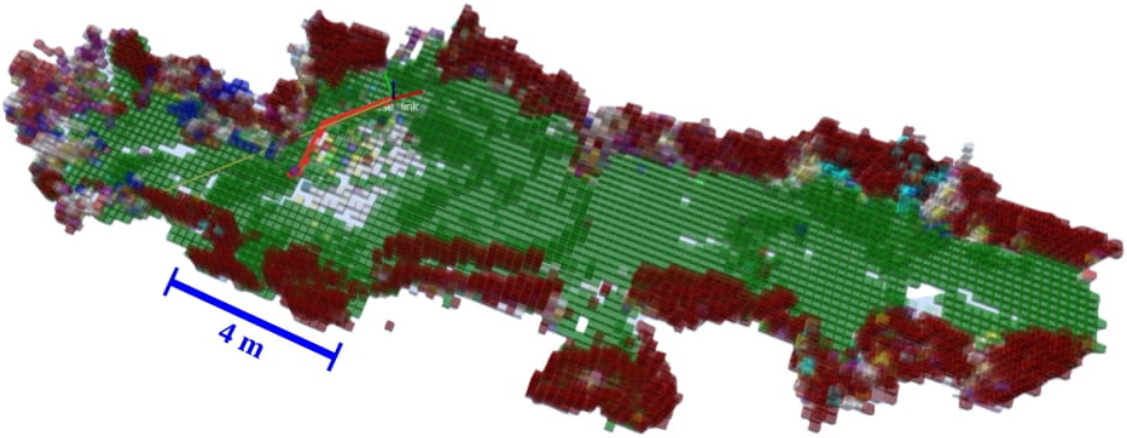
We implemented SSMI on a ground wheeled robot to autonomously map an indoor office environment. Fig. 4.9e shows the robot equipped with a NVIDIA Xavier NX computer, a Hokuyo UST-10LX LiDAR, and an Intel RealSense D435i RGBD camera. Similar to the experiments in Chapter 3, ROS was used for software deployment on the robot, and ICP laser scan matching provided localization. This time, we utilized a *ResNet18* [67] neural network architecture pre-



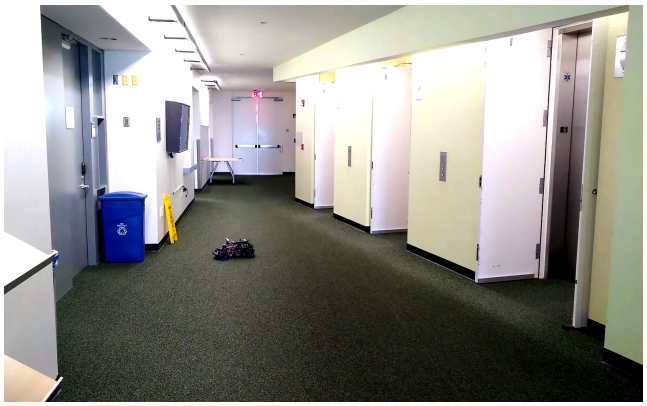
(a) The robot begins exploration.



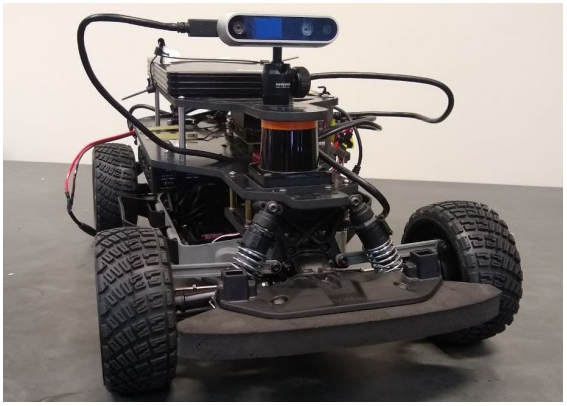
(b) The robot visits neighboring unexplored regions while trying to refine the map of visited areas.



(c) Semantic OctoMap after 20 exploration iterations.



(d) Office environment featuring corridors, furniture, signs, and doors.



(e) Robot car used in indoor real-world experiments.

Figure 4.9. Time lapse of autonomous exploration and multi-class mapping in the environment depicted in (d), using the mobile robot shown in (e). The exploration is run for 20 iterations. Different colors represent different semantic categories (floor, wall, furniture, etc.).

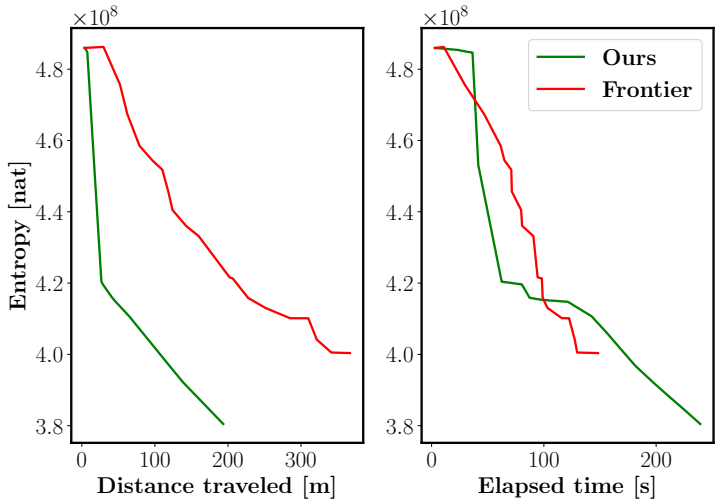


Figure 4.10. Real-world experiment results for active mapping for 20 exploration iterations.

trained on the SUN RGB-D dataset [135] for semantic segmentation. In particular, we employed the deep learning inference ROS nodes provided by NVIDIA [112], which are optimized for Xavier NX computers via TensorRT acceleration. Due to limited computational power available on the mobile platform, we operated the RGBD camera at a lower frame rate of 15 Hz with color and depth image size set to 640×480. The semantic segmentation algorithm was able to produce pixel classification images (resized to 512×400) at an average rate of 9.8 frames per second. Our implementation was able to publish semantic octomap ROS topics every $0.34s$, on average, with all of the processing occurred on the mobile platform.

Fig. 4.9 depicts the exploration process, while Fig. 4.10 shows the performance of SSMI compared to frontier-based over 20 exploration iterations. We observe that, similar to the simulations, SSMI outperforms frontier-based exploration in terms of distance traveled. Also, SSMI shows on par performance compared to Frontier in terms of entropy reduction per time. This can be explained by the fact that large depth measurement noise and classification error in the real-world experiments result in (a) the need for re-visiting explored areas in order to estimate an accurate map, leading to poor entropy reduction for the frontier-based method and (b) a small number of safe candidate trajectories, leading to fewer computations to be performed by SSMI. Overall, our experiments show that SSMI outperforms Frontier in indoor exploration

scenarios where the number, and length, of candidate trajectories is constrained by the size of the environment.

4.6 Summary

In this chapter, we presented Semantic Shannon Mutual Information (SSMI) computation for robot exploration. Our results enable efficient mutual information computation over multi-class maps and make it possible to optimize for per-class uncertainty. Our experiments show that SSMI performs on par with the state of the art FSMI method in binary active mapping scenarios. However, when semantic information is considered SSMI outperforms existing algorithms and leads to efficient exploration and accurate multi-class mapping even in the presence of measurement noise and classification error. Experiments in both simulated and real-world environments showed the scalability of SSMI for large-scale 3-D exploration scenarios.

Acknowledgments

Chapter 4, in part, is a reprint of the material as it appears in A. Asgharivaskasi and N. Atanasov, "Active Bayesian multi-class mapping from range and semantic segmentation observations," IEEE International Conference on Robotics and Automation (ICRA), pp. 1-7, 2021, and in A. Asgharivaskasi and N. Atanasov, "Semantic octree mapping and Shannon mutual information computation for robot exploration," IEEE Transactions on Robotics (T-RO), vol. 39, no.3, pp. 1910-1928, 2023. The dissertation author was the primary investigator and author of these papers.

Chapter 5

Differentiable Occlusion and Collision-Aware Active Mapping

Mapping an unknown environment using sensor-equipped mobile robots has been widely studied motivated by many real-world applications such as search and rescue operations, planetary exploration, security and surveillance. While the traditional mapping methods have been developed for a given robot trajectory, optimizing the path to increase the accuracy of the constructed map is significant for the operation of autonomous robots in highly unstructured environments. Such an active mapping problem renders challenges in the computation of objective function and its reliable optimization method in terms of enhancing the performance and computational feasibility. In particular, the problem of active mapping aims to plan an informative sequence of sensing views given a limited budget such as distance traveled. This chapter considers active occupancy grid mapping using a range sensor, such as LiDAR or depth camera. State-of-the-art methods optimize information-theoretic measures relating the occupancy grid probabilities with the range sensor measurements. The non-smooth nature of ray-tracing within a grid representation makes the objective function non-differentiable, forcing existing methods to search over a discrete space of candidate trajectories. Based on our work in [10], this chapter proposes a differentiable approximation of the Shannon mutual information (SMI) between a grid map and ray-based observations that enables gradient ascent optimization in the continuous space of *SE(3)* sensor poses. Our main **contributions** are:

1. a differentiable interpolation of the SMI as well as a closed-form gradient expression,

2. decomposition of the SMI into additive terms over a robot trajectory in *SE(3)*, under sufficient assumptions for the interpolation method.

Our gradient-based formulation of active mapping leads to more informative sensing trajectories, while avoiding occlusions and collisions. The proposed method is demonstrated in simulated and real-world experiments in 2-D and 3-D environments.

5.1 Related Works

One of the most common procedures for incremental estimation of occupancy grid maps is to utilize a narrow-ray range sensor, such as Laser range finder [143, Ch. 9]. However, the resulting observation model occurs to be a piecewise constant, namely a non-smooth function of measurement poses due to the discrete nature of ray-tracing over a grid map [75, Ch. 5]. Consequently, the objective function is also piecewise-constant, making the pose derivatives either 0 or undefined. In addition to the problem of non-smoothness of grid map representations, occlusion, which is caused when a field of view intersects with objects, is a significant challenge in sensing and planning to accomplish several tasks in robotics such as 3-D reconstruction [96], object classification [16], and target tracking [91]. Koga *et al.* [80, 82] has developed dense active SLAM using iterative Covariance Regulation (iCR), which utilizes a differentiable field of view that enables deriving an explicit gradient of an information-theoretic objective with respect to a multi-step control sequence. Incorporating occlusion in iCR makes the field of view dependent on the map estimate, which renders a significant challenge for sensing model formulation since it violates the linear-Gaussian property of the mapping by Kalman Filter. Nevertheless, the current work borrows one of the key ideas from iCR: In order to perform gradient-based optimization of the mutual information, one needs to apply a smooth proximity operator among the elements involved in its computation.

The work by Rocha *et al.* [122] was among the first that attempted to derive a gradient-based strategy for active mapping of occupancy grids. The authors proposed a gradient of map entropy with respect to the robot pose at a cell center via finite difference of entropy values at adjacent cells. Julian [75] proposed a divergent beam sensor model, where the width of a beam increases radially as it travels farther through space. While the derived mutual information formula was shown to be differentiable, it suffers from high computational complexity as it requires numerical integration of the objective function. Charrow *et al.* [34] proposed a numerical evaluation for the gradient of Cauchy-Schwarz mutual information (CSQMI) [35] using finite differences of CSQMI evaluated at cell centers. Our work in this chapter is most similar to [46] and [47], where the authors formulate the information gain as a sum of informative elements weighted by a discount factor. In particular, [46] defines informative elements as frontier cells between free and unexplored areas visible from a candidate pose. However, unlike the mutual information between the map and a sensor observation, using visible frontier size as a proxy for information gain does not take into account the effect of sensor noise which is inevitable in real-world sensing applications [35, 134].

The present work distinguishes itself from the prior methods by proposing an active mapping strategy that allows gradient ascent optimization of the SMI between the grid map and a sequence of beam-based observations. As opposed to the discrete-space active mapping methods [8, 35, 161, 170] that aim to plan an informative robot path through evaluating a finite set of candidate trajectories, the current work finds an optimal trajectory over the continuous space of the robot state. Moreover, gradient-based methods allow augmenting the objective function with other differentiable terms (e.g. localization accuracy [79], path cost [99]). With the addition of a differentiable collision penalty to the objective function, we propose an occlusion and collision-aware robot exploration. Fig. 5.1 illustrates the proposed gradient-based active mapping using a depth sensor.

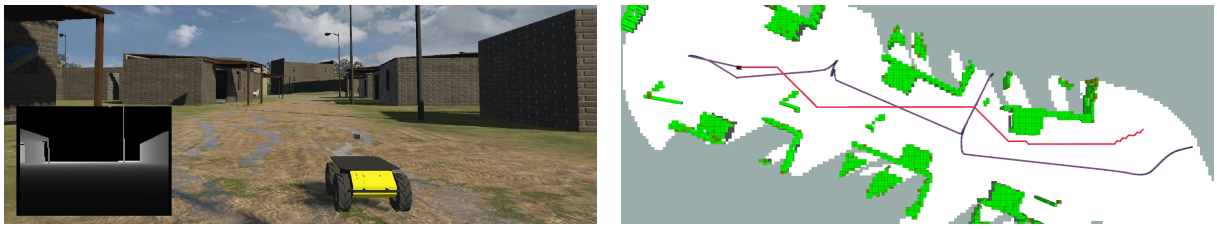


Figure 5.1. Occlusion and collision-aware active mapping. Left: Wheeled robot exploring an unknown environment equipped with a depth sensor used for estimating a 3-D map. Right: Path planning given the current map. The red trajectory is A^* planning towards a frontier. The purple trajectory is obtained via gradient ascent optimization of the objective function.

5.2 Problem: Continuous-Space *SE(3)* Optimization of Shannon Mutual Information between a Voxel Map and Range Observations

Consider a robot with pose $\mathbf{X}_t \in SE(3)$ at time t:

$$\mathbf{X}_t := \begin{bmatrix} \mathbf{R}_t & \mathbf{p}_t \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

The robot is navigating in an environment composed of occupied and free space. A mounted range sensor, e.g. LiDAR or depth camera, provides the robot with a stream of beam-based observations $\mathbf{z}_t \in \mathbb{R}^B$, where B is the number of beams in a laser scan or pixels in a depth image, measuring the distance from the robot's position to the closest obstacle along the beam. We model the map $\mathbf{m}$ as a grid of cells m_i, $i \in \mathscr{I} := \{1,\dots,N\}$, where each cell can take one of the two states: *free* or *occupied*. To model measurement noise, we consider a PDF $p(\mathbf{z}_t \mid \mathbf{m}, \mathbf{X}_t)$ for each observation. Let $p_t(\mathbf{m}) = p(\mathbf{m} \mid \mathscr{H}_t)$ be the PMF of the map $\mathbf{m}$ given the history of robot poses and observations $\mathscr{H}_t = \{(\mathbf{X}_\tau, z_\tau)\}_{\tau=1}^t$. A new observation $\mathbf{z}_{t+1}$ made from robot pose $\mathbf{X}_{t+1}$ can then be integrated into the map estimation process using Bayes rule:

$$p_{t+1}(\mathbf{m}) \propto p(\mathbf{z}_{t+1} \mid \mathbf{m}, \mathbf{X}_{t+1}) p_t(\mathbf{m}). \tag{5.1}$$

The goal is to choose a collision-free pose trajectory to obtain maximally informative measurements for constructing an accurate map. As shown by Julian *et al.* [76], maximizing the Shannon Mutual Information (SMI) between the map $\mathbf{m}$ and a sequence of potential future measurements $\mathbf{z}_{t+1:t+T}$ yields an efficient active mapping strategy. The SMI is defined as:

$$I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t) := \sum_{\mathbf{m} \in 2^N} \int \cdots \int p(\mathbf{m}, \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t) \times \log \frac{p(\mathbf{m}, \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t)}{p(\mathbf{m} \mid \mathscr{H}_t) p(\mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t)} \prod_{\tau=1}^{T} d\mathbf{z}_{t+\tau}, \tag{5.2}$$

where $\mathscr{H}_t$ represents the realized history of robot poses and observations and, hence, does not appear as an integration variable. Throughout this chapter, we assume that the robot pose $\mathbf{X}_t$ is known for all t. Note that the SMI can be considered as a function $I(.) : SE(3)^T \to \mathbb{R}_{\geq 0}$ of the robot trajectory $\mathbf{X}_{t+1:t+T}$ parameterized by $\mathscr{H}_t$.

Problem 3. Given a map PMF $p_t(\mathbf{m})$ obtained from prior robot poses and observations $\mathscr{H}_t$ and a finite planning horizon T, find a pose trajectory $\mathbf{X}_{t+1:t+T} \in SE(3)^T$ that maximizes the SMI between the map $\mathbf{m}$ and the future observations $\mathbf{z}_{t+1:t+T}$ with PDF in (5.1):

$$\max_{\mathbf{X}_{t+1:t+T} \in SE(3)^T} \Big\{ I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t) - \gamma_c C(\mathbf{X}_{t+1:t+T}) \Big\}, \tag{5.3}$$

where $C(\mathbf{X}_{t+1:t+T})$ is a penalty term capturing the cost of collisions along $\mathbf{X}_{t+1:t+T}$ and $\gamma_c \geq 0$ is the weight of the collision penalty.

In the next section, we propose a differentiable approximation of the SMI function that can be utilized for gradient-based optimization of (5.3). In Sec. 5.4 we introduce a collision and occlusion-aware active mapping method based on gradient of the approximated SMI. Lastly, in Sec. 5.5 the performance of the gradient-based active mapping method is demonstrated in simulated and real-world experiments.

5.3 Differentiable Approximation of the Shannon Mutual Information

5.3.1 One Step Ahead Planning

We first study the case where $T = 1$, i.e., the robot view is optimized only one step into the future. The core idea is to introduce the notion of a viewpoint grid, which is a discrete set of candidate robot poses. Then, the SMI with respect to an arbitrary robot pose is approximated as a linear combination of the SMI with respect to the candidate poses with a differentiable function with respect to an arbitrary robot pose.

Definition 4. A viewpoint grid $\mathscr{G}$ is a set of robot poses $\mathbf{X} \in SE(3)$ with position $\mathbf{p} \in \mathbb{R}^3$ and orientation $\mathbf{R} \in SO(3)$ such that $(\mathbf{p}, \mathbf{R}) \in \mathscr{P} \times \mathscr{R}$, where $\mathscr{P}$ is the set of all map cell centers and $\mathscr{R}$ is a finite set of orientations.

We approximate the SMI at pose $\mathbf{X}$ as a convex combination of the SMI computed over all poses in the view grid $\mathscr{G}$. Namely, $I(\mathbf{m}; \mathbf{z} \mid \mathbf{X}, \mathscr{H}_t) \approx \tilde{I}(\mathbf{m}; \mathbf{z} \mid \mathbf{X}, \mathscr{H}_t)$, and

$$\begin{aligned} \tilde{I}(\mathbf{m}; \mathbf{z} \mid \mathbf{X}, \mathscr{H}_t) &:= \sum_{\mathbf{V} \in \mathscr{G}} \alpha_{\mathbf{V}}(\mathbf{X}) I(\mathbf{m}; \mathbf{z} \mid \mathbf{V}, \mathscr{H}_t), \\ \text{where } \sum_{\mathbf{V} \in \mathscr{G}} \alpha_{\mathbf{V}}(\mathbf{X}) &= 1, \ \forall \mathbf{V} \in \mathscr{G} : \ 0 \leq \alpha_{\mathbf{V}}(\mathbf{X}), \end{aligned} \tag{5.4}$$

and $\alpha_{\mathbf{V}}(\mathbf{X})$ is a differentiable function with respect to robot pose $\mathbf{X}$. In practice, evaluating the SMI for all $\mathbf{V} \in \mathscr{G}$ during each planning phase would be computationally expensive. Therefore, one needs to design a distance metric for $\alpha_{\mathbf{V}}(\mathbf{X})$ that is only non-zero in close vicinity of the robot pose $\mathbf{X}$. The idea of pulling the robot pose $\mathbf{X}$ out of the SMI function $I(.)$ using a differentiable weighting function $\alpha_{\mathbf{V}}(\mathbf{X})$ makes it possible to obtain non-zero derivatives for the approximate SMI function $\tilde{I}(.)$ with respect to the robot pose. Fig. 5.2 illustrates the SMI approximated from 3 viewpoints; note how each viewpoint creates a 'field' of information in its vicinity, while the approximate SMI equals the net influence of all 3 viewpoints.

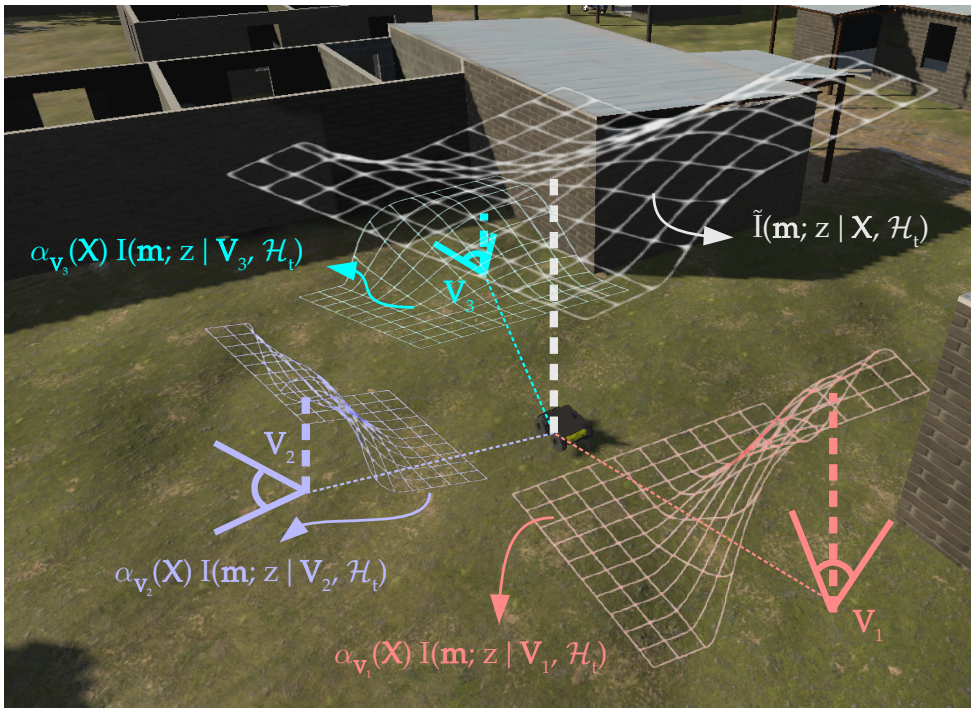


Figure 5.2. The SMI approximation via 3 viewpoints, colored differently for each $\mathbf{V}_i$, $i \in \{1,2,3\}$. The field of view of each viewpoint determines their corresponding SMI $I(\mathbf{m}, \mathbf{z} \mid \mathbf{V}_i, \mathscr{H}_t)$, while the weight $\alpha_{\mathbf{V}_i}(\mathbf{X})$ dictates the contribution of $\mathbf{V}_i$ to the approximated SMI $\tilde{I}(\mathbf{m}, \mathbf{z} \mid \mathbf{X}, \mathscr{H}_t)$, colored white.

The differentiable property of the approximate SMI enables gradient-based optimization of the robot pose in order to generate maximally informative observations. This is done via applying a small perturbation $\psi \in \mathbb{R}^6$ in the robot frame to the pose $\mathbf{X}$ along the direction of the gradient with a step size of l:

$$\begin{aligned} \mathbf{X}^{(k+1)} &= \mathbf{X}^{(k)} \exp(l\hat{\psi}^{(k)}), \\ \psi^{(k)} &= \nabla_{\psi} \tilde{I}(\mathbf{m}; \mathbf{z} \mid \mathbf{X}^{(k)} \exp(\hat{\psi}), \mathscr{H}_t)|_{\psi=0} \\ &= \sum_{\mathbf{V} \in \mathscr{G}} \nabla_{\psi} \alpha_{\mathbf{V}}(\mathbf{X}^{(k)} \exp(\hat{\psi}))|_{\psi=0} I(\mathbf{m}; \mathbf{z} \mid \mathbf{V}, \mathscr{H}_t). \end{aligned} \tag{5.5}$$

Note that the gradient is a 6-dimensional vector since the robot pose in *SE(3)* has 6 degrees of freedom. In Sec. 5.4, we derive a closed form expression of the gradient for a particular selection of the weighting function $\alpha_{\mathbf{V}}(\mathbf{X})$. The gradient ascent rule of (5.5) concludes our

method proposition for finding the (locally) most informative next robot pose when the planning horizon T is set to 1. In the following, we discuss the case where we are interested in optimizing a multi-step sequence of robot poses, given a history of prior observations $\mathscr{H}_t$.

5.3.2 Trajectory Optimization for Active Mapping

Here, we present the approximation of the SMI for a trajectory, rather than a single robot pose. Analogous to (5.4), we define the approximated SMI for robot trajectory $\mathbf{X}_{t+1:t+T}$ as a convex combination over all trajectories in the set $\mathscr{G}^T$. Namely:

$$\begin{aligned} &I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t) \approx \tilde{I}(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t), \\ &\tilde{I}(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t) := \sum_{\mathscr{V} \in \mathscr{G}^T} A_{\mathscr{V}}(\mathbf{X}_{t+1:t+T}) I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathscr{V}, \mathscr{H}_t), \end{aligned} \tag{5.6}$$

where the weighting function $A_{\mathscr{V}}(\mathbf{X}_{t+1:t+T})$ for a trajectory $\mathscr{V} = V_{t+1:t+T}$ is defined as:

$$A_{\mathscr{V}}(\mathbf{X}_{t+1:t+T}) := \prod_{\tau=1}^{T} \alpha_{\mathbf{V}_{t+\tau}}(\mathbf{X}_{t+\tau}). \tag{5.7}$$

The terms $\alpha_{\mathbf{V}_{t+\tau}}(\mathbf{X}_{t+\tau})$ follow the same properties as in (5.4), making (5.6) a convex combination of the SMI terms. Computing $I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathscr{V}, \mathscr{H}_t)$ requires integration over all instances of combined observations $\mathbf{z}_{t+1:t+T}$ and should be repeated for all $\mathscr{V} \in \mathscr{G}^T$, which is computationally infeasible during the planning time. We aim to impose a structure on the weighting function $\alpha_{\mathbf{V}_{t+\tau}}(\mathbf{X}_{t+\tau})$ that allows breaking down $I(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathscr{V}, \mathscr{H}_t)$ into independent additive terms. The following conditions enable such a decomposition:

Condition 1. Given a robot pose $\mathbf{X}$, $\alpha_{\mathbf{V}}(\mathbf{X})$ is non-zero only for a subset $\bar{\mathscr{G}}(\mathbf{X}) \subset \mathscr{G}$ of viewpoints within a distance $\xi_{\max}$ from $\mathbf{X}$.

Condition 2. Let $\mathscr{F} \subset \mathbb{R}^3$ be the unobstructed field of view (FOV) of the sensor in robot frame with homogeneous representation $\underline{\mathscr{F}}$. Also, let $\mathbb{U}(\mathbf{X}) := \cup_{\mathbf{V} \in \bar{\mathscr{G}}(\mathbf{X})} \mathbf{V} \underline{\mathscr{F}} \subset \mathbb{R}^3$ be the union of all FOVs belonging to $\bar{\mathscr{G}}(\mathbf{X})$ in the world frame. This condition states that, for any pair of robot

poses $\mathbf{X}_i$ and $\mathbf{X}_j$ ($i \neq j$) in trajectory $\mathbf{X}_{t+1:t+T}$, we have $\mathbb{U}(\mathbf{X}_i) \cap \mathbb{U}(\mathbf{X}_j) = \emptyset$.

The above conditions are sufficient for decomposing the approximate SMI of trajectory $\mathbf{X}_{t+1:t+T}$ to T independent additive terms, resulting in a computationally feasible trajectory optimization formula. The main idea comes from the fact that, given the above conditions, observations $\mathbf{z}_i$ and $\mathbf{z}_j$ ($i \neq j$) made from viewpoints inside $\mathcal{G}(\mathbf{X}_i)$ and $\mathcal{G}(\mathbf{X}_j)$, respectively, are independent random variables. In practice, active sensors such as lasers have bounded FOV which can meet these conditions. For passive sensors (e.g. cameras), it is commonplace to limit the effective range since the estimation accuracy diminishes as we get farther from the sensor; leading to a limited *applicable* FOV.

Proposition 4. *Under Cond. 1 and 2, the approximated SMI can be expressed as the sum of individual SMI approximations for each pose in the trajectory* $\mathbf{X}_{t+1:t+T}$*:*

$$\tilde{I}(\mathbf{m}; \mathbf{z}_{t+1:t+T} \mid \mathbf{X}_{t+1:t+T}, \mathscr{H}_t) = \sum_{\tau=1}^{T} \tilde{I}(\mathbf{m}; \mathbf{z}_{t+\tau} \mid \mathbf{X}_{t+\tau}, \mathscr{H}_t). \tag{5.8}$$

Proof. See Appendix D. □

The result of Prop. 4 enables computationally feasible trajectory optimization for robot exploration in an unknown environment. Since each term in (5.8) is only dependent upon a single pose in the trajectory, the gradient ascent rule in (5.5) can be directly employed to update each robot pose $\mathbf{X}_{t+\tau}$, $\tau \in \{1, \ldots, T\}$. In the following part, we introduce a practical gradient-based solution to the problem of active mapping stated in (5.3).

5.4 Active Voxel Mapping via Gradient-Ascent

A key advantage of gradient-based optimization is the possibility of adding various reward or penalty terms to the objective function, enabling achievement of a more complex optimization goal. We begin by defining the collision penalty term $C(\mathbf{X}_{t+1:t+T})$ in the objective function of (5.3), which is responsible for driving the optimized robot pose away from obstacles

within the environment. However, since we do not know the map a priori, we resort to the estimation of the map to extract the obstacles.

Definition 5. Let $\hat{\mathscr{E}}_f(p_t(\mathbf{m}))$ be the maximum-likelihood estimation of the free space at time t. For a position $\mathbf{p} \in \mathbb{R}^3$, we define *free distance* as follows:

$$d(\mathbf{p}, p_t(\mathbf{m})) = \min_{\mathbf{b} \in \partial \hat{\mathscr{E}}_f(p_t(\mathbf{m}))} \|\mathbf{p} - \mathbf{b}\|_2. \tag{5.9}$$

It is important to consider that large mutual information occurs near the boundary between the free space and the unknown parts of the map [76]. Therefore, one should seek a balance between large clearance from obstacles and informativeness of observation made from the resulting robot pose. We define the collision cost $C(\mathbf{X}_{t+1:t+T})$ as sum of the log-values of inverse free distance for each pose $\mathbf{X}_{t+\tau}$ with position $\mathbf{p}_{t+\tau}$ in the trajectory:

$$C(\mathbf{X}_{t+1:t+T}) = -\sum_{\tau=1}^{T} \log\left(d(\mathbf{p}_{t+\tau}, p_t(\mathbf{m}))\right). \tag{5.10}$$

Using a logarithmic scale causes a large penalty for poses close to obstacles, while it does not discourage approaching the unknown region from a safe distance due to its suppressed gradient over large inputs.

In addition to the collision cost, we add a penalty term to the objective function to enforce Cond. (2) during each planning phase, minimizing the overlap among the sensor FOVs $\mathbf{X}_{t+\tau}\mathscr{F}$ in the candidate trajectory. We consider a pair-wise penalty term for poses within the trajectory as follows:

$$q(\mathbf{p}_i, \mathbf{p}_j) = \max\left\{0, 2\delta_q - \|\mathbf{p}_i - \mathbf{p}_j\|_2\right\}^2, \qquad \delta_q = |\mathscr{F}| + \xi_{\max}, \tag{5.11}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are robot positions for poses $\mathbf{X}_i$ and $\mathbf{X}_j$, respectively, $|\mathscr{F}|$ is the diameter of $\mathscr{F}$, and $\xi_{\max}$ denotes the maximum distance from robot pose $\mathbf{X}$ to a viewpoint in $\bar{\mathscr{G}}(\mathbf{X})$ (cf. Cond. 1). The penalty term (5.11) effectively discourages the case where the 2-norm ball $\mathscr{B}(\mathbf{p}_i, \delta_q)$ centered around $\mathbf{p}_i$ with radius δ_q coincides with $\mathscr{B}(\mathbf{p}_j, \delta_q)$. Since $\mathscr{B}(\mathbf{p}_i, \delta_q)$ contains

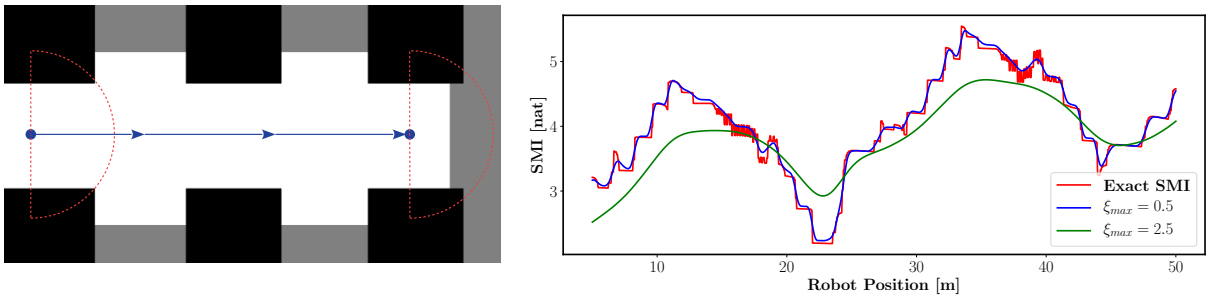


Figure 5.3. Example of the SMI approximation using (5.13). Left: Partially known environment where black, white, and gray regions depict occupied, free, and unknown areas. The robot equipped with a 180° range sensor moves along a straight line shown as the blue arrowed segment. Right: Exact value of the SMI evaluated at each robot position alongside approximate SMI values for various radii ξ_{max}.

$\mathbb{U}(\mathbf{X}_i)$, $q(\mathbf{p}_i, \mathbf{p}_j) = 0$ is sufficient to ensure Cond. (2) is not violated for a pair of poses $\mathbf{X}_i$ and $\mathbf{X}_j$. Note that Cond. (1) is an inherent property of the weighting function $\alpha_{\mathbf{V}}(\mathbf{X})$ and can be evaluated offline. Putting all the components together, the differentiable objective function for gradient-based active mapping is expressed as follows:

$$f(\mathbf{X}_{t+1:t+T}) = \sum_{\tau=1}^{T} \Bigg(\sum_{\mathbf{V} \in \mathscr{G}(\mathbf{X}_{t+\tau})} \alpha_{\mathbf{V}}(\mathbf{X}_{t+\tau}) [I(\mathbf{m}; \mathbf{z} \mid \mathbf{V}, \mathscr{H}_t) + \gamma_c \log(d(\mathbf{v}, p_t(\mathbf{m})))] - \frac{\gamma_q}{2} \sum_{\substack{\tau'=1 \\ \tau' \neq \tau}}^{T} q(\mathbf{p}_{t+\tau}, \mathbf{p}_{t+\tau'}) \Bigg), \tag{5.12}$$

where $\mathbf{v}$, $\mathbf{p}_{t+\tau}$, and $\mathbf{p}_{t+\tau'}$ are the corresponding positions of poses $\mathbf{V}$, $\mathbf{X}_{t+\tau}$, and $\mathbf{X}_{t+\tau'}$.

So far we assumed a general definition for the differentiable weighting function $\alpha_{\mathbf{V}}(\mathbf{X})$ that satisfies Cond. 1 and 2. We use the following definition for $\alpha_{\mathbf{V}}(\mathbf{X})$:

$$\alpha_{\mathbf{V}}(\mathbf{X}) = \frac{\upsilon(\delta(\xi_{\mathbf{X},\mathbf{V}}))(1 + \cos \delta(\xi_{\mathbf{X},\mathbf{V}}))}{\sum_{\mathbf{U} \in \mathscr{G}} \upsilon(\delta(\xi_{\mathbf{X},\mathbf{U}}))(1 + \cos \delta(\xi_{\mathbf{X},\mathbf{U}}))}, \tag{5.13}$$

where $\delta(\xi_{\mathbf{X},\mathbf{V}})$ is the distance between poses $\mathbf{V}, \mathbf{X} \in SE(3)$:

$$\delta(\xi_{\mathbf{X},\mathbf{V}}) = \frac{\pi}{\xi_{\max}} \sqrt{\xi_{\mathbf{X},\mathbf{V}}^{\top} \Gamma \xi_{\mathbf{X},\mathbf{V}}}, \quad \xi_{\mathbf{X},\mathbf{V}} = \log(\mathbf{X}^{-1}\mathbf{V})^{\vee}. \tag{5.14}$$

Here, Γ is a diagonal matrix containing positive coefficients and $\xi_{\mathbf{X},\mathbf{V}}$ is the difference of two *SE(3)* poses in the local frame of $\mathbf{X}$. The indicator function $\upsilon(\delta(\xi_{\mathbf{X},\mathbf{V}}))$ in (5.13) is equal to one only when $0 \leq \delta(\xi_{\mathbf{X},\mathbf{V}}) \leq \pi$ and zero otherwise, which in effect limits the SMI approximation to the viewpoints within the radius $\xi_{\max}$ from $\mathbf{X}$, satisfying Cond. 1. Note that the discontinuity of $\upsilon(\delta(\xi_{\mathbf{X},\mathbf{V}})$ occurs at the same point where $1+\cos\delta(\xi_{\mathbf{X},\mathbf{V}}) = 0$; hence $\alpha_{\mathbf{v}}(\mathbf{X})$ is differentiable with respect to all poses $\mathbf{X} \in$ *SE(3)*. Fig. 5.3 shows an example of the accurate SMI evaluation at different robot positions compared to the approximate SMI of (5.4) using the weighting function of (5.13). Note that, while being differentiable, the approximations follow the occlusion-aware behavior of the exact SMI, i.e. they peak at the positions where more of the unexplored region is visible. With $\alpha_{\mathbf{v}}(\mathbf{X})$ specified as (5.13), we can compute a closed-form expression for the gradient of the approximate SMI as follows.

Proposition 5. *Using the weighting function of* (5.13)*, the gradient of the approximate SMI with respect to the robot pose can be obtained as follows:*

$$
\begin{aligned}
&\nabla_{\psi}\tilde{I}(\mathbf{m};\mathbf{z} \mid \mathbf{X}\exp(\hat{\psi}),\mathscr{H}_t)|_{\psi=0} = \left(\frac{\pi}{\xi_{\max}\eta(\mathbf{X})}\right)^2 \mathscr{J}_R^{\top}(\xi_{\mathbf{X}}) \sum_{\mathbf{V}\in\bar{\mathscr{G}}(\mathbf{X})} \Lambda_{\mathbf{V}}(\mathbf{X})\xi_{\mathbf{X},\mathbf{V}},\\
&\Lambda_{\mathbf{V}}(\mathbf{X}) = [\eta(\mathbf{X})I(\mathbf{m};\mathbf{z} \mid \mathbf{V},\mathscr{H}_t) - \beta(\mathbf{X})]\frac{\sin\delta(\xi_{\mathbf{X},\mathbf{V}})}{\delta(\xi_{\mathbf{X},\mathbf{V}})}\mathscr{J}_R^{\top}(-\xi_{\mathbf{X},\mathbf{V}})\Gamma,\\
&\beta(\mathbf{X}) = \sum_{\mathbf{U}\in\bar{\mathscr{G}}(\mathbf{X})} I(\mathbf{m};\mathbf{z} \mid \mathbf{U},\mathscr{H}_t)(1+\cos\delta(\xi_{\mathbf{X},\mathbf{U}})),\\
&\eta(\mathbf{X}) = |\bar{\mathscr{G}}(\mathbf{X})| + \sum_{\mathbf{U}\in\bar{\mathscr{G}}(\mathbf{X})} \cos\delta(\xi_{\mathbf{X},\mathbf{U}}),\\
&\xi_{\mathbf{X}} = \log(\mathbf{X})^{\vee},
\end{aligned}
\tag{5.15}
$$

where $\bar{\mathscr{G}}(\mathbf{X})$ is the subset of viewpoints $\mathbf{V} \in \mathscr{G}$ that $\upsilon(\delta(\xi_{\mathbf{X},\mathbf{V}})) = 1$, and $\mathscr{J}_R(.)$ is the right Jacobian of SE(3).

Proof. See Appendix E. □

The objective function (5.12) can used with any Bayesian sensing model (5.1). In the

context of occupancy grid mapping via a range sensor, the log-odds technique using a narrow beam [143, Ch. 9] is a natural choice for the sensing model which provides a computationally simple yet accurate representation of the environment. Using such a model, the SMI between the grid map and a beam z can be written as a weighted sum of occlusion probabilities [8, 76, 170]:

$$I(\mathbf{m}; z \mid \mathbf{X}, \mathscr{H}_t) = \sum_{i=1}^{N_\mathbf{X}} w_i(\mathscr{H}_t) p(i \mid \mathscr{H}_t) \tag{5.16}$$

where $N_\mathbf{X}$ is the maximum number of map cells that can be visited by a sensor beam emitted from pose $\mathbf{X}$ and $p(i|\mathscr{H}_t)$ is the probability that the beam is occluded at i-th cell along its path. Hence, the corresponding SMI expression is occlusion-aware, namely the value of the mutual information is directly controlled by whether or not the observations are occluded.

The benefit of using the occlusion-aware SMI formulation of (5.16) comes with the computational burden of ray-tracing during each evaluation of the SMI, since (5.16) requires identifying the map cells along the observation beam. Going back to the objective function (5.12), it is required to perform $B\sum_{\tau=1}^{T} |\hat{\mathscr{G}}(\mathbf{X}_{t+\tau})|$ ray-tracings for a range sensor with B beams during every evaluation of $f(\mathbf{X}_{t+1:t+T})$. Since all viewpoints in $\mathscr{G}$ are located at map cell centers, the ray-tracing for a viewpoint $\mathbf{V}$ with position $\mathbf{v}$ and orientation $\mathbf{R}$ can be obtained from a ray-tracing from the map origin with the same orientation and simply translated by $\mathbf{v}$. Considering that all viewpoints in $\mathscr{G}$ have a fixed set of orientations $\mathscr{R}$, we skip online ray-tracing by computing $|\mathscr{R}|$ ray-tracings from the origin only once and query ray-tracings by applying translation $\mathbf{v}$ for each viewpoint $\mathbf{V}$. To further accelerate the optimization, we avoid repeated evaluations of $I(\mathbf{m}; \mathbf{z} \mid \mathbf{V}, \mathscr{H}_t)$ by caching the values for each viewpoint $\mathbf{V}$ until the map is updated. For computing the free distance $d(\mathbf{v}, p_t(\mathbf{m}))$, we once again use the fact that all viewpoints are located at map cell centers; therefore we can obtain $d(\mathbf{v}, p_t(\mathbf{m}))$ from the distance transform of $\hat{\mathscr{E}}_f(p_t(\mathbf{m}))$ in Def. 5 scaled by the map resolution. This needs to be computed only once for each planning step, since the distance transform provides values of $d(\mathbf{v}, p_t(\mathbf{m}))$ for all cell centers $\mathbf{v}$.

Gradient-based optimization of the objective function (5.12) allows local maximization

Algorithm 4. Active Mapping via Gradient Ascent

Input: $\mathbf{X}_t$, $p_t(\mathbf{m})$, T, $n_{\max}$, $pcrt$

if $pcrt$ is None **then** $pcrt = \text{PRECOMPUTERAYTRACING}()$
$\mathbf{d} = \text{DISTANCETRANSFORM}(p_t(\mathbf{m}))$
$k \leftarrow 0$
$\mathbf{X}^{(k)}_{t+1:t+T} = \text{INITPATH}(\mathbf{X}_t, p_t(\mathbf{m}), T)$
while not $\text{TERMINATE}(f(\mathbf{X}^{(k)}_{t+1:t+T}), n_{\max})$ **do**
  $\mathbf{X}^{(k+1)}_{t+1:t+T} = \text{GRADASCENT}(f(\mathbf{X}^{(k)}_{t+1:t+T}), pcrt, \mathbf{d})$
  $k \leftarrow k+1$
return $\mathbf{X}^{(k)}_{t+1:t+T}$

of trajectory informativeness as well as the distance from obstacles. Given an initial trajectory which can be provided by frontier-based exploration [161], we perform gradient ascent for $n_{\max}$ steps or until the improvement in $f(\mathbf{X}_{t+1:t+T})$ is less than 0.1%. The output of this optimization will be a kinematically feasible trajectory $\mathbf{X}^*_{t+1:t+T} \in SE(3)^T$ which can be tracked by a low-level controller specified by the robot dynamics. Alg. 4 summarizes our procedure for occlusion and collision-aware active mapping. We evaluate the performance of the proposed method in the next section.

5.5 Experiments

In this section we evaluate the performance of our proposed active mapping method in several simulated and real-world experiments in comparison to baseline exploration strategies: frontier-based exploration (Frontier) [161], FSMI [170], SSMI [8], and optimized next best view using RRT (O-NBV-*RRT*) [46]. In Sec. 5.5.1, we perform 2-D active mapping using the proposed method alongside the baselines in a set of 10 randomly generated 2-D environments. Sec. 5.5.2 contains large-scale active mapping using octomap representation [70] in a simulated 3-D Unity environment. Lastly, in Sec. 5.5.3 we demonstrate the performance of our method in a real-world environment using a wheeled ground robot.

Across all experiments, each method uses the log-odds mapping from range measurements [143, Ch. 9] but selects robot trajectories based on their own criteria. Moreover, during

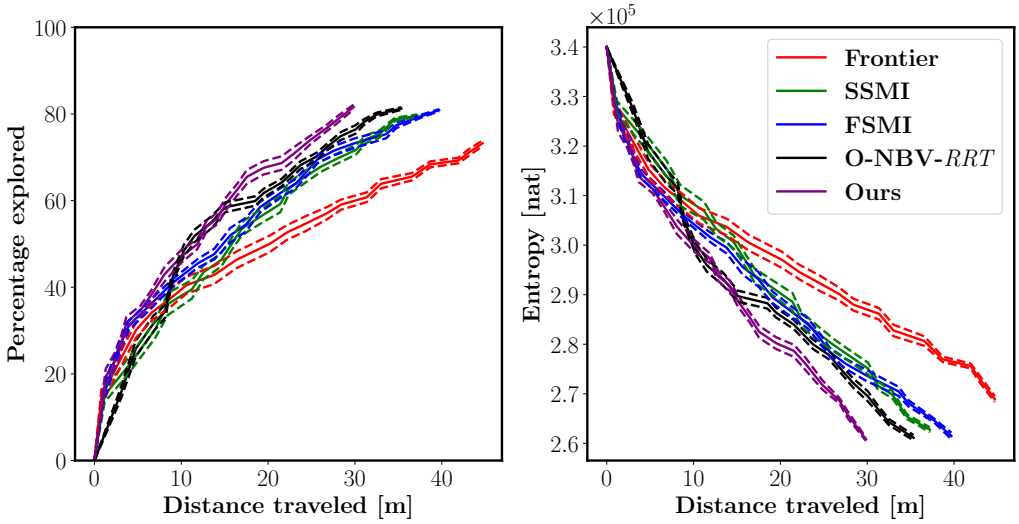
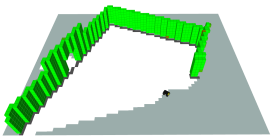
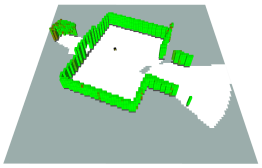


Figure 5.4. 2-D simulated active mapping performance compared among various methods. The results are averaged over 10 randomly generated environments with 3 random starting positions. The dashed lines represent $\pm 0.25\sigma$ deviation from the mean.

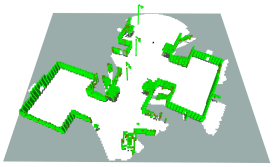
each planning step for 2-D exploration, we identify the frontiers using edge detection over the most likely map at time t, i.e. the mode of $p_t(\mathbf{m})$. Then, we cluster the frontier map cells by detecting the connected components. We plan a path from robot pose $\mathbf{X}_t$ to the center of each frontier using A^* graph search to produce candidate paths for FSMI and SSMI. For Frontier, we pick the path that maximizes the ratio between the frontier size and path length. For O-NBV-*RRT*, we pick the goal of the same path used by Frontier, but instead plan an *RRT* path towards the goal. Our method also uses the same path used by Frontier as the initial trajectory. For exploration in 3-D environments, we first project the most likely octomap at time t onto the ground level in order to derive a 2-D occupancy map and proceed with similar steps as in 2-D exploration. We selected maximum size of neighboring viewpoints $\xi_{\max} = 2$, maximum number of gradient ascent iterations $n_{\max} = 50$, step size $l = 10$, collision penalty weight of $\gamma_c = 5 \times 10^{-4}$, and $\gamma_q = 1$. For Γ in (5.14), we use a diagonal matrix with $[1, 1, 0.1]$ as the elements on the diagonal.



(a) The robot begins exploration.

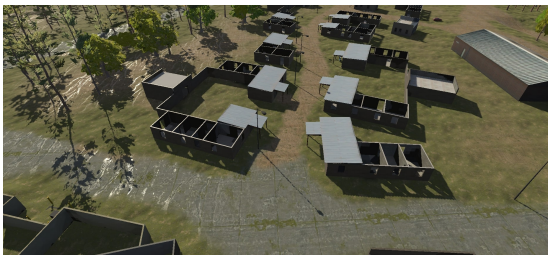


(b) During the first 3 iterations, the robot tries to build the map of its immediate vicinity.



(c) The robot explores the boundaries of the known region at iteration 10.

(d) Octomap after 20 exploration iterations



(e) Photo-realistic Unity simulation environment

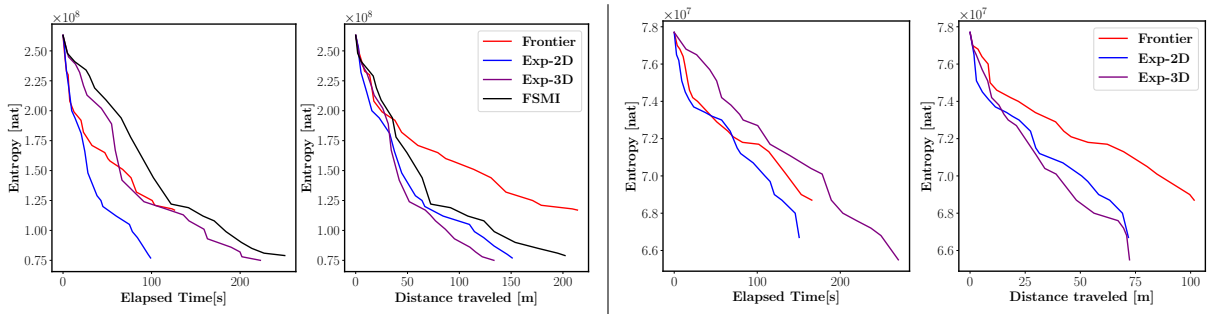
Figure 5.5. Time lapse of 3-D active mapping using the proposed method in simulated Unity environment. The robot receives depth measurements using an RGBD camera, and incrementally builds an octomap as it explores the unknown environment.

5.5.1 2-D Active Mapping in Simulation

Here we compare our proposed method to Frontier, FSMI, SSMI, and O-NBV-*RRT*. The experiments are performed in 10 randomly generated 2-D environments with dimensions $60m \times 60m$ and 3 random starting positions for each instance. We consider a robot equipped with a LiDAR sensor of range $10m$ and 90° field of view where each measurement beam is added with Gaussian noise of $\mathcal{N}(0, 0.1)$. Fig. 5.4 shows the 2-D simulation results among all methods. We witness superior exploration performance of our proposed method, which can be attributed to the fact that our method applies occlusion-aware optimization for each pose in a candidate trajectory; resulting in high utilization of information gathering opportunities. The subpar performance of O-NBV-*RRT* can be caused by the fact that O-NBV-*RRT* does not account for sensor noise during pose optimization. Robot trajectories obtained by Frontier, FSMI, and SSMI are computed using A^* search, and no further optimization is performed over single robot poses along the trajectory; resulting in a very restricted set of candidate solutions.

5.5.2 Exploration in 3-D Unity Simulation

In this part, we test our proposed method in a photo-realistic 3-D Unity simulated environment, shown in Fig. 5.5e. We use a Husky robot equipped with a depth camera that provides a 3-D point cloud used for building an octomap. Octomap [70] is occupancy grid mapping method based on octree data structure and the log-odds technique that provides a scalable way to store the 3-D map for large environments through compressing the map cells with similar occupancy probability. For experiments in 3-D environments, we operate our proposed method in two different modes. In the first mode (Exp-2D), we project both the octomap and 3-D point clouds onto the $z = 0$ surface, obtaining a 2-D representation of the map and the observations. Then we compute informative trajectories similar to the 2-D active mapping in Sec. 5.5.1. By executing the optimal trajectory, we receive new 3-D point cloud observations that incrementally update the octomap. The first mode of active mapping only requires ray-tracing in a 2-D grid map, saving computation time while trading for accuracy in evaluating mutual information. In the second mode (Exp-3D), we directly use the 3-D sensor model for ray-tracing within the octomap. This mode of operation is expected to evaluate mutual information more accurately since the exact sensor model and 3-D occlusion have been taken into account. Fig. 5.5 shows several exploration iterations of active mapping process using Exp-3D. Furthermore, we compare our method with FSMI, where run length encoding is utilized for accelerated evaluation of the SMI over an octree. Fig. 5.6a shows simulation results for experiments in the 3-D Unity environment for 20 exploration iterations. We observe Exp-2 finds a good balance between exploration efficiency and computation time; while Exp-3D is slightly more efficient in terms of distance traveled, it takes longer computation time to perform exploration compared to Exp-2D. This comes from the fact that the objects in the simulation have a uniform profile in the z direction, hence the information in the 2-D and 3-D maps are almost the same. Table. 5.1 (middle column) compares the average clearance from the obstacles among different methods. Compared to Frontier, it is clear that the proposed method chooses



(a) Unity simulation **(b)** Real-world

Figure 5.6. 3-D active mapping performance compared between Frontier, FSMI, and our method for 20 exploration iterations. Exp-2D uses a projected 2-D perception model for planning, while Exp-3D performs ray-tracing directly on the octomap.

Table 5.1. Clearance from obstacles compared between Frontier and our proposed method averaged over 20 exploration iterations.

Algorithm	3-D Unity Simulation	Real World
Frontier	1.2 m	0.48 m
Exp-2D	***3.9 m***	1.7 m
Exp-3D	3.4 m	***1.84 m***

informative trajectories that avoid approaching obstacles.

5.5.3 Real-World Experiments

We deployed Exp-2D and Exp-3D on a ground wheeled robot to autonomously explore an indoor environment. The robot was equipped with an NVIDIA Xavier NX GPU, a Hokuyo UST-10LX LiDAR, and an Intel RealSense D435i RGBD camera. We implemented our software stack using *Robot Operating System* (ROS) [120]. Robot localization was carried out using ICP scan matching of LiDAR measurements [31]. 3-D point clouds from the depth images were used to build an octomap. The complete implementation was able to update the octomap every $0.11s$ on average. Fig. 5.6b and Table. 5.1 (right column) show the performance of active mapping for 20 exploration iterations and the average clearance from obstacles, respectively. The real-world experiments confirm the findings in the simulations in terms of the efficiency of the proposed method.

5.6 Summary

This chapter developed a differentiable approximation of the Shannon mutual information between a probabilistic occupancy grid map and range sensor measurements. Our formulation enables gradient-based optimization of informative occlusion-aware sensing trajectories in 3-D and allows the inclusion of additional differentiable penalty terms, such as collision cost. We demonstrated in simulated and real-world experiments that our method outperforms the state-of-the-art techniques due to its ability to optimize the sensing views in continuous space. In the next chapter, we will investigate extending the proposed method to multiple agents, where we expect gradient-based optimization to also be significantly more efficient than discrete space search.

Acknowledgments

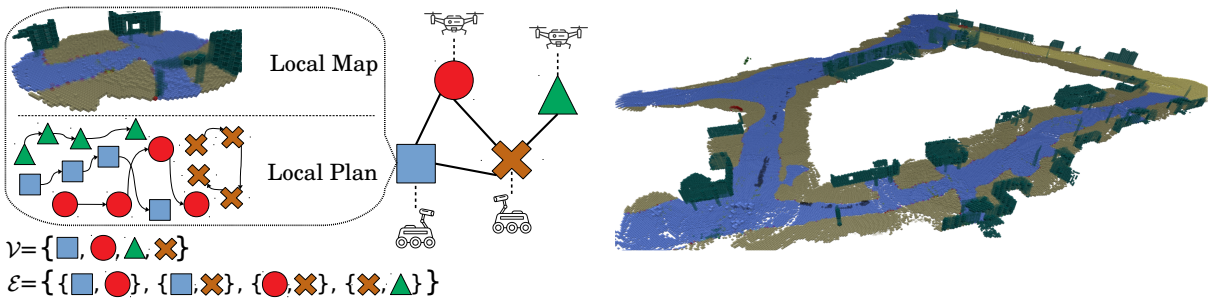
Chapter 5, in part, is a reprint of the material as it appears in A. Asgharivaskasi, S. Koga, and N. Atanasov, "Active mapping via gradient ascent optimization of Shannon mutual information over continuous *SE(3)* trajectories," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 12994-13001, 2022. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Distributed Multi-Robot Semantic Active Mapping

The ability to explore unknown environments and discover objects of interest is a prerequisite for autonomous execution of complex tasks by mobile robots. Active mapping methods [117] consider joint optimization of the motion of a robot team and the fidelity of the map constructed by the team. The goal is to compute maximally informative robot trajectories under a limited exploration budget (e.g., time, energy, etc.).

Time-critical applications, such as search and rescue [119, 124] and security and surveillance [65], as well as large-scale operations, such as environmental monitoring [73], substantially benefit if exploration is carried out by a team of coordinating robots. This is traditionally done via multi-robot systems relying on centralized estimation and control [5, 50, 53, 73, 107], where each robot receives local sensor observations, builds its own map, and sends it to a central node for map aggregation and team trajectory computation. The availability of powerful computation onboard small robot platforms makes it possible to develop autonomous exploration algorithms without the need for a central processing node [89]. Removing the central unit improves the resilience of a multi-robot system with respect to communication-based faults and central node failures [118] but brings up new challenges related to distributed storage, computation, and communication. How can one guarantee that the performance of decentralized active mapping would be on par with a centralized architecture in terms of global map consistency and team



(a) Robot network and the communicated messages

(b) Local maps become globally consistent over time

Figure 6.1. Overview of our distributed multi-robot active mapping approach. (a) A team of robots, denoted by vertex set $\mathcal{V}$, collaboratively explores an unknown environment. Each robot builds a local map using onboard sensor measurements and computes a local plan for the team, with the goal of maximizing the collective information gathered by the team. The local maps and plans are communicated over a peer-to-peer network whose connectivity is represented by the edge set $\mathcal{E}$. (b) As the robot team continues communication, the local maps stored by different robots become globally consistent in that they store similar information about the environment.

trajectory optimality?

To address this question, in this chapter we propose ROAM: **R**iemannian **O**ptimization for **A**ctive **M**apping with robot teams. Based on the papers [9] and [7], ROAM is a decentralized Riemannian optimization algorithm that operates on a communication graph with node variables belonging to a Riemannian manifold and ensures consensus among the node variables. The graph nodes correspond to different robots, while the graph edges model the communication among the robots. In the context of mapping, the node variables are categorical PMFs representing probabilistic maps with different semantic classes (e.g., building, vegetation, terrain) at each robot. The consensus constraint requires that the local maps of different robots agree with each other. In the context of planning, the node variables are trajectories of $SE(3)$ robot poses. Each robot plans trajectories for the whole team using its local information, while the consensus constraint requires that the team trajectories computed by different robots agree. See Fig. 6.1 for an overview of ROAM.

We demonstrate the performance of ROAM in a variety of simulation and real-world experiments using a team of wheeled robots with on-board sensing and processing hardware.

Specifically, each robot gathers range and semantic segmentation measurements using an RGBD sensor, and incrementally builds a local 3-D semantic grid map of the environment, where each map cell maintains a probability distribution over object classes. To achieve memory and communication efficiency, an octree data structure is employed to represent the 3-D semantic maps [8]. The robots cooperatively find the most informative set of $SE(3)$ paths for the team to efficiently improve the map and explore the unknown areas while avoiding obstacle collisions. Both multi-robot mapping and planning are performed in the absence of a central estimation and control node and only involve peer-to-peer communication among neighboring robots.

Our gradient-based distributed Riemannian optimization approach extends the scope of previous works in multi-robots estimation and planning to enable continuous non-Euclidean state and control spaces, as well as non-linear and non-Gaussian perception models. Our main **contributions** include:

1. a distributed Riemannian optimization algorithm for multi-robot systems using only one-hop communication, with consensus and optimality guarantees,

2. a distributed semantic octree mapping approach utilizing local semantic point cloud observations at each robot,

3. a distributed collaborative planning algorithm for robot exploration, where the search domain is defined as the continuous space of $SE(3)$ robot pose trajectories,

4. an open-source implementation,[1] achieving real-time performance onboard resource-constrained robots in simulation and real-world experiments.

6.1 Related Works

Multi-robot active mapping is in essence an optimization problem, with the goal of finding maximally informative robot trajectories, while simultaneously maintaining globally

[1] Open-source software and videos supplementing this chapter are available at https://existentialrobotics.org/ROAM_webpage/.

consistent map estimates. Thus, we begin our literature review with identifying relevant works in distributed optimization. The algorithms introduced in [37, 60, 110, 121, 148] provide a class of approaches for decentralized gradient-based optimization in the Euclidean space under a variety of constraints such as time variation or communication asymmetry between agents in the network. The survey by Halsted et al. [66] provides a comprehensive study of distributed optimization methods for multi-robot applications. In this work, we decompose the task of multi-robot active mapping to two consensus-constrained Riemannian optimization problems, i.e. distributed mapping and distributed path planning. However, naive utilization of the Euclidean optimization techniques in Riemannian manifolds might violate the structure of the optimization domain, leading to infeasible solutions. Therefore, it is required to employ a special family of distributed optimization methods specific to Riemannian manifolds.

Absil et al. [1] presents the foundations of optimization over matrix manifolds, giving rise to many centralized and distributed algorithms in subsequent works. As examples, Chen et al. [39] and Wang et al. [156] devise decentralized optimization algorithms for Stiefel manifolds where a Lagrangian function is used to enforce consensus and maintain the manifold structure. The papers by Sarlette et al. [84, 126, 127] formulate consensus (i.e. synchrony) as the objective function of a distributed optimization problem. Our work allows inclusion of an additional objective function, while simultaneously incorporating consensus as a constraint of the optimization. The work in [127] formulates coordination of rigid body attitudes as a distributed optimization problem, where the cost function is the sum of chordal distances between each pair of agents. The same problem, for the more general context of Lie groups, have been studied in [126] and [84]

Manifold optimization also allows designing efficient learning algorithms where model parameters can be learned using unconstrained manifold optimization as opposed to Euclidean space optimization with projection to the parameter manifold. Zhang et al. [167] and Li et al. [87] introduce stochastic learning algorithms for Riemannian manifolds in centralized and federated formats, respectively. Related to our work, Tian et al. [146] present a multi-robot pose-graph SLAM algorithm which employs gradient-descent local to each robot directly over the $SE(3)$

space of poses. Our work is inspired by the distributed Riemannian gradient optimization method introduced by Shah [132]. We develop a distributed gradient-descent optimization method for general Riemannian manifolds, and derive specific instantiations for two particular cases, namely the space of probability distributions over semantic maps and the space of $SE(3)$ robot pose trajectories.

Distributed mapping is a special case of distributed estimation, where a model of the environment is estimated via sensor measurements. Distributed estimation techniques are used in multi-robot localization [17], multi-robot mapping [176], or multi-robot SLAM [145]. Paritosh et al. [114] define Bayesian distributed estimation as maximizing sensor data likelihood from all agents, while enforcing consensus in the estimates. The present work follows a similar methodology in that we achieve multi-robot Bayesian semantic mapping via distributed maximization of local sensor observation log-likelihood with a consensus constraint on the estimated maps.

Regarding collaborative mapping, an important consideration is the communication of local map estimates among the robots. Corah et al. [43] propose distributed Gaussian mixture model (GMM) mapping, where a GMM map is globally estimated, and each robot uses this global map to extract occupancy maps for planning. The use of GMM environment representation for multi-robot exploration is motivated by its lower communication overhead compared to uniform resolution occupancy grid maps. Subsequent works in [49] and [160] have similarly used distributed GMM mapping for place recognition and relative localization alongside exploration. Alternative techniques for communication-efficient multi-robot mapping include distributed topological mapping [169], sub-map-based grid mapping [164], and distributed truncated signed distance field (TSDF) estimation [51]. More recently, the work in [48] extends neural implicit signed distance mapping to a distributed setting via formulating multi-robot map learning as a consensus-constrained minimization of the loss function. In this case, the robots need to share the neural network parameters to achieve consensus. In our work, we use a semantic octree data structure introduced in our prior work [8] to alleviate the communication burden by using a

lossless octree compression. Relevant to our work, the authors in [74] propose merging of two binary octree maps via summing the occupancy log-odds of corresponding octree leaves. Our work distinguishes itself from [74] through a different formulation of multi-robot mapping as a consensus-based Riemannian optimization problem, which enables *a*) extension to multi-class octree representations, and *b*) combination of map merging with online map updates from local observations.

Similar to multi-robot mapping, many multi-robot planning methods utilize distributed optimization techniques. The work in [92] outlines various trajectory planning methods used in multi-robot systems, including graph-based, sampling-based, model-based, and bio-inspired approaches. In particular, graph neural networks (GNNs) have been utilized in [64, 172] for learning to extract, communicate, and accumulate features from local observations in the context of collaborative multi-robot planning in a distributed way. Coordination and plan deconfliction for multi-robot cooperative tasks is discussed in [159], where robots are assigned priorities in a decentralized manner in order to reach a Pareto equilibrium. In our work, we introduce a decentralized gradient-based negotiation mechanism to resolve $SE(3)$ path conflicts.

Path planning for autonomous exploration has been extensively studied in the field of active SLAM. Atanasov et al. [14] propose a distributed active SLAM method for robots with linear-Gaussian observation models and a finite set of admissible controls. The authors exploit the conditional entropy formula for the Gaussian noise model to derive an open-loop control policy, called reduced value iteration (RVI), with the same performance guarantees as a closed-loop policy. An anytime version of RVI is proposed in [129] using a tree search that progressively reduces the suboptimality of the plan. In contrast to [14, 129], we use a probabilistic range-category observation model that accounts for occlusion in sensing. Sampling-based solutions to multi-robot active SLAM have been presented in [78] and [150], with asymptotic optimality guarantees. Cai et al. [27] consider collision safety and energy as additional factors in the cost function for active SLAM using a heterogeneous team of robots. Zhou and Kumar [171] propose robust multi-robot active target tracking with performance guarantees in regard to sensing and

communication attacks, however, the estimation and control are carried out centrally. Tzes et al. [149] develop a learning-based approach for multi-robot target estimation and tracking, used a GNN to accumulate and process information communicated among one-hop neighbors. The works in [88, 165] aim to maintain multi-robot network connectivity and collision avoidance via control barrier functions. Another line of research [19, 41] uses decentralized Monte-Carlo tree search for multi-robot path planning for exploration. The interested reader is encouraged to refer to [117] for a comprehensive survey of active SLAM methods. Our work distinguishes itself by considering continuous-space planning on a Riemannian manifold, generalizing the previous works in terms of the finite number of controls and the Euclidean robot states.

Related to active SLAM with continuous-space planning, Koga et al. [80, 82] introduce iterative covariance regulation, an $SE(3)$ trajectory optimization algorithm for single-robot active SLAM with a Gaussian observation model. Model-based [162] and model-free [163] deep reinforcement learning techniques have been applied to similar single-robot active SLAM problems. Extending to a team of robots, Hu et al. [72] propose Voronoi-based decentralized exploration using reinforcement learning, where coordination among the robots takes place via distributed assignment of each Voronoi region to a robot, and the policy generates a 2-D vector of linear and angular velocities. In our work, we formulate multi-robot planning for exploration as a distributed optimization problem in $SE(3)$ space with a consensus constraint to enforce agreement among the robot plans.

6.2 Problem: Consensus-Constrained Riemannian Optimization for Multi-Agent Systems

We begin our analysis by formulating consensus-constrained Riemannian optimization for multi-agent systems. Consider a network of agents represented by an undirected connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ encodes the existence of communication links between pairs of agents. Each agent $i \in \mathcal{V}$ has state x^i which belongs to

a compact Riemannian manifold $\mathscr{M}$. Let $T_{x^i}\mathscr{M}$ denote the tangent space of $\mathscr{M}$ at x^i and let $\langle v,u\rangle_{x^i} \in \mathbb{R}$ with $u,v \in T_{x^i}\mathscr{M}$ be a Riemannian metric on $\mathscr{M}$. The norm of a tangent vector $v \in T_{x^i}\mathscr{M}$ is defined by the Riemannian metric as $\|v\|_{x^i} = \sqrt{\langle v,v\rangle_{x^i}}$. Additionally, let $\mathrm{Exp}_{x^i}(\cdot) : T_{x^i}\mathscr{M} \to \mathscr{M}$ denote the exponential map on $\mathscr{M}$ at x^i, and denote its inverse as $\mathrm{Exp}_{x^i}^{-1}(\cdot) : \mathscr{M} \to T_{x^i}\mathscr{M}$.

We associate a local objective function $f^i(\cdot) : \mathscr{M} \to \mathbb{R}$ with each agent $i \in \mathscr{V}$. Our goal is to maximize the cumulative objective function over the joint agent state $\mathbf{x} = (x^1,\ldots,x^{|\mathscr{V}|})$:

$$F(\mathbf{x}) = \frac{1}{|\mathscr{V}|}\sum_{i\in\mathscr{V}} f^i(x^i). \tag{6.1}$$

The global objective can be maximized using $|\mathscr{V}|$ independent local optimizations. However, in many applications it is necessary to find a common solution among all agents. For example, in multi-robot mapping, the robots need to ensure that their local maps are consistent and take into account the observations from other robots. Therefore, the global optimization problem needs to be constrained such that the agents reach consensus on $\mathbf{x}$ during optimization. For this aim, we define an aggregate distance function $\phi(\mathbf{x}) : \mathscr{M}^{|\mathscr{V}|} \to \mathbb{R}_{\geq 0}$:

$$\phi(\mathbf{x}) = \sum_{\{i,j\}\in\mathscr{E}} A_{ij} d^2(x^i,x^j), \tag{6.2}$$

where A is a symmetric weighted adjacency matrix corresponding to the graph $\mathscr{G}$, and $d(\cdot) : \mathscr{M}\times\mathscr{M} \to \mathbb{R}_{\geq 0}$ is a distance function on the Riemannian manifold $\mathscr{M}$, i.e., computes the length of the geodesic (shortest path) between pairs of elements in $\mathscr{M}$. The definition of the aggregate distance function in (6.2) implies that consensus will be reached if and only if $\phi(\mathbf{x}) = 0$. Hence, adding $\phi(\mathbf{x}) = 0$ as a constraint to (6.1) would require feasible joint states $\mathbf{x} = (x^1,\ldots,x^{|\mathscr{V}|})$ to satisfy $x^i = x^j$ for all $i,j \in \mathscr{V}$.

Problem 4. Consider a connected graph $\mathscr{G} = (\mathscr{V},\mathscr{E})$ where each node $i \in \mathscr{V}$ represents an agent with state $x^i \in \mathscr{M}$ and local objective function $f^i(x^i)$. Find a joint state $\mathbf{x}$ that maximizes the

following objective function:

$$\begin{aligned} \max_{\mathbf{x}} \quad & F(\mathbf{x}) = \frac{1}{|\mathscr{V}|} \sum_{i \in \mathscr{V}} f^i(x^i), \\ \text{s.t.} \quad & x^i \in \mathscr{M}, \ \forall i \in \mathscr{V}, \ \text{and} \ \phi(\mathbf{x}) = 0, \end{aligned} \tag{6.3}$$

where $\phi(\mathbf{x}) = 0$ is the consensus constraint defined in (6.2).

In the next section, we develop a distributed gradient-based optimization algorithm to solve (6.3) using only local computation and single-hop communication, with consensus and optimality guarantees. As we discuss in Sec. 6.4 and Sec. 6.5, both multi-robot mapping and multi-robot trajectory optimization can be formulated as consensus-constrained optimization problems as in (6.3). In the case of mapping, the manifold $\mathscr{M}$ is the probability simplex capturing map density functions while the local objective $f^i(x^i)$ is the log-likelihood of the observations made by robot i. In the case of trajectory optimization, $\mathscr{M}$ represents the space of 3-D pose (rotation and translation) trajectories in $SE(3)$, and $f^i(x^i)$ is a collision and perception-aware objective for the robot pose trajectories. Lastly, in Sec. 6.6 we evaluate the performance of our proposed distributed multi-robot exploration in several simulation and real-world experiments.

6.3 Distributed Riemannian Optimization

The problem in (6.3) has a specific structure, maximizing a sum of local objectives subject to a consensus constraint among all x^i, $i \in \mathscr{V}$. We develop a distributed gradient-based algorithm to solve (6.3). The idea is to interleave gradient updates for the local objectives with gradient updates for the consensus constraint at each agent. Alg. 5 formalizes this idea. The update step in line 4 guides the local state x^i towards satisfaction of the consensus constraint, with a step size of ε. The gradient of $\phi(\mathbf{x})$ with respect to x^i, denoted as $\text{grad}_{x^i} \phi(\mathbf{x})$, lies in the tangent space $T_{x^i}\mathscr{M}$. Hence, the exponential map is used to retract the gradient update $-\varepsilon \, \text{grad}_{x^i} \phi(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{(k)}}$ to the manifold $\mathscr{M}$. The gradient $\text{grad}_{x^i} \phi(\mathbf{x})$ can be expressed as a sum of

Algorithm 5. Distributed Riemannian Optimization

Input: Network $\mathcal{G}(\mathcal{V},\mathcal{E})$ and initial state $x^{i(0)}$
Output: Consensus optimal solution to (6.3)
1: **for** $k \in \mathbb{Z}_{\geq 0}$ **do**
2: **for** each agent $i \in \mathcal{V}$ **do**
3: ▷ Promote consensus with step size ε:
4: $\tilde{x}^{i(k)} = \mathrm{Exp}_{x^{i(k)}}\left(-\varepsilon \,\mathrm{grad}_{x^i}\, \phi(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{(k)}}\right)$
5: ▷ Optimize local objective with step size $\alpha^{(k)}$:
6: $x^{i(k+1)} = \mathrm{Exp}_{\tilde{x}^{i(k)}}\left(\alpha^{(k)}\,\mathrm{grad}\, f^i(x^i)|_{x^i=\tilde{x}^{i(k)}}\right)$
7: **return** $x^{i(k)}$

gradients with respect to the neighbors $\mathcal{N}_i = \{j | A_{ij} > 0\}$ of agent i:

$$\mathrm{grad}_{x^i}\,\phi(\mathbf{x}) = \sum_{j\in\mathcal{N}_i} A_{ij}\,\mathrm{grad}_{x^i}\, d^2(x^i,x^j) = -2\sum_{j\in\mathcal{N}_i} A_{ij}\,\mathrm{Exp}_{x^i}^{-1}(x^j).$$

Therefore, line 4 requires *only single-hop communication* between agent i and its neighbors $\mathcal{N}_i$. Line 6 carries out an update with step size $\alpha^{(k)}$ in the direction of the gradient of the local objective $f^i(\cdot)$, computed at the updated state $\tilde{x}^{i(k)}$. Similar to the consensus update step, the exponential map is used in to retract $\mathrm{grad}\, f^i(x^i)$ and apply it to the point $\tilde{x}^{i(k)}$. Line 6 is local to each agent i and does not require communication. The two update steps are continuously applied until a maximum number of iterations is reached or the update norm is smaller than a threshold.

Example. Consider a sensor network where several agents gather data that is not supposed to be shared over the network, due to either privacy reasons or bandwidth limitations. Our Riemannian optimization algorithm enables distributed processing of the global data, accumulated over all agents, without actually sharing the data. As an example, Fig. 6.2 illustrates applying Alg. 5 to compute the leading eigenvector of the covariance of data. Fig. 6.2a depicts the global data distribution $Z = [Z_1^\top Z_2^\top]^\top$, such that different segments of the data Z_1 and Z_2 are known to agent

1 and agent 2, separately. This problem can be formulated as:

$$\begin{aligned} \max_{x^1,x^2} \quad & \sum_{i\in\{1,2\}} (Z_i x^i)^\top Z_i x^i, \\ \text{s.t.} \quad & x^1, x^2 \in \mathbb{S}^1 \ \text{ and } \ \arccos({x^1}^\top x^2) = 0, \end{aligned} \tag{6.4}$$

where the domain manifold is the unit circle $\mathbb{S}^1$, and cosine distance is used as the distance function. Note that for all x^1 and x^2 that satisfy the consensus constraint $\arccos({x^1}^\top x^2) = 0$, the objective function is equivalent to the one for the centralized leading eigenvector problem. Hence, we expect to find the eigenvector for the covariance of the global data matrix Z by employing Alg. 5 to (6.4). Fig. 6.2b shows an initialization of x^1 and x^2 over the unit circle $\mathbb{S}^1$. While the Riemannian gradients of $\phi(\cdot)$ and $f^i(\cdot)$ are tangent vectors to $\mathbb{S}^1$, the consensus and local objective function update steps keep the state on the $\mathbb{S}^1$ manifold thanks to the exponential map (see the circular arcs in Fig. 6.2c):

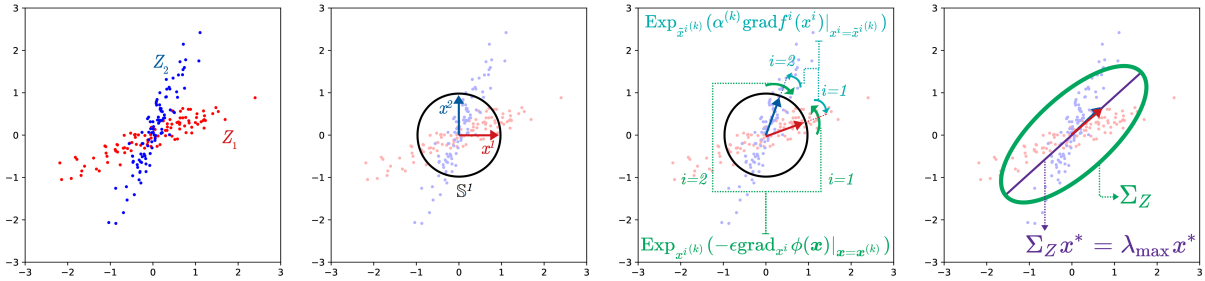
$$\mathrm{Exp}_{x^i}(v) = \cos(\sqrt{v^\top v})x^i + \sin(\sqrt{v^\top v})\frac{v}{\sqrt{v^\top v}}. \tag{6.5}$$

Note that the consensus update (green arc) acts in the direction of agreement between x^1 and x^2, whereas the local objective function gradient tries to steer the states x^i towards the leading eigenvector of their respective data Z_i. Although each agent has only partial access to Z, both x^1 and x^2 eventually converge to x^*, namely the leading eigenvector of the covariance for the global data matrix Z, as Fig. 6.2d suggests. •

Next, we study whether Alg. 5 achieves consensus and optimality. We make several assumptions to ensure that the problem is well-posed in accordance with prior work on distributed optimization [37, 60, 86, 108, 110, 121, 131, 137, 148].

Definition 6. A differentiable function $f : \mathscr{M} \to \mathbb{R}$ is geodesically convex if and only if for any $x, y \in \mathscr{M}$:

$$f(x) \geq f(y) + \langle \mathrm{grad}\, f(y), \mathrm{Exp}_y^{-1}(x) \rangle_y.$$



(a) Data distribution **(b)** State initialization **(c)** Update steps **(d)** Convergence

Figure 6.2. Application of Alg. 5 to the leading eigenvector problem. (a) Data distribution $Z = [Z_1^\top Z_2^\top]^\top$, where Z_1 and Z_2 are separately known to agent 1 and agent 2. (b) State initialization, limited to the unit circle $\mathbb{S}^1$ since we are only interested in eigenvector directions. (c) Consensus and local objective function updates, shown in green and teal arcs, respectively. The exponential mapping of $\mathbb{S}^1$ maintains the manifold structure of the states throughout the update steps. (d) Convergence to the global leading eigenvector, where the level-set of the covariance matrix of the global data Z is shown, alongside its leading eigenvector x^*.

See Sec. 2.5 of Chapter 2 for an equivalent definition of geodesic convexity. The function f is geodesically concave if the above inequality is flipped.

Assumption 1. Assume the following statements hold for Problem 4 and Alg. 5.

- The Riemannian manifold $\mathscr{M}$ is compact.
- The local objective functions f^i, $\forall i \in \mathscr{V}$, are smooth, geodesically concave, and their Riemannian gradients are bounded by some constant C:

$$\|\operatorname{grad} f^i(x^i)\|_{x^i} \leq C,\ \forall x^i \in \mathscr{M},\ \forall i \in \mathscr{V}.$$

- The weighted adjacency matrix A of the graph $\mathscr{G}$ is row-stochastic, i.e., $\sum_{j\in\mathscr{V}} A_{ij} = 1$.
- The squared distance function $d^2 : \mathscr{M} \times \mathscr{M} \to \mathbb{R}_{\geq 0}$ is geodesically convex.
- The step sizes $\alpha^{(k)} > 0$ for the update step in line 6 satisfy the Robbins-Monro conditions:

$$\sum_{k=0}^{\infty} \alpha^{(k)} = \infty, \qquad \sum_{k=0}^{\infty} {\alpha^{(k)}}^2 < \infty, \qquad \forall k \geq 0.$$

In addition to the assumptions above, we require an additional condition to prove that Alg. 5 achieves consensus, i.e., $\phi(\mathbf{x}) = 0$. Let $\mathscr{T}_{y^i}^{x^i} : T_{y^i}\mathscr{M} \to T_{x^i}\mathscr{M}$ denote parallel transport [22, Ch.10] from the tangent space at y^i to the tangent space at x^i. For points $x^i, x^j, y^j, y^i \in \mathscr{M}$, consider the geodesic loop $x^i \to x^j \to y^j \to y^i \to x^i$ with corresponding tangent vectors $v_x^{ij}, v_y^{ij}, v_{xy}^i, v_{xy}^j$ defined as:

$$v_x^{ij} = \mathrm{Exp}_{x^i}^{-1}(x^j),\ v_{xy}^i = \mathrm{Exp}_{x^i}^{-1}(y^i),$$

with similar definitions for v_y^{ij} and v_{xy}^j. Let $v_{xy}^{ij} \in T_{x^i}\mathscr{M}$ be the net tangent vector transported to $T_{x^i}\mathscr{M}$:

$$v_{xy}^{ij} = v_x^{ij} + \mathscr{T}_{x^j}^{x^i} v_{xy}^j - v_{xy}^i - \mathscr{T}_{y^i}^{x^i} v_y^{ij}. \tag{6.6}$$

Assumption 2. For a $\rho > 0$ and any 4-tuple $(x^i, x^j, y^j, y^i) \in \mathscr{M}$, assume the norm of the net tangent vector v_{xy}^{ij} is bounded by the lengths of the opposite geodesics along the loop:

$$\|v_{xy}^{ij}\|_{x^i} \leq \rho \min\{\|v_{xy}^i\|_{x^i} + \|v_{xy}^j\|_{x^j}, \|v_x^{ij}\|_{x^i} + \|v_y^{ij}\|_{y^i}\}. \tag{6.7}$$

In Euclidean space, the net tangent vector v_{xy}^{ij} is equivalent to zero linear displacement; hence, the assumption holds for any $\rho \geq 0$. Similarly, the manifold $\mathbb{S}^1$ of example (6.4) satisfies (6.7) due to zero angular displacement. For a general case, v_{xy}^{ij} can be non-zero due to the curvature of the manifold. This is dual to the fact that, for a zero net tangent vector v_{xy}^{ij}, the corresponding geodesics might not form a loop. The assumption in (6.7) essentially imposes a condition over curvature of the manifold so that the norm of v_{xy}^{ij} is limited by the length of the geodesic loop. Based on the above assumptions, we show consensus and optimality for Alg. 5.

Theorem 1. *Consider the consensus-constrained Riemannian optimization problem in* (6.3) *and the distributed Riemannian optimization algorithm in Alg. 5. Suppose that Assumptions 1 and 2 hold and step size ε is chosen such that $\varepsilon \in (0, 2/L)$ with $L = 4(1+\rho)$. Then, Alg. 5 provides a solution to* (6.3) *with the following properties.*

1. *The joint state $\mathbf{x}^{(k)}$ converges to $\mathbf{x}^{(\infty)} \in \mathscr{M}^{|\mathscr{V}|}$, where $\mathbf{x}^{(\infty)}$ is a consensus configuration, i.e.,*

$x^{i(\infty)} = x^{j(\infty)}$ *for all* $i, j \in \mathcal{V}$.

2. *Let* $\mathbf{x}^*$ *be an optimal solution to* (6.3)*. The optimal value* $F(\mathbf{x}^*)$ *is a lower-bound for the maximum of* $F(\mathbf{x}^{(k)})$ *across all iterations:*

$$F(\mathbf{x}^*) \leq \lim_{k_{max} \to \infty} \max_{0 \leq k \leq k_{max}} F(\mathbf{x}^{(k)}). \tag{6.8}$$

Proof. See Appendix F. □

We stress that, while the optimal solution $\mathbf{x}^*$ and the convergence point $\mathbf{x}^{(\infty)}$ of Alg. 5 are both consensus configurations, the optimality bound of (6.8) can potentially admit a solution $\mathbf{x}^{(k)}$ that does not satisfy the consensus constraint. For the Euclidean case, Nedić [109, Ch.5] shows that $d^2(\mathbf{x}^*, \mathbf{x}^{(k)})$ is a Lyapunov function, and subsequently, $F(\mathbf{x}^*) = F(\mathbf{x}^{(\infty)})$ holds. However, a similar derivation for $d^2(\mathbf{x}^*, \mathbf{x}^{(k)})$ has not been found for a general Riemannian manifold, due to the complexity added by the curvature.

Alg. 5 establishes consensus and an optimality bound without requiring identical initial states $x^{i(0)} = x_0$ for all $i \in \mathcal{V}$ or parallel transport of the gradients between neighboring agents. Hence, our distributed Riemannian optimization provides an approach to solve multi-robot problems with communication constraints. The main requirement to use Alg. 5 is to express a multi-robot optimization problem in the form of (6.3), with local objectives $f^i(\cdot)$ and distance measure $\phi(\cdot)$ defined as smooth concave and convex functions in $\mathscr{M}$ and $\mathscr{M}^{|\mathcal{V}|}$, respectively. In the absence of concavity for the objective functions or convexity for the consensus constraint, Alg. 5 can still be utilized to obtain a solution with local consensus and optimality guarantees.

In the next two sections, we apply Alg. 5 to achieve simultaneous multi-robot mapping and planning. We refer to our approach as *Riemannian Optimization for Active Mapping (ROAM)*. In Sec. 6.4, we apply Alg. 5 to multi-robot estimation of semantic octree maps, while in Sec. 6.5 we use Alg. 5 to achieve multi-robot motion planning for exploration and active estimation of semantic octree maps.

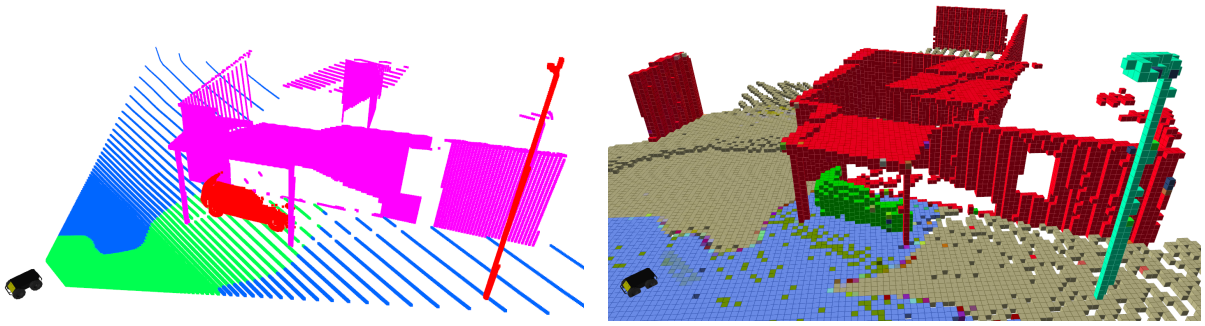


Figure 6.3. Semantically annotated point cloud (left) obtained from an RGBD sensor, where each object category is shown with a unique color and an octree map (right) obtained from the semantic point cloud.

6.4 Distributed Semantic Octree Mapping

In this section, we design a decentralized multi-robot octree mapping algorithm using the results from Sec. 6.3. We consider a team of robots gathering local sensor measurements and communicating map estimates with one-hop neighbors in order to build a globally consistent common map. The robots are navigating in an environment consisting of disjoint sets $\mathscr{S}_c \subset \mathbb{R}^3$, each associated with a semantic category $c \in \mathscr{C} := \{0, 1, \ldots, C\}$. Let $\mathscr{S}_0$ represent the free space and let each $\mathscr{S}_c$ for $c > 0$ represent a different category, such as building, vegetation, terrain. Each robot $i \in \mathscr{V}$ is equipped with a mounted sensor that provides a stream of semantically-annotated point clouds in the sensor frame. Such information may be obtained by processing the measurements of an RGBD camera [97] or a LiDAR with a semantic segmentation algorithm [98]. We model a point cloud as a set $\mathbf{z}_t^i = \{(r_{t,b}^i, y_{t,b}^i)\}_{b=1}^B$ of B rays at time t, containing the distance $r_{t,b}^i \in \mathbb{R}_{\geq 0}$ from the sensor's position to the closest obstacle along the ray in addition to object category $y_{t,b}^i \in \mathscr{C}$ of the obstacle (see Fig. 6.3).

We represent the map $\mathbf{m}$ as a 3-D grid of N independent cells, where each individual cell m is labeled with a category in $\mathscr{C}$. To model measurement noise, we use a PDF $q^i(\mathbf{z}_t^i \mid \mathbf{m})$ as the observation model of each robot. The observation model $q^i(\mathbf{z}_t^i \mid \mathbf{m})$ depends on the sensor pose as well but we assume that accurate sensor poses are available from localization and calibration between the robot body frame and the sensor frame. We intend to perform probabilistic mapping,

which requires maintaining a PDF of the map, and updating it based on sensor observations. For this aim, we maximize the sum of expected log-likelihood of the measurements up to time t, i.e. local observations $\mathbf{z}^i_{1:t}$ collected from each robot $i \in \mathscr{V}$[2]:

$$\max_{p\in\mathscr{P}} \sum_{i\in\mathscr{V}} \mathbb{E}_{\mathbf{m}\sim p}\big[\log q^i(\mathbf{z}^i_{1:t}|\mathbf{m})\big], \tag{6.9}$$

where $\mathscr{P}$ is the space of all PMFs over the set of possible maps:

$$\mathscr{P} = \{p(\cdot)|\sum_{\mathbf{m}} p(\mathbf{m}) = 1,\ p(\mathbf{m}) \geq 0\ \ \forall \mathbf{m} \in \mathscr{C}^N\} \tag{6.10}$$

The map cell independence assumption allows for decomposing the measurement log-likelihood as a sum over individual map cells m, as indicated in the following lemma.

Lemma 1. *The objective function in* (6.9) *can be expressed as a sum over all map cells and all observations:*

$$\sum_{i\in\mathscr{V}} \sum_{\tau=1}^{t} \Big(\log q^i(\mathbf{z}^i_\tau) + \sum_{n=1}^{N} \mathbb{E}_{m\sim p_n}\big[\log \frac{q^i(m|\mathbf{z}^i_\tau)}{p_n(m)}\big]\Big), \tag{6.11}$$

where $q^i(\mathbf{z}^i_\tau)$ is the marginal density of the observation, $p_n(\cdot)$ denotes the PMF of the n-th map cell, and $q^i(m|\mathbf{z}^i_\tau)$ is an inverse observation model *that represents the sensor noise properties (see* (3.7) *in Chapter 3).*

Proof. See Appendix G. □

The log-density term $\log q^i(\mathbf{z}^i_\tau)$ in (6.11) does not depend on any of the map probabilities $p_n(\cdot)$, $n \in \{1,\ldots,N\}$; hence, it can be removed from the objective without affecting the solution. Moreover, each term in the innermost summation in (6.11) only depends on a single map cell probability $p_n(\cdot)$. Therefore, the maximization of the objective can be carried out separately for

[2]It can be shown that maximizing the sum of expected log-likelihood of the data is equivalent to minimizing the KL-divergence between the true and the evaluated observation models. See [114] for more details.

each cell m:

$$\max_{p_n \in \mathscr{P}_{\mathscr{C}}} \sum_{i \in \mathscr{V}} \mathbb{E}_{m \sim p_n} \big[\log \frac{q_t^i(m)}{p_n(m)} \big], \tag{6.12}$$

where $\mathscr{P}_{\mathscr{C}}$ is the space of categorical distributions over $\mathscr{C}$ and

$$q_t^i(m) = \prod_{\tau=1}^{t} q^i(m|\mathbf{z}_\tau^i)^{1/t}. \tag{6.13}$$

In order to remove the constraint $p_n \in \mathscr{P}_{\mathscr{C}}$, we utilize a multi-class log-odds ratio representation of the categorical distribution similar to Chapter 3:

$$\mathbf{h}_n := \begin{bmatrix} \log \frac{p_n(m=0)}{p_n(m=0)} & \cdots & \log \frac{p_n(m=C)}{p_n(m=0)} \end{bmatrix}^\top \in \mathbb{R}^{C+1}. \tag{6.14}$$

A PMF and its log-odds representation have a one-to-one correspondence through the softmax function $\sigma : \mathbb{R}^{C+1} \to \mathbb{R}^{C+1}$:

$$p_n(m=c) = \sigma_{c+1}(\mathbf{h}_n) := \frac{\mathbf{e}_{c+1}^\top \exp(\mathbf{h}_n)}{\mathbf{1}^\top \exp(\mathbf{h}_n)},$$

where $\mathbf{e}_c$ is the standard basis vector with c-th element equal to 1 and 0 elsewhere, $\mathbf{1}$ is the vector with all elements equal to 1, and $\exp(\cdot)$ is applied element-wise to the vector $\mathbf{h}_n$. In order to enable distributed optimization of the objective (6.12) via the framework of Sec. 6.3, we introduce a constraint that requires the robots to agree on a common map estimate using only one-hop communication.

Problem 5. Let $\mathscr{G}(\mathscr{V}, \mathscr{E})$ be a network of robots, where each robot $i \in \mathscr{V}$ collects semantic point cloud observations $\mathbf{z}_t^i$. Construct local estimates of the map log-odds $\mathbf{h}^i$ at each robot i that are

consistent among the robots via the following optimization:

$$\begin{aligned} &\max_{\mathbf{h}^{1:|\mathscr{V}|}\in\mathbb{R}^{(C+1)\times|\mathscr{V}|}} \sum_{i\in\mathscr{V}} f^i(\mathbf{h}^i), \\ &\text{s.t.} \quad \phi(\mathbf{h}^{1:|\mathscr{V}|}) = \sum_{\{i,j\}\in\mathscr{E}} A_{ij}\|\mathbf{h}^j - \mathbf{h}^i\|_2^2 = 0, \end{aligned} \tag{6.15}$$

where $f^i(\mathbf{h}^i) = \sum_{c\in\mathscr{C}} \sigma_{c+1}(\mathbf{h}^i) \log \frac{q_t^i(c)}{\sigma_{c+1}(\mathbf{h}^i)}$ and $q_t^i(c)$ is defined in (6.13).

The multi-robot mapping problem in (6.15) has the same structure as the general distributed optimization in (6.3). Therefore, the distributed Riemannian optimization algorithm (Alg. 5) can be employed to perform multi-robot semantic mapping. Note that $\phi(\cdot)$ is globally convex because of the flatness of Euclidean space. Thus, Theorem 1 guarantees that Alg. 5 can achieves consensus in the map estimates of all robots. The application of Alg. 5 to solve (6.15) in a distributed manner is presented in Alg. 6. The update step in line 4 guides the local log-odds towards satisfaction of the consensus constraint, which only requires single-hop communication between neighboring robots $j \in \mathscr{N}_i$. Line 9 incorporates the local observations via γ^i and β^i, where $\odot$ is element-wise multiplication. This step is local to each robot i and does not require communication. Note that lines 4 and 11 resemble the log-odds equivalent of Bayes rule for updating multi-class probabilities (see (3.5) in Chapter 3).

The distributed semantic mapping algorithm we developed assumes a regular grid representation of the environment. To reduce the storage and communication requirements, we may utilize a semantic octree data structure which provides a lossless compression of the original 3-D multi-class map. In this case, the update rules in Alg. 6 should be applied to all leaf nodes in the semantic octree map of each robot i. Refer to Alg. 1 in Chapter 3 for the semantic octree equivalents of the update steps in lines 4 and 11.

In this section, we presented the mapping component of ROAM as distributed construction of semantic octree maps given local semantic point cloud observations at each robot. In the next section, we introduce the multi-robot planning component of ROAM, where robots coopera-

Algorithm 6. Distributed Semantic Mapping

Input: Local observations $\mathbf{z}^i_{1:t}$ and initial multi-class map estimate $\mathbf{h}^{i(0)}$
Output: Globally consistent semantic map

for $k \in \mathbb{Z}_{\geq 0}$ **do**
  for each cell in **m do**
    ▷ Promote consensus with step size ε_m:
    $\tilde{\mathbf{h}}^{i(k)} = \mathbf{h}^{i(k)} + \varepsilon_m \sum_{j \in \mathscr{N}_i} A_{ij}(\mathbf{h}^{j(k)} - \mathbf{h}^{i(k)})$
    ▷ Local gradient computation:
    $\Delta^i = \tilde{\mathbf{h}}^{i(k)} - \log \mathbf{q}^i_t$ ▷ $\log \mathbf{q}^i_t = [\log q^i_t(c)]_{c=0}^C$
    $\gamma^i = (\exp(\tilde{\mathbf{h}}^{i(k)})^\top \Delta^i)\mathbf{1}$
    $\beta^i = (\exp(\tilde{\mathbf{h}}^{i(k)})^\top \mathbf{1})\Delta^i$
    $\mathbf{g}^i = (\gamma^i - \beta^i) \odot \frac{\exp(\tilde{\mathbf{h}}^{i(k)})}{(\exp(\tilde{\mathbf{h}}^{i(k)})^\top \mathbf{1})^2}$
    ▷ Apply gradient with step size $\alpha^{(k)}_m$:
    $\mathbf{h}^{i(k+1)} = \tilde{\mathbf{h}}^{i(k)} + \alpha^{(k)}_m \mathbf{g}^i$
    $h^{i\,(k+1)}_1 = 0$ ▷ $h^{i\,(k+1)}_1 = \log \frac{p(m=0)}{p(m=0)} = 0$
return $\mathbf{h}^{i(k)}$

tively find trajectories along which their observations are maximally informative. Employing ROAM for simultaneous distributed mapping and planning closes the loop for autonomously exploring an unknown environment with a team of robots.

6.5 Distributed Planning for Exploration

We discussed the case where observations are collected passively along the robot trajectories and used for distributed mapping. In this section, we consider planning the motion of the robots to collect observations that reduce map uncertainty and uncover an unknown environment. This active mapping process prevents redundant observations that may not improve the map accuracy or increase the overall covered area.

Let $\mathbf{X}^i_t \in SE(3)$ be the pose of robot $i \in \mathscr{V}$, at time t:

$$\mathbf{X}^i_t = \begin{bmatrix} \mathbf{R}^i_t & \mathbf{p}^i_t \\ \mathbf{0}^\top & 1 \end{bmatrix},$$

where $\mathbf{R}_t^i \in SO(3)$ and $\mathbf{p}_t^i \in \mathbb{R}^3$ are the robot's orientation and position, respectively. The distance between two poses $\mathbf{X}_t^i$ and $\mathbf{X}_{t'}^j$ is defined as:

$$d^2(\mathbf{X}_t^i, \mathbf{X}_{t'}^j) = \xi_{\mathbf{X}_t^i, \mathbf{X}_{t'}^j}^\top \Gamma \xi_{\mathbf{X}_t^i, \mathbf{X}_{t'}^j}, \;\; \xi_{\mathbf{X}_t^i, \mathbf{X}_{t'}^j} = \log(\mathbf{X}_t^{i^{-1}} \mathbf{X}_{t'}^j)^\vee,$$

where the functions $\log(\cdot) : SE(3) \to \mathfrak{se}(3)$ and $(\cdot)^\vee : \mathfrak{se}(3) \to \mathbb{R}^6$ denote the inverse mappings associated with $\exp(\cdot)$ and $(\cdot)^\wedge$, respectively. See Sec.2.4 of Chapter 2 for an overview of *SO(3)* and $SE(3)$ Lie groups. Also, $\Gamma \in \mathbb{R}^{6\times 6}$ is a diagonal matrix with positive diagonal entries that account for the difference in scale between the linear and angular elements of $\xi_{\mathbf{X}_t^i, \mathbf{X}_{t'}^j}$.

To enable gradient-based pose trajectory optimization, we introduce differentiable cost functions to quantify the safety and the informativeness of a pose trajectory. We use a distance field $D(\mathbf{X}_t^i, p_t^i(\mathbf{m}))$ as a measure of path safety derived from the map $p_t^i(\mathbf{m})$ of robot i given observations up to time t. To obtain the distance field, we extract a maximum likelihood occupancy map from $p_t^i(\mathbf{m})$ and compute the distance transform. Furthermore, we use the Shannon mutual information $I(\mathbf{m}; \mathbf{z} | \mathbf{X}_t^i, p_t^i(\mathbf{m}))$ to quantify the informativeness of an observation $\mathbf{z}$ made from pose $\mathbf{X}_t^i$ with respect to the current map $p_t^i(\mathbf{m})$ of robot i. As discussed in the previous chapter, in the case of semantic octree mapping with a range sensor, mutual information is not differentiable with respect to the pose $\mathbf{X}_t^i$. As a solution, we use the approach in Chapter 5 to obtain a differentiable approximation of mutual information by interpolating its values at several nearby poses $\mathbf{V} \in SE(3)$. Specifically, the collision and informativeness score of a pose $\mathbf{X}_t^i$ is expressed as a convex combination of poses $\mathbf{V}$ on a grid $\mathscr{X}(\mathbf{X}_t^i)$ inside a geodesic ball centered around $\mathbf{X}_t^i$ with radius $\xi_{\max}$:

$$\mathfrak{f}(\mathbf{X}_t^i, p_t^i(\mathbf{m})) = \sum_{\mathbf{V} \in \mathscr{X}(\mathbf{X}_t^i)} \lambda_{\mathbf{V}}(\mathbf{X}_t^i) s^i(\mathbf{V}),$$

$$s^i(\mathbf{V}) = I(\mathbf{m}; \mathbf{z} | \mathbf{V}, p_t^i(\mathbf{m})) + \gamma_c \log D(\mathbf{V}, p_t^i(\mathbf{m})),$$

where the safety constant $\gamma_c > 0$ trades off informativeness with collision avoidance and the

convex combination coefficients $\lambda_{\mathbf{V}}(\mathbf{X}_t^i)$ adjust the influence of the terms corresponding to $\mathbf{V}$ based on distance to $\mathbf{X}_t^i$:

$$\lambda_{\mathbf{V}}(\mathbf{X}_t^i) = \frac{1+\cos(\bar{d}(\mathbf{X}_t^i, \mathbf{V}))}{\sum_{\mathbf{U}\in\mathscr{X}(\mathbf{X}_t^i)}(1+\cos(\bar{d}(\mathbf{X}_t^i, \mathbf{U})))},$$
$$\bar{d}(\mathbf{X}_t^i, \mathbf{V}) = \frac{\pi}{\xi_{\max}} d(\mathbf{X}_t^i, \mathbf{V}).$$

Cooperative planning requires the robots to take into account the plans of their peers in order to avoid actions that provide redundant information. Let $\mathfrak{X} = [\mathbf{X}_{t+1:t+T}^1, \ldots, \mathbf{X}_{t+1:t+T}^{|\mathscr{V}|}]^\top \in SE(3)^{|\mathscr{V}|\times T}$ be the concatenated T-length trajectories of all robots in $\mathscr{V}$, where T is the planning horizon. In the remainder of this section, we use $\mathfrak{X}_{i,\tau}$ as an alternative notation for $\mathbf{X}_{t+\tau}^i$, namely the $SE(3)$ pose of robot i at time $t+\tau$. The function $\mathfrak{q}(\mathfrak{X}_{i,\tau}, \mathfrak{X}_{j,\tau'})$ quantifies the observation redundancy as the overlap between sensor field of views (FoVs) for two poses $\mathfrak{X}_{i,\tau}$ and $\mathfrak{X}_{j,\tau'}$:

$$\mathfrak{q}(\mathfrak{X}_{i,\tau}, \mathfrak{X}_{j,\tau'}) = \max\left\{0, 2d_{\mathfrak{q}} - \|\mathbf{Q}(\mathfrak{X}_{i,\tau} - \mathfrak{X}_{j,\tau'})\mathbf{e}\|_2\right\}^2,$$

where:

$$d_{\mathfrak{q}} = |\mathscr{F}| + \xi_{\max}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{I}_{3\times 3} & 0_{3\times 1} \\ 0_{1\times 3} & 0 \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} 0_{3\times 1} \\ 1 \end{bmatrix},$$

and $|\mathscr{F}|$ is the diameter of the sensor FoV.

The local objective function for robot i is defined using the collision and informativeness score $\mathfrak{f}$ and the FoV overlap $\mathfrak{q}$:

$$f^i(\mathfrak{X}, p_t^i(\mathbf{m})) = \sum_{\tau=1}^{T}\left[\mathfrak{f}(\mathfrak{X}_{i,\tau}, p_t^i(\mathbf{m})) - \gamma_{\mathfrak{q}}\sum_{j\in\mathscr{V}}\sum_{\tau'=1}^{T}[1-\delta_{ij}\delta_{\tau\tau'}][1-\frac{\delta_{ij}}{2}]\mathfrak{q}(\mathfrak{X}_{i,\tau}, \mathfrak{X}_{j,\tau'})\right], \quad (6.16)$$

where δ_{ij} is the Kronecker delta which takes value 1 if and only if $i=j$, and 0 otherwise. Also, the constant $\gamma_{\mathfrak{q}} > 0$ trades off trajectory collision avoidance and informativeness with sensor FoV overlap.

The goal of multi-robot planning is to maximize the sum of local objective functions f^i over $\mathscr{V}$. Since we intend to perform the maximization in a distributed manner, we consider *local plans* $\mathfrak{X}^i \in SE(3)^{|\mathscr{V}|\times T}$ for each robot $i \in \mathscr{V}$, representing an individual robot's plan for the collective trajectories of the team. Eventually, these local plans should reach consensus so that the team members act in agreement. To quantify the disagreement among the robot plans, we define an aggregate distance function $\phi(\cdot) : SE(3)^{|\mathscr{V}|\times T\times|\mathscr{V}|} \to \mathbb{R}_{\geq 0}$ that accumulates the pairwise distances between all local plans $\mathfrak{X}^i$, $i \in \mathscr{V}$:

$$\phi(\mathfrak{X}^{1:|\mathscr{V}|}) = \sum_{\{i,j\}\in\mathscr{E}} A_{ij} d^2(\mathfrak{X}^i, \mathfrak{X}^j), \tag{6.17}$$

where $d : SE(3)^{|\mathscr{V}|\times T\times 2} \to \mathbb{R}_{\geq 0}$ is defined via extension of the distance in $SE(3)$ to the product manifold $SE(3)^{|\mathscr{V}|\times T}$.

Problem 6. Let $\mathscr{G}(\mathscr{V},\mathscr{E})$ be a network of robots, where each robot $i \in \mathscr{V}$ maintains a local map $p_t^i(\mathbf{m})$ obtained by solving (6.15). Determine $SE(3)$ pose trajectories for all robots that maximize the cost function in (6.16) subject to the consensus constraint in (6.17):

$$\begin{aligned} \max_{\mathfrak{X}^{1:|\mathscr{V}|}\in SE(3)^{|\mathscr{V}|\times T\times|\mathscr{V}|}} & \sum_{i\in\mathscr{V}} f^i(\mathfrak{X}^i, p_t^i(\mathbf{m})), \\ \text{s.t.} \qquad & \phi(\mathfrak{X}^{1:|\mathscr{V}|}) = 0. \end{aligned} \tag{6.18}$$

The structure of the planning problem in (6.18) is compatible with the distributed Riemannian optimization method of Sec. 6.3. We formulate a version of Alg. 5 specialized for the $SE(3)$ manifold. Due to the positive curvature of the $SE(3)$ manifold, the aggregate distance function $\phi(\cdot)$ has local minima (see Appendix A.3 in [40]). Thus, if the initial trajectories $\mathfrak{X}^{i^{(0)}}$, $i \in \mathscr{V}$, are not similar, the algorithm may converge to a local optimum of the consensus constraint (6.17). Furthermore, the local objective functions f^i, $i \in \mathscr{V}$, are only locally concave [76]. Hence, Theorem 1 guarantees only a locally optimal consensus solution.

Our distributed planning algorithm for solving (6.18) is presented in Alg. 7. Given its

Algorithm 7. Distributed Planning for Exploration

Input: Local map $p_t^i(\mathbf{m})$ of robot i
Output: Collaborative robot team plan for exploration

1: $\mathfrak{X}^{i^{(0)}} = \text{FRONTIER}(p_t^i(\mathbf{m})) \quad \forall i \in \mathscr{V}$ ▷ Initialization
2: **for** $k \in \mathbb{Z}_{\geq 0}$ **do**
3: ▷ Promote consensus with step size ε_p:
4: **for** every $l \in \mathscr{V}$ and $\tau \in \{1,\dots,T\}$ **do**
5: $\tilde{\mathfrak{X}}_{l,\tau}^{i^{(k)}} = \mathfrak{X}_{l,\tau}^{i^{(k)}} \exp\Big(\varepsilon_p \sum_{j\in\mathscr{N}_i} A_{ij} \big(J_L^{-\top}(\xi_{\mathfrak{X}_{l,\tau}^{i^{(k)}}, \mathfrak{X}_{l,\tau}^{j^{(k)}}}) \Gamma \xi_{\mathfrak{X}_{l,\tau}^{i^{(k)}}, \mathfrak{X}_{l,\tau}^{j^{(k)}}}\big)^{\wedge}\Big)$
6: ▷ Local gradient computation:
7: $\mathbf{g}_{l,\tau} = 0$ ▷ Initialize for all $l \in \mathscr{V}$, $\tau \in \{1,\dots,T\}$
8: **for** every $\tau' \in \{1,\dots,T\}$ **do**
9: $c_{\text{set}} = |\mathscr{X}(\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}})| + \sum_{\mathbf{V}\in\mathscr{X}(\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}})} \cos\big(\bar{d}(\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}}, \mathbf{V})\big)$
10: $\mathbf{g}_{i,\tau'} \mathrel{+}= \sum_{\mathbf{V}\in\mathscr{X}(\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}})} \big[(s(\mathbf{X}) - \mathfrak{f}(\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}}, p_t^i(\mathbf{m}))) \frac{\sin(\bar{d}(\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}}, \mathbf{V}))}{c_{\text{set}} \bar{d}(\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}}, \mathbf{V})} J_L^{-\top}(\xi_{\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}}, \mathbf{V}}) \Gamma \xi_{\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}}, \mathbf{V}}\big]$
11: **for** every $l \in \mathscr{V}$ and $\tau \in \{1,\dots,T\}$ **do**
12: $\mathbf{p}_{i,\tau'} = \mathbf{Q}\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}}\mathbf{e}, \quad \mathbf{p}_{l,\tau} = \mathbf{Q}\tilde{\mathfrak{X}}_{l,\tau}^{i^{(k)}}\mathbf{e}$
13: $\mathbf{R}_{i,\tau'} = \mathbf{Q}\tilde{\mathfrak{X}}_{i,\tau'}^{i^{(k)}}\mathbf{E}, \quad \mathbf{R}_{l,\tau} = \mathbf{Q}\tilde{\mathfrak{X}}_{l,\tau}^{i^{(k)}}\mathbf{E}$
14: $\mathbf{c}_{\text{disp}} = [1 - \delta_{il}\delta_{\tau\tau'}][1 - \frac{\delta_{il}}{2}](\mathbf{p}_{i,\tau'} - \mathbf{p}_{l,\tau})$
15: $\mathbf{c}_{\text{tot}} = \gamma_{\text{q}} \mathbf{c}_{\text{disp}} \max\Big\{0, \frac{2d_{\text{q}}}{\|\mathbf{p}_{i,\tau'} - \mathbf{p}_{l,\tau}\|_2} - 1\Big\}$
16: $\mathbf{g}_{i,\tau'} \mathrel{+}= \begin{bmatrix} \mathbf{R}_{i,\tau'}^{\top}\mathbf{c}_{\text{tot}} \\ 0_{3\times 1} \end{bmatrix}, \quad \mathbf{g}_{l,\tau} \mathrel{-}= \begin{bmatrix} \mathbf{R}_{l,\tau}^{\top}\mathbf{c}_{\text{tot}} \\ 0_{3\times 1} \end{bmatrix}$
17: ▷ Apply gradient with step size $\alpha_p^{(k)}$
18: **for** every $l \in \mathscr{V}$ and $\tau \in \{1,\dots,T\}$ **do**
19: $\mathfrak{X}_{l,\tau}^{i^{(k+1)}} = \tilde{\mathfrak{X}}_{l,\tau}^{i^{(k)}} \exp(\alpha_p^{(k)} \hat{\mathbf{g}}_{l,\tau})$
20: **return** $\mathfrak{X}^{i^{(k)}}$

current local map $p_t^i(\mathbf{m})$, each robot $i \in \mathscr{V}$ computes an initial plan $\mathfrak{X}^{i^{(0)}}$ for the whole team using frontier-based exploration [161]. In line 5, each pose in the local plan $\mathfrak{X}^{i^{(k)}}$ is guided towards consensus with the plans of neighboring robots $j \in \mathscr{N}_i$. The update in this line is carried out via a perturbation in the robot frame, where $J_L(\cdot)$ denotes the left Jacobian of $SE(3)$, and only involves communication between neighbors. To compute the local objective function gradients with respect to each pose in the local plan, we first initialize the gradients with zero in line 7, and then populate them with proper values in lines 10 and 16. The gradient of the collision and

informativeness score $\mathfrak{f}(\tilde{\mathfrak{X}}^{i^{(k)}}_{i,\tau'}, p^i_t(\mathbf{m}))$ is computed in lines 9-10, while lines 12-16 derive the gradient of the sensor overlap $\mathfrak{q}(\tilde{\mathfrak{X}}^{i^{(k)}}_{i,\tau'}, \tilde{\mathfrak{X}}^{j^{(k)}}_{l,\tau})$ with respect to both inputs, where $\mathbf{E} = [\mathbf{I}_{3\times 3}\ 0_{3\times 1}]^\top$. Note that, for the $\mathfrak{f}$ terms, we only need to compute the gradient with respect to robot i's own trajectory $\tilde{\mathfrak{X}}^{i^{(k)}}_{i,\tau'}$, $\tau' \in \{1,\dots,T\}$, whereas for the $\mathfrak{q}$ terms, robot i should locally obtain gradients with respect to both its own trajectory as well as the trajectories of all other robots in $\mathscr{V}$. Since each robot stores the trajectory of the whole team, the computation for the gradients of the $\mathfrak{q}$ terms does not require any communication among the robots. Lastly, in line 19, we apply the computed gradients to each pose in the local plan, using a right perturbation in the robot frame.

Solving (6.18) via Alg. 7 leads to two types of behaviors.

1. Locally, the robots attempt to maximize information and distance from obstacles along their trajectories. This encourages each robot to visit unvisited parts of the environment, and corresponds to the $\mathfrak{f}$ terms of (6.16).

2. Within each neighborhood, the robots *negotiate* with their peers to minimize redundant observations. This prevents the trajectories to amass at certain regions of the map, and corresponds to the $\mathfrak{q}$ terms of (6.16).

We emphasize that each local plan $\mathfrak{X}^i$ stores the paths for all robots in $\mathscr{V}$, instead of only robot i's and its immediate neighbors. This is because storing all $|\mathscr{V}|$ paths in each robot allows propagation of the mentioned behaviors on a global scale, due to the consensus constraint of 6.18. Therefore, the global solution of (6.18) corresponds to a Pareto optimum where agents find an optimal trade-off between their own information and safety maximization on one hand and avoiding observation overlap with their peers on the other hand.

In this section, we developed the distributed planning component of ROAM. The robot trajectories are chosen to maximize information and safety for cooperative estimation of a semantic octree map. Combined with the distributed mapping method of Sec. 6.4, the overall system can be utilized for efficient multi-robot exploration of an unknown environment. In the

next section, we demonstrate the performance of ROAM in a variety of simulation and real-world experiments.

6.6 Experiments

This section describes the implementation of ROAM on multi-robot systems. Next, we evaluate the performance of ROAM using several measures that quantify optimality, convergence to consensus, and communication overhead. The evaluations are done in both simulation and real-world.

6.6.1 Implementation of ROAM for Distributed Active Mapping

We deploy our approach on a team of ground wheeled robots, each equipped with an RGB-D sensor. Fig. 6.4 shows an overview of the software stack, implemented using the *Robot Operating System* (ROS) [120]. The RGB-D sensor provides synchronized RGB and depth images. The RGB image is processed with a semantic segmentation algorithm to label each pixel with an object category. The segmented image is fused with the depth image to obtain a 3-D semantically annotated point cloud in the sensor frame of robot i.

Multi-robot localization

It is required to perform multi-robot localization in order to find the transformation *a*) from robot i's sensor frame to a static *world* frame $\mathscr{W}_i$ for point cloud registration, and *b*) from $\mathscr{W}_i$ to $\mathscr{W}_j$ for distributed multi-robot mapping and planning. Our implementation of multi-robot localization in the simulation and real-world experiments is explained in Sec. 6.6.2 and Sec. 6.6.3, respectively.

Multi-robot mapping

The semantic point cloud is used to build and update a semantic octree map for each robot $i \in \mathscr{V}$ via lines 6-12 of Alg. 6. The semantic map of each robot i is broadcasted to its neighboring robots $\mathscr{N}_i$ once every t_m^{pub} seconds. Moreover, each robot pushes any newly received map to

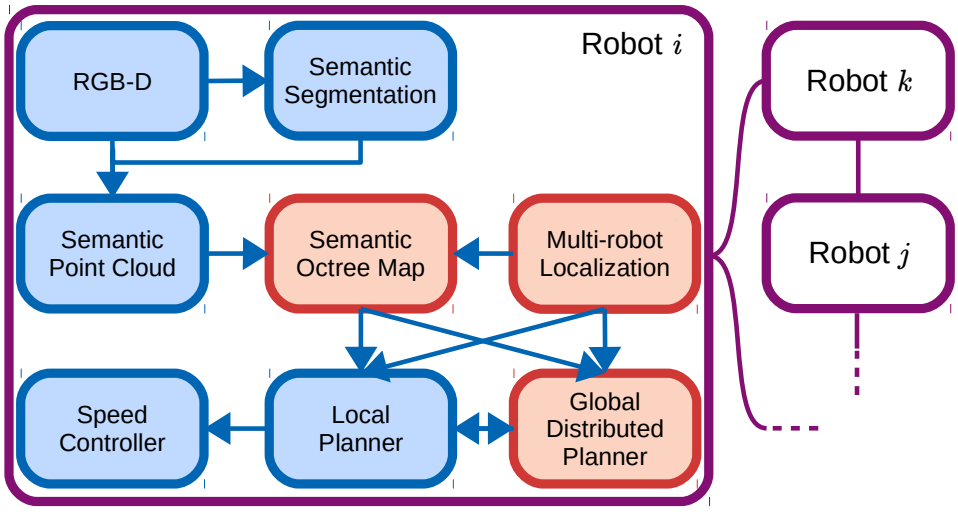


Figure 6.4. Software stack for multi-robot distributed active mapping. The blue blocks are local to each robot, whereas the red blocks require communication with neighboring robots. The communication links between pairs of robots are represented by violet lines.

a local buffer memory, and performs line 4 of Alg. 6 every t_m^{int} seconds to integrate neighbors' maps into its local map. The buffer is cleared after each successful iteration of Alg. 6.

Multi-robot viewpoint planning

To decouple low-frequency informative planning from high-frequency planning for collision-avoidance, we perform two separate planning stages, namely on global viewpoint level and on local trajectory level. On the viewpoint level, the distributed collaborative planning in Alg. 7 is employed to find informative viewpoints for each robot in $\mathscr{V}$. To coordinate viewpoint planning across all robots, every robot $i \in \mathscr{V}$ maintains a ledger $\mathscr{L}$ composed of $|\mathscr{V}|$ binary values each indicating whether the corresponding robot in the team is ready for planning. Due to the decentralized nature of our method, each robot sends its own copy of the ledger $\mathscr{L}_i$ to its neighbors every t_p^{pub} second, and updates $\mathscr{L}_i$ using the incoming ledgers, as well as its status with respect to the current plan. Alg. 8 details the process of decentralized ledger synchronization for each robot i. In line 1 robot i makes a copy of the incoming ledger $\mathscr{L}_{inc}$. Then, in line 2, the function CHECKREADY() determines whether or not the robot is ready to compute a new plan. A robot would declare ready to plan only when it has finished its previous plan and also

Algorithm 8. Distributed Ledger Synchronization

Input: Incoming ledger $\mathscr{L}_{inc}$
Output: Synchronized ledger

1: $\mathscr{L}_i = \mathscr{L}_{inc}$
2: **if** CHECKREADY() **then**
3: $\quad \mathscr{L}_i[i] = 1$
4: $\quad$ **if** MEAN$(\mathscr{L}_i) \geq thresh_p$ **then**
5: $\quad\quad$ STARTPLANNING() ▷ Viewpoint planning via Alg. 7
6: **else**
7: $\quad \mathscr{L}_i[i] =$ ISPLANNING()
8: **return** $\mathscr{L}_i$

it is currently not planning. The global distributed planning of Alg. 7 would start only after a minimum fraction of robots, denoted by $thresh_p$, are ready to plan. Line 7 is used to stabilize the ledger synchronization process. During the global distributed planning of Alg. 7, each robot i broadcasts its local plan $\mathfrak{X}^i$ after each optimization iteration. Incoming local plans $\mathfrak{X}^j$, $j \in \mathscr{N}_i$, are pushed to a local buffer memory to be used during the consensus step (line 5 of Alg. 7). The buffer is cleared after each optimization iteration. Lastly, the planning terminates after reaching k_p iterations.

Local trajectory optimization

After computing a sequence of viewpoints $\mathfrak{X}^{1:|\mathscr{V}|}$, each robot i locally computes a trajectory to visit its portion of the viewpoints $\mathfrak{X}^i_{i,1:T}$. The separation of the viewpoint planning from the trajectory optimization allows the robots to rapidly react to environment changes or mapping errors via local path re-planning, without the need to coordinate with their peers in viewpoint planning via Alg. 7. Furthermore, the two stage planning allows accounting for dynamical constraints of each robot in heterogeneous robot teams, such that the low-level trajectory optimizer takes the viewpoint set $\mathfrak{X}^i_{i,1:T}$ and computes a dynamically feasible path. In our experiments, each robot i projects its own semantic octree map onto a 2-D plane to obtain an occupancy grid map of the environment. Given its viewpoint set $\mathfrak{X}^i_{i,1:T}$, the robot computes a sequence of collision-free positions and orientations that connect its current pose to $\mathfrak{X}^i_{i,1}$, and

Table 6.1. Parameter set for multi-robot exploration.

Planning				Mapping	
ε_p	0.1	$\alpha_p^{(k)}$	$\frac{0.1}{k+1}$	ε_m	0.1
$d_{\mathfrak{q}}$	20m	$\xi_{\max}$	16m	$\alpha_m^{(k)}$	$\frac{1}{k+1}$
γ_c	10^{-3}	$\gamma_{\mathfrak{q}}$	10^{-2}	t_m^{pub}	5
T	5	k_p	20	t_m^{int}	5
t_p^{pub}	1sec	$thresh_p$	0.4	Voxel side length	0.2m
Γ		$diag(1,1,1,0.1,0.1,0.1)$			

each $\mathfrak{X}_{i,\tau}^i$ to $\mathfrak{X}_{i,\tau+1}^i$ for $\tau \in \{1,\ldots,T-1\}$. For this purpose, the trajectory optimizer uses A^* graph search over the 2-D occupancy map. If a collision is detected during execution of the path, the corresponding path segment is re-planned using another A^* call. The local trajectory is then used by a low-level speed controller to generate velocity commands.

An open-source implementation of ROAM is available on GitHub[3]. The rest of this section describes the simulation and real-world experiments. Table 6.1 summarizes the parameters used across all experiments.

6.6.2 Simulation Experiments

We carry out experiments in a photo-realistic 3-D simulation powered by the Unity engine [151]. The environment resembles an outdoor village area with various types of terrain (e.g., grass, dirt road, asphalt, etc.) and object classes, such as buildings, cars, and street lighting. Our experiments utilize $|\mathscr{V}| = 6$ ClearPath Husky wheeled robots, each equipped with an RGB-D sensor. We assume known robot poses and perfect semantic segmentation over the RGB input in the simulation experiments. Fig. 6.5 shows the simulation setup.

Each robot uses its local semantic octree map to extract traversable regions, while other object and terrain classes are considered as obstacles. In particular, *Asphalt* and *Dirt road* classes are selected as traversable terrain classes. Fig. 6.6 visualizes a time lapse of the distributed multi-robot active mapping experiment. The consistency between the local map of robot 1

[3]https://github.com/ExistentialRobotics/ROAM.

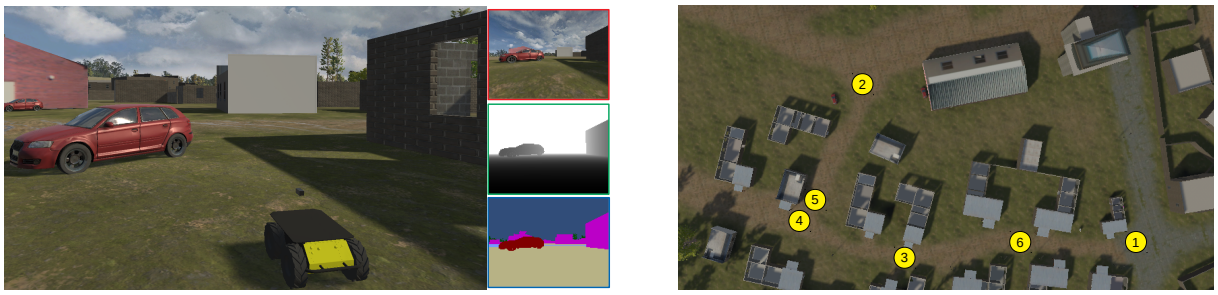


Figure 6.5. Simulation environment for multi-robot distributed active mapping. Left: A Husky robot receiving RGB, depth, semantic segmentation images. Right: A top-down view of the simulated environment, where the numbered circles show the starting positions of six robots.

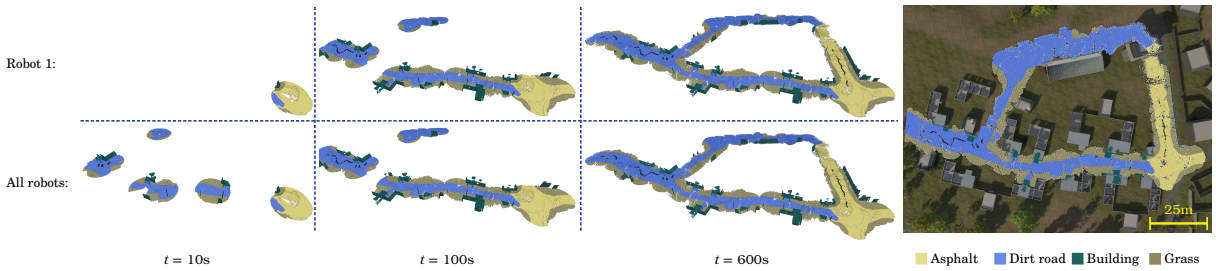
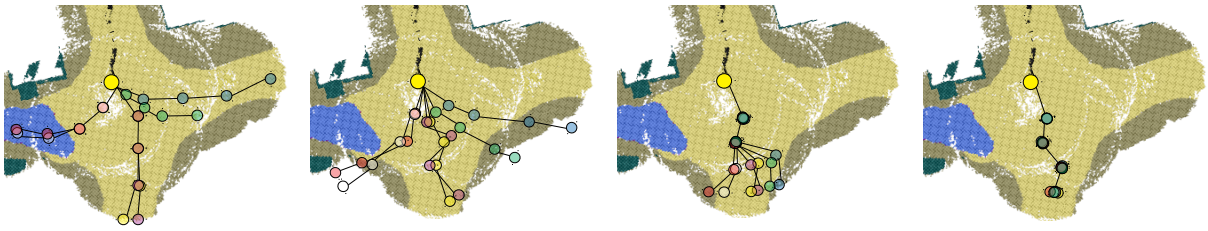


Figure 6.6. Time lapse of the multi-robot active mapping experiment. The local map of robot 1 (in Fig. 6.5) is compared against the combined maps of all robots. The right sub-figure shows the estimated semantic octree map of robot 1 overlayed on the ground-truth simulation environment. The exploration is carried out using a fully-connected network of robots.

and the combined map of all robots can be seen as a qualitative example of the map consensus achieved by the distributed mapping method in Alg. 6. Analogously, Fig. 6.7 illustrates consensus achieved by the distributed multi-robot planning in Alg. 7. As described in Sec. 6.5, each robot computes its local plan based on its local map. Hence, differences in the local maps can cause variation across the local plans, as seen in Fig. 6.7a. However, during each iteration of distributed planning, line 5 in Alg.7 steers the local plans towards a consensus plan, as is evident in Fig. 6.7d.

The performance of ROAM is evaluated quantitatively under various robot network configurations and planning parameters. We consider 3 different network topologies: 1. *Full*, where all robots can communicate with each other in a fully-connected network, 2. *Hierarchical*, where robots can only communicate with their team leaders, and 3. *Ring*, where each robot has exactly 2 neighbors. Fig. 6.8 depicts the 3 network configurations. For each network



(a) Initial local plans **(b)** 5th iteration **(c)** 15th iteration **(d)** 20th iteration

Figure 6.7. Time lapse of viewpoint planning for robot 1 from Fig. 6.5. Each color corresponds to a robot $j \in \mathcal{V}$ computing a local plan $\mathcal{X}^j_{1,1:T}$ for robot 1. The planned trajectories contain both position and orientation, however only the positions are visualized for clarity. The planning is carried out over a fully-connected network of robots.

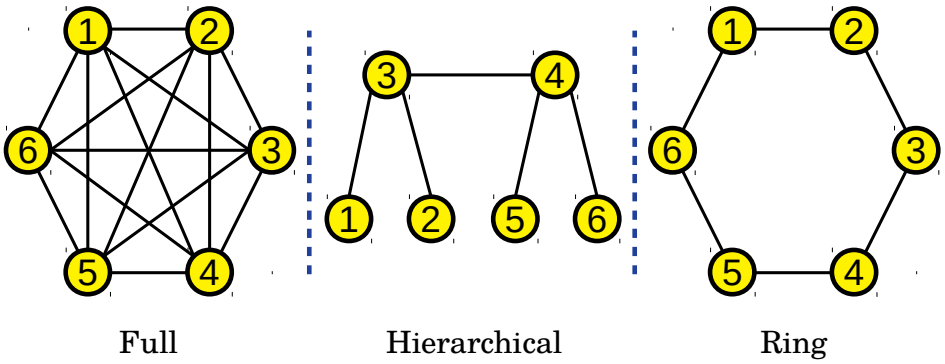


Figure 6.8. Network topologies used in the simulation experiments.

configuration, we perform exploration under 3 variants of Alg. 7: 1. *Collaborative*, which is the original version of Alg. 7, 2. *Egocentric*, where each robot only maximizes its own path informativeness and safety by choosing $\varepsilon_p = \gamma_{\mathfrak{q}} = 0$, and 3. *Frontier*, where robots perform frontier-based exploration by choosing $k_p = 0$.

Fig. 6.9 quantifies the coverage achieved by each network topology and planning parameter set. For *Collaborative* and *Egocentric* planning configurations, *Full* network configuration leads to faster coverage while traveling less distance compared to *Hierarchical* and *Ring* topologies. This is expected since *Full* is the only network topology that allows one-hop exchange of information between any pair of robots. On the other hand, the network configuration does not play a significant role for *Frontier* exploration in terms of total covered area, since robots usually choose a frontier that is nearby their current position, and do not utilize information

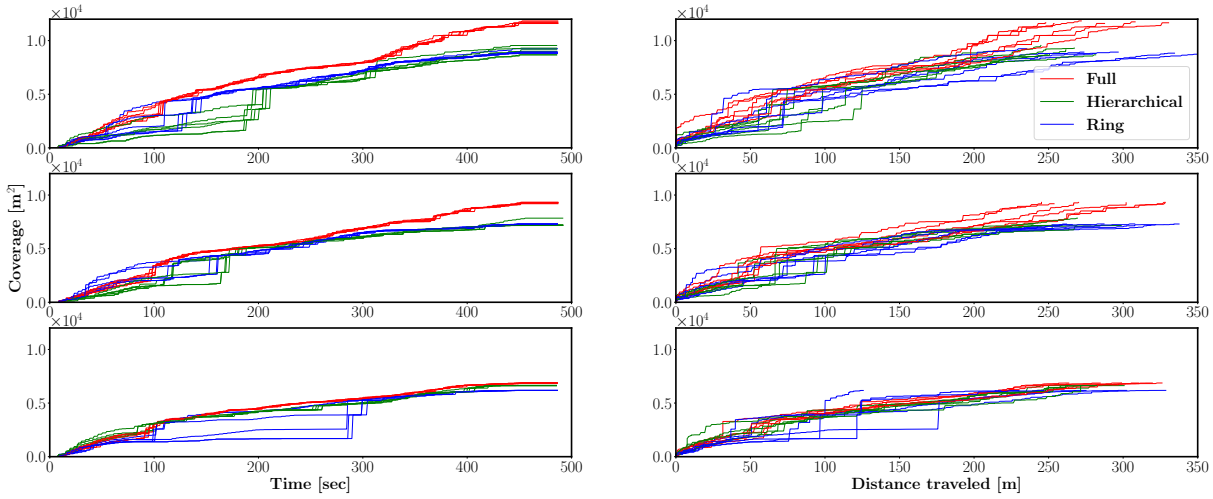


Figure 6.9. Coverage versus time (left column) and distance traveled (right column) for the simulation experiments. The top, middle, and bottom rows show the results for *Collaborative*, *Egocentric*, and *Frontier* modes of planning, respectively. In each plot, lines with the same color correspond to robots participating in the same multi-robot exploration experiment, while the experiments are separated by the type of network topology in Fig. 6.8.

coming from their peers' local maps. The most interesting takeaway from Fig. 6.9 is the similar performance of *Collaborative* planning with *Hierarchical* and *Ring* topologies, compared to *Egocentric* planning with *Full* topology. This observation suggests that effective coordination among agents via *Collaborative* planning can alleviate the longer multi-hop communication routes caused by the sparse connectivity of *Hierarchical* and *Ring* topologies.

Similar insights can be obtained from Fig. 6.10, where normalized map entropy is measured against elapsed time and distance traveled, for each network topology and planning mode. Normalized map entropy for robot $i \in \mathcal{V}$ is defined as the sum of Shannon entropies of all map voxels divided by the number of voxels:

$$H^i_{norm} = \frac{-1}{N^i} \sum_{n=1}^{N^i} \sum_{c \in \mathscr{C}} p^i_n(m=c) \log p^i_n(m=c),$$

where N^i denotes the number of voxels in the local map of robot i, and $\mathscr{C}$ as well as $p^i_n(m)$ are defined in the previous sections. Note that, unlike total map entropy, normalized entropy can increase as the robots register unvisited voxels into their map. As Fig. 6.10 shows, for each

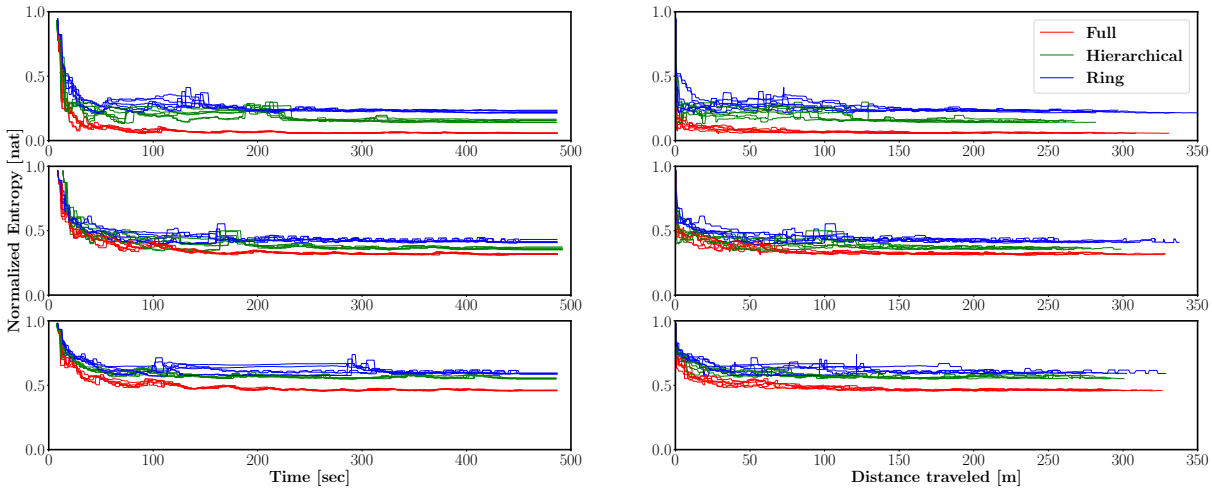


Figure 6.10. Normalized map entropy versus time (left column) and distance traveled (right column) for the simulation experiments. The top, middle, and bottom rows show the results for *Collaborative*, *Egocentric*, and *Frontier* modes of planning, respectively. In each plot, lines with the same color correspond to robots participating in the same multi-robot exploration experiment, while the experiments are separated by the type of network topology in Fig. 6.8.

planning mode, *Full* network topology outperforms *Hierarchical* and *Ring* configurations. Also, *Collaborative* planning with *Hierarchical* and *Ring* configurations have similar performance to *Egocentric* planning with *Full* network topology. The same reasoning used for Fig. 6.9 can be utilized to justify these observations. However, unlike coverage, network topology plays a more significant role in terms of normalized map entropy for *Frontier* planning mode. This is due to the relatively more distributed mapping consensus steps for the *Full* topology that lead to more certainty in the map estimation and, hence, smaller entropy compared to *Hierarchical* and *Ring*. Since coverage does not take map uncertainty into account, such behavior is only noticeable in the bottom row of Fig. 6.10 but not in Fig. 6.9.

Additional quantitative metrics specific to multi-robot active mapping are reported in Fig. 6.11. The left column of Fig. 6.11 shows the aggregate distance $\phi(\mathbf{h}^{1:6})$, which represents the total discrepancy across all local maps. Despite robots discovering distinct unexplored regions during exploration, which can increase the difference among the local maps, it can be seen that the map discrepancy tends to decrease overall. The long-term value of the map discrepancy depends on the ratio of exploration rate and information exchange rate. Hence,

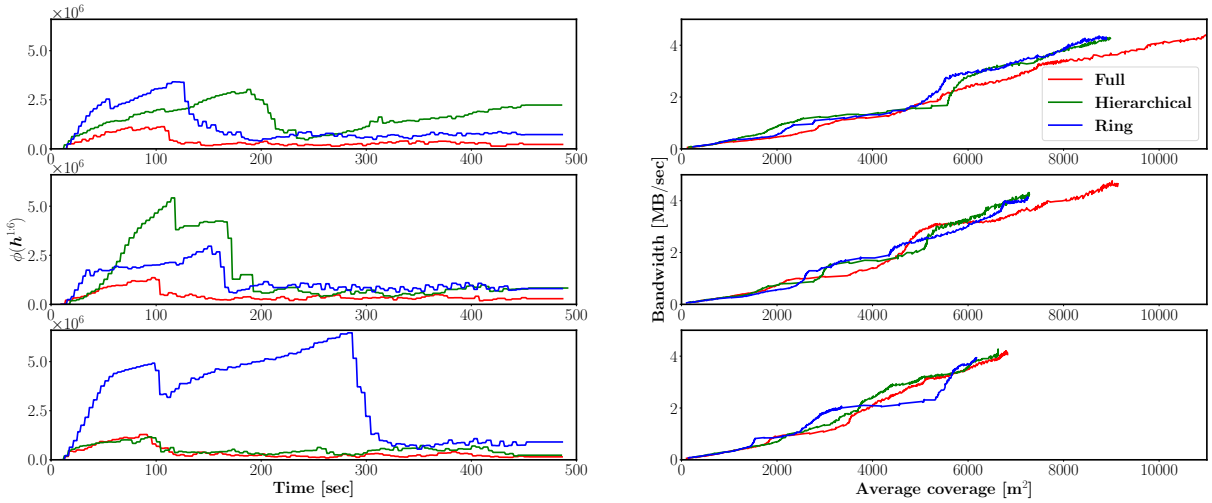


Figure 6.11. Multi-robot exploration performance metrics in the simulation experiments. The left column shows evolution of the map discrepancy $\phi(\mathbf{h}^{1:6})$ across the robot networks $\mathscr{G}$ in Fig. 6.8 over time, and the right column displays bandwidth requirements for the distributed mapping with respect to average coverage. The top, middle, and bottom rows show the results for *Collaborative*, *Egocentric*, and *Frontier* modes of planning, respectively. The average coverage is computed by averaging the area covered at each timestamp over all robots participating in an experiment.

the *Full* topology yields the closest performance to map consensus due to its relatively faster rate of exchanging the local maps amongst the robots. The right column of Fig. 6.11 displays the bandwidth required for communicating the local maps within the robot network. Since the simulation environment uses a centralized network scheme to register the broadcasted local maps, there is no significant variation in terms of bandwidth use across different network topologies and planning modes. Nevertheless, the results show scalability of the semantic octree mapping for multi-robot applications, where an average 97 Bytes/sec of bandwidth is needed for each 1 m^2 of covered area for voxel dimensions of $0.2 \times 0.2 \times 0.2$ m^3.

The quantitative results of Fig. 6.9 and Fig. 6.10 demonstrate the effective performance of ROAM, while Fig. 6.11 showcases the consensus and communication properties of our method. In the next subsection, we evaluate the distributed multi-robot active mapping in real-world experiments.

6.6.3 Real-World Experiments

We deployed ROAM on a team of ground robots to achieve autonomous exploration and mapping of an unknown indoor area. The robot team was comprised of two ClearPath Jackal robots (robot 1 and robot 2), and an F1/10 race car robot (robot 3). The Jackals were each equipped with an Ouster OS1-32 3D LiDAR, an Intel RealSense D455 RGB-D camera, and an NVIDIA GTX 1650 GPU. The F1/10 race car was equipped with a Hokuyo UST-10LX 2D LiDAR, an Intel RealSense D455 RGB-D camera, and an NVIDIA Xavier NX computer. Fig. 6.12 shows the three robots participating in the experiment. We utilize a ResNet18 [67] neural network architecture pre-trained on the SUN RGB-D dataset [135] for semantic segmentation. To achieve real-time segmentation, we employed the deep learning inference ROS nodes provided by NVIDIA [112], which are optimized for NVIDIA GPUs via TensorRT acceleration. The semantic segmentation module processes the RGB image stream from the D455 camera, and fuses the segmentation results with the depth image stream, to publish semantic point cloud ROS topics. For localization, the Jackal robots used the direct LiDAR odometry of [38], while the F1/10 race car used iterative closest point (ICP) scan matching [31]. In order to align the world frames $\mathscr{W}_i$, $i \in \{1,2,3\}$, we used AprilTag detection [155], where the estimated $SE(3)$ transformation between the RGB sensor frame of a robot and the detected tag is defined as the *world-to-sensor* transformation. Communication was handled via a Wi-Fi network and multi-master ROS architecture, such that each robot i runs its own ROS master, and shares its planning ledger $\mathscr{L}_i$, local plan $\mathfrak{X}_i$, local semantic octree map $\mathbf{m}^i$, and estimated pose with respect to $\mathscr{W}_i$. With all the mapping and planning computations carried out using the on-board robot computers, we obtain an average frame rate of 2.44Hz for distributed semantic octree mapping, and an average distributed planning iteration time of 0.014s.

Our experiments took place in a basement area, consisting of a lobby room connected to a corridor and a large laboratory, shown in Fig. 6.13. Similar to the simulation experiments, the local semantic octree maps were utilized to analyze the terrain traversability, where in this

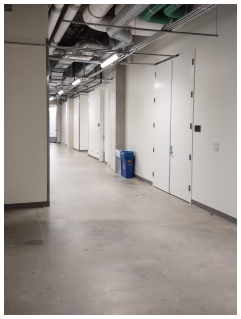
(a) Robot 1

(b) Robot 2

(c) Robot 3

Figure 6.12. Ground robot team used in our real-world multi-robot active mapping experiments.

(a) Lobby

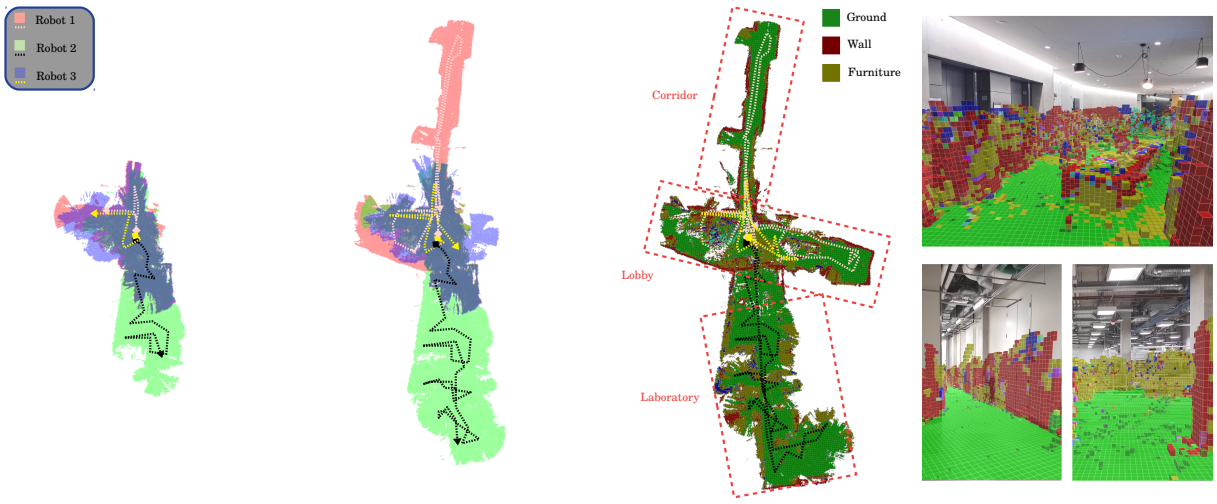


(b) Corridor

(c) Laboratory

Figure 6.13. Indoor environment used in our real-world multi-robot active mapping experiments.

case the *Ground* object class was selected as the traversable region. Exploration was performed using the *Collaborative* planning parameter set. The communication network topology was *Full*, however, there were intermittent disconnections due to signal attenuation and occlusion by the walls. Fig. 6.14 shows a time lapse of the real-world multi-robot active mapping experiment. The robots started at nearby positions, all facing the same AprilTag to align their world frames. During the first 400s of exploration, the robots mostly explored the same areas in their immediate vicinity. Each robot gradually separated from the others after $t = 400$s, and focused on a specific part of the environment, as it can be seen in Fig. 6.14a. In particular, robot 1 explored the corridor area, while robot 2 and robot 3 visited the laboratory and the lobby, respectively. Around $t = 500$s, as the robots got farther away from each other, they temporarily lost communication. During the disconnection period, the robots could not plan collaboratively and relied only on



(a) At $t = 400$s, the robots effectively choose different sections of the environment to explore.

(b) At $t = 900$s, the robots continue to explore the environment.

(c) At $t = 1450$s, the robots are back to their initial locations and perform a final map exchange to ensure consensus.

(d) First-person view of the semantic octree map of robot 1.

Figure 6.14. Qualitative results from the real-world multi-robot active mapping experiment. (a)-(b): Snapshots of exploration at $t = 400$s and $t = 900$s. The explored region and the path of each robot are identified by a distinct color (legend at the top left). (c): The final semantic octree map of robot 1 at $t = 1450$s. The three sections of the environment (i.e. lobby, laboratory, and corridor) have been marked on the map. (d) First-person views of the map overlaying the ground-truth environment.

their local maps for planning. Nonetheless, this did not deteriorate the exploration performance since the robots were so far away from each other that they would not revisit a region which had already been explored by a peer. Fig. 6.14b is a good example of such situations. At $t = 1200$s, the team was ordered to return to base, where the robots individually planed paths from their current positions back to their initial positions. Communication was automatically re-established as soon as the robots arrived near the starting locations, and distributed mapping was resumed leading to agreement in the semantic octree maps, as shown in Fig. 6.14c. First-person views of the semantic octree map of robot 1 overlaid on the real-world environment is illustrated in Fig. 6.14d.

Fig. 6.15 and Fig. 6.16 show the quantitative results of the same real-world experiment.

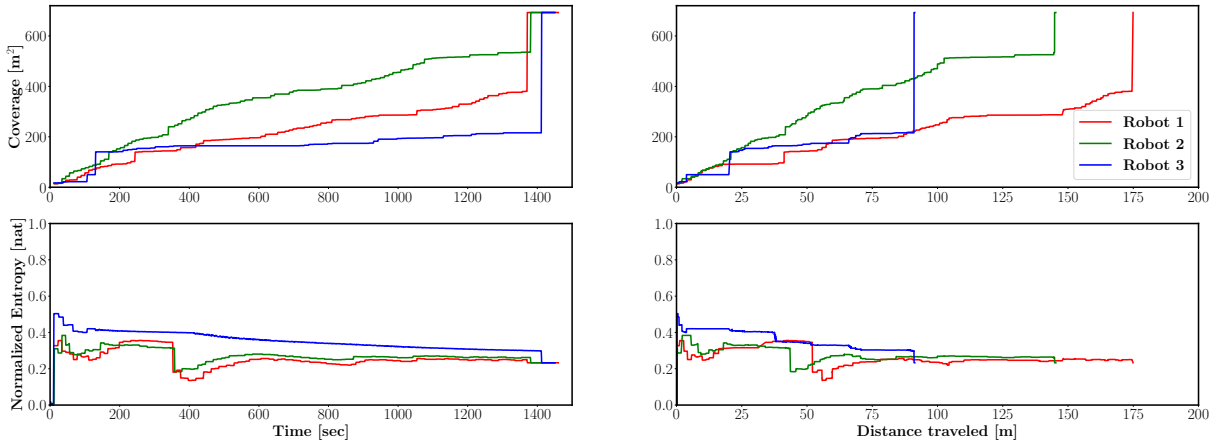


Figure 6.15. Coverage (top row) and normalized map entropy (bottom row) versus time (left column) and distance traveled (right column) for the real-world multi-robot active mapping experiment. Each color corresponds to one robot in the team.

Note the sudden jump in the map coverage plot of Fig. 6.15, corresponding to reaching map consensus amongst the robots when network connection was re-established. This can be clearly observed in Fig. 6.16 where the map discrepancy $\phi(\mathbf{h}^{1:3})$ shrinks to zero as the robots reach consensus. Unlike the simulation experiments, the bandwidth use of the robots can be measured separately by probing the communication packages sent to/from each robot, reported in Fig. 6.16. The flat lines represent the jump in map coverage due to the eventual connection after a period of intermittent disconnections during the exploration. Overall, the real-world experiments show the practicality of ROAM for autonomous mapping of large unstructured areas using a team of robots and consumer-grade communication infrastructure.

6.7 Summary

In this chapter, we developed a distributed Riemannian optimization method that achieves consensus among the variables estimated by different nodes in a communication graph. We used this method to formulate distributed techniques for multi-robot semantic mapping and information-theoretic viewpoint planning. The resulting Riemannian Optimization for Active Mapping (ROAM) enables fully distributed collaborative active mapping of an unknown envi-

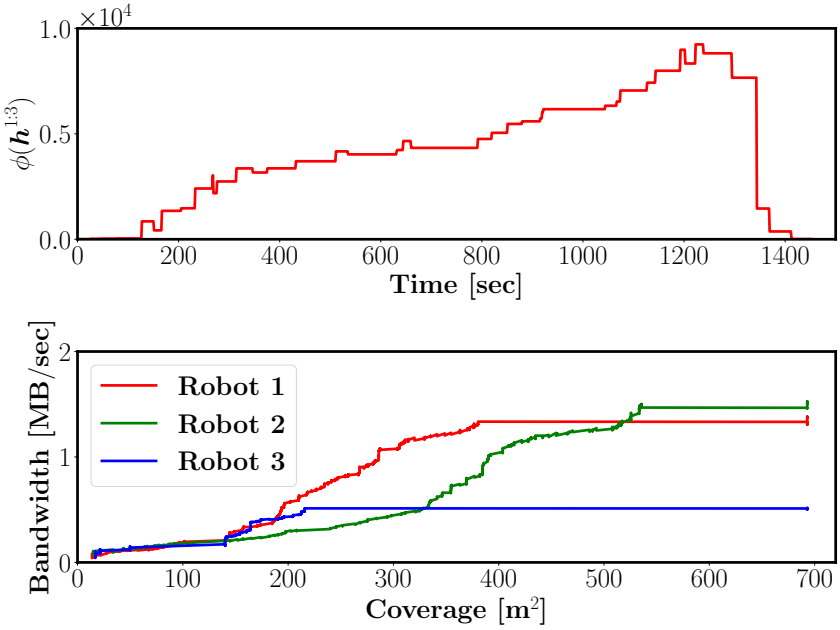


Figure 6.16. Multi-robot exploration performance metrics for the real-world experiment. Top: Evolution of the map discrepancy $\phi(\mathbf{h}^{1:3})$ across the robot network $\mathscr{G}$ over time. Bottom: Bandwidth requirement of each robot for the distributed mapping with respect to coverage.

ronment without the need for central estimation and control. Our experiments demonstrated scalability and efficient performance even with sparse communication, and corroborated the theoretical guarantees of ROAM to achieve consensus and convergence to an optimal solution. ROAM offers the possibility to generalize many single-robot non-Euclidean problems to distributed multi-robot applications, for example formulating multi-robot control for $SE(3)$ robot dynamics [52]. An important future research direction involves analyzing the distance to consensus as well as sub-optimality bounds for ROAM in the case of non-convex distance measures and non-concave objective functions. Additionally, further research effort is needed to study faster variants of ROAM using Nesterov accelerated [11] and second-order [115] gradient methods.

Acknowledgments

Chapter 6, in part, is a reprint of the material as it appears in A. Asgharivaskasi and N. Atanasov, "Distributed optimization with consensus constraint for multi-robot semantic octree

mapping," IEEE International Conference on Robotics and Automation (ICRA) workshop on Collaborative Perception and Learning (CoPerception), 2023. Chapter 6, in part, has been submitted for publication of the material as it may appear in A. Asgharivaskasi, F. Girke, and N. Atanasov, "Riemannian optimization for active mapping with robot teams," IEEE Transactions on Robotics (T-RO), 2024. The dissertation author was the primary investigator and author of these papers.

Chapter 7

Conclusions and Future Work

Sophisticated mobile robot tasks require knowledge of objects of interest, traversability of terrain, restricted areas, as well as other semantic cues in the environment. Hence, robot situational awareness would only be possible if we go beyond the limitations of binary occupancy representations. Moreover, we require a methodology to autonomously build such high-level semantic understanding in the face of unknown environments. This dissertation presented techniques for autonomous construction of 3-D probabilistic multi-class maps, by computing a maximally informative sensing trajectory for a general 6-degrees of freedom mobile robot. We further extended our mapping and planning approach to accomodate a team of collaborating robots with only point-to-point communication, without the need for a central estimation or planning unit.

In Chapter 2, we reviewed the mathematical concepts that are instrumental for understanding the contributions of this thesis. Topics include Bayesian grid mapping, information theory, matrix Lie groups, and Riemannian manifolds.

In Chapter 3, we introduced Bayesian semantic octree mapping from a sequence of semantically-annotated 3-D point clouds. We first generalized the log-odds mapping technique to multi-class cases, leading to a categorical probability distribution over different object classes represented by a vector of log-odds values. Then, we employed an octree data structure to significantly reduce the memory footprint of the map representation, via compressing the

neighboring voxels with similar multi-class probabilities. The performance of the semantic octree mapping method was demonstrated in both simulation and real-world.

In Chapter 4, the multi-class octree map of a partially explored environment was used to compute the Shannon mutual information between the current map estimate and a sequence of future sensor observations. The presented information measure, named Semantic Shannon Mutual Information (SSMI), explicitly considers the per-class uncertainty of the map, as well as both range sensing noise and image semantic segmentation error rate. Additionally, we showed that during calculation of the mutual information, the computational complexity with respect to the map resolution can be substantially reduced by exploiting the octree data structure of the map representation. The SSMI was utilized for identifying informative sensor poses, used as navigation goal for exploration planning. The resulting exploration strategy was then compared with several active mapping baselines in simulation and real-world mobile robot experiments.

Chapter 5 addressed the issue of non-differentiability of mutual information between grid map representations and ray-based observations. As a solution, we introduced a differentiable approximation to SSMI, by using a convex combination of exact SSMI values sampled around a given sensor pose. The differentiable formulation of information allows defining an active mapping objective function that can be augmented with other reward terms such as collision avoidance, enabling occlusion and collision-aware active mapping via gradient-ascent. The method was deployed on a ground wheeled robot, and its performance was compared with other autonomous exploration algorithms.

Finally, in Chapter 6, we showed how gradient-based active mapping can be extended to a team of collaborating robots with a single-hop communication network. We decomposed multi-robot exploration to two sub-problems, namely multi-robot mapping and multi-robot planning, and formulated both sub-problems as instances of multi-agent Riemannian optimization. A general distributed Riemannian optimization algorithm with consensus and optimality guarantees was presented, and utilized to achieve mapping and planning. The resulting distributed mapping leads to global consistency among the local map estimates for each robot, while distributed

planning allows coordination among the robots during exploration. Lastly, we demonstrated the performance of Riemannian Optimization for Active Mapping (ROAM) in simulation and real-world scenarios.

We envision several directions for future research, inspired by the work presented in this thesis.

Geometric grounding of natural language

Our 3-D semantic map representation can be used in order to identify physical coordinates of objects involved in a natural language task specification. Using a Large Language Model (LLM), such as the GPT [25] family of models, one can extract objects of interest from a natural language specification, and query the corresponding voxels in the semantic octree map. The work presented in [44] employs a given scene graph of an indoor environment for natural language grounding. However, online and accurate construction of the scene graph for a novel environment requires high computational power, which is usually not available on mobile robots. On the other hand, as shown in Chapter 3, our semantic map representation can be built on resource-constrained robots on the fly, while containing a similar type of information that scene graphs provide for geometric grounding of natural language. Hence, one can utilize our online multi-class mapping method in order to perform task planning with natural language mission specification on board of a robot, without prior knowledge of the environment.

Information-theoretic abstraction for multi-robot mapping

The distributed multi-robot mapping presented in Chapter 6 utilizes the octree data compression in order to reduce the communication bandwidth required for broadcasting the local semantic maps. Thanks to the probabilistic representation of the multi-class map, one can employ more sophisticated strategies in order to systematically abstract away semantic classes that are less relevant to a particular task. The work in [85] is a step this direction, in that the authors use an information bottleneck formulation to prune a semantic octree in an online manner. The pruned semantic octree retains high-resolution information that is relevant to the task, while

irrelevant information is abstracted away in the form of low-resolution octree leafs. This helps to significantly improve the communication burden of distributed multi-robot mapping, without sacrificing the performance, due to the selective class-aware pruning.

Learning-based approaches

With the ubiquity of GPU-accelerated computers on board mobile robots, we are motivated to use deep learning as a function approximation for SSMI, presented in Chapter 4. Using learning methods may allow predicting the mutual information beyond the limited field of view of range sensors, by recognizing the geometric and semantic patterns seen in a large collection of real-world scenes. See [116, 141] for examples. One of the main challenges for leaning SSMI would be non-smoothness with respect to sensor pose, which renders neural networks as inefficient function approximators. However, using the technique in Chapter 5 will alleviate the non-differentiability problem using convex combination of exact SSMI values. Furthermore, it is curious to see how using behavioural cloning could help to find more efficient exploration policies by imitating a rule-based method or an exploring human agent, similar to the work in [33]. Lastly, using GNNs can enable training novel distributed multi-robot exploration policies. The work by Tzes et al. [149] can be considered as a suitable inspiration for this direction, however, it is of great practical interest to extend the linear-Gaussian observation model.

Appendix A

Proof of Proposition 1

Applying Bayes rule in (3.1) and the factorization in (3.2) to $p_t(\mathbf{m})$ for some $\mathbf{z} \in \mathscr{Z}_{t+1}$ leads to:

$$\prod_{i=1}^{N} p(m_i|\mathscr{Z}_{1:t},\mathbf{z}) = \frac{p(\mathbf{z})}{p(\mathbf{z}|\mathscr{Z}_{1:t})} \prod_{i=1}^{N} \frac{p(m_i|\mathbf{z})}{p(m_i)} p(m_i|\mathscr{Z}_{1:t}). \tag{A.1}$$

The term $\frac{p(\mathbf{z})}{p(\mathbf{z}|\mathscr{Z}_{1:t})}$ may be eliminated by considering the odds ratio of an arbitrary category $m_i = k_i \in \mathscr{K}$ versus the free category $m_i = 0$ for each cell i:

$$\prod_{i=1}^{N} \frac{p(m_i = k_i|\mathscr{Z}_{1:t},\mathbf{z})}{p(m_i = 0|\mathscr{Z}_{1:t},\mathbf{z})} = \prod_{i=1}^{N} \frac{p(m_i = k_i|\mathbf{z})}{p(m_i = 0|\mathbf{z})} \frac{p(m_i = 0)}{p(m_i = k_i)} \frac{p(m_i = k_i|\mathscr{Z}_{1:t})}{p(m_i = 0|\mathscr{Z}_{1:t})}.$$

Since each term in both the left- and right-hand side products only depends on one map cell m_i, the expression holds for each individual cell. Re-writing the expression for cell m_i in vector form, with elements corresponding to each possible value of $k_i \in \mathscr{K}$, and taking an element-wise log leads to:

$$\left[\log \frac{p(m_i=0|\mathscr{Z}_{1:t},\mathbf{z})}{p(m_i=0|\mathscr{Z}_{1:t},\mathbf{z})} \cdots \log \frac{p(m_i=K|\mathscr{Z}_{1:t},\mathbf{z})}{p(m_i=0|\mathscr{Z}_{1:t},\mathbf{z})}\right]^{\top} = (\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i}) + \mathbf{h}_{t,i}. \tag{A.2}$$

Applying (A.2) recursively for each element $\mathbf{z} \in \mathscr{Z}_{t+1}$ leads to the desired result in (3.5). □

Appendix B

Proof of Proposition 2

Let $\mathscr{R}_{t+1:t+T}(r_{max}) := \cup_{\tau,b}\mathscr{R}_{\tau,b}(r_{max})$ be the set of map indices which can potentially be observed by $\underline{\mathscr{Z}}_{t+1:t+T}$. Using the factorization in (3.2) and the fact that Shannon entropy is additive for mutually independent random variables, the mutual information only depends on the cells whose index belongs to $\mathscr{R}_{t+1:t+T}(r_{max})$, i.e.:

$$I(\mathbf{m}; \underline{\mathscr{Z}}_{t+1:t+T} \mid \mathscr{Z}_{1:t}) = \sum_{\tau=t+1}^{t+T} \sum_{b=1}^{B} \sum_{i \in \mathscr{R}_{\tau,b}(r_{max})} I(m_i; \mathbf{z}_{\tau,b} \mid \mathscr{Z}_{1:t}). \tag{B.1}$$

This is true because the measurements $\mathbf{z}_{\tau,b} \in \underline{\mathscr{Z}}_{t+1:t+T}$ are independent by construction and the terms $I(m_i; \underline{\mathscr{Z}}_{t+1:t+T} \mid \mathscr{Z}_{1:t})$ can be decomposed into sums of mutual information terms between single-beam measurements $\mathbf{z}_{\tau,b}$ and the respective observed map cells m_i. The mutual information between a single map cell m_i and a sensor ray $\mathbf{z}$ is:

$$I(m_i; \mathbf{z} \mid \mathscr{Z}_{1:t}) = \int p(\mathbf{z} \mid \mathscr{Z}_{1:t}) \sum_{k=0}^{K} p(m_i = k \mid \mathbf{z}, \mathscr{Z}_{1:t}) \log \frac{p(m_i = k \mid \mathbf{z}, \mathscr{Z}_{1:t})}{p_t(m_i = k)} d\mathbf{z}. \tag{B.2}$$

Using the inverse observation model in (3.7) and the Bayesian multi-class update in (3.5), we have:

$$\begin{aligned}
&\sum_{k=0}^{K} p(m_i = k \mid \mathbf{z}, \mathscr{Z}_{1:t}) \log \frac{p(m_i = k \mid \mathbf{z}, \mathscr{Z}_{1:t})}{p_t(m_i = k)} \\
&\quad = (\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i})^\top \sigma(\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i} + \mathbf{h}_{t,i}) + \log \frac{p(m_i = 0 \mid \mathbf{z}, \mathscr{Z}_{1:t})}{p_t(m_i = 0)} \\
&\quad = f(\mathbf{l}_i(\mathbf{z}) - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}),
\end{aligned} \tag{B.3}$$

where (3.7) and (3.5) were applied a second time to the log term above. Plugging (B.3) back into the mutual information expression in (B.2) and returning to (B.1), we have:

$$\begin{aligned}
&I(\mathbf{m}; \mathscr{Z}_{t+1:t+T} \mid \mathscr{Z}_{1:t}) \\
&\quad = \sum_{\tau=t+1}^{t+T} \sum_{b=1}^{B} \sum_{y=1}^{K} \int_0^{r_{max}} \left(p(\mathbf{z}_{\tau,b} = (r,y) \mid \mathscr{Z}_{1:t}) \sum_{i \in \mathscr{R}_{\tau,b}(r_{max})} f(\mathbf{l}_i((r,y)) - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}) \right) dr.
\end{aligned} \tag{B.4}$$

For $\mathbf{z}_{\tau,b} = (r,y)$, the second term inside the integral above can be simplified to:

$$\begin{aligned}
\tilde{C}_{\tau,b}(r,y) &:= \sum_{i \in \mathscr{R}_{\tau,b}(r_{max})} f(\mathbf{l}_i((r,y)) - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}) \\
&= f(\phi^+ + \mathbf{E}_{y+1}\psi^+ - \mathbf{h}_{0,i^*_{\tau,b}}, \mathbf{h}_{t,i^*_{\tau,b}}) + \sum_{i \in \mathscr{R}_{\tau,b}(r) \setminus \{i^*_{\tau,b}\}} f(\phi^- - \mathbf{h}_{0,i}, \mathbf{h}_{t,i})
\end{aligned} \tag{B.5}$$

because for map indices $i \in \mathscr{R}_{\tau,b}(r_{max}) \setminus \mathscr{R}_{\tau,b}(r)$ that are not observed by $\mathbf{z}_{\tau,b}$, we have $\mathbf{l}_i((r,y)) = \mathbf{h}_{0,i}$ according to (3.7) and $f(\mathbf{h}_{0,i} - \mathbf{h}_{0,i}, \mathbf{h}_{t,i}) = 0$.

Next, we apply the definition of (4.4) for the first term in the integral in (B.4), which turns it into an integration over $\tilde{p}_{\tau,b}(r,y)\tilde{C}_{\tau,b}(r,y)$. Note that $\tilde{p}_{\tau,b}(r,y)$ and $\tilde{C}_{\tau,b}(r,y)$ are piecewise-constant functions since $\mathscr{R}_{\tau,b}(r)$ is constant with respect to r as long as the beam $\mathbf{z}$ lands in cell m_{i^*}. Hence, we can partition the integration domain over r into a union of intervals where the beam $\mathbf{z}$ hits the same cell, i.e., $\mathscr{R}_{\tau,b}(r)$ remains constant:

$$\int_0^{r_{max}} \tilde{p}_{\tau,b}(r,y)\tilde{C}_{\tau,b}(r,y)\, dr = \sum_{n=1}^{N_{\tau,b}} \int_{r_{n-1}}^{r_n} \tilde{p}_{\tau,b}(r,y)\tilde{C}_{\tau,b}(r,y)\, dr,$$

where $N_{\tau,b} = |\mathscr{R}_{\tau,b}(r_{max})|$, $r_0 = 0$, and $r_N = r_{max}$. From the piecewise-constant property of $\tilde{p}_{\tau,b}(r,y)$ and $\tilde{C}_{\tau,b}(r,y)$ over the interval $(r_{n-1}, r_n]$, one can obtain:

$$\int_{r_{n-1}}^{r_n} \tilde{p}_{\tau,b}(r,y)\tilde{C}_{\tau,b}(r,y)\,dr = \tilde{p}_{\tau,b}(r_n,y)\tilde{C}_{\tau,b}(r_n,y)\gamma(n) = p_{\tau,b}(n,y)C_{\tau,b}(n,y),$$

where $p_{\tau,b}(n,y)$ and $C_{\tau,b}(n,y)$ are defined in the statement of Proposition 2. Substituting y with k and plugging the integration result into (B.4) yields the lower bound in (4.6) for the mutual information between $\mathbf{m}$ and $\mathscr{Z}_{t+1:t+T}$. □

Appendix C

Proof of Proposition 3

Consider a single beam $\mathbf{z}_{\tau,b}$, passing through cells $\{m_i\}_i$, $i \in \mathscr{R}_{\tau,b}(r_{max})$. As shown in Appendix B, the mutual information between the map $\mathbf{m}$ and a beam $\mathbf{z}_{\tau,b}$ can be computed as:

$$I(\mathbf{m};\mathbf{z}_{\tau,b}|\mathscr{Z}_{1:t}) = \sum_{k=1}^{K}\sum_{n=1}^{N_{\tau,b}} p_{\tau,b}(n,k)C_{\tau,b}(n,k). \tag{C.1}$$

Assuming piece-wise constant class probabilities, we have:

$$\sum_{n=1}^{N_{\tau,b}} p_{\tau,b}(n,k)C_{\tau,b}(n,k) = \sum_{q=1}^{Q_{\tau,b}}\sum_{n=\omega_{\tau,b,1:q-1}+1}^{\omega_{\tau,b,1:q}} p_{\tau,b}(n,k)C_{\tau,b}(n,k), \tag{C.2}$$

where $\omega_{\tau,b,1:q} = \sum_{j=1}^{q}\omega_{\tau,b,j}$. For each $\omega_{\tau,b,1:q-1} < n \leq \omega_{\tau,b,1:q}$, the terms $p_{\tau,b}(n,k)$ and $C_{\tau,b}(n,k)$ are expressed as:

$$\begin{aligned} p_{\tau,b}(n,k) &= \pi_t(q,k)\pi_t^{(n-1-\omega_{\tau,b,1:q-1})}(q,0)\prod_{j=1}^{q-1}\pi_t^{\omega_{\tau,b,j}}(j,0), \\ C_{\tau,b}(n,k) &= f(\phi^+ + \mathbf{E}_{k+1}\psi^+ - \chi_{0,q},\chi_{t,q}) + (n-1-\omega_{\tau,b,1:q-1})f(\phi^- - \chi_{0,q},\chi_{t,q}) \\ &\qquad + \sum_{j=1}^{q-1}\omega_{\tau,b,j}f(\phi^- - \chi_{0,j},\chi_{t,j}). \end{aligned} \tag{C.3}$$

Plugging this into the inner summation of (C.2) leads to:

$$\sum_{n=\omega_{\tau,b,1:q-1}+1}^{\omega_{\tau,b,1:q}} p_{\tau,b}(n,k)C_{\tau,b}(n,k) = \rho_{\tau,b}(q,k)\big[\beta_{\tau,b}(q,k)\sum_{j=0}^{\omega_{\tau,b,q}-1}\pi_t^j(q,0)+f(\phi^- - \chi_{0,q},\chi_{t,q})\sum_{j=0}^{\omega_{\tau,b,q}-1} j\pi_t^j(q,0)\big], \tag{C.4}$$

The summations in (C.4) can be computed explicitly, leading to the following closed-form expression:

$$\begin{aligned}\beta_{\tau,b}(q,k)\sum_{j=0}^{\omega_{\tau,b,q}-1}\pi_t^j(q,0)+f(\phi^- - \chi_{0,q},\chi_{t,q})\sum_{j=0}^{\omega_{\tau,b,q}-1} j\pi_t^j(q,0) = \\ \beta_{\tau,b}(q,k)\frac{1-\pi_t^{\omega_{\tau,b,q}}(q,0)}{1-\pi_t(q,0)}+\frac{f(\phi^- - \chi_{0,q},\chi_{t,q})}{(1-\pi_t(q,0))^2}\Big[(\omega_{\tau,b,q}-1)\pi_t^{\omega_{\tau,b,q}+1}(q,0) \\ -\omega_{\tau,b,q}\pi_t^{\omega_{\tau,b,q}}(q,0)+\pi_t(q,0)\Big] = \Theta_{\tau,b}(q,k).\end{aligned} \tag{C.5}$$

Therefore, the Shannon mutual information between a semantic OctoMap $\mathbf{m}$ and a range-category measurement $\mathbf{z}_{\tau,b}$ can be computed as in (4.8). □

Appendix D

Proof of Proposition 4

Conditions (1) and (2) state that for any pair of viewpoints $\mathbf{V} \in \hat{\mathscr{G}}(\mathbf{X}_i)$ and $\mathbf{U} \in \hat{\mathscr{G}}(\mathbf{X}_j)$, the two sets of map cells inside the FOVs of $\mathbf{V}$ and $\mathbf{U}$ do not intersect. This is true since the spaces inside $\mathbf{V}_{\underline{\mathscr{F}}}$ and $\mathbf{U}_{\underline{\mathscr{F}}}$ are always subsets of $\mathbb{U}(\mathbf{X}_i)$ and $\mathbb{U}(\mathbf{X}_j)$, respectively, while Cond. (2) states that $\mathbb{U}(\mathbf{X}_i) \cap \mathbb{U}(\mathbf{X}_j) = \emptyset$. Consequently, the observations made from $\mathbf{V}$ and $\mathbf{U}$ are independent random variables, resulting in the following decomposition of the SMI:

$$I(\mathbf{m}; \mathbf{z}_v, \mathbf{z}_u \mid \mathbf{V}, \mathbf{U}, \mathscr{H}_t) = I(\mathbf{m}; \mathbf{z}_v \mid \mathbf{V}, \mathscr{H}_t) + I(\mathbf{m}; \mathbf{z}_u \mid \mathbf{U}, \mathscr{H}_t).$$

Following Cond. (2), the above decomposition can be applied for any set of viewpoints $\mathscr{V} := \mathbf{V}_{t+1:t+T}$ where $\mathbf{V}_{t+\tau} \in \hat{\mathscr{G}}(\mathbf{X}_{t+\tau})$, $\tau \in \{1, \ldots, T\}$. Hence we have:

$$\sum_{\tau=1}^{T} \sum_{\mathscr{V} \in \mathscr{G}^T} A_{\mathscr{V}}(\mathbf{X}_{t+1:t+T}) I(\mathbf{m}; \mathbf{z}_{t+\tau} \mid \mathbf{V}_{t+\tau}, \mathscr{H}_t) = \sum_{\tau=1}^{T} \tilde{I}(\mathbf{m}; \mathbf{z}_{t+\tau} \mid \mathbf{X}_{t+\tau}, \mathscr{H}_t) \sum_{\mathscr{V} - \mathbf{V}_{t+\tau} \in \mathscr{G}^{T-1}} \frac{A_{\mathscr{V}}(\mathbf{X}_{t+1:t+T})}{\alpha_{\mathbf{V}_{t+\tau}}(\mathbf{X}_{t+\tau})}.$$

Based on the definition of $A_{\mathscr{V}}(\mathbf{X}_{t+1:t+T})$, the inner sum is equal to 1, which yields the expression in (5.6). □

Appendix E

Proof of Proposition 5

Since the approximate SMI in (5.4) is linear with respect to $\alpha_{\mathbf{V}}(\mathbf{X})$ terms, the overall gradient computation can be reduced to a weighted sum of individual gradients of $\alpha_{\mathbf{V}}(\mathbf{X})$ with respect to robot pose $\mathbf{X}$. Also, it is only needed to compute gradients for viewpoints where $\upsilon(\delta(\xi_{\mathbf{X},\mathbf{V}})) = 1$ since the rest of the viewpoints do not affect the derivations:

$$[\nabla_{\psi}\alpha_{\mathbf{V}}(\mathbf{X}\exp(\hat{\psi}))]|_{\psi=0}^{\top} = \frac{\frac{\partial \cos\delta(\xi_{\mathbf{X}\exp(\hat{\psi}),\mathbf{V}})}{\partial\psi}\eta(\mathbf{X}) - (1+\cos\delta(\xi_{\mathbf{X},\mathbf{V}}))\frac{\partial\eta(\mathbf{X}\exp(\hat{\psi}))}{\partial\psi}}{\eta^2(\mathbf{X})},$$

where $\eta(\mathbf{X})$ is defined in (5.15). Both partial derivations in the numerator require computing $\frac{\partial \cos\delta(\xi_{\mathbf{X}\exp(\hat{\psi}),\mathbf{V}})}{\partial\psi}$. Applying the chain rule, we have:

$$\frac{\partial \cos\delta(\xi_{\mathbf{X}\exp(\hat{\psi}),\mathbf{V}})}{\partial\psi} = \frac{\partial\cos\delta}{\partial\delta}\bigg|_{\delta=\delta(\xi_{\mathbf{X},\mathbf{V}})}\frac{\partial\delta(\xi)}{\partial\xi}\bigg|_{\xi=\xi_{\mathbf{X},\mathbf{V}}}\frac{\partial\log(\exp(-\hat{\psi})\mathbf{X}^{-1}\mathbf{V})^{\vee}}{\partial\psi}\bigg|_{\psi=0}.$$

The first two partial derivatives can be obtained via differentiation in $\mathbb{R}$ and $\mathbb{R}^6$, respectively. The last partial derivative can be obtained via applying small perturbation ψ in the robot frame:

$$\frac{\partial\log(\exp(-\hat{\psi})\mathbf{X}^{-1}\mathbf{V})^{\vee}}{\partial\psi}\bigg|_{\psi=0} = \mathscr{J}_R(-\xi_{\mathbf{X},\mathbf{V}})\,\mathscr{J}_R(\xi_{\mathbf{X}}).$$

Summing over all $\mathbf{V}\in\hat{\mathscr{G}}$ leads to the expression in (5.15). □

Appendix F

Proof of Theorem 1

We organize the proof into three main steps. First, we show that the aggregate distance function $\phi(\cdot)$ is geodesically L-smooth. Second, we show that Alg. 5 converges to a consensus configuration. Last, optimality properties of Alg. 5 are derived.

Step 1. We begin the proof by showing that the aggregate distance function $\phi(\cdot)$ is geodesically L-smooth. Namely, for any pair of joint states $\mathbf{x}, \mathbf{y} \in \mathscr{M}^{|\mathscr{V}|}$ we prove:

$$\|g_\phi(\mathbf{x}) - \mathscr{T}_{\mathbf{y}}^{\mathbf{x}} g_\phi(\mathbf{y})\|_{\mathbf{x}} \leq L d(\mathbf{x}, \mathbf{y}),$$

with $g_\phi(\mathbf{x})$ as a shorthand notation for $\operatorname{grad}\phi(\mathbf{x})$.

For two joint states $\mathbf{x}, \mathbf{y} \in \mathscr{M}^{|\mathscr{V}|}$ we have:

$$\operatorname{grad}_{x^i}\phi(\mathbf{x}) - \mathscr{T}_{y^i}^{x^i} \operatorname{grad}_{y^i}\phi(\mathbf{y}) = -2\sum_{j\in\mathscr{V}} A_{ij}(v_x^{ij} - \mathscr{T}_{y^i}^{x^i} v_y^{ij}),$$

where the vectors v_x^{ij} and v_y^{ij} follow the notation in Assumption 2. Using row-stochasticity of A and the fact that $\mathscr{M}^{|\mathscr{V}|}$ is the product manifold of $\mathscr{M}$ leads to:

$$\|g_\phi(\mathbf{x}) - \mathscr{T}_{\mathbf{y}}^{\mathbf{x}} g_\phi(\mathbf{y})\|_{\mathbf{x}}^2 \leq 4 \sum_{(i,j)\in\mathscr{V}^2} A_{ij}\|v_x^{ij} - \mathscr{T}_{y^i}^{x^i} v_y^{ij}\|_{x^i}^2, \tag{F.1}$$

Adding and subtracting tangent terms v_{xy}^i and v_{xy}^j as well as utilizing the Cauchy-Schwarz

inequality in $T_{x^i}\mathscr{M}$ results in the following decomposition:

$$\|v_x^{ij} - \mathscr{T}_{y^i}^{x^i} v_y^{ij}\|_{x^i}^2 \leq \|v_{xy}^i - \mathscr{T}_{x^j}^{x^i} v_{xy}^j\|_{x^i}^2 + \|v_{xy}^{ij}\|_{x^i}^2 + 2\|v_{xy}^i - \mathscr{T}_{x^j}^{x^i} v_{xy}^j\|_{x^i} \|v_{xy}^{ij}\|_{x^i}, \tag{F.2}$$

such that the vector $v_{xy}^{ij} \in T_{x^i}\mathscr{M}$ is defined in (6.6). The vector v_{xy}^{ij} contains sum of 4 vectors corresponding to a geodesic loop. Assumption 2, in addition to applying the triangle inequality, allows finding an upper bound for (F.2) that does not involve v_x^{ij} and v_y^{ij} terms:

$$\|v_x^{ij} - \mathscr{T}_{y^i}^{x^i} v_y^{ij}\|_{x^i}^2 \leq (\rho+1)^2 (\|v_{xy}^i\|_{x^i} + \|v_{xy}^j\|_{x^j})^2.$$

Plugging into (F.1) and summing over $(i,j) \in \mathscr{V}^2$ yields:

$$\|g_\phi(\mathbf{x}) - \mathscr{T}_{\mathbf{y}}^{\mathbf{x}} g_\phi(\mathbf{y})\|_{\mathbf{x}}^2 \leq 8(\rho+1)^2 \Big(d^2(\mathbf{x},\mathbf{y}) + \sum_{(i,j)\in\mathscr{V}^2} A_{ij} \|v_{xy}^i\|_{x^i} \|v_{xy}^j\|_{x^j} \Big),$$

The summation term in the above inequality is upper-bounded by the induced norm of A, i.e. its largest eigenvalue. Using the fact that the largest eigenvalue of row-stochastic matrices is 1, we derive the following:

$$\frac{\|g_\phi(\mathbf{x}) - \mathscr{T}_{\mathbf{y}}^{\mathbf{x}} g_\phi(\mathbf{y})\|_{\mathbf{x}}}{d(\mathbf{x},\mathbf{y})} \leq 4(1+\rho) := L. \tag{F.3}$$

Therefore, the aggregate distance function $\phi(\mathbf{x})$ is geodesically L-smooth, with $L = 4(1+\rho)$.

Step 2. This step proves convergence of Alg. 5 to a consensus configuration. We use the L-smoothness of $\phi(\cdot)$ to find a bound for the values of $\phi(\mathbf{x})$. Consider two points $\mathbf{x}$ and $\mathbf{y}$ in $\mathscr{M}^{|\mathscr{V}|}$, and the geodesic $\mathbf{s}(\cdot) : [0,1] \to \mathscr{M}^{|\mathscr{V}|}$ connecting $\mathbf{x}$ to $\mathbf{y}$, i.e. $\mathbf{s}(0) = \mathbf{x}$ and $\mathbf{s}(1) = \mathbf{y}$. Using the fundamental theorem of calculus for line integrals we have:

$$\phi(\mathbf{y}) - \phi(\mathbf{x}) - \langle g_\phi(\mathbf{x}), \mathrm{Exp}_{\mathbf{x}}^{-1}(\mathbf{y}) \rangle_{\mathbf{x}} = \int_0^1 \langle \mathscr{T}_{\mathbf{s}(t)}^{\mathbf{x}} g_\phi(\mathbf{s}(t)) - g_\phi(\mathbf{x}), \mathrm{Exp}_{\mathbf{x}}^{-1}(\mathbf{y}) \rangle_{\mathbf{x}} \, dt.$$

Applying the Cauchy-Schwarz inequality and using the L-smoothness of $\phi(\mathbf{x})$ results in:

$$\phi(\mathbf{y}) - \phi(\mathbf{x}) - \langle g_\phi(\mathbf{x}), \mathrm{Exp}_{\mathbf{x}}^{-1}(\mathbf{y})\rangle_{\mathbf{x}} \leq L d^2(\mathbf{x},\mathbf{y}) \int_0^1 t\, dt = \frac{L}{2} d^2(\mathbf{x},\mathbf{y}). \tag{F.4}$$

The above bound helps to analyze the dynamics of the joint state $\mathbf{x}$ over the iterations of Alg. 5. Consider line 4 of Alg. 5. Using (F.4) leads to:

$$\phi(\tilde{\mathbf{x}}^{(k)}) \leq \phi(\mathbf{x}^{(k)}) + \langle g_\phi(\mathbf{x}^{(k)}), -\varepsilon g_\phi(\mathbf{x}^{(k)})\rangle_{\mathbf{x}^{(k)}} + \frac{L\varepsilon^2}{2} \|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}}. \tag{F.5}$$

Similarly, for line 6 of Alg. 5 we have:

$$\phi(\mathbf{x}^{(k+1)}) \leq \phi(\tilde{\mathbf{x}}^{(k)}) + \langle g_\phi(\tilde{\mathbf{x}}^{(k)}), \alpha^{(k)} g_F(\tilde{\mathbf{x}}^{(k)})\rangle_{\tilde{\mathbf{x}}^{(k)}} + \frac{L{\alpha^{(k)}}^2}{2} \|g_F(\tilde{\mathbf{x}}^{(k)})\|^2_{\tilde{\mathbf{x}}^{(k)}},$$

where, analogous to $g_\phi(\mathbf{x}^{(k)})$, $g_F(\tilde{\mathbf{x}}^{(k)})$ is shorthand notation for $\mathrm{grad} F(\mathbf{x})|_{\mathbf{x}=\tilde{\mathbf{x}}^{(k)}}$. Using the positive definiteness of the Riemannian metric $\langle v,u\rangle_{\mathbf{x}}$, we have $2\langle v,u\rangle_{\mathbf{x}} \leq \eta\|v\|^2_{\mathbf{x}} + \|u\|^2_{\mathbf{x}}/\eta$ for any $v,u \in T_{\mathbf{x}}\mathscr{M}^{|\mathscr{V}|}$ and $\eta > 0$. Hence:

$$\phi(\mathbf{x}^{(k+1)}) \leq \phi(\tilde{\mathbf{x}}^{(k)}) + \frac{\eta}{2}\|g_\phi(\tilde{\mathbf{x}}^{(k)})\|^2_{\tilde{\mathbf{x}}^{(k)}} + \frac{{\alpha^{(k)}}^2}{2}(L + \frac{1}{\eta})\|g_F(\tilde{\mathbf{x}}^{(k)})\|^2_{\tilde{\mathbf{x}}^{(k)}}.$$

By adding and subtracting $\mathscr{T}^{\tilde{\mathbf{x}}^{(k)}}_{\mathbf{x}^{(k)}} g_\phi(\mathbf{x}^{(k)})$ from $g_\phi(\tilde{\mathbf{x}}^{(k)})$, and using the fact that $\|v+u\|^2_{\mathbf{x}}/2 \leq \|v\|^2_{\mathbf{x}} + \|u\|^2_{\mathbf{x}}$, we have:

$$\begin{aligned}\phi(\mathbf{x}^{(k+1)}) \leq \phi(\tilde{\mathbf{x}}^{(k)}) &+ \frac{{\alpha^{(k)}}^2}{2}(L + \frac{1}{\eta})\|g_F(\tilde{\mathbf{x}}^{(k)})\|^2_{\tilde{\mathbf{x}}^{(k)}} \\ &+ \eta\|g_\phi(\tilde{\mathbf{x}}^{(k)}) - \mathscr{T}^{\tilde{\mathbf{x}}^{(k)}}_{\mathbf{x}^{(k)}} g_\phi(\mathbf{x}^{(k)})\|^2_{\tilde{\mathbf{x}}^{(k)}} + \eta\|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}}.\end{aligned}$$

Using the bound for gradients of the local objective functions $f^i(x^i)$ alongside utilizing the

L-smoothness of $\phi(\mathbf{x})$ and plugging (F.5) into the above inequality, yields:

$$\phi(\mathbf{x}^{(k+1)}) \leq \phi(\mathbf{x}^{(k)}) + \frac{{\alpha^{(k)}}^2}{2|\mathscr{V}|}(L+\frac{1}{\eta})C^2 + [\varepsilon(\frac{L\varepsilon}{2}-1) + \eta(1+L^2\varepsilon^2)]\|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}}.$$

Choosing $\eta = \frac{\varepsilon(2-L\varepsilon)}{4(1+L^2\varepsilon^2)}$ we have:

$$\frac{\varepsilon}{2}(1-\frac{L\varepsilon}{2})\|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}} \leq \phi(\mathbf{x}^{(k)}) - \phi(\mathbf{x}^{(k+1)}) + {\alpha^{(k)}}^2(\frac{1+\varepsilon^2L^2}{\varepsilon(1-\frac{L\varepsilon}{2})} + \frac{L}{2})\frac{C^2}{|\mathscr{V}|}. \tag{F.6}$$

Summing over $k_{\max}$ first iterations of Alg. 5 yields:

$$\frac{\varepsilon}{2}(1-\frac{L\varepsilon}{2})\sum_{k=0}^{k_{\max}}\|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}} \leq \phi(\mathbf{x}^{(0)}) - \phi(\mathbf{x}^{(k_{\max}+1)}) + \frac{C^2}{|\mathscr{V}|}(\frac{1+\varepsilon^2L^2}{\varepsilon(1-\frac{L\varepsilon}{2})} + \frac{L}{2})\sum_{k=0}^{k_{\max}}{\alpha^{(k)}}^2.$$

Because $\phi(\mathbf{x}^{(k_{\max}+1)})$ is always non-negative, we have:

$$\frac{\varepsilon}{2}(1-\frac{L\varepsilon}{2})\sum_{k=0}^{k_{\max}}\|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}} \leq \phi(\mathbf{x}^{(0)}) + \frac{C^2}{|\mathscr{V}|}(\frac{1+\varepsilon^2L^2}{\varepsilon(1-\frac{L\varepsilon}{2})} + \frac{L}{2})\sum_{k=0}^{k_{\max}}{\alpha^{(k)}}^2.$$

As a consequence of the compactness of $\mathscr{M}$, $\phi(\mathbf{x}^{(0)})$ will be bounded for any choice of $\mathbf{x}^{(0)}$. Moreover, setting $k_{\max} \to \infty$, due to the convergence property for the sum of the squared step sizes ${\alpha^{(k)}}^2$, we have:

$$\frac{\varepsilon}{2}\left(1-\frac{L\varepsilon}{2}\right)\sum_{k=0}^{\infty}\|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}} \leq \phi(\mathbf{x}^{(0)}) + \frac{C^2}{|\mathscr{V}|}\left(\frac{1+\varepsilon^2L^2}{\varepsilon(1-\frac{L\varepsilon}{2})} + \frac{L}{2}\right)\sum_{k=0}^{\infty}{\alpha^{(k)}}^2 < \infty. \tag{F.7}$$

Therefore, for $\varepsilon \in (0, 2/L)$, $g_\phi(\mathbf{x}^{(k)})$ shrinks to zero as k goes to infinity. Since $\alpha^{(k)}$ is a decaying sequence and $\|g_F(\tilde{\mathbf{x}}^{(k)})\|_{\tilde{\mathbf{x}}^{(k)}} \leq \frac{C}{\sqrt{|\mathscr{V}|}}$ for all $k \geq 0$, Alg. 5 converges to a first-order critical point of $\phi(\mathbf{x})$.

Let $\mathbf{x}^{(\infty)}$ be the joint state that Alg. 5 converges to. Also, let $\mathbf{x}_c$ be a consensus state, which can be constructed by setting all x^i, $i \in \mathscr{V}$, to an arbitrary state $x_c \in \mathscr{M}$. Since the squared distance $d^2(\cdot)$ is geodesically convex, and the adjacency matrix A is symmetric and row

stochastic, it is possible to show that $\phi(\cdot)$ is also geodesically convex. Thus we have:

$$\phi(\mathbf{x}_c) \geq \phi(\mathbf{x}^{(\infty)}) + \langle g_\phi(\mathbf{x}^{(\infty)}), \mathrm{Exp}^{-1}_{\mathbf{x}^{(\infty)}}(\mathbf{x}_c)\rangle_{\mathbf{x}^{(\infty)}}.$$

Since $\phi(\mathbf{x}_c)$ is zero, $\phi(\mathbf{x}^{(\infty)})$ is non-negative, and (F.7) indicates that $\mathbf{x}^{(\infty)}$ is a first-order critical point of $\phi(\cdot)$, we arrive at the following for Alg. 5:

$$\phi(\mathbf{x}^{(\infty)}) = 0 \tag{F.8}$$

Note that (F.7) and (F.8) hold even for a disconnected graph $\mathscr{G}$. However, for the global asymptotic consensus, $\mathscr{G}$ should be a connected graph; otherwise, consensus occurs separately for each connected component of $\mathscr{G}$.

Step 3. In the last step, we show the optimality properties of Alg. 5. We utilize the Riemannian manifold version of the law of cosines, which can be expressed for a geodesic triangle with side lengths a, b, and c as follows (Lemma 5, [168]):

$$a^2 \leq \frac{c\sqrt{|\kappa_{\min}|}}{\tanh(c\sqrt{|\kappa_{\min}|})} b^2 + c^2 - 2bc\cos(\angle bc), \tag{F.9}$$

where $\kappa_{\min}$ is a lower bound on the sectional curvature of the manifold. Now, consider a geodesic triangle specified by $x^{i(k+1)}$, $\tilde{x}^{i(k)}$, and x^*, where $x^* \in \mathscr{M}$ is an element of the centralized optimal solution of (6.3), denoted by $\mathbf{x}^* \in \mathscr{M}^{|\mathscr{V}|}$:

$$d^2(x^{i(k+1)}, x^*) \leq d^2(\tilde{x}^{i(k)}, x^*) + \zeta {\alpha^{(k)}}^2 \|g_{f^i}(\tilde{x}^{i(k)})\|^2_{\tilde{x}^{i(k)}} - 2\alpha^{(k)} \langle g_{f^i}(\tilde{x}^{i(k)}), \mathrm{Exp}^{-1}_{\tilde{x}^{i(k)}}(x^*)\rangle_{\tilde{x}^{i(k)}},$$

where $\zeta = \frac{d_{\max}\sqrt{|\kappa_{\min}|}}{\tanh(d_{\max}\sqrt{|\kappa_{\min}|})}$ with $d_{\max} \geq \max_{x,y\in\mathscr{M}} d(x,y)$. Since $\mathscr{M}$ is compact, $d_{\max}$ is well-defined. Also, the law of cosines still holds using $d_{\max}$ instead of side length $d^2(\tilde{x}^{i(k)}, x^*)$ due to the strict monotonicity of the function $\frac{y}{\tanh(y)}$ for $y \in \mathbb{R}_{\geq 0}$. As a result of the local objective

function concavity and gradient boundedness, we have:

$$2\alpha^{(k)}(f^i(x^*) - f^i(\tilde{x}^{i(k)})) \leq d^2(\tilde{x}^{i(k)}, x^*) - d^2(x^{i(k+1)}, x^*) + \zeta C^2 {\alpha^{(k)}}^2.$$

Summing over all agents in $\mathscr{V}$ yields:

$$2\alpha^{(k)}|\mathscr{V}|(F(\mathbf{x}^*) - F(\tilde{\mathbf{x}}^{(k)})) \leq d^2(\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^*) - d^2(\mathbf{x}^{(k+1)}, \mathbf{x}^*) + \zeta|\mathscr{V}|C^2{\alpha^{(k)}}^2. \tag{F.10}$$

Now, we repeat the same steps for the geodesic triangle specified by $\tilde{\mathbf{x}}^{(k)}$, $\mathbf{x}^{(k)}$, and $\mathbf{x}^*$:

$$d^2(\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^*) \leq d^2(\mathbf{x}^{(k)}, \mathbf{x}^*) + \zeta\varepsilon^2\|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}} + 2\varepsilon\langle g_\phi(\mathbf{x}^{(k)}), \mathrm{Exp}^{-1}_{\mathbf{x}^{(k)}}(\mathbf{x}^*)\rangle_{\mathbf{x}^{(k)}}.$$

Using the convexity of $\phi(\cdot)$, the fact that $\phi(\mathbf{x}^*)$ is zero by definition, and $\phi(\mathbf{x}^{(k)})$ is always non-negative, we have:

$$d^2(\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^*) \leq d^2(\mathbf{x}^{(k)}, \mathbf{x}^*) + \zeta\varepsilon^2\|g_\phi(\mathbf{x}^{(k)})\|^2_{\mathbf{x}^{(k)}}.$$

Plugging (F.6) into the above inequality yields:

$$d^2(\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^*) \leq d^2(\mathbf{x}^{(k)}, \mathbf{x}^*) + \frac{2\zeta\varepsilon}{1-\frac{L\varepsilon}{2}}\Big[\phi(\mathbf{x}^{(k)}) - \phi(\mathbf{x}^{(k+1)}) + \frac{{\alpha^{(k)}}^2C^2}{|\mathscr{V}|}[\frac{1+\varepsilon^2L^2}{\varepsilon(1-\frac{L\varepsilon}{2})} + \frac{L}{2}]\Big].$$

Plugging once more into (F.10) leads to:

$$\begin{aligned}2\alpha^{(k)}|\mathscr{V}|(F(\mathbf{x}^*) - F(\tilde{\mathbf{x}}^{(k)})) \leq\ & d^2(\mathbf{x}^{(k)}, \mathbf{x}^*) - d^2(\mathbf{x}^{(k+1)}, \mathbf{x}^*) \\ & + \zeta|\mathscr{V}|C^2{\alpha^{(k)}}^2 + \frac{2\zeta\varepsilon}{1-\frac{L\varepsilon}{2}}\Big[\phi(\mathbf{x}^{(k)}) - \phi(\mathbf{x}^{(k+1)}) + \frac{{\alpha^{(k)}}^2C^2}{|\mathscr{V}|}[\frac{1+\varepsilon^2L^2}{\varepsilon(1-\frac{L\varepsilon}{2})} + \frac{L}{2}]\Big].\end{aligned}$$

Summing over $k_{\max}$ first iterations of Alg. 5 yields:

$$2|\mathscr{V}|\sum_{k=0}^{k_{\max}}\alpha^{(k)}(F(\mathbf{x}^*)-F(\tilde{\mathbf{x}}^{(k)}))\leq d^2(\mathbf{x}^{(0)},\mathbf{x}^*)+\frac{4\zeta\phi(\mathbf{x}^{(0)})}{2/\varepsilon-L}$$
$$+\zeta C^2\Big[\frac{2\varepsilon}{|\mathscr{V}|(1-\frac{L\varepsilon}{2})}[\frac{1+\varepsilon^2L^2}{\varepsilon(1-\frac{L\varepsilon}{2})}+\frac{L}{2}]+|\mathscr{V}|\Big]\sum_{k=0}^{k_{\max}}{\alpha^{(k)}}^2.$$

It is straightforward to show:

$$\min_{0\leq k'\leq k_{\max}}\{F(\mathbf{x}^*)-F(\tilde{\mathbf{x}}^{(k')})\}\sum_{k=0}^{k_{\max}}\alpha^{(k)}\leq\sum_{k=0}^{k_{\max}}\alpha^{(k)}(F(\mathbf{x}^*)-F(\tilde{\mathbf{x}}^{(k)})).$$

Therefore, we have:

$$F(\mathbf{x}^*)\leq\max_{0\leq k'\leq k_{\max}}\{F(\tilde{\mathbf{x}}^{(k')})\}+\frac{1}{2|\mathscr{V}|\sum_{k=0}^{k_{\max}}\alpha^{(k)}}\Big[d^2(\mathbf{x}^{(0)},\mathbf{x}^*)+\frac{4\zeta\phi(\mathbf{x}^{(0)})}{2/\varepsilon-L}$$
$$+\zeta C^2\Big(\frac{2\varepsilon}{|\mathscr{V}|(1-\frac{L\varepsilon}{2})}(\frac{1+\varepsilon^2L^2}{\varepsilon(1-\frac{L\varepsilon}{2})}+\frac{L}{2})+|\mathscr{V}|\Big)\sum_{k=0}^{k_{\max}}{\alpha^{(k)}}^2\Big].$$

Since $d^2(\mathbf{x}^{(0)},\mathbf{x}^*)$, $\phi(\mathbf{x}^{(0)})$, and $\sum_{k=0}^{\infty}{\alpha^{(k)}}^2$ are bounded and $\sum_{k=0}^{\infty}\alpha^{(k)}=\infty$, the term inside the brackets in the right hand side of the above inequality vanishes as $k_{\max}\to\infty$. Therefore, $\max_{0\leq k'\leq k_{\max}}\{F(\tilde{\mathbf{x}}^{(k')})\}$ will be asymptotically lower-bounded by $F(\mathbf{x}^*)$. Moreover, since $\|g_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}$ and $\alpha^{(k)}$ both shrink to zero as $k\to\infty$, we derive the same asymptotic lower bound for $F(\mathbf{x}^{(k)})$:

$$F(\mathbf{x}^*)\leq\lim_{k_{\max}\to\infty}\max_{0\leq k\leq k_{\max}}\{F(\mathbf{x}^{(k)})\}. \tag{F.11}$$

In summary, the relations in (F.7) and (F.8) show convergence to a consensus configuration for step size $\varepsilon\in(0,2/L)$, where $L=4(1+\rho)$. Furthermore, (F.11) expresses the optimality of Alg. 5 with respect to a centralized solution. □

Appendix G

Proof of Lemma 1

The conditional independence of observations given the map $\mathbf{m}$ allows decomposing the observation model $q^i(\mathbf{z}^i_{1:t}|\mathbf{m})$ into a product of single observation models $q^i(\mathbf{z}^i_\tau|\mathbf{m})$, $\tau \in \{1,\dots,t\}$. By applying Bayes rule to the decomposed observation model, the objective function in (6.9) can be written as:

$$\sum_{i\in\mathscr{V}}\sum_{\tau=1}^{t}\Big(\log q^i(\mathbf{z}^i_\tau) + \mathbb{E}_{\mathbf{m}\sim p}\big[\log\frac{q^i(\mathbf{m}|\mathbf{z}^i_\tau)}{p(\mathbf{m})}\big]\Big).$$

Using the map independence assumption, the log term inside the expectation can be expressed as the sum of log terms with respect to single cells. The additivity of expected value yields:

$$\sum_{i\in\mathscr{V}}\sum_{\tau=1}^{t}\Big(\log q^i(\mathbf{z}^i_\tau) + \sum_{n=1}^{N}\mathbb{E}_{\mathbf{m}\sim p}\big[\log\frac{q^i(m_n|\mathbf{z}^i_\tau)}{p_n(m_n)}\big]\Big).$$

Since every term inside the expectation only depends on a single cell $m_n \sim p_n$, $n \in \{1,\dots,N\}$, the expectation can thus be simplified to only incorporate one cell instead of the joint distribution $\mathbf{m} \sim p$. This leads to the expression in (6.11). □

Bibliography

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[2] Amirhossein Afsharrad, Ahmadreza Moradipari, and Sanjay Lall. Convex methods for constrained linear bandits. In *2024 European Control Conference (ECC)*, pages 2111–2118. IEEE, 2024.

[3] Mark Agate, Richard L. Grimsdale, and Paul F. Lister. The HERO algorithm for ray-tracing octrees. In *Advances in Computer Graphics Hardware IV*, pages 61–73, 1991.

[4] Mohammad Akbari, Bahman Gharesifard, and Tamás Linder. Individual regret bounds for the distributed online alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, 64(4):1746–1752, 2018.

[5] Randa Almadhoun, Tarek Taha, Lakmal Seneviratne, and Yahya Zweiri. A survey on multi-robot coverage path planning for model reconstruction and mapping. *SN Applied Sciences*, 1:1–24, 2019.

[6] Arash Asgharivaskasi and Nikolay Atanasov. Active Bayesian multi-class mapping from range and semantic segmentation observations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2021.

[7] Arash Asgharivaskasi and Nikolay Atanasov. Distributed optimization with consensus constraint for multi-robot semantic octree mapping. In *Co-Perception: ICRA workshop on collaborative perception and learning*, 2023.

[8] Arash Asgharivaskasi and Nikolay Atanasov. Semantic octree mapping and Shannon mutual information computation for robot exploration. *IEEE Transactions on Robotics*, 39(3):1910–1928, 2023.

[9] Arash Asgharivaskasi, Fritz Girke, and Nikolay Atanasov. Riemannian optimization for active mapping with robot teams. *arXiv preprint arXiv:2404.18321*, 2024.

[10] Arash Asgharivaskasi, Shumon Koga, and Nikolay Atanasov. Active mapping via gradient ascent optimization of Shannon mutual information over continuous SE(3) trajectories. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12994–13001, 2022.

[11] Mahmoud Assran and Mike Rabbat. On the convergence of Nesterov's accelerated gradient method in stochastic settings. In *International Conference on Machine Learning*, pages 410–420, 2020.

[12] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J. Pappas. Information acquisition with sensing robots: Algorithms and error bounds. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6447–6454, 2014.

[13] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J. Pappas. Decentralized active information acquisition: Theory and application to multi-robot SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4775–4782, 2015.

[14] Nikolay Atanasov, Jerome Le Ny, Kostas Daniilidis, and George J. Pappas. Decentralized active information acquisition: Theory and application to multi-robot SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4775–4782, 2015.

[15] Nikolay Atanasov, Jerome Le Ny, and George J. Pappas. Distributed algorithms for stochastic source seeking with mobile robot networks. *Journal of Dynamic Systems, Measurement, and Control*, 137(3), 2014.

[16] Nikolay Atanasov, Bharath Sankaran, Jerome Le Ny, George J. Pappas, and Kostas Daniilidis. Non-myopic view planning for active object classification and pose estimation. *IEEE Transactions on Robotics*, 30(5):1078–1090, 2014.

[17] Nikolay Atanasov, Roberto Tron, Victor M. Preciado, and George J Pappas. Joint estimation and localization in sensor networks. In *IEEE Conference on Decision and Control*, pages 6875–6882, 2014.

[18] Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2017.

[19] Graeme Best, Oliver M. Cliff, Timothy Patten, Ramgopal R. Mettu, and Robert Fitch. Dec-MCTS: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research*, 38(2-3):316–337, 2019.

[20] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon "next-best-view" planner for 3D exploration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1462–1468, 2016.

[21] Hermann Blum, Silvan Rohrbach, Marija Popovic, Luca Bartolomei, and Roland Y. Siegwart. Active learning for UAV-based semantic mapping. *ArXiv*, abs/1908.11157, 2019.

[22] Nicolas Boumal. An introduction to optimization on smooth manifolds. To appear with Cambridge University Press, Jun 2022.

[23] Frederic Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 540–545, 2002.

[24] Jack E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[25] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[26] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics (TRO)*, 21(3):376–386, 2005.

[27] Xiaoyi Cai, Brent Schlotfeldt, Kasra Khosoussi, Nikolay Atanasov, George J. Pappas, and Jonathan P. How. Energy-aware, collision-free information gathering for heterogeneous robot teams. *IEEE Transactions on Robotics*, 39(4):2585–2602, 2023.

[28] Chao Cao, Hongbiao Zhu, Howie Choset, and Ji Zhang. TARE: A hierarchical framework for efficiently exploring complex 3D environments. In *Robotics: Science and Systems (RSS)*, 2021.

[29] Luca Carlone, Jingjing Du, Miguel Kaouk Ng, Basilio Bona, and Marina Indri. Active SLAM and exploration with particle filters using Kullback-Leibler divergence. *Journal of Intelligent & Robotic Systems*, 75(2):291–311, 2014.

[30] Henry Carrillo, Ian Reid, and José A. Castellanos. On the comparison of uncertainty criteria for active SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2080–2087, 2012.

[31] Andrea Censi. An ICP variant using a point-to-line metric. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 19–25, 2008.

[32] Ping Chao. Fully convolutional HarDNet for segmentation in Pytorch. https://github.com/PingoLH/FCHarDNet. Accessed: 2022-07-01.

[33] Devendra S. Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural SLAM. In *International Conference on Learning Representations*, 2020.

[34] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-theoretic planning with trajectory optimization for dense 3D mapping. In *Robotics: Science and Systems*, volume 11, pages 3–12, 2015.

[35] Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using Cauchy-Schwarz quadratic mutual information. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[36] Fanfei Chen, Paul Szenher, Yewei Huang, Jinkun Wang, Tixiao Shan, Shi Bai, and Brendan Englot. Zero-shot reinforcement learning on graphs for autonomous exploration under uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5193–5199, 2021.

[37] Jianshu Chen and Ali H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, 60(8):4289–4305, 2012.

[38] Kenny Chen, Brett T. Lopez, Ali-akbar Agha-mohammadi, and Ankur Mehta. Direct LiDAR odometry: Fast localization with dense point clouds. *IEEE Robotics and Automation Letters*, 7(2):2000–2007, 2022.

[39] Shixiang Chen, Alfredo Garcia, Mingyi Hong, and Shahin Shahrampour. Decentralized Riemannian gradient descent on the Stiefel manifold. In *International Conference on Machine Learning*, pages 1594–1605, 2021.

[40] Thomas B. Cohn, Mark Petersen, Max Simchowitz, and Russ Tedrake. Non-Euclidean motion planning with graphs of geodesically-convex sets. In *Proceedings of Robotics: Science and Systems*, 2023.

[41] Micah Corah and Nathan Michael. Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice. *Autonomous Robots*, 43:485–501, 2019.

[42] Micah Corah and Nathan Michael. Volumetric objectives for multi-robot exploration of three-dimensional environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9043–9050, 2021.

[43] Micah Corah, Cormac O'Meadhra, Kshitij Goel, and Nathan Michael. Communication-efficient planning and mapping for multi-robot exploration in large environments. *IEEE Robotics and Automation Letters*, 4(2):1715–1721, 2019.

[44] Zhirui Dai, Arash Asgharivaskasi, Thai Duong, Shusen Lin, Maria-Elizabeth Tzes, George Pappas, and Nikolay Atanasov. Optimal scene graph planning with large language model guidance. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14062–14069, 2024.

[45] Tung Dang, Frank Mascarich, Shehryar Khattak, Christos Papachristos, and Kostas Alexis. Graph-based path planning for autonomous robotic exploration in subterranean environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3105–3112, 2019.

[46] Di Deng, Runlin Duan, Jiahong Liu, Kuangjie Sheng, and Kenji Shimada. Robotic exploration of unknown 2D environment using a frontier-based automatic-differentiable information gain measure. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1497–1503, 2020.

[47] Di Deng, Zhefan Xu, Wenbo Zhao, and Kenji Shimada. Frontier-based automatic-differentiable information gain measure for robotic exploration of unknown 3D environments. *arXiv preprint arXiv:2011.05288*, 2020.

[48] Yinan Deng, Yujie Tang, Yi Yang, Danwei Wang, and Yufeng Yue. MACIM: Multi-agent collaborative implicit mapping. *IEEE Robotics and Automation Letters*, 9(5):4369–4376, 2024.

[49] Haolin Dong, Jincheng Yu, Yuanfan Xu, Zhilin Xu, Zhaoyang Shen, Jiahao Tang, Yuan Shen, and Yu Wang. MR-GMMapping: Communication efficient multi-robot mapping system via Gaussian mixture model. *IEEE Robotics and Automation Letters*, 7(2):3294–3301, 2022.

[50] Siyan Dong, Kai Xu, Qiang Zhou, Andrea Tagliasacchi, Shiqing Xin, Matthias Nießner, and Baoquan Chen. Multi-robot collaborative dense scene reconstruction. *ACM Transactions on Graphics (TOG)*, 38(4):1–16, 2019.

[51] Thibaud Duhautbout, Julien Moras, and Julien Marzat. Distributed 3D TSDF manifold mapping for multi-robot systems. In *European Conference on Mobile Robots (ECMR)*, pages 1–8, 2019.

[52] Thai Duong, Abdullah Altawaitan, Jason Stanley, and Nikolay Atanasov. Port-Hamiltonian neural ODE networks on Lie groups for robot dynamics learning and control. *arXiv preprint arXiv:2401.09520*, 2024.

[53] Ayan Dutta, Anirban Ghosh, and O. Patrick Kreidl. Multi-robot informative path planning with continuous connectivity constraints. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3245–3251, 2019.

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

[55] Alberto Elfes. Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework. In *Reasoning With Uncertainty in Robotics*, pages 93–130, 1995.

[56] Péter Fankhauser, Michael Bloesch, and Marco Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters*, 3(4):3019–3026, 2018.

[57] Lu Gan, Ray Zhang, Jessy W. Grizzle, Ryan M. Eustice, and Maani Ghaffari. Bayesian spatial kernel smoothing for scalable dense semantic mapping. *IEEE Robotics and Automation Letters*, 5(2):790–797, 2020.

[58] Miguel Angel Garcia and Agusti Solanas. 3D simultaneous localization and modeling from stereo vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 847–853. IEEE, 2004.

[59] Georgios Georgakis, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, and Kostas Daniilidis. Learning to map for active semantic goal navigation. *ArXiv*, abs/2106.15648, 2021.

[60] Bahman Gharesifard and Jorge Cortés. Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Transactions on Automatic Control*, 59(3):781–786, 2014.

[61] Bahman Gharesifard and Stephen L Smith. Distributed submodular maximization with limited information. *IEEE transactions on control of network systems*, 5(4):1635–1645, 2017.

[62] Clara Gómez, Marius Fehr, Alejandra C. Hernández, Juan Nieto, Ramon Barber, and Roland Siegwart. Hybrid topological and 3D dense mapping through autonomous exploration for large indoor environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[63] Héctor H. González-Baños and Jean-Claude Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research (IJRR)*, 21(10-11):829–848, 2002.

[64] Walker Gosrich, Siddharth Mayya, Rebecca Li, James Paulos, Mark Yim, Alejandro Ribeiro, and Vijay Kumar. Coverage control in multi-robot systems via graph neural networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8787–8793, 2022.

[65] Ben Grocholsky, James Keller, Vijay Kumar, and George Pappas. Cooperative air and ground surveillance. *IEEE Robotics & Automation Magazine*, 13(3):16–25, 2006.

[66] Trevor Halsted, Ola Shorinwa, Javier Yu, and Mac Schwager. A survey of distributed optimization methods for multi-robot systems. *arXiv preprint arXiv:2103.12840*, 2021.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[68] Theia Henderson, Vivienne Sze, and Sertac Karaman. An efficient and continuous approach to information-theoretic exploration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[69] A. Hong, O. Igharoro, Yugang Liu, Farzad Niroui, Goldie Nejat, and Beno Benhabib. Investigating human-robot teams for learning-based semi-autonomous control in urban search and rescue environments. *Journal of Intelligent & Robotic Systems*, 94:669–686, 2019.

[70] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at https://octomap.github.io.

[71] Junyan Hu, Hanlin Niu, Joaquin Carrasco, Barry Lennox, and Farshad Arvin. Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12):14413–14423, 2020.

[72] Junyan Hu, Hanlin Niu, Joaquin Carrasco, Barry Lennox, and Farshad Arvin. Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12):14413–14423, 2020.

[73] Xiang Huang, Min Sun, Hang Zhou, and Shuai Liu. A multi-robot coverage path planning algorithm for the environment with multiple land cover types. *IEEE Access*, 8:198101–198117, 2020.

[74] J. Jessup, Sidney N. Givigi, and Alain Beaulieu. Merging of octree based 3D occupancy grid maps. In *2014 IEEE International Systems Conference Proceedings*, pages 371–377, 2014.

[75] Brian J. Julian. *Mutual information-based gradient-ascent control for distributed robotics*. PhD thesis, Massachusetts Institute of Technology, 2013.

[76] Brian J. Julian, Sertac Karaman, and Daniela Rus. On mutual information-based control of range sensing robots for mapping applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5156–5163, 2013.

[77] Xinyue Kan, Hanzhe Teng, and Konstantinos Karydis. Online exploration and coverage planning in unknown obstacle-cluttered environments. *IEEE Robotics and Automation Letters*, 5(4):5969–5976, 2020.

[78] Yiannis Kantaros, Brent Schlotfeldt, Nikolay Atanasov, and George J. Pappas. Asymptotically optimal planning for non-myopic multi-robot information gathering. In *Robotics: Science and Systems*, pages 22–26, 2019.

[79] Joseph Knuth and Prabir Barooah. Collaborative 3D localization of robots from relative pose measurements using gradient descent on manifolds. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[80] Shumon Koga, Arash Asgharivaskasi, and Nikolay Atanasov. Active exploration and mapping via iterative covariance regulation over continuous SE(3) trajectories. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2735–2741, 2021.

[81] Shumon Koga, Arash Asgharivaskasi, and Nikolay Atanasov. Active SLAM over continuous trajectory and control: A covariance-feedback approach. *arXiv preprint:2110.07546*, 2022.

[82] Shumon Koga, Arash Asgharivaskasi, and Nikolay Atanasov. Active SLAM over continuous trajectory and control: A covariance-feedback approach. In *American Control Conference (ACC)*, pages 5062–5068, 2022.

[83] Thomas Kollar and Nicholas Roy. Efficient optimization of information-theoretic exploration in SLAM. In *AAAI Conference on Artificial Intelligence*, page 1369–1375, 2008.

[84] Christian Lageman, Alain Sarlette, and Rodolphe Sepulchre. Synchronization with partial state feedback on SO(n). In *IEEE Conference on Decision and Control (CDC)*, pages 1696–1701, 2009.

[85] Daniel T. Larsson, Arash Asgharivaskasi, Jaein Lim, Nikolay Atanasov, and Panagiotis Tsiotras. Information-theoretic abstraction of semantic octree models for integrated perception and planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6937–6943, 2023.

[86] Husheng Li and Zhu Han. Competitive spectrum access in cognitive radio networks: Graphical game and learning. In *IEEE Wireless Communication and Networking Conference*, pages 1–6, 2010.

[87] Jiaxiang Li and Shiqian Ma. Federated learning on Riemannian manifolds. *arXiv preprint arXiv:2206.05668*, 2022.

[88] Jiazhen Liu, Lifeng Zhou, Ragesh Ramachandran, Gaurav S. Sukhatme, and Vijay Kumar. Decentralized risk-aware tracking of multiple targets. In *Distributed Autonomous Robotic Systems*, pages 408–423. Springer Nature Switzerland, 2024.

[89] Xu Liu, Ankit Prabhu, Fernando Cladera, Ian D. Miller, Lifeng Zhou, Camillo J. Taylor, and Vijay Kumar. Active metric-semantic mapping by multiple aerial robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3282–3288, 2023.

[90] Max Lodel, Bruno Brito, Alvaro Serra-Gómez, Laura Ferranti, Robert Babuška, and Javier Alonso-Mora. Where to look next: Learning viewpoint recommendations for informative trajectory planning. *arXiv preprint arXiv:2203.02381*, 2022.

[91] Hongxuan Ma, Wei Zou, Siyang Sun, Zheng Zhu, and Zhaobing Kang. FOV constraint region analysis and path planning for mobile robot with observability to multiple feature points. *International Journal of Control, Automation and Systems*, 19(11):3785–3800, 2021.

[92] Angel Madridano, Abdulla Al-Kaff, David Martín, and Arturo De La Escalera. Trajectory planning for multi-robot systems: Methods and applications. *Expert Systems with Applications*, 173:114660, 2021.

[93] Renan Maffei, Marcos P. Souza, Mathias Mantelli, Diego Pittol, Mariana Kolberg, and Vitor A. M. Jorge. Exploration of 3D terrains using potential fields with elevation-based

local distortions. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[94] Richard M. Marino, W. R. Davis, G. C. Rich, J. L. McLaughlin, E. I. Lee, B. M. Stanley, J. W. Burnside, G. S. Rowe, R. E. Hatch, T. E. Square, et al. High-resolution 3D imaging laser radar flight test experiments. In *Laser Radar Technology and Applications X*, volume 5791, pages 138–151. SPIE, 2005.

[95] Larry Matthies and Alberto Elfes. Integration of sonar and stereo range data using a grid-based representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1988.

[96] Jasna Maver and Ruzena Bajcsy. Occlusions as a guide for planning the next view. *IEEE transactions on pattern analysis and machine intelligence*, 15(5):417–433, 1993.

[97] Andres Milioto and Cyrill Stachniss. Bonnet: An open-source training and deployment framework for semantic segmentation in robotics using CNNs. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[98] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet++: Fast and accurate LiDAR semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220, 2019.

[99] Naoki Mizuno, Kazunori Ohno, Ryunosuke Hamada, Hiroyoshi Kojima, Jun Fujita, Hisanori Amano, Thomas Westfechtel, Takahiro Suzuki, and Satoshi Tadokoro. Enhanced path smoothing based on conjugate gradient descent for firefighting robots in petrochemical complexes. *Advanced Robotics*, 33(14):687–698, 2019.

[100] Stewart J Moorehead, Reid Simmons, and William L. Whittaker. Autonomous exploration using multiple sources of information. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3098–3103, 2001.

[101] Ahmadreza Moradipari, Sanae Amani, Mahnoosh Aliradeh, and Christos Thrampoulidis. Safe linear bandits. In *Annual Conference on Information Sciences and Systems (CISS)*, pages 1–1. IEEE, 2021.

[102] Ahmadreza Moradipari, Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. Safe linear Thompson sampling with side information. *IEEE Transactions on Signal Processing*, 69:3755–3767, 2021.

[103] Ahmadreza Moradipari, Mohammad Ghavamzadeh, and Mahnoosh Alizadeh. Collaborative multi-agent stochastic linear bandits. In *2022 American Control Conference (ACC)*, pages 2761–2766. IEEE, 2022.

[104] Ahmadreza Moradipari, Christos Thrampoulidis, and Mahnoosh Alizadeh. Stage-wise conservative linear bandits. *Advances in neural information processing systems*, 33:11191–11201, 2020.

[105] Ahmadreza Moradipari, Berkay Turan, Yasin Abbasi-Yadkori, Mahnoosh Alizadeh, and Mohammad Ghavamzadeh. Feature and parameter selection in stochastic linear bandits. In *International Conference on Machine Learning*, pages 15927–15958. PMLR, 2022.

[106] Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61, 1988.

[107] David Morilla-Cabello, Luca Bartolomei, Lucas Teixeira, Eduardo Montijano, and Margarita Chli. Sweep-your-map: Efficient coverage planning for aerial teams in large-scale environments. *IEEE Robotics and Automation Letters*, 7(4):10810–10817, 2022.

[108] Angelia Nedić. Asynchronous broadcast-based convex optimization over a network. *IEEE Transactions on Automatic Control*, 56(6):1337–1351, 2011.

[109] Angelia Nedić. *Convergence Rate of Distributed Averaging Dynamics and Optimization in Networks*. Now Publishers Inc., 2015.

[110] Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2014.

[111] Ngoc Thinh Nguyen, Lars Schilling, Michael Sebastian Angern, Heiko Hamann, Floris Ernst, and Georg Schildbach. B-spline path planner for safe navigation of mobile robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 339–345, 2021.

[112] NVIDIA. Deep learning nodes for ROS/ROS2. https://github.com/dusty-nv/ros_deep_learning. Accessed: 2021-07-10.

[113] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. ISDF: Real-time neural signed distance fields for robot perception. In *Robotics: Science and Systems (RSS)*, 2022.

[114] Parth Paritosh, Nikolay Atanasov, and Sonia Martinez. Distributed Bayesian estimation of continuous variables over time-varying directed networks. *IEEE Control Systems Letters*, 6:2545–2550, 2022.

[115] Thomas Pierrot, Nicolas Perrin-Gilbert, and Olivier Sigaud. First-order and second-order variants of the gradient descent in a unified framework. In *International Conference on Artificial Neural Networks*, pages 197–208, 2021.

[116] Julio A Placed and José A. Castellanos. A deep reinforcement learning approach for active SLAM. *Applied Sciences*, 10(23):8386, 2020.

[117] Julio A. Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A. Castellanos. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, 39(3):1686–1705, 2023.

[118] Amanda Prorok, Matthew Malencia, Luca Carlone, Gaurav S Sukhatme, Brian M Sadler, and Vijay Kumar. Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems. *arXiv preprint arXiv:2109.12343*, 2021.

[119] Jorge Pena Queralta, Jussi Taipalmaa, Bilge Can Pullinen, Victor Kathan Sarker, Tuan Nguyen Gia, Hannu Tenhunen, Moncef Gabbouj, Jenni Raitoharju, and Tomi Westerlund. Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision. *IEEE Access*, 8:191617–191643, 2020.

[120] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Y Ng. ROS: An open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.

[121] Srinivasan S. Ram, Angelia Nedić, and Venugopal V. Veeravalli. A new class of distributed optimization algorithms: Application to regression of distributed data. *Optimization Methods and Software*, 27(1):71–88, 2012.

[122] Rui Rocha, Jorge Dias, and Adriano Carvalho. Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theory. *Robotics and Autonomous Systems*, 53(3-4):282–311, 2005.

[123] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: An open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696, 2020.

[124] Tomáš Rouček, Martin Pecka, Petr Čížek, Tomáš Petříček, Jan Bayer, Vojtěch Šalanskỳ, Daniel Heřt, Matěj Petrlík, Tomáš Báča, Vojěch Spurnỳ, et al. DARPA subterranean challenge: Multi-robotic exploration of underground environments. In *Modelling and Simulation for Autonomous Systems: 6th International Conference, MESAS 2019, Palermo, Italy, October 29–31, 2019, Revised Selected Papers 6*, pages 274–290. Springer, 2020.

[125] Hanan Samet. Implementing ray tracing with octrees and neighbor finding. *Computers & Graphics*, 13(4):445–460, 1989.

[126] Alain Sarlette, Silvère Bonnabel, and Rodolphe Sepulchre. Coordinated motion design on Lie groups. *IEEE Transactions on Automatic Control*, 55(5):1047–1058, 2010.

[127] Alain Sarlette, Rodolphe Sepulchre, and Naomi Ehrich Leonard. Autonomous rigid body attitude synchronization. *Automatica*, 45(2):572–577, 2009.

[128] Kelsey Saulnier, Nikolay Atanasov, George J. Pappas, and Vijay Kumar. Information theoretic active exploration in signed distance fields. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[129] Brent Schlotfeldt, Dinesh Thakur, Nikolay Atanasov, Vijay Kumar, and George J. Pappas. Anytime planning for decentralized multirobot active information gathering. *IEEE Robotics and Automation Letters (RA-L)*, 3(2):1025–1032, 2018.

[130] Stephen Se and Piotr Jasiobedzki. Stereo-vision based 3D modeling and localization for unmanned vehicles. *International Journal of Intelligent Control and Systems*, 13(1):47–58, 2008.

[131] Rodolphe Sepulchre, Alain Sarlette, and Pierre Rouchon. Consensus in non-commutative spaces. In *IEEE conference on decision and control (CDC)*, pages 6596–6601, 2010.

[132] Suhail M. Shah. Distributed optimization on Riemannian manifolds for multi-agent networks. *arXiv preprint arXiv:1711.11196*, 2017.

[133] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[134] Shaojie Shen, Nathan Michael, and Vijay Kumar. Stochastic differential equation-based exploration algorithm for autonomous indoor 3D exploration with a micro-aerial vehicle. *The International Journal of Robotics Research (IJRR)*, 31(12):1431–1444, 2012.

[135] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015.

[136] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1754, 2017.

[137] Kunal Srivastava and Angelia Nedić. Distributed asynchronous constrained stochastic optimization. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):772–790, 2011.

[138] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems*, pages 65–72, 2005.

[139] Shreyas Sundaram and Bahman Gharesifard. Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, 64(3):1063–1076, 2018.

[140] Vincenzo Suriani, Sara Kaszuba, Sandeep R. Sabbella, Francesco Riccio, and Daniele Nardi. S-AvE: Semantic active vision exploration and mapping of indoor environments for mobile robots. In *2021 European Conference on Mobile Robots (ECMR)*, pages 1–8, 2021.

[141] Yuezhan Tao, Yuwei Wu, Beiming Li, Fernando Cladera, Alex Zhou, Dinesh Thakur, and Vijay Kumar. SEER: Safe efficient exploration for aerial robots using learning to predict information gain. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1235–1241, 2023.

[142] Sebastian Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2):111–127, 2003.

[143] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press Cambridge, 2005.

[144] Yafu Tian, Ke Wang, Ruifeng Li, and Lijun Zhao. A fast incremental map segmentation algorithm based on spectral clustering and quadtree. *Advances in Mechanical Engineering*, 10(2):1–15, 2018.

[145] Yulun Tian, Yun Chang, Fernando Herrera Arias, Carlos Nieto-Granda, Jonathan P. How, and Luca Carlone. Kimera-multi: Robust, distributed, dense metric-semantic SLAM for multi-robot systems. *IEEE Transactions on Robotics*, 38(4), 2022.

[146] Yulun Tian, Alec Koppel, Amrit Singh Bedi, and Jonathan P How. Asynchronous and parallel distributed pose graph optimization. *IEEE Robotics and Automation Letters*, 5(4):5819–5826, 2020.

[147] Behrouz Touri and Bahman Gharesifard. Continuous-time distributed convex optimization on time-varying directed networks. In *IEEE Conference on Decision and Control (CDC)*, pages 724–729, 2015.

[148] Konstantinos I. Tsianos, Sean Lawlor, and Michael G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *IEEE Conference on Decision and Control (CDC)*, pages 5453–5458, 2012.

[149] Mariliza Tzes, Nikolaos Bousias, Evangelos Chatzipantazis, and George J. Pappas. Graph neural networks for multi-robot active information acquisition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3497–3503, 2023.

[150] Mariliza Tzes, Yiannis Kantaros, and George J. Pappas. Distributed sampling-based planning for non-myopic active information gathering. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5872–5877, 2021.

[151] Unity Technologies. Unity Engine. https://unity.com/products/unity-engine. Accessed: 2023-11-01.

[152] Tiago S. Veiga, Miguel Silva, Rodrigo Ventura, and Pedro U. Lima. A hierarchical approach to active semantic mapping using probabilistic logic and information eeward POMDPs. *Proceedings of the International Conference on Automated Planning and Scheduling*, 29(1):773–781, May 2021.

[153] Arnoud Visser and Bayu Slamet. Balancing the information gain against the movement cost for multi-robot frontier exploration. In *European Robotics Symposium*, pages 43–52, 2008.

[154] Jinkun Wang, Tixiao Shan, and Brendan Englot. Virtual maps for autonomous exploration with pose SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[155] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198, 2016.

[156] Lei Wang and Xin Liu. Decentralized optimization over the Stiefel manifold by an approximate augmented Lagrangian function. *IEEE Transactions on Signal Processing*, 70:3029–3041, 2022.

[157] Tianyu Wang, Vikas Dhiman, and Nikolay Atanasov. Learning navigation costs from demonstration with semantic observations. *Proceedings of Machine Learning Research vol*, 120:1–11, 2020.

[158] Maggie Wigness, Sungmin Eum, John G. Rogers, David Han, and Heesung Kwon. A RUGD dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007, 2019.

[159] Wenying Wu, Subhrajit Bhattacharya, and Amanda Prorok. Multi-robot path deconfliction through prioritization by path prospects. In *IEEE international conference on robotics and automation (ICRA)*, pages 9809–9815, 2020.

[160] Yichun Wu, Qiuyi Gu, Jincheng Yu, Guangjun Ge, Jian Wang, Qingmin Liao, Chun Zhang, and Yu Wang. MR-GMMExplore: Multi-robot exploration system in unknown environments based on Gaussian mixture model. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1198–1203, 2022.

[161] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, 1997.

[162] Pengzhi Yang, Shumon Koga, Arash Asgharivaskasi, and Nikolay Atanasov. Policy learning for active target tracking over continuous SE(3) trajectories. In *Learning for Dynamics and Control Conference*, pages 64–75, 2023.

[163] Pengzhi Yang, Yuhan Liu, Shumon Koga, Arash Asgharivaskasi, and Nikolay Atanasov. Learning continuous control policies for information-theoretic active perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2098–2104, 2023.

[164] Jincheng Yu, Jianming Tong, Yuanfan Xu, Zhilin Xu, Haolin Dong, Tianxiang Yang, and Yu Wang. SMMR-Explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8779–8785, 2021.

[165] Rahul Zahroof, Jiazhen Liu, Lifeng Zhou, and Vijay Kumar. Multi-robot localization and target tracking with connectivity maintenance and collision avoidance. In *2023 American Control Conference (ACC)*, pages 1331–1338, 2023.

[166] Hao Zhang, Jiyu Cheng, Lin Zhang, Yibin Li, and Wei Zhang. H2GNN: Hierarchical-hops graph neural networks for multi-robot exploration in unknown environments. *IEEE Robotics and Automation Letters*, 7(2):3435–3442, 2022.

[167] Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 29, 2016.

[168] Hongyi Zhang and Suvrit Sra. First-order methods for geodesically convex optimization. In *29th Annual Conference on Learning Theory*, volume 49, pages 1617–1638, 2016.

[169] Zhaoliang Zhang, Jincheng Yu, Jiahao Tang, Yuanfan Xu, and Yu Wang. MR-TopoMap: Multi-robot exploration based on topological map in communication restricted environment. *IEEE Robotics and Automation Letters*, 7(4):10794–10801, 2022.

[170] Zhengdong Zhang, Theia Henderson, Vivienne Sze, and Sertac Karaman. FSMI: Fast computation of Shannon mutual information for information-theoretic mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6912–6918, 2019.

[171] Lifeng Zhou and Vijay Kumar. Robust multi-robot active target tracking against sensing and communication attacks. *IEEE Transactions on Robotics*, 39(3):1768–1780, 2023.

[172] Lifeng Zhou, Vishnu D. Sharma, Qingbiao Li, Amanda Prorok, Alejandro Ribeiro, Pratap Tokekar, and Vijay Kumar. Graph neural networks for decentralized multi-robot target tracking. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 195–202, 2022.

[173] Hongbiao Zhu, Chao Cao, Yukun Xia, Sebastian Scherer, Ji Zhang, and Weidong Wang. DSVP: Dual-stage viewpoint planner for rapid exploration by dynamic expansion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7623–7630, 2021.

[174] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R. Oswald, and Marc Pollefeys. NICE-SLAM: Neural implicit scalable encoding for SLAM. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022.

[175] Ehsan Zobeidi, Alec Koppel, and Nikolay Atanasov. Dense incremental metric-semantic mapping via sparse gaussian process regression. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6180–6187, 2020.

[176] Ehsan Zobeidi, Alec Koppel, and Nikolay Atanasov. Dense incremental metric-semantic mapping for multiagent systems via sparse Gaussian process regression. *IEEE Transactions on Robotics*, 38(5):3133–3153, 2022.

[177] Elchanan Zwecher, Eran Iceland, Sean R. Levy, Shmuel Y. Hayoun, Oren Gal, and Ariel Barel. Integrating deep reinforcement and supervised learning to expedite indoor

mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 10542–10548, 2022.