Environment Model Adaptation for Autonomous Exploration

Erik Arthur Nelson
May 14, 2015
CMU-RI-TR-15-12

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania

Thesis Committee:
Nathan Michael, *chair*
Artur Dubrawski
Daniel Maturana

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

This thesis proposes adapting a mobile robot's environment model as a means of increasing the speed at which it is able to explore an unknown environment. Exploration is a useful capability for autonomous mobile robots that must operate outside of controlled factories and laboratories. Recent advances in exploration employ techniques that compute control actions by analyzing information-theoretic metrics on the robot's map. Information-theoretic metrics are generally computationally expensive to evaluate, ultimately limiting the speed at which a robot is able to explore.

To reduce the computational cost of exploration, this thesis develops an information-theoretic strategy for simplifying a robot's environment representation, in turn allowing information-based reward to be evaluated more efficiently. To remain effective for exploration, this strategy must adapt the environment model in a way that sacrifices a minimal amount of information about expected future sensor measurements. Adapting the robot's map representation in response to local environment complexity, and propagating the efficiency gains through to planning frequency and velocity gives rise to intelligent behaviors such as speeding up in open expanses. These methods are used to demonstrate information-theoretic exploration through mazes and cluttered indoor environments at speeds of 3 m/s in simulation, and 1.6 m/s on a ground robot.

Acknowledgements

I have had the great fortune of working alongside many incredibly talented people during my time in the Robotics Institute, all of whom contributed in some part to this thesis and to my future. I am thankful to those who have given me support and encouragement through what most consider to be a selfish endeavor.

I would first like to thank Prof. Nathan Michael, my advisor, for his immense help in shaping my research interests, providing critical writing and presentation feedback, and for the significant one-on-one time that he put into teaching me the art and science of robotics.

The members of the Robust Adaptive Systems Lab (RASL) have been an invaluable resource for solving difficult problems, and for providing hope and comfort during stressful times. I am thankful to Vishnu Desaraju, John Yao, Derek Mitchell, and Shihyun Lo for putting everything into perspective, and for guidance through tricky problems in planning, state estimation, and control theory.

My experiences at Carnegie Mellon University would not have been nearly as enjoyable or productive without the friends that I made along the way. Thank you Nick Gisolfi, Zach Batts, and Allie Del Giorno for balancing my life with music, philosophy, politics, and even more robotics.

Finally, I would like to thank the people in my life that guided me into doing what I now love. Thanks to Profs. Zoë Wood, Jane Lehr, and Chris Clark, who encouraged and persuaded me to attend graduate school in the first place. Thanks to Alex Viksne, my parents, and my sister, who supported me unconditionally.

We gratefully acknowledge the support of ARL grant W911NF-08-2-0004.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Robots are emerging from controlled factories and laboratories into our homes, workplaces, roads, and public airspaces. Alongside their transition into these unstructured and transient environments comes their need to be able to explore, characterize, and catalog their surroundings. Mobile robot autonomy is generally accomplished by referring to a map - a 2D or 3D probabilistic representation of the locations of obstacles in the robot's workspace. With access to a map, robots can localize to determine their position, plan collision-free trajectories to goals, locate objects for interaction, and make decisions by reasoning about the geometry and dynamics of the world. Given that a robot's map is of critical importance for most autonomy tasks, robots that find themselves initialized without a priori access to a map should be capable of autonomously, efficiently, and intelligently creating one.

The exercise of choosing and executing actions that lead a robot to learn more about its own map is known as *active perception* or *exploration*, and is the central topic of this thesis. Active perception has previously been studied with a multitude of sensor models, environment representations, and robot dynamics models. The active perception task itself can be dissected into two components [41]:

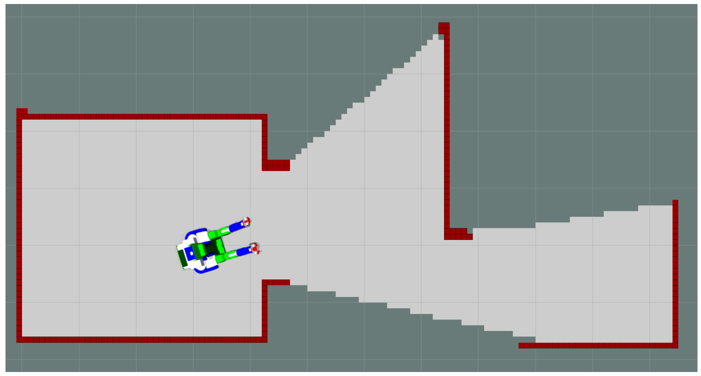


Figure 1.1: A household service robot awakens in an unknown environment. Prior to accomplishing its main functionalities, it will require a map of its surroundings. What sequence of actions should it take to minimize the time it spends exploring?

component 1: Identifying and ranking actions that the robot can take to spatially extend or reduce uncertainty in its map

component 2: Autonomously navigating according to the most informative action, while simultaneously localizing to the map and updating it with incoming sensor measurements

A motivating example is depicted in Fig. 1.1, where a household service robot is initialized in an unknown environment. Prior to accomplishing tasks that a human might ask it to perform, the robot must learn its surroundings and build a map of the house. Ideally this phase of initialization would be fast, as it is a prerequisite to the main functionality of the robot, and also might be required when furniture is moved or household objects are displaced. Where should the robot travel to observe the most of the environment in the shortest amount of time? Virtually any autonomous robot operating in an unknown environment will require a map-building initialization phase, welcoming strategies that enable high-speed and intelligent exploration.

This thesis explores one method for making **component 1** more efficient, thereby allowing an exploring robot to either consider more future actions in the same amount of time, or the same amount of actions in a shorter amount of time. By considering more actions, the robot has a higher likelihood of finding one that is informative. By considering actions in a shorter amount of time, the robot is able to allot more computational resources to other tasks, such as planning, mapping, and control, thereby allowing it to achieve **component 2** at higher speeds. Combined, these implications allow a robot to explore an unknown environment more efficiently than it would otherwise be able to.

Simplifying the robot's environment representation (through compression or approximation) is proposed as a means of reducing the computational cost of **component 1**. This thesis will examine the ties between the complexity of an environment representation and the expected informativeness of planned actions, and use this relationship to formulate methods for optimally compressing or approximating a robot's map based on the local environment and incoming sensor measurements. These considerations ultimately allow an exploring robot to navigate more quickly and select more informative actions for navigation.

1.1 Previous Work

Prior approaches to mobile robot active perception fall into two broad categories: *geometric* approaches that reason about the locations and presence of obstacles and free space in the robot's map [2, 9, 10, 43, 47, 49], and *information-theoretic* approaches that treat the map as a multivariate random variable and choose actions that will maximally reduce its uncertainty [4, 8, 11, 14, 21]. Both categories of approaches solve **component 1** of active perception, and assume that a planner and Simultaneous Localization and Mapping (SLAM) framework are available to accomplish **component 2**.

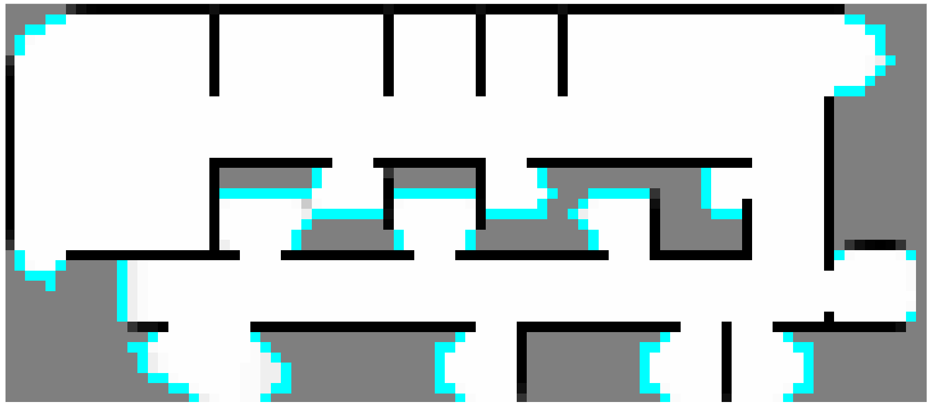


Figure 1.2: A partially explored map with frontiers between free and unknown space highlighted in blue.

1.1.1 Geometric Exploration Strategies

Many successful geometric exploration approaches build upon the seminal work of Yamauchi [49], guiding the robot to *frontiers* - regions on the boundary between free and unexplored space in the map (Fig. 1.2). Since multiple frontiers often exist simultaneously in a partially explored map, a variety of heuristics and spatial metrics can be used to decide which frontier to travel towards [28]. For example, an agent may decide to visit the frontier whose path through the configuration space from the agent's current position has minimum length, or requires minimal time or energy input to traverse. Similarly, an agent may decide to only plan paths to locations from which frontiers can be observed by its onboard sensors.

While effective in 2D environments, frontier exploration algorithms have several restrictive qualities. First, the naïve extension of frontier exploration from 2D to 3D maps poses a non-trivial challenge; as the dimensionality of the workspace increases, frontiers are distributed more densely throughout the environment due to occlusions, sensing resolution, and field-of-view, resulting in poor exploration performance [41]. Second, planning a path to a frontier does not imply that the path itself will be

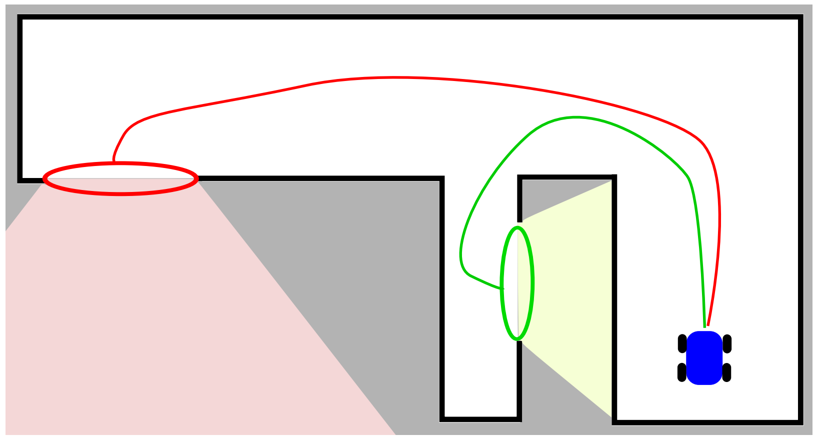


Figure 1.3: Traditional frontier exploration would visit the green location first because it is closest. A simple extension involves simulating sensor measurements from frontiers and examining their informativeness [17]. Applying this extension would cause robot to visit the red frontier first, since that location will provide more information about the map per unit time.

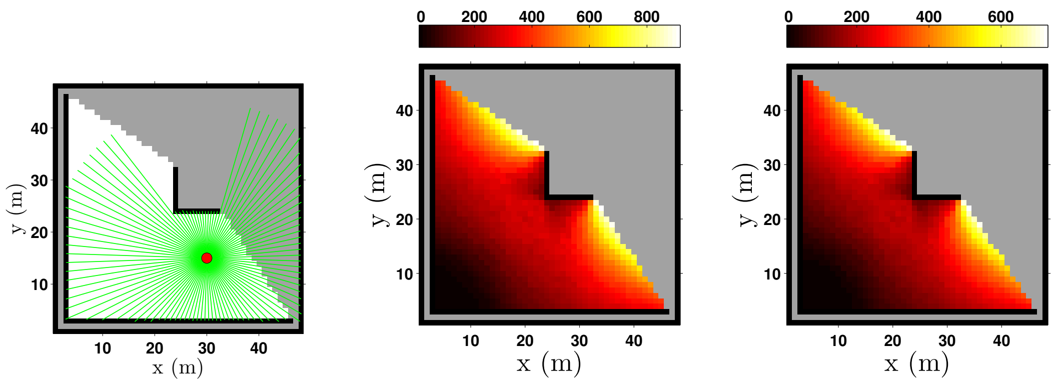
information-rich. Trajectory optimization techniques that consider information acquired by the robot's sensors along a planned path can be used as extensions to improve exploration performance [24, 42]. Finally, although the robot is guaranteed to learn new information upon reaching a frontier, the amount of information learned is dependent on the robot's sensor model, which is not considered when identifying frontiers. It may therefore be more efficient to visit a frontier that is suboptimal according to heuristics such as path length if the robot's sensors will provide more information from that location (Fig. 1.3). This limitation was first overcome by evaluating the informativeness of simulated sensor measurements taken from frontier locations [17], and was the original motivation for developing a category of information-theoretic exploration strategies.

More thorough surveys of frontier exploration algorithms and heuristics are provided by Basilico et al. [7] and Holz et al. [19].

1.1.2 Information-Theoretic Exploration Strategies

Information-theoretic exploration strategies cast the active perception task as an optimization, and choose actions for the robot that maximize an information-based objective function such as Shannon's entropy or mutual information [8, 11, 21, 23] (Fig. 1.4). Entropic measures like these are appealing because unlike geometric methods, they capture the relationship between sensor placement and uncertainty in the map. In addition, they can be computed without a maximum likelihood map estimate, and therefore do not discard probabilistic information known to the robot. Control policies that maximize mutual information have been proven to guide robots towards unexplored space [21], and weighting frontiers by the expected mutual information between the map and a sensor measurement acquired at the frontier location has been shown to result in more efficient exploration behaviors than traditional frontier exploration [11]. The same calculation (involving raycasting along beams from a sensor measurement) can be used to evaluate information-theoretic objective functions in both 2D and 3D environments.

The utility afforded by information-theoretic approaches comes at the cost of computational inefficiency. As a point of comparison, frontiers and other geometrically-defined landmarks need only to be computed once per map update, and can be computed (at worst, using a brute force search) with time complexity linear in the number of cells in the robot's map. One may alternatively choose to identify and cache frontiers every time a sensor measurement is used to update the map, yielding a constant time frontier identification step that is bounded by the number of map voxels within the maximum sensor range. By contrast, information-theoretic objective functions typically consider the probabilistic uncertainty associated with the sensor and environment models, and therefore require expensive sensor-related operations



(a) (b) (c)

Figure 1.4: Two variants of mutual information (Fig. 1.4b: Shannon; Fig. 1.4c: Cauchy-Schwarz Quadratic) densely computed in free space over an occupancy grid (Fig. 1.4a) using a 100-beam omnidirectional 2D laser with 30 m range. An exemplary sensor measurement is depicted in Fig. 1.4a. Controlling the robot towards locations that maximize either variant of mutual information would attract the robot to locations from which it could observe unknown areas of the map.

such as raycasts or sampling a large number of times from the distribution of possible future measurements. Approximations to mutual information between a map and beam-based sensor measurements can be evaluated with time complexity linear in the number of map voxels intersected by a sensor's beams [11, 22, 29]. This already-expensive operation must be performed for every future location that the robot might wish to travel to, or even multiple times along a single planned action. Julian et al. report that densely calculating mutual information over the free space in a large 1500 m^2 map requires approximately ten seconds with a parallelized implementation using a 3.4 Ghz quad-core CPU and NVIDIA GTX 690 graphics card [21]. While impressive, these results demonstrate that dense evaluation of mutual information over a robot's map is not feasible for high-speed applications that require subsecond planning frequencies.

1.2 Thesis Problem

This thesis focuses on the problem of enabling a mobile robot to *efficiently* explore its environment without human intervention. A robot that explores efficiently will learn the largest map possible for a fixed amount of time or energy input. The robot is assumed to be initialized without a priori access to a map, and is assumed to be capable of creating one in real-time from incoming sensor measurements with a Simultaneous Localization and Mapping (SLAM) subsystem [44]. The robot's task is to select control actions that maximize the likelihood that it will acquire informative sensor measurements in the future, given that it only knows a partial representation of the environment through which it is navigating. An *informative sensor measurement* is one that reduces uncertainty in the robot's map, allowing it to make more informed subsequent decisions. The exploration task is challenging because determining actions that will result in a minimum-time or minimum-energy traversal of the environment is impossible (or highly unlikely) without full a priori knowledge of the map. Instead, the environment must be modeled as a random quantity whose structure is only understood after it is observed by the robot's sensors.

State-of-the-art solutions to the exploration problem involve choosing control actions that drive the robot to states from which it will receive sensor measurements that maximally reduce uncertainty in its map (i.e. maximize an *information-theoretic reward function*). Although computationally inefficient, this category of solutions drives a robot to unexplored territories in a smaller amount of time than geometric exploration approaches [11]. However, as evidenced by the infeasibility of dense information-theoretic reward evaluation, there remains room for improvement. For example, a highly-efficient recent approach by Charrow et al. requires eleven minutes to explore a 17 m $\times$18 m $\times$3 m building with a quadrotor - enough time for the robot's batteries to deplete twice [11]. Any increase to the efficiency of information-theoretic exploration will allow the robot to consider more future actions before selecting and executing one. By considering more actions, a robot might discover a highly informative location in the environment that it previously would not have, increasing exploration performance.

This thesis addresses the computational costs of mobile robot exploration, summarized in the following statement:

> **Thesis Problem:** The computational cost of exploration with a mobile robot scales with the robot's speed, environment complexity, and dynamic model complexity. Exploring at high speeds or in complex regimes therefore requires either increased computational resources, or methods to increase the efficiency of identifying and navigating to informative locations.

1.3 Contributions and Outline

The thesis problem in Section 1.2 is addressed by developing methods to increase the efficiency of identifying and ranking the informativeness of future actions that the robot can take (**component 1** of the active perception task). Environment model simplification is proposed as a means of reducing the cost of evaluating information-theoretic reward along planned actions. Environment model simplification refers to the process of approximating or compressing the robot's map while retaining as much information as possible about obstacles and free space. Unfortunately, reducing the memory required to store the map necessarily causes structural information about the environment to be discarded (referred to as *distortion*). Simplification therefore decreases the accuracy of information-gain computed with respect to a sensor measurement. The tradeoff between information accuracy and redundancy is investigated to characterize situations in which the robot's map may be approximated while sacrificing a minimal amount of information about an expected future sensor measurement.

The structure of this thesis follows. Chapter 2 provides an overview of occupancy grid mapping, active perception, several planning strategies, and foundational elements from information theory. These ideas will be used to rigorously define the active perception problem and exact aspects of it which are the most computationally inefficient. Chapter 3 develops a strategy for simplifying (compressing) one specific environment model, an occupancy grid map, to a lower resolution using ideas from rate distortion theory. Since the compression strategy in Chapter 3 does not depend on the robot's sensor model, information-theoretic rewards computed on a compressed map will necessarily be distorted from those computed on an uncompressed map. Chapter 4 introduces an optimization to minimize environment model complexity for efficiency while simultaneously minimizing sensing distortion. Concluding remarks and avenues of future research are provided in Chapter 5.

Chapter 2

Active Perception Foundations

This thesis draws upon prior research from the robotics, information theory, and signal processing domains to develop its formulations. Sections 2.1 - 2.3 review relevant topics within robotics including occupancy grid mapping, active perception as an optimization, and several planning strategies that are suitable for the exploration task. These foundational topics will be used to develop a theory of optimal occupancy grid simplification as well as methods for guiding a robot to explore uncertain areas of its map efficiently. The formulations developed in Chapters 3 - 5 will also borrow heavily from information theory and rate distortion theory. These domains are frequently concerned with evaluating the effect one random variable (e.g. a sensor measurement) has on another (e.g. a map) or with simplifying a random variable to a reduced representation in such a way that the reduced form preserves the structure of the original form. Sections 2.4 and 2.5 review concepts from these domains that will be used when developing theories to determine an optimal complexity for the environment model.

2.1 Occupancy Grid Mapping

Occupancy grids (OGs) are a common and useful probabilistic map model for representing and reasoning about an unknown environment [13]. The remainder of this thesis assumes that the robot's environment is represented as an OG. Figures 1.1, 1.2, 1.3 and 1.4a depict occupancy grids, where black cells represent areas of the environment occupied by an obstacle, white cells represent areas that do not contain obstacles, and grey cells represent locations with unknown occupancy status.

OGs decompose the robot's workspace into a discrete set of 2D or 3D cells with a specified resolution. The presence or absence of obstacles within these cells is modeled as a K-tuple binary random variable, $\mathbf{m} = \{m_i\}_{i=1}^{K}$, with support set $\{\texttt{EMP}, \texttt{OCC}\}$. The probability that an individual cell is occupied is given by $p\left(m_i \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t}\right)$, where $\mathbf{x}_{1:t}$ denotes the robot's history of states, and $\mathbf{z}_{1:t}$ denotes the history of range observations accumulated by the robot. The OG representation treats cells as independent from one another, allowing one to express the joint occupancy probability of a specific map as the product of individual cell occupancy probabilities:

$$p\left(\mathbf{m} \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t}\right) = \prod_i p\left(m_i \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t}\right). \tag{2.1}$$

For notational simplicity, the map conditioned on random variables $\mathbf{x}_{1:t}$ and $\mathbf{z}_{1:t}$ will henceforth be written as $p\left(\mathbf{m}\right) \equiv p\left(\mathbf{m} \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t}\right)$, and the probability of occupancy for a grid cell m_i as $o_i \equiv p\left(m_i = \texttt{OCC} \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t}\right)$. Unobserved grid cells are assigned a uniform prior such that $\{o_i = 1 - o_i = 0.5\}_{i=1}^{K}$. This implies that the robot is initially unaware of its surroundings prior to accumulating sensor measurements. To prevent numerical precision issues, the occupancy status of a grid cell m_i is represented by the log-odds ratio

$$l_i \equiv \log \frac{o_i}{1 - o_i}. \tag{2.2}$$

The log-odds ratio maps from occupancy probabilities existing on $[0, 1]$ to $\mathbb{R}$, which is more suitable for floating-point arithmetic. In addition, the log-odds ratio makes updates to a cell occupancy probability additive rather than multiplicative. When a new measurement $\mathbf{z}_t$ is obtained, a cell's log-odds occupancy probability may be updated with

$$l_i \leftarrow l_i + L\left(m_i \left| \mathbf{z}_t\right.\right), \tag{2.3}$$

where the term $L\left(m_i \left| \mathbf{z}_t\right.\right)$ represents the robot's inverse sensor model [44].

2.2 Active Perception as an Optimization

Active perception is the idea that a machine should continually guide itself to states in which it is able to acquire better sensor measurements [5, 6]. Active perception draws inspiration from biological sensors that adapt in response to external stimuli. The human eye, for example, has muscles that constrict the pupil in response to bright light (adaptation), and others that distort the curvature of its lens to focus on nearby or far-away objects (accommodation). Adaptation and accommodation allow humans to see light varying nine orders of magnitude in brightness, and focus on objects an infinite distance away. Similarly, a man-made sensor such as a camera should not passively collect and report incoming photons, but should adapt its aperture, CMOS gains, and shutter speed based on the properties of the incoming light.

To extend this idea to mobile robotics, one must consider the robot system itself as a sensor that is able to move and actuate for the purpose of collecting better sensor measurements. From this perspective, the robot's task is to choose and execute *actions* that optimize the quality of its sensor measurements. An action can be defined as a sequence of configurations $\mathbf{x}_\tau \equiv (\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+T})$ that the robot will achieve over a future time interval $\tau \equiv (t+1, \ldots, t+T)$. From configurations $\mathbf{x}_\tau$ the robot will acquire future sensor measurements $\mathbf{z}_\tau \equiv (\mathbf{z}_{t+1}(\mathbf{x}_{t+1}), \ldots, \mathbf{z}_{t+T}(\mathbf{x}_{t+T}))$. This thesis is concerned primarily with ground robots constrained to $SE(2)$, and will therefore use $\mathbf{x}_i$ to refer to a pose in 2D space: $\mathbf{x}_i \equiv (x_i, y_i, \theta_i)^T$.

In the context of exploring an unknown environment, the active perception problem can be framed as an optimization over possible future actions that the robot can take:

$$\mathbf{x}_\tau^* = \underset{\mathbf{x}_\tau \in \mathcal{X}}{\operatorname{argmax}} \; \mathcal{J}\left(\mathbf{m}, \mathbf{z}_\tau(\mathbf{x}_\tau)\right), \tag{2.4}$$

where $\mathcal{J}\left(\mathbf{m}, \mathbf{z}_\tau(\mathbf{x}_\tau)\right)$ is a reward function expressing the new information learned by sequentially moving the robot to configurations $\mathbf{x}_\tau$, collecting sensor measurements $\mathbf{z}_\tau$, and updating the map $\mathbf{m}$. $\mathcal{X}$ is the set of all collision-free and dynamically feasible actions that the robot can take. In addition to evaluating the pure information content of $\mathbf{z}_\tau$, $\mathcal{J}$ commonly incorporates the time or energy expenditure required to carry out the action $\mathbf{x}_\tau$.

Unfortunately, the active perception optimization faces the *curse of dimensionality*; the size of $\mathcal{X}$ grows exponentially with the length of the time horizon τ. As τ increases in size, it quickly becomes infeasible to evaluate $\mathcal{J}$ over all possible actions in $\mathcal{X}$. This inefficiency is the primary reason that dense evaluation of a reward function over a map is infeasible for high planning frequencies. Instead, to remain computationally tractable, one can generate a fixed-size set of candidate actions that are likely to be informative prior to optimizing (2.4). Because the candidate action set is a subset of $\mathcal{X}$, it is possible, and in many situations likely, that the most informative possible action will not be considered.

2.3 Action Generation

The primary concern of action generation is to suggest a fixed-size set of feasible actions $\hat{\mathcal{X}}$ that are likely to be informative. A suitable choice of $\mathcal{J}$ can be evaluated on these actions to choose an optimal exploration action using (2.4). Several action generation options exist.

2.3.1 Frontier Seeding

Recent works by Charrow et al. [11] and Vallvé et al. [46] suggest seeding information-theoretic exploration by identifying frontiers and then evaluating a reward function from frontier locations. Because frontier identification is efficient, this two-pass approach is useful for locating potentially informative locations prior to performing the comparatively more expensive reward evaluation step. This strategy has the added benefit that frontiers are computed globally across the robot's map, guaranteeing that the robot will never become trapped in a dead-end or a location where its local map is already fully explored (i.e. a local minimum in information space).

Identifying frontiers before planning to them avoids planning trajectories to many future locations in a single planning cycle. Frontiers can be ranked by the information-theoretic reward offered from their locations, and the resulting sorted list of frontiers can be iterated through until a dynamically feasible and collision-free trajectory is found. Planning from an initial state to a goal state subject to dynamic and obstacle constraints becomes especially expensive in high-dimensional configuration spaces, and should be performed as few times as possible.

After selecting a location that will yield high reward, one may use a real-time pathfinding algorithm such as A* [18], RRT [27], or their many variants to generate a trajectory from the robot's initial state.

2.3.2 Forward-Arc Motion Primitives

Actions can also be generated by sampling from a set of pre-computed motion primitives. A simple strategy for generating motion primitives for a ground vehicle constrained to $SE(2)$ involves simulating the robot's path when moving at a constant linear and angular velocity for a specified amount of time. Actions resulting from this approach form arcs of a circle with a radius that is a function of the specified linear and angular velocity (Fig. 2.1).

Consider a robot following the arc of a circle with velocity v and rotational velocity ω. Assuming the robot's current position is given by $\mathbf{x}_t = (x_t, y_t, \theta_t)^T$, forward-arc motion primitives can be generated by specifying the future robot state, $\mathbf{x}_{t+T}$, as a function of v and ω for a sequence of uniformly varying times $T \in \mathbb{R}^+$. These paths are described by a set of nonlinear differential equations:

$$\dot{\mathbf{x}}_{t+T} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}_{t+T} = \begin{bmatrix} v\cos(\theta_{t+T}) \\ v\sin(\theta_{t+T}) \\ \omega \end{bmatrix}, \tag{2.5}$$

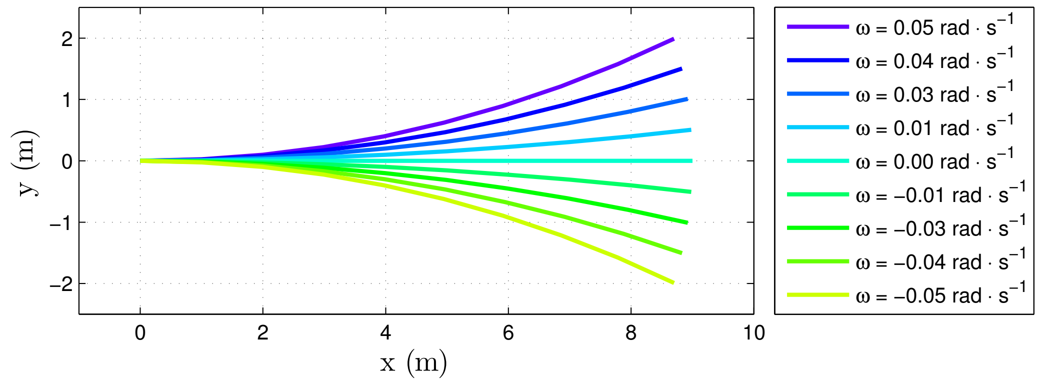


Figure 2.1: Nine motion primitives generated with $\omega = \{-0.05, -0.04, \ldots, 0.05\}$ rad/s, $v = 1.0$ m/s.

the solution of which is given by

$$\mathbf{x}_{t+T} = \begin{bmatrix} \frac{v}{\omega}\left(\sin\left(\omega T + \theta_t\right) - \sin\left(\theta_t\right)\right) \\ \frac{v}{\omega}\left(\cos\left(\theta_t\right) - \cos\left(\omega T + \theta_t\right)\right) \\ \omega T \end{bmatrix} + \mathbf{x}_t. \tag{2.6}$$

Sequentially incrementing T in (2.6) produces a sampling of poses lying along an arc parameterized by the robot's velocity and angular velocity, with origin $\mathbf{x}_t$.

A sampling of actions with varying v and w values (such as that depicted in Fig. 2.1) is referred to as a primitive *dictionary*. To generate more actions, one can construct a primitive *library*. This is accomplished by forming a tree with nodes corresponding to poses at the endpoints of actions. The tree is initialized by adding the robot's current position as the root node. Then, a dictionary of motion primitives is rotated and translated to leaf nodes in the tree until a specified depth is reached. A primitive library is shown in Fig. 2.2.

Forward-arc motion primitives are pre-computed prior to deployment into an un-

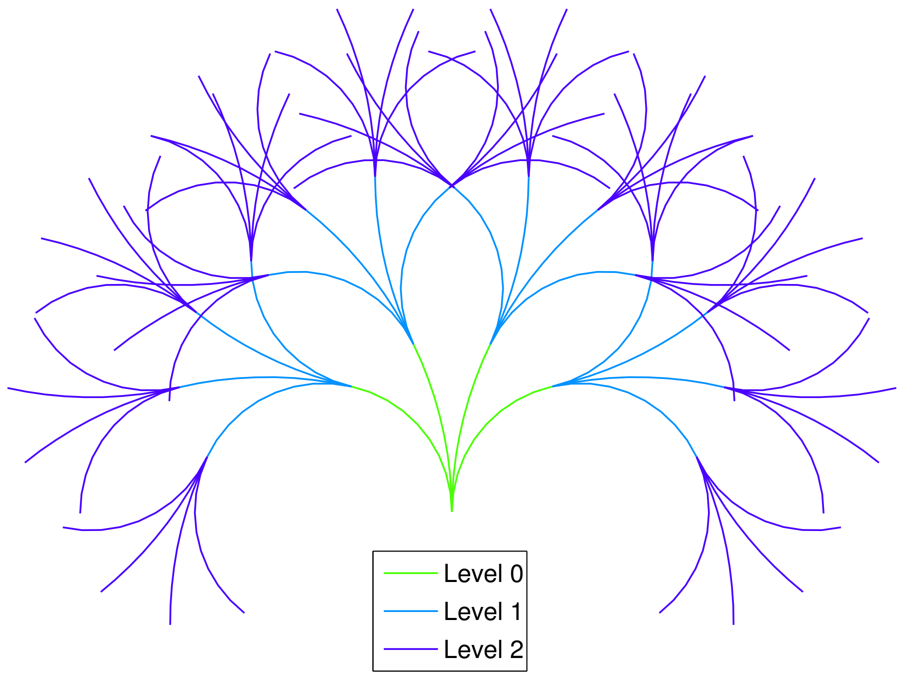


Figure 2.2: A primitive library with a depth of three constructed from a dictionary of four motion primitives.

known environment, making them an efficient choice for real-time exploration. Collision checking involves stepping along actions during a breadth-first or depth-first search and pruning all nodes (actions) that lie below those that contain a collision.

2.3.3 Lattice Graph Motion Primitives

A third method for generating actions is lattice graph planning. Lattice graph planners define a discrete set of goal states, and solve Boundary Value Problems (BVPs) to find trajectories from ($\emptyset$) to each goal [30–32] (Fig. 2.3). The resulting set of motion primitives can be rotated and translated to the robot's current position at run-time, and sampled from to produce candidate actions. Like forward-arc motion primitives, lattice graph motion primitives can be pre-computed and are therefore a suitable choice for real-time exploration. Collision checking for motion primitives in the lattice graph involves stepping along the action and checking for poses that lie outside of the robot's configuration space.

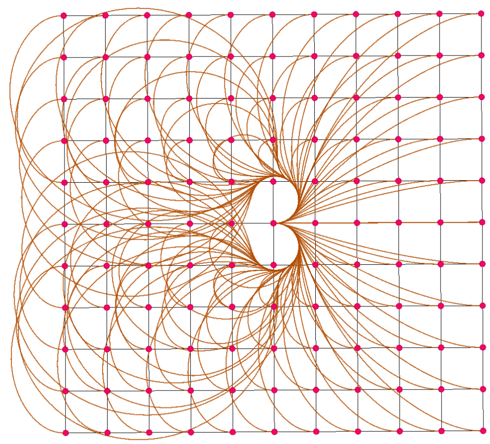


Figure 2.3: An $11 \times 11 \times 1$ lattice graph generated by solving many BVPs from the robot's initial pose (middle, facing right) to a lattice of final poses (with final angle equal to initial angle) subject to linear and angular velocity constraints.

2.4 Generalized Entropies and Divergences

Two fundamental building blocks of information theory that will be used to formulate environment model simplification in Chapters 3 and 4 are entropy and divergence. The former describes the amount of uncertainty in a random variable, or equivalently, the random variable's information content. The latter is a distance metric between probability distributions that describes the information lost when one distribution is used to describe another. The most well-known forms of entropy and divergence are the Shannon entropy [39], and Kullback-Leibler divergence [26]. For a random variable X, the Shannon entropy, H, and Kullback-Leibler divergence, $\mathrm{D_{KL}}$, are given by

$$\begin{aligned} \mathrm{H}\left(X\right) &= -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x), \\ \mathrm{D_{KL}}\left(p||q\right) &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)}, \end{aligned} \tag{2.7}$$

where $\mathcal{X}$ is the sample space of X, and p and q are discrete probability distributions over X. While Shannon entropy and Kullback-Leibler divergence succinctly describe critical concepts of information theory, alternative definitions of these concepts exist.

Shannon entropy is one solution to a more general parametric family of entropies introduced by Rényi [36] that take the form

$$\begin{aligned} \mathrm{H}_{\alpha}(X) &= \frac{1}{1-\alpha} \log_2 \sum_{x \in \mathcal{X}} p^{\alpha}(x) \quad \text{(discrete)} \\ \mathrm{H}_{\alpha}(X) &= \frac{1}{1-\alpha} \log_2 \int_{\mathcal{X}} p^{\alpha}(x) \quad \text{(continuous).} \end{aligned} \tag{2.8}$$

Rényi's so-called α-*entropy* approaches the Shannon entropy as $\alpha \to 1$, but allows one to express the information content of a random variable using any choice from a family of functions. H_{∞} entropy or H_2 entropy, for example, carry a similar meaning to Shannon entropy, but in some cases may be easier to evaluate. Rényi's α-entropy will be used for an optimization that is difficult to solve using Shannon's entropy in Chapter 3.

In a similar nature, there exists a spectrum of divergence measures that generalize and extend the properties of the Kullback-Leibler divergence. The Cauchy-Schwarz (CS) divergence is one measure that is of particular importance to this thesis. Cauchy-Schwarz divergence can be derived by substituting two distributions, p and q into the Cauchy-Schwarz inequality [37]:

$$\begin{aligned} \sqrt{\sum_{x \in \mathcal{X}} p^2(x) \sum_{x \in \mathcal{X}} q^2(x)} &\geq \sum_{x \in \mathcal{X}} p(x) q(x) \quad \text{(discrete)} \\ \sqrt{\int_{\mathcal{X}} p^2(x) dx \int_{\mathcal{X}} q^2(x) dx} &\geq \int_{\mathcal{X}} p(x) q(x) dx \quad \text{(continuous).} \end{aligned} \tag{2.9}$$

Cauchy-Schwarz divergence measures the extent of this inequality.

$$\mathrm{D_{CS}}\left(p||q\right) = \log \frac{\sum\limits_{x\in\mathcal{X}} p^2(x) \sum\limits_{x\in\mathcal{X}} q^2(x)}{\left(\sum\limits_{x\in\mathcal{X}} p(x)q(x)\right)^2} \quad \text{(discrete)}$$

$$\mathrm{D_{CS}}\left(p||q\right) = \log \frac{\int_{\mathcal{X}} p^2(x)dx \int_{\mathcal{X}} q^2(x)dx}{\left(\int_{\mathcal{X}} p(x)q(x)dx\right)^2} \quad \text{(continuous)}. \tag{2.10}$$

Cauchy-Schwarz divergence is a non-negative distance metric that takes on a value of zero when its arguments are the same distribution. Unlike Kullback-Leibler divergence, Cauchy-Schwarz divergence is symmetric in its arguments. It can equivalently be written in terms of Rényi's α-entropy for $\alpha = 2$.

$$\begin{aligned} \mathrm{D_{CS}}\left(p(x)||q(y)\right) &= -2\log \int_{\mathcal{X},\mathcal{Y}} p(x)q(y)dxdy + \log \int_{\mathcal{X}} p^2(x)dx + \log \int_{\mathcal{Y}} q^2(y)dy \\ &= 2\mathrm{H}_2\left(X;Y\right) - \mathrm{H}_2\left(X\right) - \mathrm{H}_2\left(Y\right), \end{aligned} \tag{2.11}$$

where $\mathrm{H}_2\left(X;Y\right)$ is the quadratic Rényi cross-entropy [35]:

$$\mathrm{H}_2\left(X;Y\right) = -\log_2 \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x)q(y) \quad \text{(discrete)}$$

$$\mathrm{H}_2\left(X;Y\right) = -\log_2 \int p(x)q(y)dxdy \quad \text{(continuous)}. \tag{2.12}$$

2.5 Cauchy-Schwarz Quadratic Mutual Information

The Cauchy-Schwarz divergence metric described in Section 2.4 can be used to define a second distance metric that measures the amount of dependence between two random variables X and Y. The amount of dependence between two random variables is synonymous with the definition of mutual information, another fundamental building block of information theory. Mutual information metrics describe the difference between a joint distribution, $p(x, y)$, and the product of its marginals, $p(x)p(y)$. Like entropy and divergence, there exists a common definition for mutual information (the *Shannon mutual information* (SMI)) that can be extended and generalized. In the context of mobile robotic exploration, a more convenient definition of mutual information is the *Cauchy-Schwarz Quadratic mutual information* (CSQMI), which is derived by substituting $p(x, y)$ for p and $p(x)p(y)$ for q in (2.10).

$$\mathrm{I}_{\mathrm{CS}}(X;Y) = \log \frac{\int_{\mathcal{X}} \int_{\mathcal{Y}} p^2(x,y)dxdy \int_{\mathcal{X}} \int_{\mathcal{Y}} p^2(x)p^2(y)dxdy}{\left(\int_{\mathcal{X}} \int_{\mathcal{Y}} p(x,y)p(x)p(y)dxdy\right)^2}. \tag{2.13}$$

Charrow et al. originally showed that the CSQMI between a robot's map and a beam-based sensor measurement is a superior reward metric to SMI for exploration [11]. This is because CSQMI can be computed analytically without requiring an expensive sampling step to calculate the expected value of a future sensor measurement. Additionally, CSQMI can be computed exactly in $\mathcal{O}(n^2)$, and to a close approximation in $\mathcal{O}(n)$, where n is the number of cells in the robot's map intersected by a sequence of beam-based sensor measurements $\mathbf{z}_\tau$. While SMI can also be approximated in time linear in n, Charrow et al. show that CSQMI has a smaller linear constant factor, allowing CSQMI to be computed in roughly one seventh of the amount of time.

Like SMI, CSQMI is non-negative and zero only when its arguments are independent (i.e. when $p(x, y) = p(x)p(y)$). Figure 1.4 shows that CSQMI and SMI are similar when evaluated on an OG with a beam-based sensor model, and control actions that maximize CSQMI guide the robot to unexplored space. The reader should refer to Charrow et al. [11] for discussion regarding explicit calculation of CSQMI between an OG map and a beam-based sensor measurement.

Given the numerous benefits of CSQMI, $\mathrm{I}_{\mathrm{CS}}(\mathbf{m}; \mathbf{z}_\tau(\mathbf{x}_\tau))$ is a suitable choice for $\mathcal{J}(\mathbf{m}; \mathbf{z}_\tau(\mathbf{x}_\tau))$ in (2.4). The CSQMI between an OG map and sequence of beam-based measurements is

$$\mathrm{I}_{\mathrm{CS}}(\mathbf{m}; \mathbf{z}_\tau) = \log \frac{\int \sum_{\mathcal{M}} p^2(\mathbf{m}, \mathbf{z}_\tau) d\mathbf{z}_\tau \int \sum_{M} p^2(\mathbf{m}) p^2(\mathbf{z}_\tau) d\mathbf{z}_\tau}{\left(\int \sum_{\mathcal{M}} p(\mathbf{m}) p(\mathbf{z}_\tau) p(\mathbf{m}, \mathbf{z}_\tau) d\mathbf{z}_\tau\right)^2}. \tag{2.14}$$

The map $\mathbf{m}$ has a discrete sample space $\mathcal{M}$ because cells may only take on a value of `OCC` or `EMP`. A map of K cells can therefore take on one of $|\mathcal{M}| = 2^K$ permutations. Substituting I_{CS} for $\mathcal{J}$ in (2.4) yields an active perception optimization that guides a robot towards unexplored locations in its map by maximizing an information information-theoretic reward functional.

$$\mathbf{x}_\tau^* = \underset{\mathbf{x}_\tau \in \hat{\mathcal{X}}}{\operatorname{argmax}} \ \mathrm{I}_{\mathrm{CS}}(\mathbf{m}, \mathbf{z}_\tau(\mathbf{x}_\tau)). \tag{2.15}$$

2.6 Summary of Foundations

Chapter 2 reviewed foundational elements that will be used for derivations in the remaining chapters. These elements culminate in an optimization that drives a robot towards unexplored space in its map by maximizing an information-theoretic reward function (2.15). More generally, and perhaps more importantly, this optimization can be used to select an action that will optimally reduce uncertainty in a distribution (e.g. a probability distribution over obstacles in an environment). The chosen reward function, CSQMI [34], resembles Shannon's original definition of mutual information, but can be computed more efficiently when its arguments are an OG map and a sequence of beam-based sensor measurements [11].

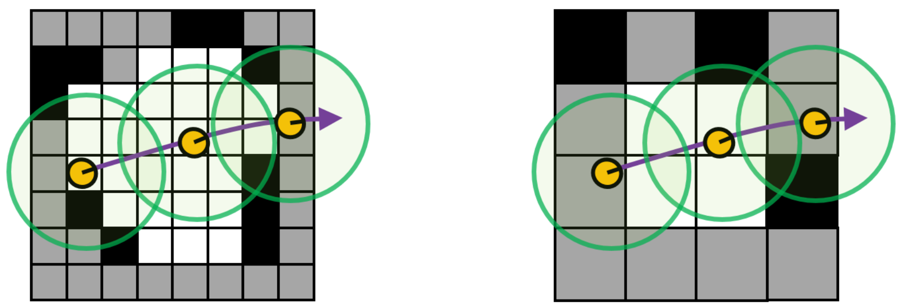
Furthermore, Chapter 2 discussed three methods for generating dynamically feasible and collision-free actions through the robot's configuration space. The action maximizing CSQMI (either integrated over the path, or at the action's final pose) will be chosen as that which optimally drives the robot towards unexplored space. The remainder of this thesis will use both forward-arc motion primitives and lattice graph motion primitives for action generation, although the third option - frontier seeding - is equally viable.

Chapter 3

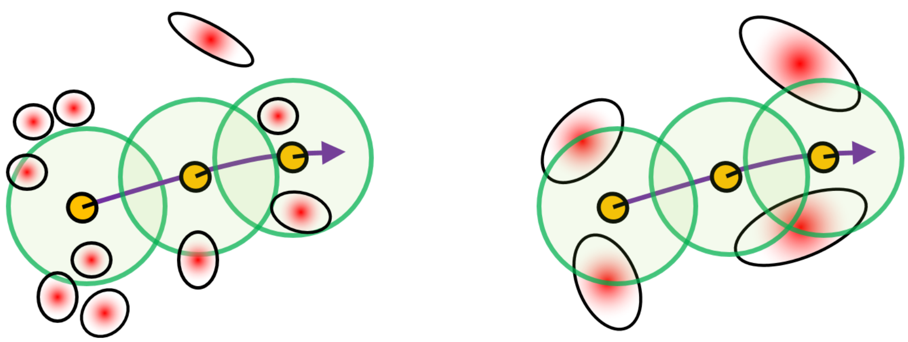
Information-Theoretic Environment Model Simplification

The remaining chapters of this thesis introduce novel extensions to active perception that make information-theoretic reward evaluation more efficient at the cost of information-theoretic reward accuracy. These extensions originate from the observation that the robot's actions, sensor model, and environment model are intricately coupled in the exploration task, and that simplifying the robot's environment model reduces the computational cost of evaluating mutual information reward (Fig. 3.1).

For example, in the case of OG maps, the cell resolution parameter shares an intricate relationship with the robot's exploration behaviors. One relationship between OG cell resolution and exploration behavior lies in the efficiency of raycasting. Most information-theoretic reward functions (e.g. SMI and CSQMI) require simulating a beam-based sensor measurement from a future position, which implicitly requires raycasting through the map. Intuitively, as the resolution of cells in the map decreases, so too does the number of cells that a raycast must traverse. Therefore the efficiency



(a) Occupancy grid environment model



(b) Mixture of Gaussian landmarks environment model

Figure 3.1: A robot plans through its map (left) and a simplified representation of the map (right), expecting to receive three future measurements. Mutual information reward evaluation is more efficient, but distorted, when computed on the simplified representation.

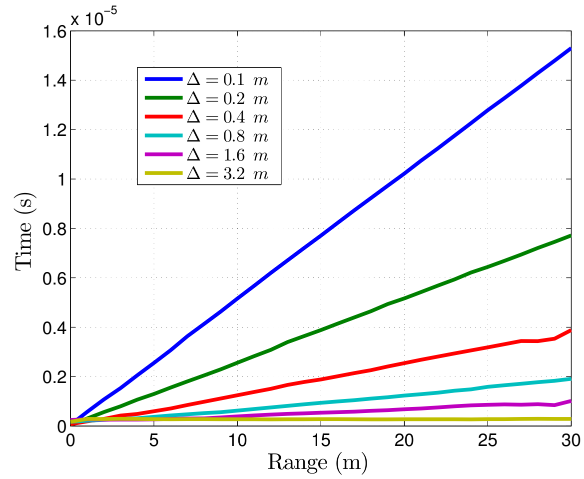


Figure 3.2: Time (median of 10^5 samples) to evaluate CSQMI for a single beam is empirically linear in both the OG cell edge length Δ, and the measurement range.

of information-theoretic reward evaluation is a function of the cell resolution of the OG. For example, using the approximate CSQMI technique from Charrow et al. [11], this relationship is linear (Fig. 3.2). As will be shown, the resolution of an OG map can typically be halved several times before a significant amount of information about free and occupied space is lost. Repeatedly halving the map's resolution exponentially increases the speed of computing CSQMI for a fixed sensor range or number of beams. This can be seen by holding range constant and tracing vertical lines down Fig. 3.2.

The following chapter investigates environment model simplification as a means of increasing the efficiency of evaluating CSQMI reward. The incentive for increasing the efficiency of information-theoretic exploration is clear and immediate; any reduction to the time required to evaluate an information-theoretic reward function will enable a robot to explore more efficiently, in turn allowing it to initialize faster (Fig. 1.1), clear a building for threats more quickly, or explore a larger amount of space on an energy budget.

Several others have proposed strategies for approximating or compressing an environment belief distribution. The OctoMap framework [48] builds an octree data structure to efficiently store the expected occupancy of cells in an environment without allocating memory for a large 3D grid. Jeong et al. [20] compress an OG by representing it with wavelets using the Haar wavelet transform. Kretzschmar et al. [25] compress pose graph maps by examining the SMI between the pose graph and sensor measurements. Most related to the formulations in this chapter is the work by Einhorn et al. [12], which adaptively chooses an OG resolution for individual cells by determining which cells are intersected by measurements.

Instead of using a previous approach, this chapter pursues a novel strategy for environment model simplification using the Principle of Relevant Information [34]. In contrast to previous works on map compression, the proposed strategy approaches the problem from an information theory perspective, optimizing a functional that describes the distortion between the map and its compressed form. When applied to the OG environment model, the proposed Principle of Relevant Information strategy yields a simple compression algorithm. The compression strategy developed can be used to complement existing multi-resolution mapping frameworks, such as octomap [48] or NDT maps [38].

3.1 The Principle of Relevant Information

The environment model simplification problem can be formulated as an information-theoretic optimization using the Principle of Relevant Information (PRI). PRI is a technique for learning a reduced representation $\hat{X}$ of a random variable X such that both the entropy of $\hat{X}$ and the divergence of $\hat{X}$ with respect to the original data are minimized [34].

$$\Lambda(\hat{X}) = \min_{\hat{X}}(\mathrm{H}_\alpha(\hat{X}) + \lambda \mathrm{D}_\alpha(X||\hat{X})). \tag{3.1}$$

The two terms of the PRI cost function are Rényi's α-entropy, which describes the amount of uncertainty in its argument (Section 2.4), and Rényi's α-divergence, which is a generalized divergence measure that describes the distortion between $p(x)$ and $p(\hat{x})$. These terms simplify to Shannon entropy and Kullback-Leibler divergence for $\alpha = 1$.

Intuitively, the PRI trades off information redundancy in $\hat{X}$ for errors induced by using the compressed form $\hat{X}$ to represent the uncompressed form X. The variational parameter $\lambda \in [0, \infty)$ balances this trade-off. Choosing $\lambda = 0$ forces the optimization to select $\hat{X}$ such that $\mathrm{H}_2(\hat{X}) = 0$. Total entropy minimization is only possible for values of $\hat{X}$ that are completely determined. In the case of an OG map, for example, entropy minimization would result in a fully determined map with no ambiguity as to whether any cell was `OCC` or `EMP`. By contrast, choosing $\lambda \to \infty$ reduces to minimizing the divergence between $p(x)$ and $p(\hat{x})$. Performing divergence minimization gives back the original data; $\hat{X} \to X$ when $\lambda \to \infty$.

Choosing Rényi's 2-entropy and Cauchy-Schwarz divergence allows one to directly relate and manipulate the two terms of the PRI cost functional through use of the equivalence in (2.11).

$$\begin{aligned} \mathrm{H}_2(\hat{X}) + \lambda \mathrm{D}_{\mathrm{CS}}(X||\hat{X}) &= \mathrm{H}_2(\hat{X}) + \lambda \left(2\mathrm{H}_2(X;\hat{X}) - \mathrm{H}_2(X) - \mathrm{H}_2(\hat{X})\right) \\ &= (1-\lambda)\mathrm{H}_2(\hat{X}) + 2\lambda \mathrm{H}_2(X;\hat{X}) - \lambda \mathrm{H}_2(X). \end{aligned} \tag{3.2}$$

Expanding the quadratic Rényi's cross-entropy term using (2.12) gives the cost function

$$(1-\lambda)\,\mathrm{H}_2(\hat{X}) - 2\lambda \log \sum_{x\in\mathcal{X}} p(x)p(\hat{x}) - \lambda \mathrm{H}_2(X). \tag{3.3}$$

The PRI optimization is a minimization over $\hat{X}$, so the third term in (3.3) has no influence and can be ignored. To simplify terms further, entropy and divergence can be given equal weight (optimizing for $\lambda = 1$).

$$\Lambda(\hat{X}) = \min_{\hat{X}} -2\log \sum_{x\in\mathcal{X}} p(x)p(\hat{x}). \tag{3.4}$$

Noting that logs and quadratic functions increase monotonically for positive arguments, and noting that the summand in (3.4) must be positive for probabilities $\in [0, 1]$, the PRI compression optimization can be simplified to:

$$\Lambda(\hat{X}) = \max_{\hat{X}} \sum_{x\in\mathcal{X}} p(x)p(\hat{x}). \tag{3.5}$$

3.2 Framing Map Compression as an Optimization

The PRI optimization is well-suited for simplifying many kinds of environment models. The following section provides one example: using the PRI to compress an OG map.

Ideally, a low-resolution compressed OG would represent its high-resolution uncompressed originator well. The divergence minimization term in the PRI accomplishes this. The entropy term in the PRI is of similar value. For example, briefly suppose that a 2×2 cell region must be compressed to a 1×1 cell region. If the 2×2 region contained two cells with a high probability of occupancy and two with a low probability of occupancy, minimizing Cauchy-Schwarz divergence would result in a 1×1 region that is a mixture of the two probability values. While this reduction in certainty through compression is instinctive in most applications, OGs are generally used for operations such as raycasting and collision-checking. In these operations it would be harmful to lose information about free and occupied regions of the environment that were known prior to compression. Minimizing entropy during compression alleviates this concern, as it forces cell occupancy probabilities in the compressed map towards `OCC` and `EMP`.

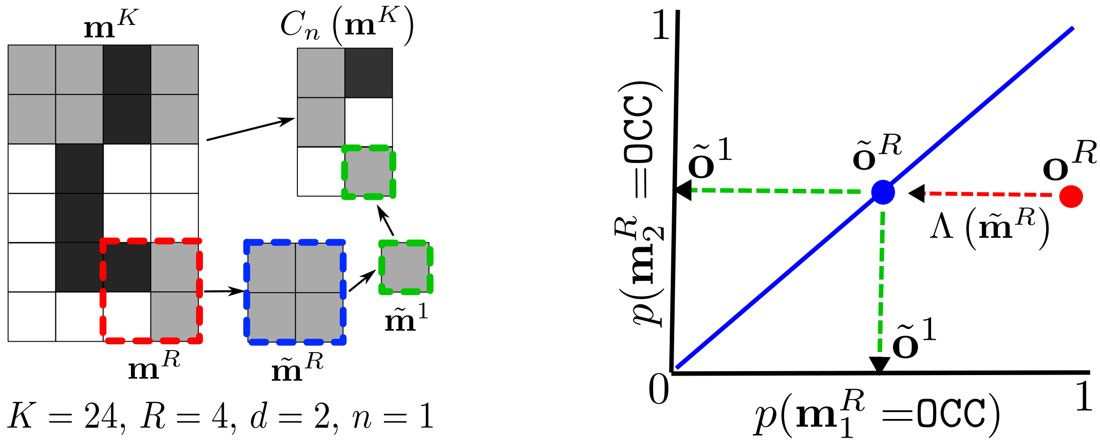
To apply the PRI method to OG compression, let X be an OG, $\mathbf{m}^K$, with K cells (for the remaining notations, superscripts will denote map cell counts). The compression problem requires three constraints.

constraint 1: Because OGs encode a 2D or 3D geometry, $\hat{X}$ must represent X well in local regions. Compression over the map can therefore be accomplished by performing compression in many small independent square (cubic in 3D) regions $\mathbf{m}^R \subseteq \mathbf{m}^K$ by exercising the OG assumption in (2.1) that individual cell occupancy probabilities are independent.

constraint 2: Only the set of compressions that reduce OG cell count in each dimension by factors of two will be considered. Therefore an OG $\mathbf{m}^K$ will be compressed to an OG $\mathbf{m}^{2^{-dn}K}$, where d is the OG dimension and n is the number of $2\times$ compressions in each dimension. The set of compressions with this property can be expressed as:

$$C_n(\mathbf{m}^K) \equiv \mathbf{m}^{2^{-dn}K}, \quad n \in \mathbb{N}_0, \tag{3.6}$$

where a compression with $n = 0$ gives the original OG: $C_0(\mathbf{m}^K) =$



(a) Example OG compression sequence.

(b) Probability space of the top two grid cells in $\mathbf{m}^R$ in 3.3a.

Figure 3.3: For each square (cubic in 3D) region $\mathbf{m}^R$ in the uncompressed OG $\mathbf{m}^K$, the PRI optimization finds a random variable $\tilde{\mathbf{m}}^R$ that minimizes (3.1) and is constrained to have uniform occupancy probability $\tilde{\mathbf{o}}^R = (\tilde{\mathbf{o}}^1, \ldots, \tilde{\mathbf{o}}^1)$.

$\mathbf{m}^K$. Both $\mathbf{m}^K$ and $C_n(\mathbf{m}^K)$ will have the same metric dimensions, but will have different cell edge lengths and cell counts when $n \geq 1$.

constraint 3: If X is an OG, $\hat{X}$ must also be an OG.

Under these constraints, the map can be compressed by decomposing it into square (or cubic) independent regions, and compressing each region. For each region $\mathbf{m}^R$, PRI can be used to find a multivariate random variable $\tilde{\mathbf{m}}^R$ that has uniform occupancy probabilities and minimizes both entropy and divergence with respect to $\mathbf{m}^R$. The occupancy probability of each $\tilde{\mathbf{m}}^R$, $\tilde{\mathbf{o}}^R \equiv p(\tilde{\mathbf{m}}^R = \{\texttt{OCC}, \ldots, \texttt{OCC}\})$, can then be reduced to a scalar, yielding the occupancy probability of a single grid cell in the compressed map, $\tilde{\mathbf{o}}^1 \equiv p(\tilde{\mathbf{m}}^1 = \texttt{OCC})$ (Fig. 3.3). The occupancy distribution of a cell in the compressed map is completely determined by knowing $\tilde{\mathbf{o}}^1$, because $p(\tilde{\mathbf{m}}^1 = \texttt{OCC}) = 1 - p(\tilde{\mathbf{m}}^1 = \texttt{EMP})$, so the set of $\tilde{\mathbf{o}}^1$ values from independent regions are all that is necessary to determine the compressed OG $C_n(\mathbf{m}^K)$.

Using these notations alongside the PRI optimization in (3.5), a compressed OG region $\tilde{\mathbf{m}}^R$ that minimizes entropy and divergence from its uncompressed counterpart $\mathbf{m}^R$ is that which maximizes $\Lambda(\tilde{\mathbf{m}}^R)$ in

$$\Lambda(\tilde{\mathbf{m}}^R) = \max_{\tilde{\mathbf{m}}^R} \sum_{\mathcal{M}^R} p(\mathbf{m}^R) p(\tilde{\mathbf{m}}^R), \tag{3.7}$$

Rather than finding the maximal value of the cost function itself, the occupancy probability for the compressed OG region can be found by determining $\tilde{\mathbf{o}}^1$:

$$\tilde{\mathbf{o}}^1_* = \operatorname*{argmax}_{\tilde{\mathbf{o}}^1} \sum_{\mathcal{M}^R} p(\mathbf{m}^R) p(\tilde{\mathbf{m}}^R), \tag{3.8}$$

In order to fully constrain the problem, it is important to note that $p(\tilde{\mathbf{m}}^R)$ is a Bernoulli distribution. All of the cells in $\tilde{\mathbf{m}}^R$ have a uniform probability of occupancy and therefore cannot differ from one another, as they must ultimately be compressed to a scalar that represents the occupancy probability of a single cell. In other words, $\tilde{\mathbf{o}}^1$ is the probability of occupancy for every cell in $\tilde{\mathbf{m}}^R$.

3.3 Solving the Optimization

Solving the optimization in (3.8) involves iterating through all permutations of maps that have cell count R and multiplying the probability that $\mathbf{m}^R$ takes on a specific permutation with the probability that $\tilde{\mathbf{m}}^R$ takes on that permutation. Fortunately, although $\mathcal{M}^R$ is large ($|\mathcal{M}^R| = 2^R$), $p(\tilde{\mathbf{m}}^R)$ is zero for all but two permutations due to the fact that $\tilde{\mathbf{m}}^R$ must have uniform occupancy. Specifically, the following is true:

- $p(\tilde{\mathbf{m}}^R \neq \{\texttt{EMP}, \ldots, \texttt{EMP}\} \wedge \tilde{\mathbf{m}}^R \neq \{\texttt{OCC}, \ldots, \texttt{OCC}\}) = 0$
- $p(\tilde{\mathbf{m}}^R = \{\texttt{EMP}, \ldots, \texttt{EMP}\}) = 1 - \tilde{\mathbf{o}}^R$

- $p(\tilde{\mathbf{m}}^R = \{\texttt{OCC}, \ldots, \texttt{OCC}\}) = \tilde{\mathbf{o}}^R$
- $p(\mathbf{m}^R = \{\texttt{EMP}, \ldots, \texttt{EMP}\}) = \prod_{i=1}^R (1 - \mathbf{o}_i^R)$
- $p(\mathbf{m}^R = \{\texttt{OCC}, \ldots, \texttt{OCC}\}) = \prod_{i=1}^R \mathbf{o}_i^R$

All map permutations in $\mathcal{M}^R$ for which $p(\tilde{\mathbf{m}}^R)$ is equal to zero do not contribute to the sum in (3.8). The two remaining non-zero terms in the summand of (3.8) can be enumerated explicitly:

$$\begin{aligned}\sum_{\mathcal{M}^R} p(\mathbf{m}^R)p(\tilde{\mathbf{m}}^R) = p(\mathbf{m}^R = \{\texttt{EMP}, \ldots, \texttt{EMP}\})p(\tilde{\mathbf{m}}^R = \{\texttt{EMP}, \ldots, \texttt{EMP}\}) \\ + p(\mathbf{m}^R = \{\texttt{OCC}, \ldots, \texttt{OCC}\})p(\tilde{\mathbf{m}}^R = \{\texttt{OCC}, \ldots, \texttt{OCC}\}) \\ = (1 - \tilde{\mathbf{o}}^1) \prod_{i=1}^R (1 - \mathbf{o}_i^R) + \tilde{\mathbf{o}}^1 \prod_{i=1}^R \mathbf{o}_i^R\end{aligned} \tag{3.9}$$

Substituting the expanded sum from (3.9) into the optimization from (3.8) gives

$$\tilde{\mathbf{o}}_*^1 = \underset{\tilde{\mathbf{o}}^1}{\operatorname{argmax}} \left((1 - \tilde{\mathbf{o}}^1) \prod_{i=1}^R (1 - \mathbf{o}_i^R) + \tilde{\mathbf{o}}^1 \prod_{i=1}^R \mathbf{o}_i^R \right), \tag{3.10}$$

which is satisfied for

$$\tilde{\mathbf{o}}_*^1 = \begin{cases} 0 & \text{if } \prod_{i=1}^R \frac{\mathbf{o}_i^R}{1 - \mathbf{o}_i^R} < 1 \\ 1 & \text{if } \prod_{i=1}^R \frac{\mathbf{o}_i^R}{1 - \mathbf{o}_i^R} > 1 \\ \frac{1}{2} & \text{otherwise} \end{cases}, \tag{3.11}$$

where the last case is undefined for $\lambda = 1$, but converges to $\frac{1}{2}$ in the limit as $\lambda \to 1^+$.

The PRI OG compression solution yields a simple compression rule: if the product of cell occupancy likelihoods in a given region is greater than 1, set the occupancy probability of the cell corresponding to that region in the compressed OG to 1. Likewise set the occupancy probability to 0 if the product of likelihoods is less than 1, and to 0.5 if the product of likelihoods is 1.

Pragmatically, it is more reasonable to use the map's occupancy and free thresholds rather than 1.0 and 0.0. This variation corresponds to optimizing for λ slightly greater than one, favoring minimal distortion to minimal entropy. Additionally, one may introduce a heuristic to increase the fraction of occupied cells that are preserved through compression by multiplying the right-hand sides of the inequalities in (3.11) by $\eta \in (0, 1)$. As η decreases, occupied cells will be preserved through compression with higher frequency. For any application involving raycasting, it is especially important to include this heuristic, as vanishing occupied cells lead to poor ray termination.

Denoting $\pi^R \equiv \prod_{i=1}^{R} \frac{\mathbf{o}_i^R}{1-\mathbf{o}_i^R}$ and applying these modifications gives a rule for determining the occupancy probability of a cell in the compressed map, given a $\sqrt{R} \times \sqrt{R}$ (in 2D) or $\sqrt[3]{R} \times \sqrt[3]{R} \times \sqrt[3]{R}$ (in 3D) region $\mathbf{m}^R$ of the uncompressed map:

$$\tilde{\mathbf{o}}_*^1 = \begin{cases} \frac{1}{2} & \text{if } \pi^R = \eta \vee \pi^R = 1 \\ p_{\text{free}} & \text{if } \pi^R < \eta \wedge \pi^R \neq 1 \\ p_{\text{occ}} & \text{if } \pi^R \geq \eta \wedge \pi^R \neq 1 \end{cases}, \tag{3.12}$$

where p_{occ} and p_{free} are the thresholds for occupied and free space in the OG implementation, respectively.

3.4 Occupancy Grid Pyramids

In some situations it is useful to store multiple versions of the robot's map that are compressed to different final resolutions. For example, Chapter 4 will compare maps

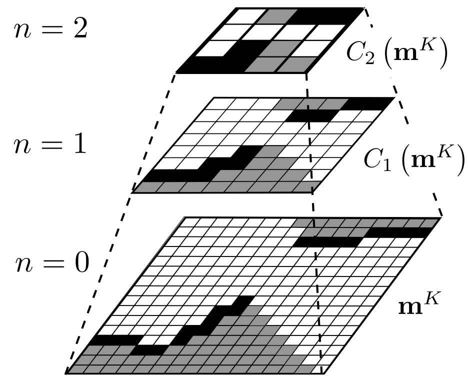


Figure 3.4: A three-level OG pyramid

of different resolutions to investigate the amount of information lost about a sensor measurement through compression. An *occupancy grid pyramid* is a multi-resolution map data structure that holds maps of several resolutions. This name is in reference to the image pyramid, a multi-resolution image data structure used commonly in computer vision [3].

To create an OG pyramid, one may simply apply increasing compressions to a base OG. Each compressed map in the pyramid has 2^{-d} times as many cells as the previous.

$$\mathcal{C}_n(\mathbf{m}^K) \equiv \{\mathbf{m}^K, C_1(\mathbf{m}^K), \ldots, C_n(\mathbf{m}^K)\}, \quad n \in \mathbb{N}_0 \tag{3.13}$$

All compressed maps are generated by applying C_n (defined in (3.6)) to the base map, rather than by applying C_1 to the previous map in the set. Depending on how one performs the compression, these may or may not result in different pyramids. A three-level OG pyramid is depicted in Fig. 3.4.

3.5 Results

Figure 3.5 shows a six-level OG pyramid built by applying the PRI optimization to a partially explored 2D map of a cluttered warehouse environment, with $\eta = 0.2$. For values of $\eta < 1$ the PRI compression strategy preserves and expands occupied cells in the map, ensuring that raycasts that terminate on the uncompressed map will also terminate on the compressed map. For $n = \{0, \ldots, 4\}$, intersections between free and unknown space in the map is preserved. Several regions that are occupied by obstacles in the uncompressed map are not represented well in the map compressed with $n = 5$.

Map compression was motivated at the start of this chapter by the observation in Fig. 3.2 that compressing the map results in increased efficiency of evaluating CSQMI. It is therefore important to ensure that a sensor measurement that is informative to the uncompressed map is still informative to a compressed map, and that map compression does not result in major perturbations to the information value returned by the CSQMI computation. Figure 3.6 shows a library of forward-arc motion primitives (Section 2.3.2) planned from a simulated robot's current position into a partially explored map. Reward is computed at the action endpoints using CSQMI between an expected future sensor measurement and the compressed maps. High CSQMI reward actions have green endpoints, while low CSQMI reward actions have red endpoints. The best action according to the active perception optimization in (2.15) is shown in blue. The relative rewards offered by actions retain their ordering until the map is compressed significantly ($n = 5$), at which point a different action is chosen. Although different, the optimal action chosen for the map with highest compression is still a high-reward path in the uncompressed map. Computing CSQMI reward on the map with highest compression is 32 times more efficient than computing CSQMI reward using the uncompressed map.

Simulated exploration trials on a maze-like 25×25 m map were conducted to examine the effects of OG compression on a ground robot's exploration path and planning frequency. To conduct these trials, a 2D dynamically constrained ground robot equipped with a laser scanner and IMU was simulated. The robot was assumed to be able to estimate its own state from incoming sensor measurements and build an accurate OG of its surroundings in real-time. To perform state estimation and mapping, the robot was outfitted with a laser- and inertial-based SLAM implementation similar to that of Shen et al. [40], leveraging ICP for laser odometry [33], a histogram filter for localization, and an unscented Kalman filter (UKF) to fuse estimates [44]. A custom dense OG implementation was used for mapping. The OG was updated at a rate of 10 Hz, and had a 0.1 m/cell uncompressed resolution. The robot's simulated laser scanner swept in a 270° arc with 1081 beams, and had a max range of 30 m.

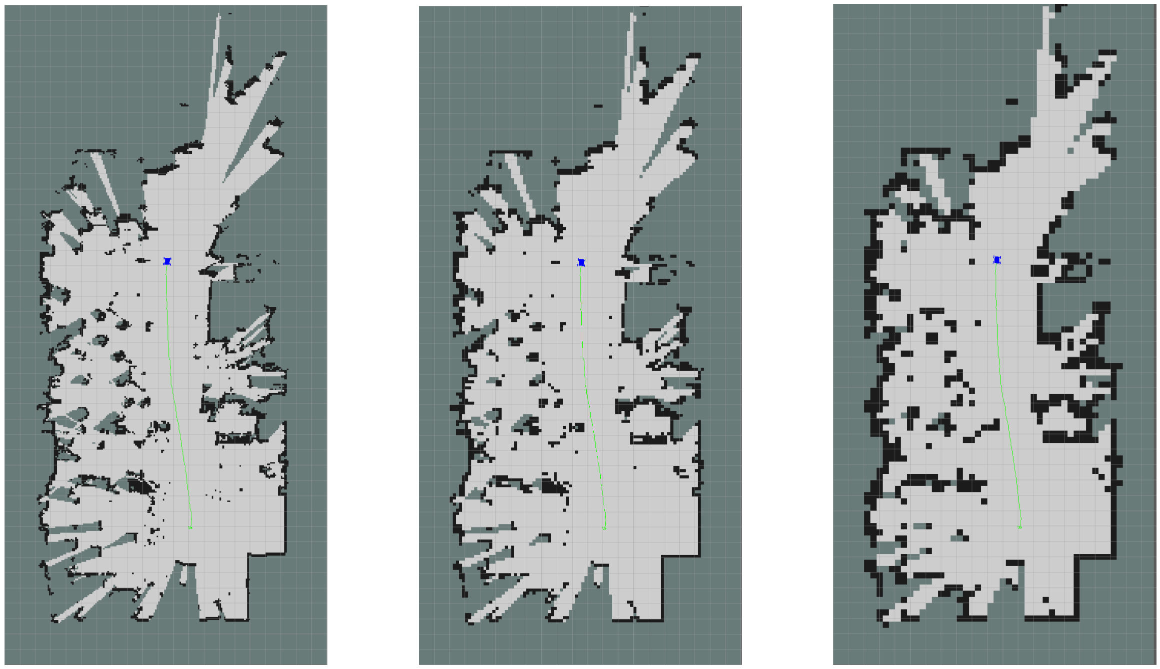
In the simulated exploration trials, the map was compressed to a fixed resolution ($n \in \{0, 2, 4\}$) using the PRI compression rule in (3.12). The compressed OG was used to calculate CSQMI along a set of planned actions. Resulting exploration paths are shown in Fig. 3.7. To measure the increase in efficiency of computing CSQMI reward from map compression, the number of actions output from the robot's planner was increased until evaluating CSQMI reward at action endpoints became prohibitively expensive (roughly 1.5 Hz planning frequency for a library containing 81 actions). Then, a goal velocity was chosen to make the robot follow its highest-reward action without colliding with walls or reaching the end of the action before a replan was triggered. With no map compression, this resulted in a velocity of 0.35 m/s, and the robot's total exploration time was 230.0 s.

The same exploration trial was run again, calculating CSQMI reward with respect

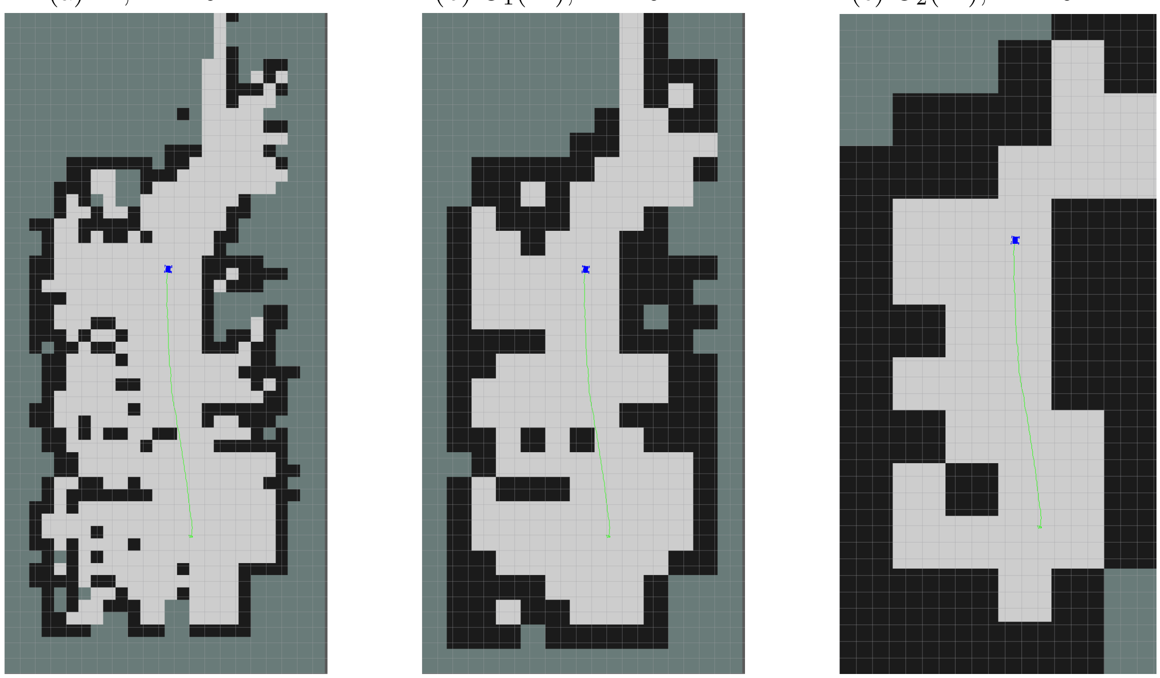
Table 3.1: Simulated exploration trial results (Fig. 3.7).

n	Δ (m)	Planning Freq. (Hz)	Maximum Velocity (m/s)	Time (s)
0	0.1	1.5	0.35	230.0
2	0.4	6.0	1.50	54.7
4	1.6	24.0	3.00	31.9

to compressed maps $C_2(\mathbf{m})$ and $C_4(\mathbf{m})$. At $n = 2$, the CSQMI reward for the 81 actions could be evaluated at approximately 6.0 Hz, resulting in a safe navigation velocity of 1.50 m/s and a total exploration time of 54.7 s. At $n = 4$, the CSQMI reward could be evaluated at 24.0 Hz, yielding a velocity of 3.00 m/s and a total exploration time of 31.9 s. Action collision checking was performed with respect to a configuration space built from the uncompressed map for all three trials. At $n = 4$, a different action is chosen in the top-right corner of the map, resulting in a different exploration path. The map is mostly complete after each of the three trials. Numerical results from the trials are shown in Table 3.1.

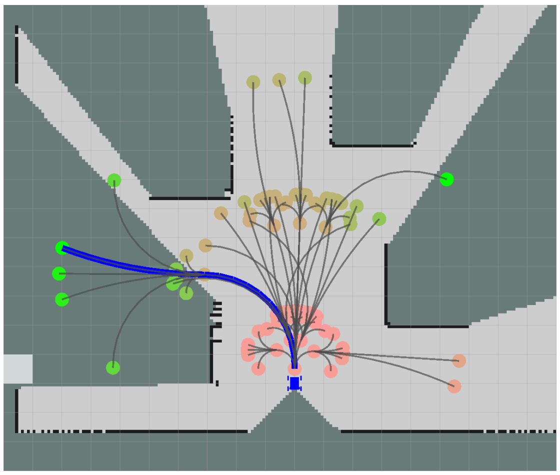


(a) $\mathbf{m}$, $\Delta = 0.1$ m (b) $C_1(\mathbf{m})$, $\Delta = 0.2$ m (c) $C_2(\mathbf{m})$, $\Delta = 0.4$ m

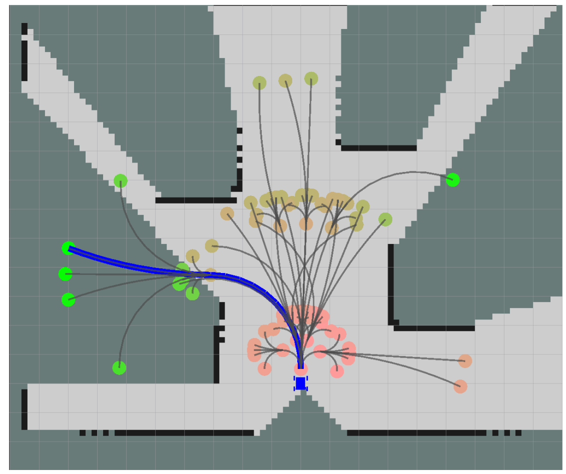


(d) $C_3(\mathbf{m})$, $\Delta = 0.8$ m (e) $C_4(\mathbf{m})$, $\Delta = 1.6$ m (f) $C_5(\mathbf{m})$, $\Delta = 3.2$ m

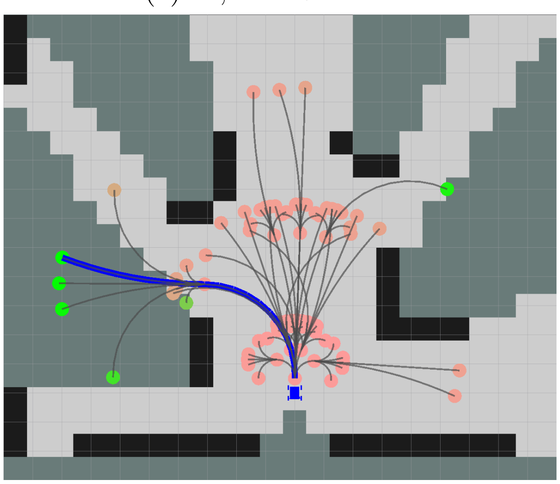
Figure 3.5: A six-level OG pyramid $\mathcal{C}_5(\mathbf{m})$, built from a partially explored 2D base OG, $\mathbf{m}$, in a cluttered warehouse environment. The PRI compression strategy preserves boundaries between free and unknown space, and expands occupied cells. Δ denotes cell edge length. The robot is shown in blue, and its path in green.



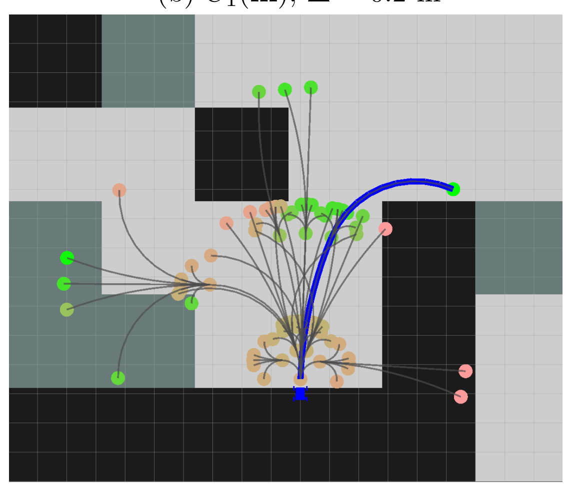
(a) $\mathbf{m}$, $\Delta = 0.1$ m



(b) $C_1(\mathbf{m})$, $\Delta = 0.2$ m

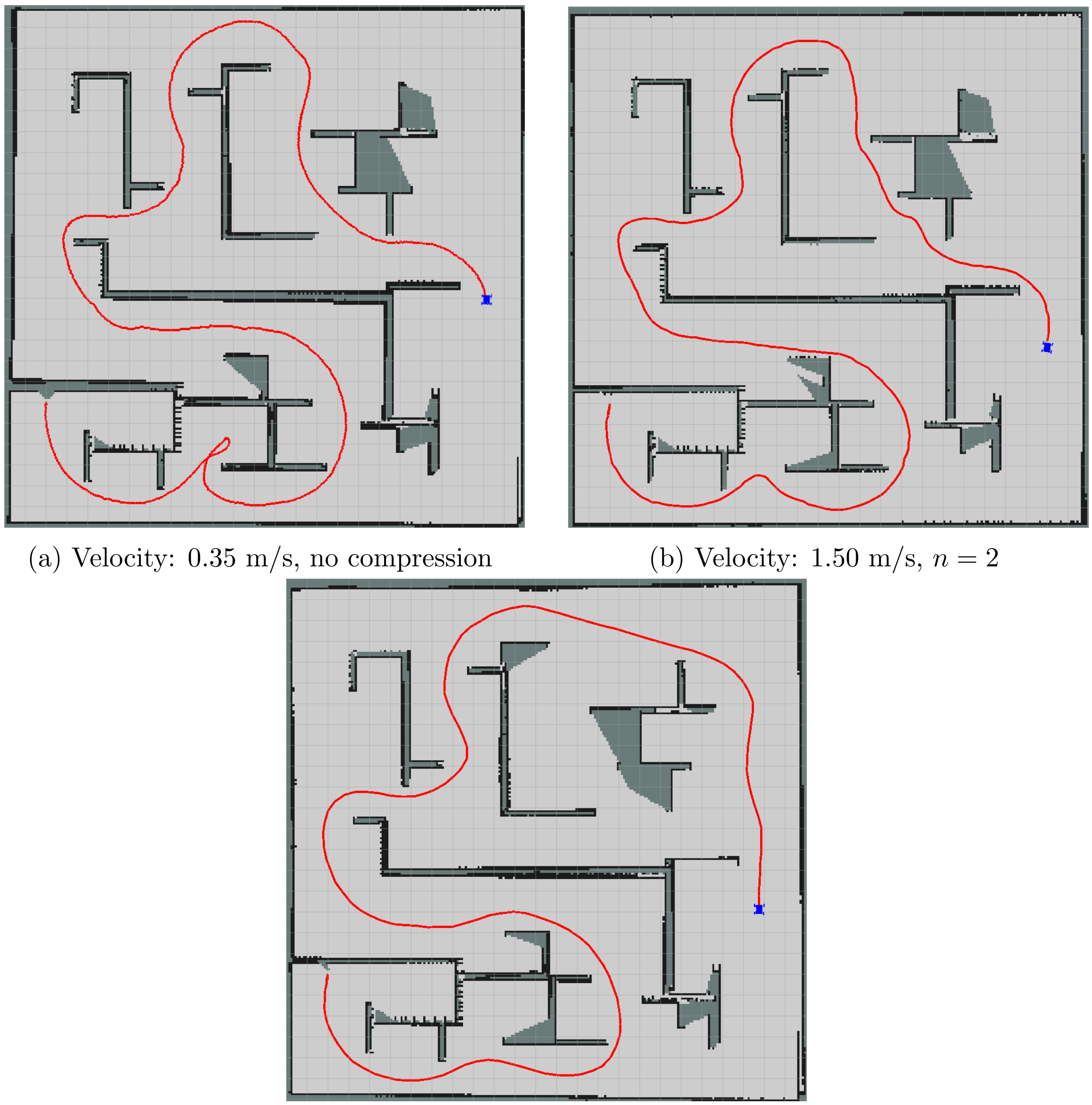


(c) $C_3(\mathbf{m})$, $\Delta = 0.8$ m



(d) $C_5(\mathbf{m})$, $\Delta = 3.2$ m

Figure 3.6: The robot plans forward-arc motion primitives into a partially explored map with varying compression, calculating CSQMI reward at primitive endpoints. The optimal exploration action (blue) is not affected until a large ($n = 5$) compression is applied. Green corresponds to high CSQMI reward, and red to low. Δ denotes cell edge length.



(a) Velocity: 0.35 m/s, no compression

(b) Velocity: 1.50 m/s, $n = 2$

(c) Velocity: 3.00 m/s, $n = 4$

Figure 3.7: Exploration paths when computing CSQMI with respect to: Fig. 3.7a an uncompressed OG; Fig. 3.7b an OG compressed with $n = 2$; Fig. 3.7c an OG compressed with $n = 4$. Compressing the OG used for CSQMI computation leads to higher planning frequencies, in turn enabling higher speed navigation. The uncompressed map is shown in each figure to demonstrate completeness after exploration.

3.6 Chapter Summary

Chapter 3 formulated environment model simplification as an information-theoretic optimization using the Principle of Relevant Information. As an example, the optimization was applied to the problem of occupancy grid compression, which was solved by decomposing an uncompressed OG into independent square (cubic in 3D) regions, and compressing each region such that both its entropy and divergence with respect to the uncompressed region were minimized. The PRI optimization lent equal weight to minimum entropy and minimum divergence (i.e. $\lambda = 1$), which gave a simple compression rule with three cases: a cell in the compressed OG can take on a value of either p_{free}, p_{occ}, or $\frac{1}{2}$. The PRI compression strategy was applied to a map of a large warehouse environment, and was shown to preserve boundaries between free and unknown space, and to expand occupied cells for $\eta \in (0, 1)$.

Heuristics were introduced to avoid zero-valued or one-valued probabilities in compressed OGs, and to ensure that occupied cells in the uncompressed map are preserved through compression. Occupancy grid pyramids were introduced as a mathematical tool for defining a set of OGs with increasing compression.

Applying the PRI environment model simplification strategy to OGs in a real cluttered warehouse environment (Fig. 3.5), and in simulated maze environments (Figs. 3.6 and 3.7) allowed the robot to plan actions more efficiently, which was hypothesized by Fig. 3.2. Propagating the gains in planning efficiency through to the robot's dynamics enabled the robot to explore faster when calculating CSQMI reward with respect to a compressed map. After significant amounts of compression, the robot began to select suboptimal exploration actions (Figs. 3.6d and 3.7c). The distortion to CSQMI reward with increasing map compression necessitates a method for choosing a compression level in response to the robot's environment.

Chapter 4

Balancing Map Simplification with Information Accuracy

This chapter examines distortions to mutual information between a sensor measurement and an environment belief distribution when the environment representation is adapted. The PRI simplification strategy described in Chapter 3 is lossy, resulting in simplified maps that have a lower information content than their originators. It should be expected, then, that information about random variables that share a dependence with the map (such as a sensor measurement) is also lost when the map undergoes simplification. Large distortions to a mutual information reward metric (e.g. CSQMI between a map and sensor measurement) induced by map simplification will cause the active perception optimization in (2.15) to select actions that would be suboptimal if CSQMI were to instead be computed using the original map. This chapter seeks to develop a strategy for balancing simplification (efficiency) with fidelity of information (accuracy) about the robot's sensor measurements.

In the context of exploration, a map could ideally be simplified significantly without causing perturbation to the ordering of rewards offered by a set of planned actions. This would allow a robot to choose optimal actions for exploration, while also enabling faster reward evaluation and therefore higher planning frequencies. However, map simplification and mutual information accuracy are two competing objectives. If one's goal is to select the best actions for exploration, one should choose not to simplify the map at all. Likewise, if one's goal is to evaluate reward over actions as efficiently as possible, one should simplify the map as much as possible. In the context of exploration, the optimal environment model adaptation strategy lies somewhere in between.

To balance these two competing objectives, a second optimization, based on the Information Bottleneck method [45], is introduced in order to select a map representation minimizing both the redundancy between the map and its simplified counterpart, and loss of mutual information with respect to a sensor measurement. Performing the Information Bottleneck optimization online allows a robot to simplify its map in a way that minimally distorts mutual information reward, and simultaneously enables more efficient reward evaluation.

As a robot explores, it is constantly evaluating reward in new areas of the environment. A map resolution that does not distort mutual information reward in one area of the environment may drastically distort it in another. An adaptive strategy is therefore developed to select an optimal map representation whenever the complexity of the robot's local environment has changed significantly.

4.1 The Information Bottleneck Method

The Information Bottleneck (IB) method is a widely used technique in signal processing for finding the optimal reduced representation $\hat{X}$ of a random variable X that

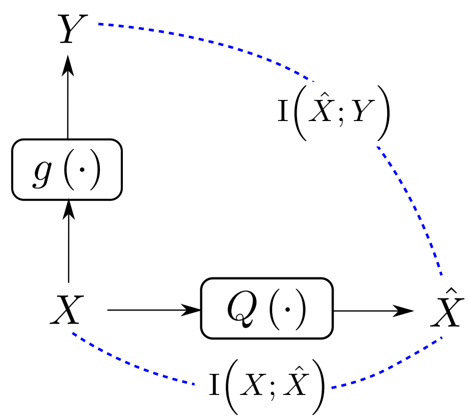


Figure 4.1: A diagram of variables relevant to the IB method. IB attempts to minimize the mutual information between X and its compressed form, $\hat{X}$, while maximizing the mutual information between X and a second variable, $Y = g(X)$. Q is a function quantizing (compressing) X.

preserves maximum information about a second random variable Y:

$$\Psi(p(\hat{x}|x)) = \min_{p(\hat{x}|x)} \mathrm{I}(X;\hat{X}) - \beta\mathrm{I}(\hat{X};Y). \tag{4.1}$$

IB resembles the PRI cost functional from Section 3.1, but considers the effects of compression on the information between two datasets, as opposed to one. Similar to λ in the PRI optimization, β is a design parameter that trades compression for conservation of information. As $\beta \to 0$, the optimization tends towards the trivial lossy compression $\{\hat{X}\} = 0$, whereas when $\beta \to \infty$, $\hat{X}$ approaches its original representation X [34]. The two variables in the information terms of the IB functional can equivalently be thought of as the information loss incurred by describing $\hat{X}$ with Y instead of with X [15, 16] (Fig. 4.1).

In most applications, the IB method is used to extract information relevant to Y from X by iteratively refining the distribution $p(\hat{x}|x)$. However, given access to a set of quantizers $\mathcal{Q}$ functioning on the uncompressed variable X such that $\hat{X} = Q(X)$, one may use the IB cost functional directly to select an optimal quantization Q^*.

$$Q^* = \underset{Q \in \mathcal{Q}}{\operatorname{argmin}} \ \mathrm{I}(X; Q(X)) - \beta \mathrm{I}(Q(X); Y). \tag{4.2}$$

4.2 Optimizing Map Resolution for Sensing

The IB method can be combined with the PRI compression strategy in Chapter 3 to determine a function, C^*, that simplifies the environment model as much as possible without sacrificing a significant amount of information relevant to a sensor measurement, $\mathbf{z}$. As a continuation to Chapter 3, the IB formulation will be applied to OG simplification, noting that it can be used for other environment models as well. To perform the IB optimization, one builds an n-level OG pyramid, $\mathcal{C}_n(\mathbf{m})$ (Section 3.4), from the robot's map, and selects the compression that minimizes the IB functional:

$$C^* = \underset{C \in \mathcal{C}_n(\mathbf{m})}{\operatorname{argmin}} \ \ \mathrm{I}_{\mathrm{CS}}\left(\mathbf{m}\,;\, C(\mathbf{m})\right) - \beta \mathrm{I}_{\mathrm{CS}}\left(C(\mathbf{m})\,;\, \mathbf{z}\right). \tag{4.3}$$

CSQMI is an appropriate choice for the mutual information terms in the IB optimization, since it allows the second term to be computed efficiently during exploration with the formulae supplied by Charrow et al. [11] (described in Section 2.5). However, a method for calculating the first term in (4.3) is not yet well-defined. It is also not yet clear what $\mathbf{m}$ itself should be. Optimizing the IB functional over the robot's entire map will cause the first term to converge onto a steady-state value as the robot explores and learns a larger and larger map. If $\mathbf{z}$ observed the entire map at once this would not be a problem, but it is more likely that $\mathbf{z}$ will only observe a small section of the environment at a time. It is more reasonable to make $\mathbf{m}$ refer to a subsection of the total map in the vicinity of the sensor measurement. This choice allows the robot to choose a map resolution specifically tailored to locations where it is also evaluating the reward $\mathrm{I}_{\mathrm{CS}}(C(\mathbf{m})\,;\mathbf{z})$. One sensible choice for $\mathbf{m}$ is a bounding box around the sensor measurement $\mathbf{z}$.

Computing the first term in the IB functional, $\mathrm{I}_{\mathrm{CS}}(\mathbf{m}\,; C(\mathbf{m}))$, requires substituting $\mathbf{m}$ and $C(\mathbf{m})$ into the definition of CSQMI in (2.13):

$$\mathrm{I}_{\mathrm{CS}}(\mathbf{m}\,; C(\mathbf{m})) = \log \frac{\left(\sum_{\mathcal{M}} \sum_{C(\mathcal{M})} p^2(\mathbf{m}, C(\mathbf{m}))\right)\left(\sum_{\mathcal{M}} \sum_{C(\mathcal{M})} p^2(\mathbf{m}) p^2(C(\mathbf{m}))\right)}{\left(\sum_{\mathcal{M}} \sum_{C(\mathcal{M})} p(\mathbf{m}) p(C(\mathbf{m})) p(\mathbf{m}, C(\mathbf{m}))\right)^2}, \quad (4.4)$$

where $C(\mathcal{M})$ is the support set of the compressed map $C(\mathbf{m})$. Due to cell independence, the sums over all possible maps reduce to products of sums over all possible cell values, `EMP` and `OCC` (abbreviated `E` and `O`):

$$\begin{aligned}\mathrm{I}_{\mathrm{CS}}(\mathbf{m}\,; C(\mathbf{m})) = \log \prod_{i,j} \sum_{o_1, o_2 \in \{\mathtt{E},\mathtt{O}\}} & p^2(\mathbf{m}_i = o_1, C(\mathbf{m}_j) = o_2) \\ + \log \prod_{i,j} \sum_{o_1, o_2 \in \{\mathtt{E},\mathtt{O}\}} & p^2(\mathbf{m}_i = o_1) p^2(C(\mathbf{m}_j) = o_2) \\ - 2 \log \prod_{i,j} \sum_{o_1, o_2 \in \{\mathtt{E},\mathtt{O}\}} & p(\mathbf{m}_i = o_1) p(C(\mathbf{m}_j) = o_2) p(\mathbf{m}_i = o_1, C(\mathbf{m}_j) = o_2),\end{aligned} \quad (4.5)$$

with products iterating over all grid cells in the map, $i, j \in \{1, \ldots, K\}$. While $p(\mathbf{m}_i)$ and $p(C(\mathbf{m}_j))$ are readily calculated using cell occupancy probabilities from the uncompressed and compressed OGs, the joint distribution $p(\mathbf{m}, C(\mathbf{m}))$ is not yet defined. Resurrecting the assumption that the map can be decomposed into independent square regions (**constraint 1** in Section 3.2) enables the joint distribution to be represented as a product of joint distributions from independent regions:

$$p(\mathbf{m}, C(\mathbf{m})) = \prod_{\mathbf{r} \in \mathbf{m}} p(\mathbf{m_r}, C(\mathbf{m_r})) = \prod_{\mathbf{r} \in \mathbf{m}} p(\mathbf{m_r}, \tilde{\mathbf{m}}_\mathbf{r}), \tag{4.6}$$

where $\mathbf{r}$ is a vector of indices into the map $\mathbf{m}$ that define a square (or cubic) region. The second equivalence in (4.6) holds by noting that the compressed region $C(\mathbf{m_r})$ has only one cell, and that the distribution $p(\tilde{\mathbf{m}}_\mathbf{r})$ is completely determined by knowing $p(C(\mathbf{m_r}))$. $p(C(\mathbf{m_r}))$ can be found by looking up the occupancy value of the single cell in the compressed map that corresponds to the region $\mathbf{r}$.

Table 4.1 defines a contingency table to aid in determining the joint distribution. The contingency table contains the random variables $\mathbf{m}^R$ and $\tilde{\mathbf{m}}^R$, written here with superscripts to refer to the cell count of the region, $R = |\mathbf{r}|$. Rows in the table enumerate the permutations that $\mathbf{m}^R$ can take (i.e. $\{\{\texttt{EMP}, \ldots, \texttt{EMP}\}, \ldots, \{\texttt{OCC}, \ldots, \texttt{OCC}\}\}$), while columns enumerate the permutations that $\tilde{\mathbf{m}}^R$ can take (again, $\{\{\texttt{EMP}, \ldots, \texttt{EMP}\}, \ldots, \{\texttt{OCC}, \ldots, \texttt{OCC}\}\}$). The marginal distributions $p(\mathbf{m}^R)$ and $p(\tilde{\mathbf{m}}^R)$ are shown in the right-most column and bottom-most row, respectively. While $p(\tilde{\mathbf{m}}^R)$ has a support set that contains non-uniform permutations, the PRI optimization dictates that $\tilde{\mathbf{m}}^R$ can only take on values $\{\texttt{EMP}, \ldots, \texttt{EMP}\}$ or $\{\texttt{OCC}, \ldots, \texttt{OCC}\}$, since ultimately $\tilde{\mathbf{m}}^R$ is compressed to a single cell, $\tilde{\mathbf{m}}^1$, taking on a value of either EMP or OCC (Fig. 3.3). All other permutations therefore have a marginal probability of zero.

The joint distribution $p(\mathbf{m}^R, \tilde{\mathbf{m}}^R)$ makes up the center cells of Table 4.1. The PRI compression strategy leaves this distribution underconstrained by two degrees. In general, there exists an infinite set of joint distributions that satisfy a set of marginal distributions. Two commonly used methods for constraining degrees of freedom are choosing the joint distribution to be a product of marginals (therefore enforcing independence between the variables), or choosing a joint distribution that maximizes the joint entropy $\mathrm{H}(\mathbf{m}^R, \tilde{\mathbf{m}}^R)$ [1]. In this situation, the first option will not suffice; forcing the variables to be independent of one another will cause $\mathrm{I}_{\mathrm{CS}}(\mathbf{m}\,;\,C(\mathbf{m}))$ to be equal to zero, eliminating all influence of the first term on the IB optimization. The second choice is viable, but a simpler third option is available.

The joint distribution $p(\mathbf{m}^R, \tilde{\mathbf{m}}^R)$ can instead be chosen to be a product of the marginal distributions $p(\mathbf{m}^R)$ and $p(\tilde{\mathbf{m}}^R)$, weighted by four extra coefficients $w_{1:4}$. These coefficients are chosen to downweigh the probability of the event that $\tilde{\mathbf{m}}^R$ takes the value $\{\texttt{EMP}, \ldots, \texttt{EMP}\}$ if any cells in $\mathbf{m}^R$ are occupied. This choice, similar to the heuristic η from Section 3.3, reflects the fact that occupied cells should be preserved through compression for common operations on OGs (raycasting and collision checking). The remaining three constants, $w_{2:4}$, balance the effects of w_1 such that the conditional distributions $p(\mathbf{m}^R \,|\, \tilde{\mathbf{m}}^R)$ and $p(\tilde{\mathbf{m}}^R \,|\, \mathbf{m}^R)$ across the rows and columns of Table 4.1 all sum to the marginal distributions on the bottom-most row and right-most column.

With the joint distribution $p(\mathbf{m}^R, \tilde{\mathbf{m}}^R)$ fixed, one can compute the first term in the IB cost function (4.3) using (4.5), where values for $p(\mathbf{m}_i, C(\mathbf{m}_j))$ can be looked up from a table of computed joint probabilities.

4.3 Adapting Map Representation Online

Rather than solving the IB optimization (4.3) upon initialization and fixing the resulting map representation, it is necessary to adapt the representation as the robot explores. An optimal map representation in one area of the environment is not necessarily optimal in another. For example, in a wide-open area consisting of mostly empty cells, a map can be simplified significantly before CSQMI between the map

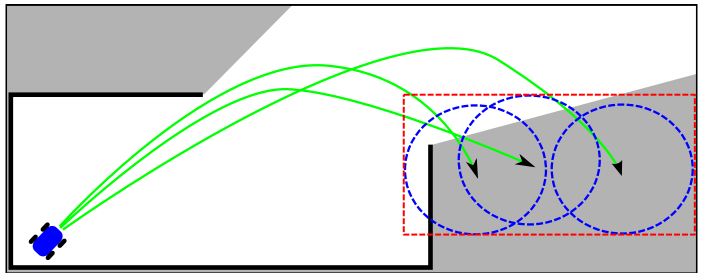


Figure 4.2: The mean entropy of cells in a submap of the robot's map is monitored to trigger IB optimizations. The submap (red) is defined by a bounding box around simulated sensor measurements (blue) acquired from planned actions (green).

and a sensor measurement is altered. Using the same reduced map representation in an area cluttered with obstacles would lead to inaccurate reward values.

The most naïve strategy for adapting the map representation to the environment is to reevaluate the IB optimization for each planned action. This strategy results in the best map representation possible for every evaluation of CSQMI reward. However, it also completely nullifies the benefits of simplification. The IB optimization itself requires computing CSQMI over a set of environment representations, making this strategy more inefficient than evaluating CSQMI on just the original representation. A more useful strategy is to reperform the IB optimization whenever the robot enters a significantly different area of the environment. Adapting to local changes in the map requires a method for detecting such changes.

One way to detect changes in the robot's local map is to look for changes to the mean entropy of cells in a submap where the robot is planning actions. Mean entropy is chosen for this section, noting that other options, such as monitoring changes to the mean or minimum distance to obstacles, changes to the ratio of free space to occupied space in a local map, or changes to feature-based map descriptors can also be used. A submap can similarly be defined in many ways; for the purposes of this section it is defined by a bounding box around the robot's planned actions, with an added buffer for sensor range (Fig. 4.2).

Every time an IB optimization is performed, the mean entropy, $\bar{\mathrm{H}}_{\mathrm{last}}$, of cells in the most recent submap is cached. Afterwards, this value is continuously monitored. A new IB optimization is triggered whenever the difference between $\bar{\mathrm{H}}_{\mathrm{last}}$ and the mean entropy of cells in the current submap surpasses a threshold, $\delta_{\mathrm{H}} \in (0, 1)$:

$$\left| \frac{1}{|\bar{\mathbf{m}}|} \sum_{i=1}^{|\bar{\mathbf{m}}|} \mathrm{H}(\bar{\mathbf{m}}_i) - \bar{\mathrm{H}}_{\mathrm{last}} \right| \geq \delta_{\mathrm{H}}, \tag{4.7}$$

where $\bar{\mathbf{m}} \subseteq \mathbf{m}$ is the robot's most recent submap.

When the criteria in (4.7) is met a random planned action is chosen, and an OG pyramid is computed in the submap defined by that action. The IB optimization (4.3) is then performed using the computed OG pyramid and randomly selected action to determine a new compression C^*. The new compression is used until the criteria in (4.7) is met again.

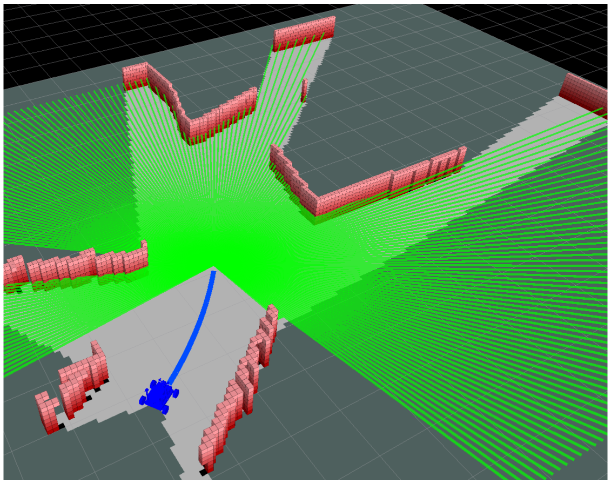
Calculating the mean entropy of cells in a submap has a bounded computational cost for actions with a fixed length, and is inexpensive in comparison to the IB optimization itself. Large values of δ_{H} make it necessary to observe large changes to the map before an IB optimization is retriggered.

4.4 Results

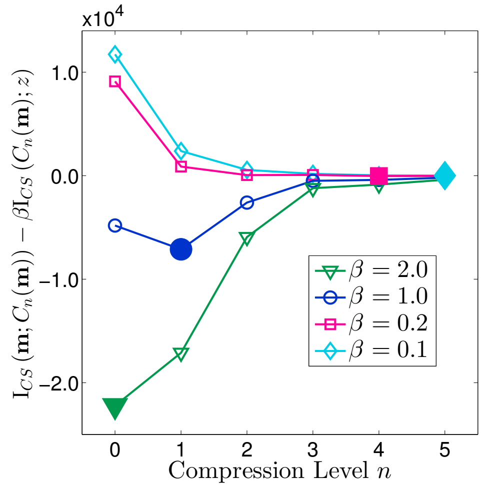
The variational parameter β plays an influential role in the IB optimization. Figure 4.3 displays a multi-beam sensor measurement simulated from the end pose of a planned action, and the IB cost functional for varying values of β. Larger values of β cause the optimization to favor no map compression (therefore preserving the entire information content of the sensor measurement). When β is small, the optimization is dominated by the minimum information term, and favors maximum compression.

The adaptive strategy introduced in Section 4.3 was tested by exploring a 35×35 m section of Carnegie Mellon University's Field Robotics Center with a ground robot. The ground robot was equipped with a MicroStrain 3DM-GX3-35 IMU, a Hokuyo URG-30LX 30 m range laser scanner, and an onboard computer with an Intel Core i5 processor and 8 GB RAM (Fig. 4.4). The robot's hardware (motor controllers, motors, and wheels) limit its maximum forward velocity to 1.6 m/s.

Figure 4.5 shows a 72 m exploration path through the environment (beginning from the bottom), and the adapted compression level and velocity estimate. Dashed lines in Fig. 4.6 correspond to times when the adaptation condition in (4.7) is met. Colored dashed lines mark times when (4.7) is met and (4.3) computes a new compression level n. The forward-arc motion primitive strategy was used for action generation (Section 2.3.2). Since generated primitives expanded in the robot's forward direction, entropy was generally computed in a submap in front of the robot. The optimal compression level remained at 4 in most of the free regions in the trial, and reduced to 0, 1, and 2 in locations where compression results in large reductions to CSQMI reward (e.g. the first $n = 0$ region occurred as the robot moved through a doorway). Although only a loose coupling was enforced in this experiment, vehicle velocity was adapted in response to the robot's estimated planning frequency. Propagating efficiency gains from map compression through to planning frequency and velocity caused the robot to accelerate and decelerate when entering highly compressible and incompressible areas, respectively. With no map compression, the maximum planning frequency would have been restricted to 2 Hz, whereas by simplifying the robot's map, the maximum planning frequency was increased to 24 Hz in areas with $n = 4$.



(a) A simulated future sensor measurement.



(b) Influence of β.

Figure 4.3: A sensor measurement is simulated from the endpoint of a planned action (Fig. 4.3a). The IB cost functional in (4.3) is shown in Fig. 4.3b for varying values of β. The optimal compression level (filled markers) decreases as β increases, favoring preservation of information about the measurement as opposed to compression.

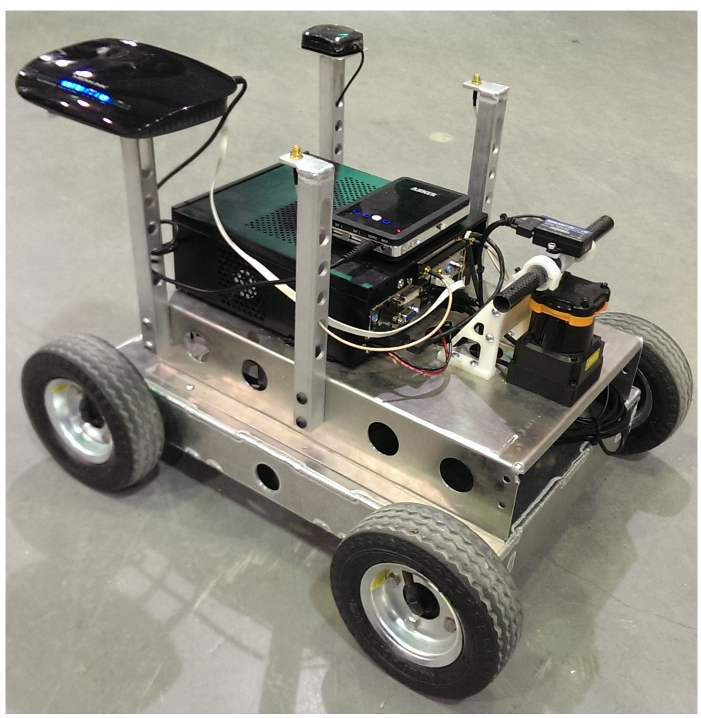


Figure 4.4: Robot platform

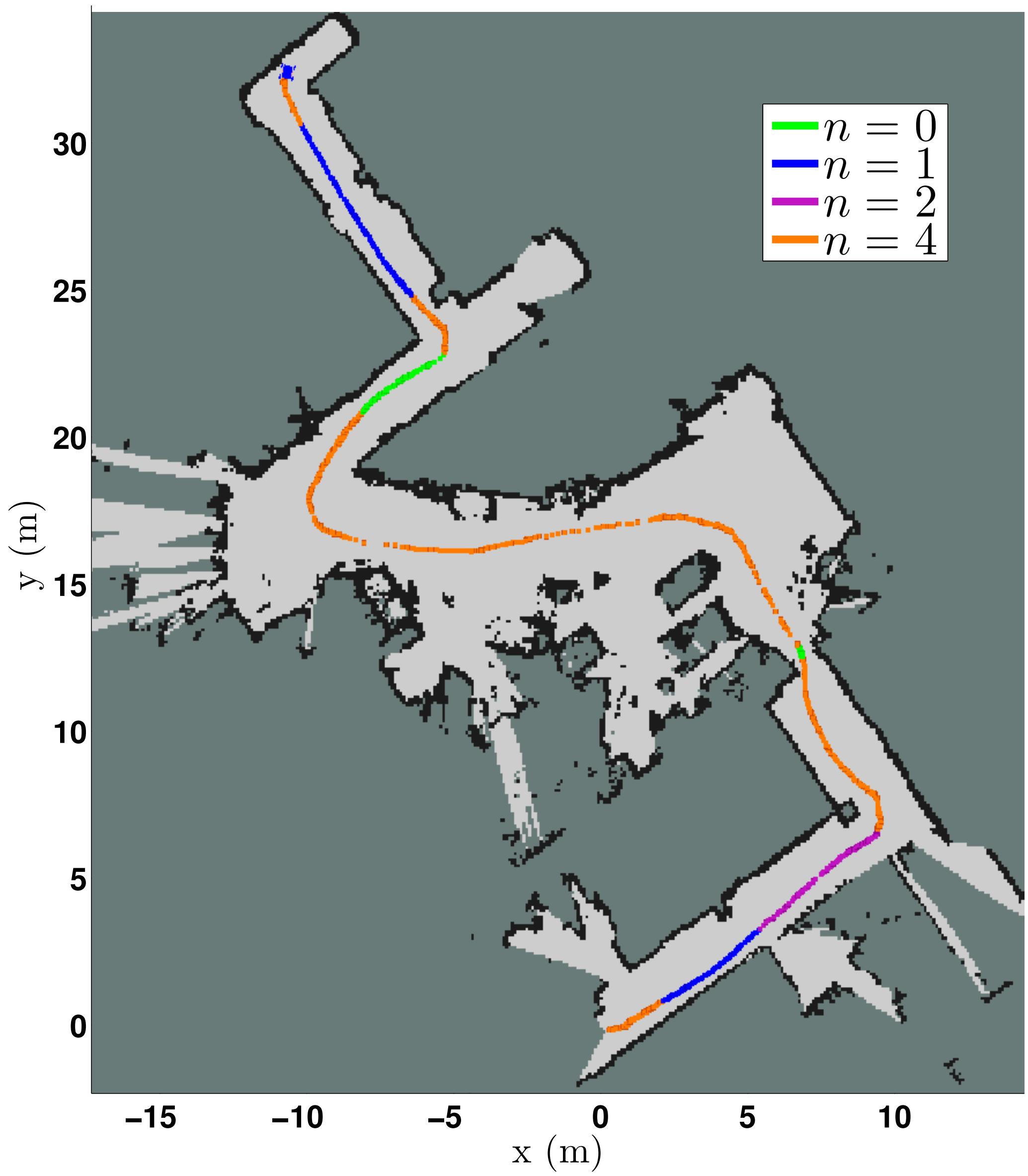


Figure 4.5: Ground robot exploration path (beginning at $(0, 0)$) with adaptive map compression.

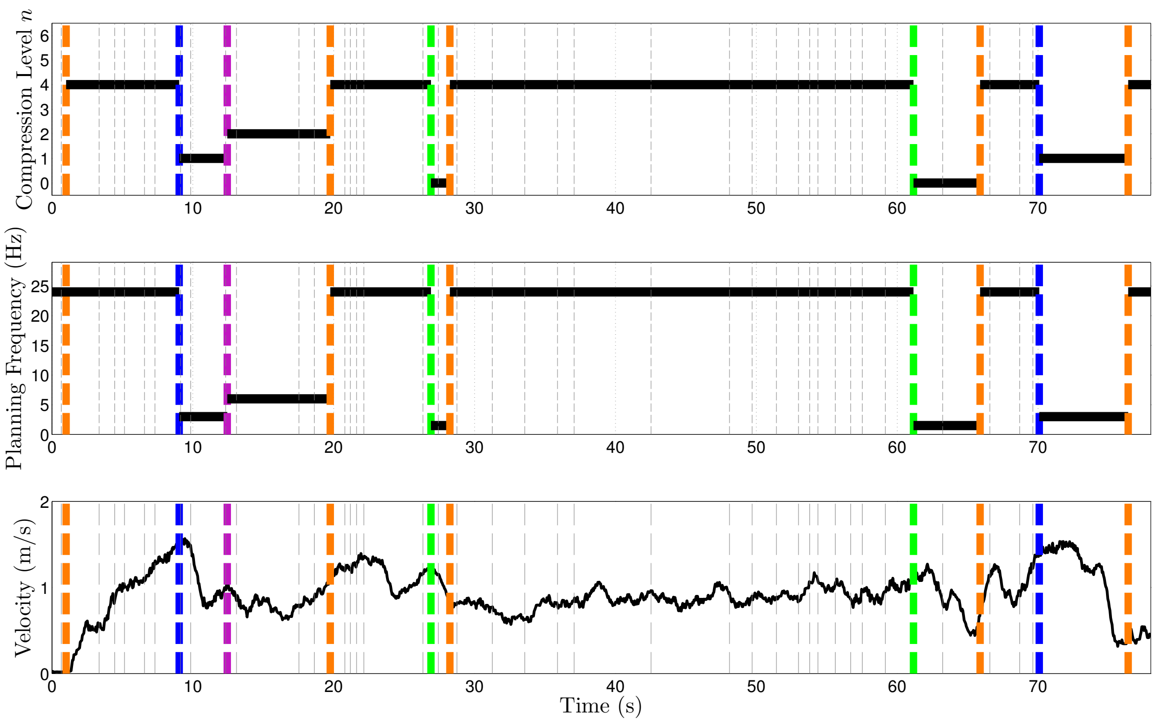


Figure 4.6: Time evolution of n, planning frequency, and velocity.

Table 4.1: Contingency table for a compression from the OG region $\mathbf{m}^R$ to $\tilde{\mathbf{m}}^R$. O and E stand for OCC and EMP.

		$\tilde{\mathbf{m}}^R$					
		E, E, …, E	E, E, …, O	…	O, O, …, E	O, O, …, O	Total
$\mathbf{m}^R$	E, E, …, E	$w_2 \cdot (1-\tilde{\mathbf{o}}^1) \cdot \prod_{i=1}^R (1-\mathbf{o}_i^R)$	0	…	0	$w_3 \cdot \tilde{\mathbf{o}}^1 \cdot \prod_{i=1}^R (1-\mathbf{o}_i^R)$	$\prod_{i=1}^R (1-\mathbf{o}_i^R)$
	E, E, …, O	$w_1 \cdot (1-\tilde{\mathbf{o}}^1) \cdot \mathbf{o}_1^R \cdot \prod_{i=2}^R (1-\mathbf{o}_i^R)$	0	…	0	$w_4 \cdot \tilde{\mathbf{o}}^1 \cdot \mathbf{o}_1 \cdot \prod_{i=2}^R (1-\mathbf{o}_i^R)$	$\mathbf{o}_1 \cdot \prod_{i=2}^R (1-\mathbf{o}_i^R)$
	⋮	⋮	⋮	⋱	⋮	⋮	⋮
	O, O, …, E	$w_1 \cdot (1-\tilde{\mathbf{o}}^1) \cdot (1-\mathbf{o}_1^R) \cdot \prod_{i=2}^R \mathbf{o}_i^R$	0	…	0	$w_4 \cdot \tilde{\mathbf{o}}^1 \cdot (1-\mathbf{o}_1^R) \cdot \prod_{i=2}^R \mathbf{o}_i^R$	$(1-\mathbf{o}_1^R) \cdot \prod_{i=2}^R \mathbf{o}_i^R$
	O, O, …, O	$w_1 \cdot (1-\tilde{\mathbf{o}}^1) \cdot \prod_{i=1}^R \mathbf{o}_i^R$	0	…	0	$w_4 \cdot \tilde{\mathbf{o}}^1 \cdot \prod_{i=1}^R \mathbf{o}_i^R$	$\prod_{i=1}^R \mathbf{o}_i^R$
	Total	$1-\tilde{\mathbf{o}}^1$	0	…	0	$\tilde{\mathbf{o}}^1$	1

4.5 Chapter Summary

Chapter 4 provided a strategy for choosing a environment representation that simultaneously minimizes the redundancy between the simplified and original maps, and minimizes distortion to CSQMI between the robot's map and expected future sensor measurements. The problem of balancing these two competing objectives was solved using the Information Bottleneck method.

Since an exploring robot can travel through multiple distinct sections of an environment in one deployment, the optimal map representation should be adapted whenever the robot enters an area that is significantly different. Section 4.3 introduced an entropy-based criteria to identify when the robot's local map has changed. Upon satisfying this criteria, an exploring robot can perform an IB optimization to recompute an optimal map representation based on the local environment.

Applied to the OG environment model, the IB optimization was shown to favor large map compressions for low values of β, and no map compression for high values (Fig. 4.3). A ground robot experiment was carried out to examine exploration performance with adaptive map compression. The robot was able to adapt its map compression, planning frequency, and velocity to changes in its local environment (Figs. 4.5 and 4.6). Without compressing its map, the computational cost of evaluating CSQMI limited the robot's planning frequency to 2 Hz. However, the adaptive map compression strategies introduced in Chapters 3 and 4 allowed for planning frequencies of up to 24 Hz in highly compressible areas.

Chapter 5

Conclusion

5.1 Summary

Autonomous mobile robots operating in unknown environments often require the ability to explore and characterize their surroundings for the purpose of generating a map, or for more complex tasks such as searching for objects, identifying threats, or tracking waypoints through unseen territory. In all of these applications, minimizing time and energy expenditure is of critical importance, suggesting that an exploring robot should be as efficient and intelligent as possible. Exploration can be broken into two components: defining and ranking informative actions that can be taken in the future, and pursuing the most informative action while simultaneously localizing and generating a map. This thesis explored one method for making the first component more efficient, thereby enabling consideration of either more actions in the same amount of time, or the same amount of actions in a shorter amount of time. The consideration of a larger number of actions leads to choosing more informative paths. The consideration of actions in a shorter amount of time leads to increased planning frequency and velocity. Combined, these implications allow a robot to explore an unknown environment more efficiently.

Chapter 2 provided an overview of foundations relevant to the active perception task. These foundations culminated in an optimization that drives a robot towards unexplored space in its map by maximizing an information-theoretic reward function. The optimization is performed over possible future actions, which reside in a space too large for dense online reward evaluation. Action generation techniques were introduced to identify actions likely to yield high amounts of information about a map, therefore constraining the space that the active perception optimization must be performed over. Frontier seeding had the advantage of providing the guarantee that all unknown space would eventually be explored, but was also limited to only choosing actions that would globally reduce map uncertainty. By contrast, the two motion primitive techniques myopically and greedily selected actions that were likely to locally be informative, but were not able to guarantee that the map would completely be explored.

Chapter 3 discussed environment model simplification as a means of increasing the efficiency of evaluating information-theoretic reward on actions. This was motivated by the fact that the time required to evaluate mutual information reward is reduced when the environment model is simplified through approximation or compression, and by the observation that a map sometimes contains large amounts of redundancy and uniformity. A map simplification strategy was developed using the Principle of Relevant Information, a technique borrowed from rate distortion theory, which minimizes divergence between the simplified and original map representations while simultaneously minimizing the entropy of the simplified map. Applied to occupancy grid mapping, several results from Chapter 3 demonstrate that map simplification leads to small, but tolerable distortions to CSQMI reward, and yields large efficiency gains for the exploration task.

Finally, Chapter 4 examined distortions to mutual information between a sensor measurement and map when the environment representation is adapted. Environment model simplification and sensing accuracy were found to be competing objectives; to minimize distortion to mutual information, one should choose to use the original map, whereas to maximize efficiency of the exploration task, one should choose a reduced map representation. These competing objectives were balanced using the Information Bottleneck method, a technique borrowed from signal processing. Performing the Information Bottleneck optimization on a set of occupancy grids compressed to different resolutions allows one to choose a compression that makes the exploration task as efficient as possible without drastically altering which actions are chosen. Since the Information Bottleneck optimization is dependent on the structure of the robot's local map, an adaptive strategy was used to recompute an optimal map resolution whenever the robot entered a significantly different area of its environment. Performing this adaptation online caused a ground robot to change its planning frequency and velocity in response to the complexity of its surroundings.

5.2 Future Work

There exist many interesting avenues for future research on the topic of efficient exploration. This section mentions several that are immediate and worth further pursuit.

The optimizations developed in Chapters 3 and 4 complement existing multi-resolution map data structures such as an octrees [48] and NDT maps [38]. The PRI simplification strategy can be used to determine occupancy probabilities in lower resolution cells, and the IB optimization can be used to determine a proper resolution for queries on the data structure. Combining the approaches from this thesis with these map representations encourages a closer examination of how to calculate information-theoretic reward on maps with cells of non-uniform resolution.

All results in this thesis were generated using 2D occupancy grids. No experiments utilized or necessitated exploration into the third dimension, since the ground robot used for experimentation was constrained to a flat floor. A natural future work would include investigating the benefits and complications of map compression in domains where a robot must explore a 3D space. It is expected that 3D environments will require a larger number of actions for exploration, which would increase the utility offered by map compression.

Highly compressible areas in a map tend to have nearly uniform occupancy probabilities in the uncompressed map. This observation could be leveraged to plan trajectories through areas that are both informative and sparse with obstacles. As the robot's velocity increases, planning safe and obstacle-free paths becomes increasingly important, welcoming strategies for avoiding obstacle-dense regions.

In Chapter 4, the coupling between compression level and velocity was simple and naïve. In reality, the maximum safe velocity that a robot can explore is a function of planning frequency, map update frequency, action count, dynamic constraints, and many other parameters. A more thorough model of the robot's dynamics and a more tightly coupled implementation might allow the robot to navigate at higher speeds.

Finally, the introduction of multiple robots into the exploration task raises interesting questions about whether robots should share a map, which robot controls the map's resolution, and how multiple robots should explore in the presence of distorted information-theoretic rewards.

Bibliography

[1] Ali E Abbas. Entropy methods for joint distributions in decision analysis. *Engineering Management, IEEE Transactions on*, 53(1):146–159, 2006.

[2] Ercan U Acar and Howie Choset. Sensor-based coverage of unknown environments: Incremental construction of morse decompositions. *The International Journal of Robotics Research*, 21(4):345–366, 2002.

[3] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

[4] Francesco Amigoni and Vincenzo Caglioti. An information-based exploration strategy for environment mapping with mobile robots. *Robotics and Autonomous Systems*, 58(5):684–699, 2010.

[5] Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[6] Ruzena Bajcsy and Mario Campos. Active and exploratory perception. *CVGIP: Image Understanding*, 56(1):31–40, 1992.

[7] Nicola Basilico and Francesco Amigoni. On evaluating performance of exploration strategies for autonomous mobile robots. In *Proceedings of the perfor-*

mance evaluation and benchmarking for intelligent robots and systems workshop at IEEE/RSJ international conference on intelligent robots and systems, 2008.

[8] Frederic Bourgault, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. Information based adaptive robotic exploration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 540–545. IEEE, 2002.

[9] Wolfram Burgard, Mark Moors, Dieter Fox, Reid Simmons, and Sebastian Thrun. Collaborative multi-robot exploration. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 476–481. IEEE, 2000.

[10] Kwong-Fai Chan and Tak Wah Lam. An on-line algorithm for navigating in an unknown environment. *International Journal of Computational Geometry & Applications*, 3(03):227–244, 1993.

[11] Ben Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *Robotics and Automation, 2015. Proceedings., 2015 IEEE International Conference on*. IEEE, 2015.

[12] Erik Einhorn, C Schroter, and H Gross. Finding the adequate resolution for grid mapping-cell sizes locally adapting on-the-fly. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1843–1848. IEEE, 2011.

[13] Alberto Elfes. *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University, 1989.

[14] Hans Jacob S Feder, John J Leonard, and Christopher M Smith. Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research*, 18(7):650–668, 1999.

[15] Bernhard C Geiger, Christian Feldbauer, and Gernot Kubin. Information loss in static nonlinearities. In *Wireless Communication Systems (ISWCS), 2011 8th International Symposium on*, pages 799–803. IEEE, 2011.

[16] Bernhard C Geiger and Gernot Kubin. Signal enhancement as minimization of relevant information loss. In *Systems, Communication and Coding (SCC), Proceedings of 2013 9th International ITG Conference on*, pages 1–6. VDE, 2013.

[17] Hector H Gonzalez-Banos and Jean-Claude Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848, 2002.

[18] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.

[19] Dirk Holz, Nicola Basilico, Francesco Amigoni, and Sven Behnke. A comparative evaluation of exploration strategies and heuristics to improve them. In *ECMR*, pages 25–30, 2011.

[20] Jeong Joon Im, Alexander Leonessa, and Andrew Kurdila. A real-time data compression and occupancy grid map generation for ground-based 3d lidar data using wavelets. In *3rd ASME Dynamic Systems and Control Conference, Boston, MA*, 2010.

[21] Brian J Julian, Sertac Karaman, and Daniela Rus. On mutual information-based control of range sensing robots for mapping applications. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5156–5163. IEEE, 2013.

[22] Brian John Julian. *Mutual information-based gradient-ascent control for distributed robotics*. PhD thesis, Massachusetts Institute of Technology, 2013.

[23] Thomas Kollar and Nicholas Roy. Efficient optimization of information-theoretic exploration in slam. In *AAAI*, volume 8, pages 1369–1375, 2008.

[24] Thomas Kollar and Nicholas Roy. Trajectory optimization using reinforcement learning for map exploration. *The International Journal of Robotics Research*, 27(2):175–196, 2008.

[25] Henrik Kretzschmar and Cyrill Stachniss. Information-theoretic compression of pose graphs for laser-based slam. *The International Journal of Robotics Research*, 31(11):1219–1230, 2012.

[26] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, pages 79–86, 1951.

[27] Steven M LaValle. Rapidly-exploring random trees a ew tool for path planning. Technical report, Compute Science Department, Iowa State University, 1998.

[28] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[29] Erik Nelson and Nathan Michael. Optimal information-theoretic occupancy grid compression for high-speed information-based exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015. (IROS 2015)*. IEEE, 2015.

[30] Mihail Pivtoraiko and Alonzo Kelly. Generating near minimal spanning control sets for constrained motion planning in discrete state spaces. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3231–3237. IEEE, 2005.

[31] Mihail Pivtoraiko, Ross A Knepper, and Alonzo Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333, 2009.

[32] Mihail Pivtoraiko, Daniel Mellinger, and Vijay Kumar. Incremental micro-uav motion replanning for exploring unknown environments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2452–2458. IEEE, 2013.

[33] Francois Pomerleau, Francis Colas, Roland Siegwart, and Stephane Magnenat. Comparing icp variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.

[34] Jose C Principe. *Information theoretic learning: Rényi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

[35] S Rao. *Unsupervised Learning: An Information Theoretic Learning Approach*. PhD thesis, University of Florida, Gainesville, 2008.

[36] Alfred Renyi. On measures of entropy and information. In *Fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 547–561, 1961.

[37] Walter Rudin. *Principles of mathematical analysis*, volume 3. McGraw-Hill New York, 1964.

[38] Jari Saarinen, Henrik Andreasson, Todor Stoyanov, Juha Ala-Luhtala, and Achim J Lilienthal. Normal distributions transform occupancy maps: Application to large-scale online 3d mapping. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2233–2238. IEEE, 2013.

[39] Claude Elwood Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

[40] Shaojie Shen, Nathan Michael, and Vijay Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 20–25. IEEE, 2011.

[41] Shaojie Shen, Nathan Michael, and Vijay Kumar. Autonomous indoor 3d exploration with a micro-aerial vehicle. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 9–15. IEEE, 2012.

[42] Robert Sim, Gregory Dudek, and Nicholas Roy. Online control policy optimization for minimizing map uncertainty during exploration. In *Robotics and Autkollar2008trajectoryomation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1758–1763. IEEE, 2004.

[43] Camillo J Taylor and D Kriegman. Exploration strategies for mobile robots. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 248–253. IEEE, 1993.

[44] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[45] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[46] Joan Vallvé and Juan Andrade-Cetto. Dense entropy decrease estimation for mobile robot exploration. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6083–6089. IEEE, 2014.

[47] Pengpeng Wang. *View planning with combined view and travel cost*. PhD thesis, School of Engineering Science-Simon Fraser University, 2007.

[48] Kai M Wurm, Armin Hornung, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2, 2010.

[49] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE, 1997.