Learning Semantic Maps from Natural Language

by

Sachithra Madhawa Hemachandra

S.M., Massachusetts Institute of Technology (2010)
B.Sc., University of Moratuwa (2006)

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2015

Author . . . . . .
Department of
Electrical Engineering and Computer Science
January 15, 2015

Certified by . . . . . .
Professor Nicholas Roy
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by . . . . . .
Professor Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses

Learning Semantic Maps from Natural Language

by

Sachithra Madhawa Hemachandra

Submitted to the Department of
Electrical Engineering and Computer Science
on January 15, 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

As robots move into human-occupied environments, the need for effective mechanisms to enable interactions with humans becomes vital. Natural language is a flexible, intuitive medium that can enable such interactions, but language understanding requires robots to learn representations of their environments that are compatible with the conceptual models used by people. Current approaches to constructing such spatial-semantic representations rely solely on traditional sensors to acquire knowledge of the environment, which restricts robots to learning limited knowledge of their local surround. Furthermore, they can only reason over the limited portion of the environment that is in the robot's field-of-view. Natural language, on the other hand, allows people to share rich properties of their environment with their robotic partners in a flexible, efficient manner. The ability to integrate such descriptions can allow the robot to learn semantic properties such as colloquial names that are difficult to infer using existing methods, and learn about the world outside its perception range. The spatial and temporal disconnect between language descriptions and the robot's onboard sensors makes fusing the two sources of information challenging.

This thesis addresses the problem of fusing information contained in natural language descriptions with the robot's onboard sensors to construct spatial-semantic representations useful for interacting with humans. The novelty lies in treating natural language descriptions as another sensor observation that informs the robot about its environment. Towards this end, we introduce the *semantic graph*, a spatial-semantic representation that provides a common framework in which we integrate information that the user communicates (e.g., labels and spatial relations) with observations from the robot's sensors. Our algorithm efficiently maintains a factored distribution over semantic graphs based upon the stream of natural language and low-level sensor information. We detail the means by which the framework incorporates knowledge conveyed by the user's descriptions, including the ability to reason over expressions that reference yet unknown regions in the environment. We evaluate the algorithm's ability to learn human-centric maps of several different environments and analyze the knowledge inferred from language and the utility of the learned maps. The results demonstrate that the incorporation of information from free-form descriptions increases the metric, topological and semantic accuracy of the recovered envi-

ronment model.

Next, we outline an algorithm that enables robots to improve their spatial-semantic representation of an environment by engaging users in dialog. The algorithm reasons over the ambiguity of language descriptions provided by the user given the current map, and selects information-gathering actions in the form of targeted questions about its local surroundings and areas distant from the robot. Our algorithm balances the information-theoretic value of candidate questions with a measure of cost associated with dialog. We demonstrate that by asking deliberate questions of the user, the method significantly improves the accuracy of the learned semantic map.

Finally, we introduce a learning framework that enables robots to successfully follow natural language navigation instructions within previously unknown environments. The algorithm utilizes information about the environment that the human conveys within the command to learn a distribution over the spatial-semantic model of the environment. We achieve this through a formulation of our semantic mapping algorithm that uses information conveyed in the command to directly reason over unobserved spatial structure. The framework then uses this distribution in place of the latent world model to interpret the natural language instruction as a distribution over the intended actions. Next, a belief space planner solves for the action that best satisfies the intent of the command. We apply this towards following directions to objects and natural language route directions in unknown environments. We evaluate this approach through simulation and physical experiments, and demonstrate its ability to follow navigation commands with performance comparable to that of a fully-known environment.

Thesis Supervisor: Professor Nicholas Roy
Title: Associate Professor of Aeronautics and Astronautics

To Seth

Acknowledgments

I would like to express my deepest gratitude to my advisors, the late Seth Teller, and Nick Roy. I was privileged to work under the guidance of two great individuals whom I hold highest in my regards. I thank them for their excellent guidance, patience and for providing me with the opportunity to be a PhD candidate under their respective lab groups.

I thank my graduate committee, Antonio Torralba and John Leonard, for their insightful comments and their rallying enthusiasm to get me through this last hurdle.

I sincerely thank Matt Walter, lab mate, collaborator but above all a true friend. Thank you for all the knowledge you have shared and the times you've spent reviewing papers/slides/thesis drafts and code and simply putting up with all my jokes.

Thank you to Felix Duvallet, Stefanie Tellex, Tom Howard, and Tom Kollar for setting the path for me, working with me, and all the valuable insights.

Thank you Janet Fisher for all the support you have shown me through out my time as a PhD student, but also for all you support this year through a difficult time. Also, thank you Leslie Kolodziejski, for being so supportive of me and making sure everything was taken care of.

A sincere thank you to Don, Marva and everyone at The Boston Home for being the source of inspiration and collaboration throughout the years. Having the opportunity to meet the individuals who might benefit from our work one day not only provided me with the practical aspect of the research, but also a great source of inspiration.

Also thanks to my lab mates in RVSN and RRG, especially William Li, Sudeep Pillai, Ross Finman, Abe Bachrach, and Albert Huang. I would also like to thank Bryt Bradley and Sophia Hasenfus for all the times they helped me to get things done. Thanks to the CSAIL and MIT community in general which provided not only an awesome atmosphere to conduct research but also a place to call home.

Last but not least I would like to thank a group of individuals who are near and dear to my heart. My parents and sibling for helping to mold the individual I am today. For always believing in me, even when I doubted my self. My wife, Dilini for being a constant presence in my life for the past five odd years. My friends, too numerous to mention, who

are my family away from home, for keeping me sane and simply being there to share all my ups and downs as a graduate student.

This work was supported by Quanta Computer and by the Robotics Consortium of the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016.

Contents

List of Figures

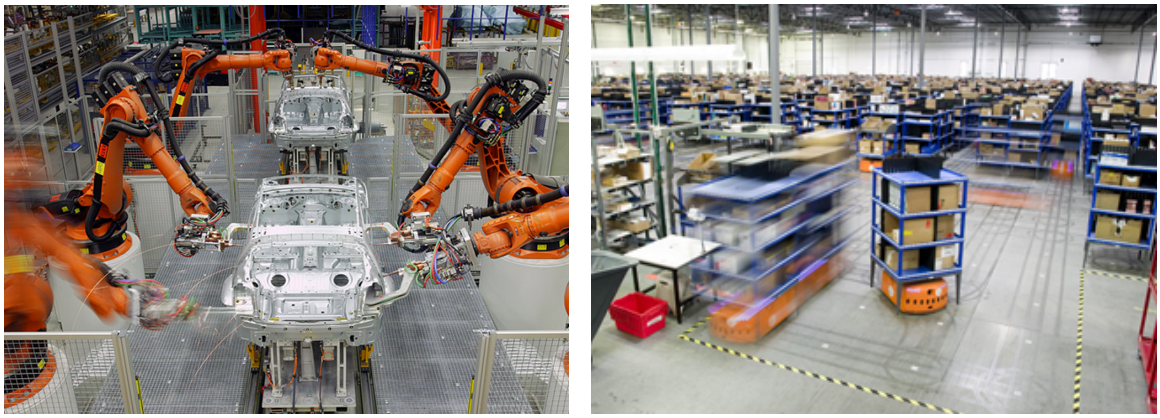
List of Tables

Chapter 1

Introduction

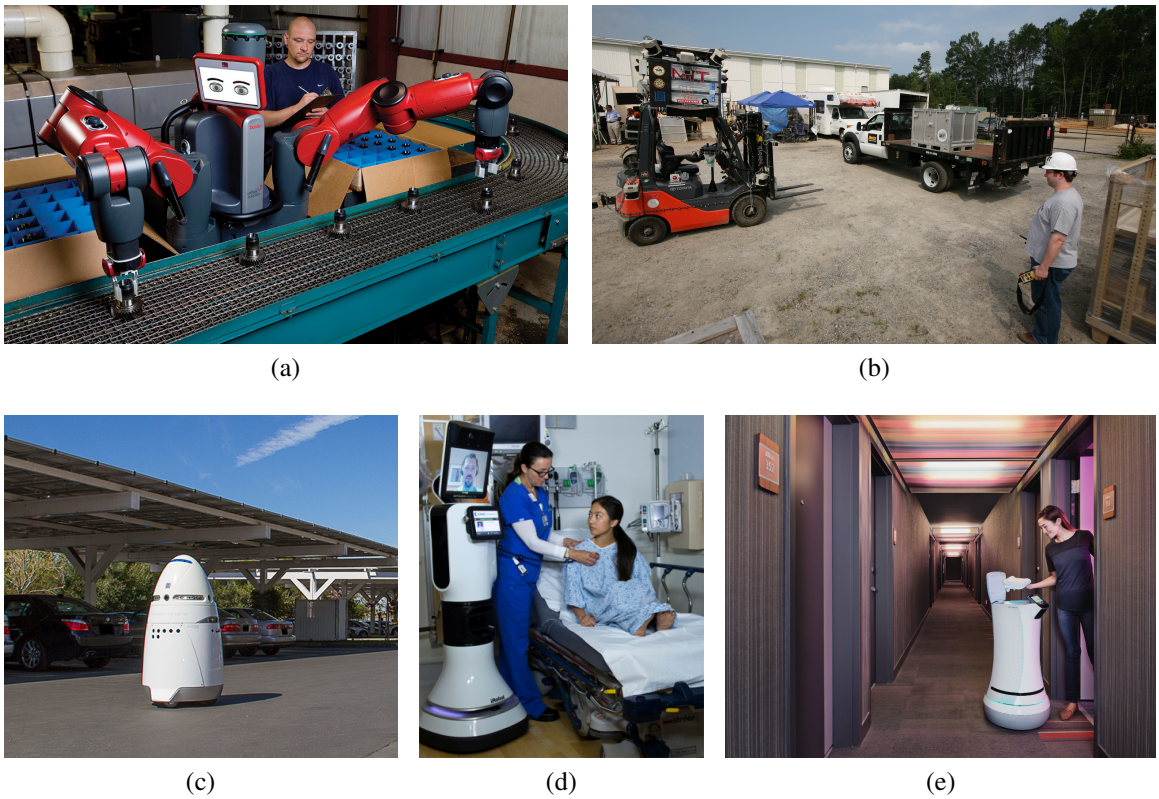
For decades robots have operated in environments such as factories, where their ability to perform precise repeatable actions was valued above all else. However, such robots lacked the ability to perceive and reason about their surroundings, could not handle uncertainty in their environments, and were not able to interact safely with humans. Due to this, they have often been separated from human occupants to ensure safety and confined to specially prepared highly controlled environments to minimize uncertainty (see Figure 1-1).

With the advent of better sensors, such as lidars and RGB-D cameras, more powerful computational resources, and more capable algorithms for mapping, localization, state-



(a) (b)

Figure 1-1: Robots operating in controlled environments. (a) Robots in automobile manufacturing, (b) Kiva systems logistics robots operating in specially prepared factories (reprinted with permission).



(a) (b)

(c) (d) (e)

Figure 1-2: Robots operating with human partners. (a) A robot in a factory, (b) A forklift robot, (c) A security robot, (d) A tele-presence robot in a hospital, (e) A delivery robot in a hotel. Images (a), (c), (d), (e) are reprinted with permission from Rethink Robotics, Inc., ©Knightscope, Inc. 2015 [39], iRobot Corporation, and Savioke, Inc. respectively.

estimation, manipulation, and planning under uncertainty, robots are better able to perceive and interact with people in unstructured environments. These advances have enabled increasing deployment of robots in populated environments, including homes, offices, and hospitals, with the potential of having significant positive impact on peoples lives. Robots capable of delivering items in offices and hotels [80] and medicine in hospitals [1, 97], providing security and surveillance in buildings [39], enabling telepresence [36, 87] and remote medicine [35], and assisting people in factories [75] are being deployed in increasing numbers (see Figure 1-2).

However, for such robots to be deployed in large-scale, they must be capable of interacting with and responding effectively to novice users. This requires effective methods for untrained users to control complex robots, without the need for specialized interfaces or extensive user training.

1.1 Enabling Behaviors with Natural Language

Natural language is one such mechanism that provide users with a flexible, intuitive medium with which to communicate with robots, without extensive prior training. For example, a voice-commandable wheelchair [30] can allow the mobility-impaired to independently and safely navigate their surroundings simply by speaking to the chair, without the need for traditional head-actuated switches or sip and puff arrays. Recognizing these advantages, much attention has been paid of late to developing algorithms that enable robots to interpret natural language expressions that provide route directions [53, 41, 8, 56], and that command navigation and manipulation [88, 34]. These algorithms have either attempted to parse free-form commands into their formal language equivalent, which a planner takes as input [82, 53, 19, 8, 58, 57] or infer the maximum likelihood mapping of free-form utterances into their corresponding object and action referents in the robot's world model [41, 88, 89, 34].

Natural language interpretation becomes particularly challenging when the expression references attributes of the environment unknown to the robot. Consider an example in which a user of the voice-commandable wheelchair directs it to "take me to the kitchen down the hallway" when the wheelchair is in an unknown environment and the hallway and kitchen are outside the field-of-view of its sensors. If the robot is unable to make use of the information about the world contained in the command, it will be reduced to following a blind exploration-based strategy until it happens upon the kitchen. However, if it reasons about the constraints over the environment imposed by information contained in the natural language command, it can take actions that follow the spoken directions more efficiently.

To effectively respond to natural language commands, a robot needs to maintain *shared situational awareness* with the human operator. Shared situational awareness can be defined as a "shared perception of elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" [21].

Building Shared Representations

In order to achieve shared situational awareness, robots need to maintain a representation of the environment compatible with that of their human partners. Such *spatial-semantic* representations consists of human compatible concepts of spatial entities (places or regions in the world), their connectivity and associated spatial properties, such as their metric locations and spatial extent, coupled with semantic attributes that are relevant to and defined by humans who inhabit the environment. These semantic attributes can range from the type of each region (e.g., "hallway," or "kitchen"), their colloquial names (e.g., "Mark's office"), the objects that they contain and types of activities that can be carried out at these places (e.g., "eat lunch").

Frameworks for constructing such representations [44, 24, 102, 40, 69] have traditionally relied on using the robot's onboard sensors to infer spatial and semantic properties of its environment. However, even when robot sensors are combined with region appearance models and object detectors, it is difficult to infer certain semantic properties, such as the colloquial names of places. Additionally, in such frameworks, due to the spatially local nature of robot sensors, the only manner in which to learn about a place in the environment would be to visit the location and observe its semantic properties. These representations are built in a bottom-up manner, where higher level concepts are inferred from lower level properties, but high level information is not used to improve lower level properties (see Figure 1-3). For example, knowing that the robot returned to the same "gym" area does not

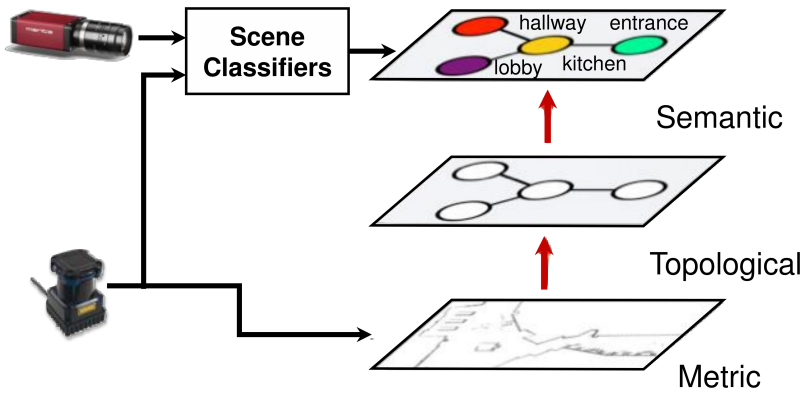


Figure 1-3: Traditional semantic mapping frameworks (information flows from lower layers to higher layers)

(a) (b)

Figure 1-4: Users providing descriptions to a robot. (a) A user providing natural language directions to a robotic wheelchair. (b) A user giving a tour to a robotic wheelchair

improve the robot's metric map.

However, a robot interpreting natural language commands might have little or no prior knowledge of its environment. To correctly interpret such instructions, the robot needs to reason over the parts of the environment that are relevant to understanding the instructions, but may not yet have been observed. Often during the course of commanding a robot (Figure 1-4a) or while introducing the robot to a new environment (Figure 1-4b), the user provides salient information regarding the environment through natural language descriptions.

A robot with the ability to use information contained in natural language to augment its spatial-semantic representation will be better able to respond to human instructions. Natural language can provide the robot with semantic information, such as colloquial names of places, that would be difficult to infer from its onboard sensors. It can also provide the robot information about spatial entities and their relationships outside the robot's sensing range. If the robot is able to fuse this information with its sensor observations in a meaningful manner it will be able to learn a more complete representation of its environment.

Natural Language Descriptions as Sensor Observations

This thesis addresses the problem of fusing natural language descriptions of the environment with a robot's onboard sensors to construct human compatible spatial-semantic rep-

resentations useful for interacting with humans. Similar to how a robot's sensors allows it to make observations about the presence of spatial entities, their associated properties and connectivity, natural language descriptions of the environment provided by human users confers information about the environment useful to building a robot's representation. These include information about the existence and semantic properties of places, such as their colloquial names (e.g., "Matt's office," "Kiva conference room"), type of region (e.g., "kitchen", "living room"), the objects that they contain, and the spatial layout of the environment. As such, the key tenet of this thesis is the treatment of natural language descriptions given to the robot as another form of sensor observation. Effective integration of natural language descriptions with robot's sensor stream can be challenging due to the inherent differences in the two sources of information.

Robot sensors observe low-level properties of the robot's local surround. Semantic mapping frameworks typically infer semantic properties such as region type or presence of objects with the use of region appearance models and object detectors respectively. Since the robot typically extends its spatial representation when visiting a particular region in the environment, the observations made by the robot's sensors can be correctly associated with the same location due to the spatial locality of the sensor observations. Coupled with odometry measurements from the robot, spatial connectivity can be established allowing the robot to infer the topology of the environment.

On the other hand, natural language descriptions are provided by humans who may have additional knowledge of the robot's environment. For example, they might be familiar with the spatial layout of the environment or be able to perceive its properties more effectively. Descriptions typically refer to human compatible spatial concepts and are ambiguous with regard to their metric associations. Such descriptions can refer to the robot's current location (*egocentric* descriptions, e.g., "This is the gym") or spatial relations and labels that are associated with non-local, potentially distant regions in the environment (*allocentric* descriptions, e.g., "The exit is next to the cafeteria"). As such, the robot can no longer assume that the referents in the description are the its immediate region, or that it has even observed them (see Figure 1-5). This spatial and temporal disconnect between language descriptions and the robot's sensor observations makes fusing the two sources of

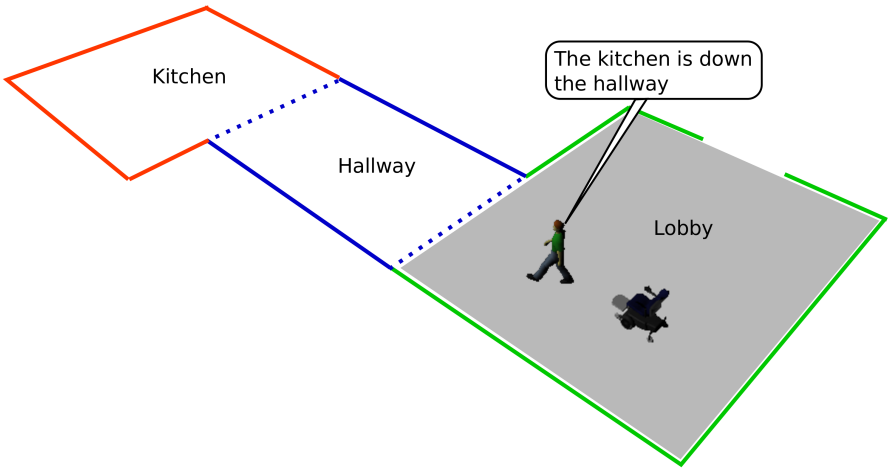


Figure 1-5: Human describes a distant kitchen region that is "down from" the hallway, neither of which have been visited by the robot (robot's visible region is shown in gray)

information challenging.

Proper integration of natural language and robot sensors requires an approach that can correctly identify and associate entities described by the user with the robot's internal representation, or even extend the representation depending on the description. While some existing frameworks for constructing spatial-semantic representations account for certain forms of natural language input, they are either limited to only inferring semantic information about the robot's immediate location [102, 69], or do so in a non-probabilistic manner [100]. The algorithms outlined in this thesis allow robots to learn about semantic properties of the robot's immediate location but also informs it about the presence of spatial entities and their relationships with each other, allowing the robot to learn more complete and accurate spatial-semantic representations.

1.2 Problem Statement

This thesis addresses the problem of fusing information contained in natural language descriptions with the robot's onboard sensors to construct spatial-semantic representations useful for human-robot interaction. To achieve this we tackle three key problems.

First, we address the problem of defining a spatial-semantic representation that is capable of combining a robot's sensor observations with descriptions of the environment provided by a human, and how this representation could be learned efficiently. Second, we address the problem of actively improving a robot's representation of the environment by taking actions, specifically by having the robot interact with a human by asking questions. Third, we address how to use our formulation to enable a robot to respond to natural language navigation commands in unknown environments. We investigate how the robot can use the information contained in natural language instructions to learn a distribution over the environment and then solve a policy given this distribution.

1.3 Thesis Contributions

This thesis makes three key contributions towards addressing the problems outlined in the previous section.

Learning Semantic Maps from Natural Language and Scene Appearance

We introduce the *semantic graph*, a representation that combines metric, topological, and semantic models of the environment, that allows robots to efficiently learn human-centric models of the environment from a narrated guided tour [30], by fusing knowledge inferred from natural language descriptions with conventional low-level sensor data. We outline a probabilistic algorithm (Chapter 3) that efficiently maintains the joint distribution over the semantic, topological and metric maps, conditioned on the language and the metric observations from the robot's proprioceptive and exteroceptive sensors using a Rao-Blackwellized particle filter [15]. We maintain the distribution over topologies using numerical sampling via particles, which are modified using spatial and semantic priors as the robot receives new descriptions and sensor measurements. Figure 1-6 shows a single particle of an environment with five places. We model the likelihood of natural language utterances with the Generalized Grounding Graph (G^3) framework [88] to infer the label

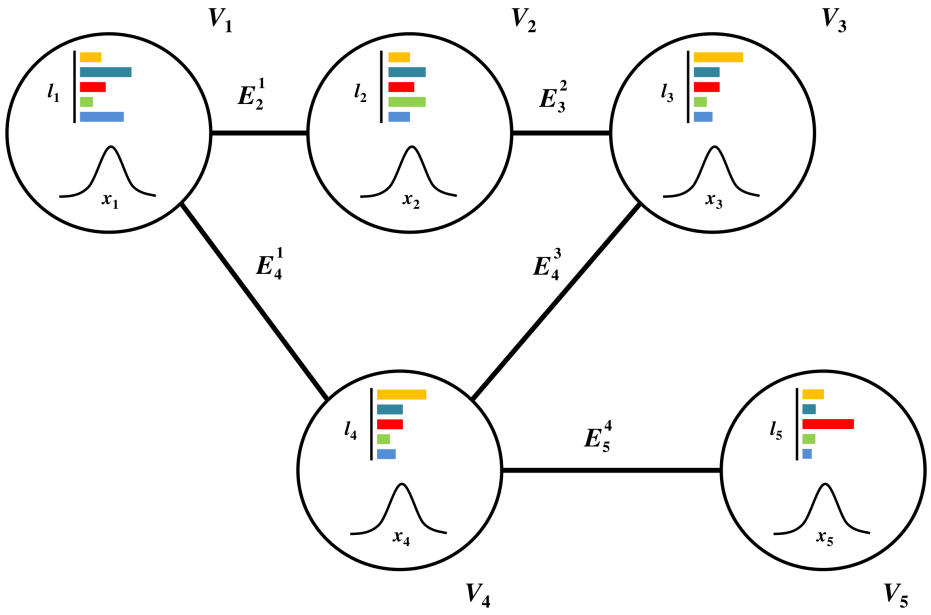
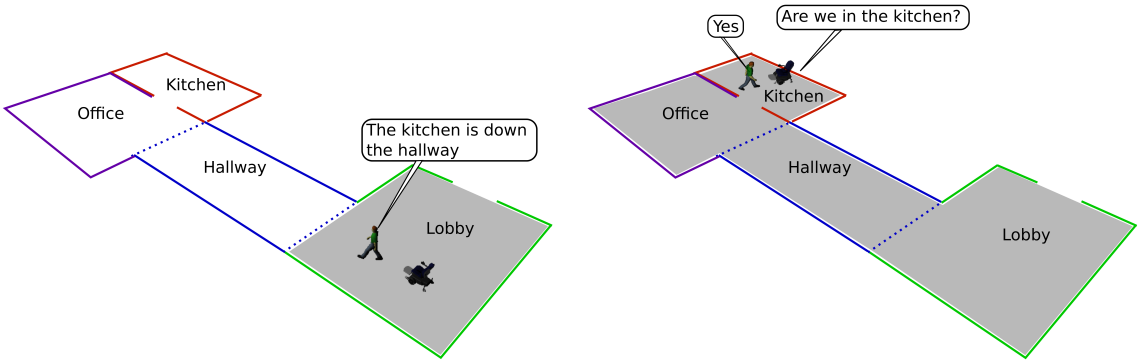


Figure 1-6: Example semantic graph particle: circles denote vertices V_i's and lines denote edges E_i's in the topology, x_i's denote the metric location and l_i's denote the label distribution.

distribution over places in the environment. We also provide mechanisms for handling descriptions that refer to as yet unobserved regions. We evaluate the algorithm's ability to learn human-centric maps of several different environments, and demonstrate its ability to incorporate information from language descriptions to improve the metric, topological and semantic accuracy of the learned environment model.

Next, we extend our algorithm (Chapter 4) to learn semantic models of the environment that reason over more semantic properties, such as the region's type and appearance, by integrating semantic information inferred from the robot's own sensor observations. We use an improved spatial representation that is better reflective of the environment's segmentation and layout, coupled with a richer semantic representation that incorporates information extracted from the natural language descriptions with the robot's own sensors (using laser and camera appearance-based models) maintained as a factor graph. By modeling the relation between an area's type and its colloquial name, the algorithm can reason over both region type and region label, even in the absence of speech. This enables more effective grounding of allocentric user utterances.



(a) (b)

Figure 1-7: How the robot can ask questions to resolve an ambiguous description: (a) The user provides ambiguous description to a robot wheelchair (there are two potential regions down the hallway that could be the kitchen). (b) The robot asks a question to resolve its ambiguous grounding.

Improving Spatial-Semantic Representations

This thesis then outlines an information-theoretic algorithm that enables a robot to improve its spatial-semantic representation of an environment by engaging users in dialog during a guided tour. At each time step, the robot decides between actions that either follow the guide or that ask a targeted question to improve its representation. We formulate the decision process as a QMDP [48], where we evaluate actions as a Markov Decision Process (MDP) for each possible configuration of the world (particle), and select the best action using the QMDP heuristic. By modeling the value of the next state using an information gain heuristic, we bias the algorithm to ask questions that are expected to help improve its understanding of the descriptions provided by the guide. Figure 1-7 shows how the robot reasons over the ambiguity over a language description as it learn about the environment, and asks a question aimed at resolving its confusion. We demonstrate that, by asking deliberate questions of the user, our algorithm results in less ambiguity over the descriptions and semantic maps that are more accurate.

Learning Models for Following Natural Language Instructions in Unknown Environments

Next, we outline how to enable a robot to follow natural language navigation instructionsin completely unknown environments, by using an improved iteration of our semantic mapping algorithm that uses natural language to learn the spatial layout of distant (as yet unobserved) parts of the environment. This semantic mapping algorithm probabilistically extends the robot's representation by creating new spatial entities based on information implicitly contained in the natural language instruction by treating language observations on par with other sensor observations on a symbolic (or semantic) level. We use this distribution over the semantic graph to ground the actions and goals from the command, resulting in a distribution over desired behaviors. We then solve for a policy that yields an action most consistent with the command, under the current map and behavior distributions. As the robot travels and senses new metric information, it updates its map prior and inferred behavior distribution, and continues to plan until it satisfies the instruction (See Figure 1-8). We apply our framework to enable robots to respond to natural language instructions in two settings, firstly to execute free-form instructions that direct a robot to unknown objects [17], and secondly to follow natural language directions in an indoor environment.

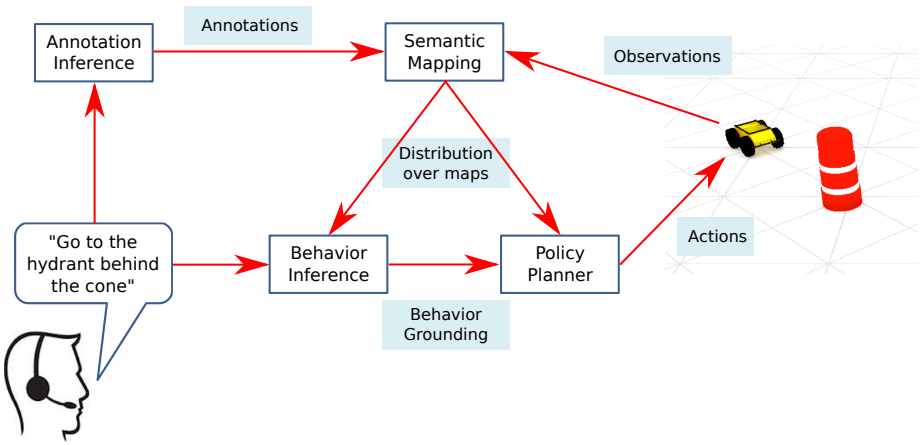


Figure 1-8: Behavior Inference Framework outline.

1.4 Thesis Overview

This section provides an overview of the remainder of the thesis.

Chapter 2 outlines prior work carried out related to constructing semantic maps, taking actions to improve robot representations and enabling robots to respond to natural language commands. The rest of the thesis will explore our contributions in-depth.

Chapter 3 introduces our semantic graph representation and our algorithm for maintaining these semantic maps using natural language descriptions and robot sensors. We use this to learn semantic properties from natural language descriptions and subsequently improve the spatial representations using this knowledge.

Chapter 4 describes several improvements to our approach outlined in the previous chapter that allows us to learn improved spatial representation, and richer semantic models by merging natural language with observations of semantic properties made by the robot's sensors.

Chapter 5 introduces an information theoretic framework that enables the robot to ask questions about spatial entities described to it during a guided tour, allowing it to improve its spatial-semantic representation.

Chapter 6 describes our approach to using natural language descriptions to enable a robot to take effective actions in unknown environments. To enable this behavior, we introduce an enhanced formulation of our spatial-semantic mapping framework that uses natural language descriptions to directly extend the robot's representation. We use our approach to enable robots to follow navigation commands given in natural language, without any prior knowledge of the environment.

Chapter 7 summarizes the contributions of this thesis and potential avenues of future research towards enabling robots to learn from natural language descriptions provided by human partners.

Chapter 2

Related Work

2.1 Learning Spatial-Semantic Representations

Robots require accurate representations of their environment in order to operate effectively. These representations are used to localize in the environment, to navigate and manipulate objects, and to interpret user instructions. Over the last few decades, robotics has tackled the problem of constructing useful spatial representations in both indoor and outdoor environments, resulting in maps that contain metric and topological information [93]. Metric maps capture the locations and geometry of objects in the environment, either in the form of location-based maps or feature-based maps. Location-based maps such as occupancy gridmaps [20] decompose the environment into an evenly spaced grid, where each location in the grid is represented by a binary variable indicating whether the location is occupied or not. Feature-based maps are composed of stable salient features or objects in the environment detectable with the robot's sensors such as lidar (e.g., corner or line features) or cameras (image features such as SIFT [50]) and their metric locations. Topological maps are coarser representations that define the environment as a graph, where vertices represent salient places in the world and edges denote their connectivity. Section 2.1.1 provides an overview of the approaches taken to construct such spatial representations.

More recent efforts have looked at constructing hybrid representations that in addition to maintaining topological and metric information, also capture higher level semantic attributes relevant to and defined by humans who inhabit the environment. These spatial-

semantic maps allow the robot to maintain shared situational awareness with its human partners, and accomplish higher level tasks such as responding to natural language commands that require reasoning about human defined concepts such as rooms, objects and high-level actions. Section 2.1.2 outlines several key semantic mapping frameworks that have contributed to the construction of these hybrid representations.

Our semantic graph algorithm introduced in this thesis also maintains a hybrid representation that jointly models the metric, topological, and semantic properties of the environment. The latter two layers are particularly useful for human-centric mapping as the semantic map models properties useful in grounding natural language commands [88], while the topology is consistent with the representation that humans use to model space [52]. We also maintain a metric representation that denotes the metric locations of each vertex and their associated spatial properties. We also infer occupancy gridmaps from these metric properties using their associated laser scans to enable the robot to navigate in the environment.

2.1.1 Learning Spatial Representations

For a robot operating in an unknown environment, the ability to construct a map of this environment while simultaneously determining its location within this map [16] is of paramount importance. This is known as Simultaneous Localization and Mapping (SLAM), first addressed in the seminal work of Smith and Cheeseman [83].

Metric SLAM Solutions

A majority of the solutions to the SLAM problem have focused on constructing metric maps that can aid robots to localize and navigate accurately in indoor and outdoor environments. Due to the inherent uncertainties in the robot's sensor measurements and odometry, the SLAM problem is often formulated probabilistically in the following manner. Having defined the robot's poses at time steps $i = 0, 1, ..., N$ as $X = \{x_i\}$, the map M as a collection of landmarks $\{m_j\}$, the odometry $u^N = \{u_1, u_2, .., u_N\}$, and sensor observations

$z^K = \{z_1, z_2, .., z_K\}$, the joint probability can be stated as:

$$p(X, M, u^N, z^K) \propto p(x_0) \prod_{i=1}^{N} p(x_i|x_{i-1}, u_i) \prod_{k=1}^{K} p(z_k|x_{i_k}, m_{j_k}), \tag{2.1}$$

where $p(x_0)$ is the prior over the initial pose, $p(x_i|x_{i-1}, u_i)$ is the robot's motion model and $p(z_k|x_{i_k}, l_{j_k})$ is the measurement model with known correspondence between x_{i_k} and the landmark l_{j_k}. Figure 2-1a shows the graphical model for the SLAM formulation.

The SLAM literature typically assumes the motion and measurement models to be Gaussian. Thus the motion model can be defined as:

$$x_i = g_i(x_{i-1}, u_i) + w_i, \tag{2.2}$$

where w_i is a normally distributed zero-mean process noise with a covariance matrix R_i. The measurement model can be defined as:

$$z_k = h_k(x_{i_k}, m_{j_k}) + v_k, \tag{2.3}$$

where v_k is a normally distributed zero-mean measurement noise with a covariance Q_k.

When the robot revisits an area in the environment, it often observes previously seen landmarks, allowing for *loop-closures*, which can be used to overcome accumulated odometry errors. However, identifying correct loop-closures are especially difficult when the robot has traveled a large distance in between.

The SLAM problem [93] can be stated as either the *online SLAM problem*, which involves estimating the posterior over the current pose and the map:

$$p(x_t, M|z^K, u^N), \tag{2.4}$$

or the *full SLAM problem*, which calculates the posterior over the entire robot path X as well as the map:

$$p(X, M|z^K, u^M). \tag{2.5}$$

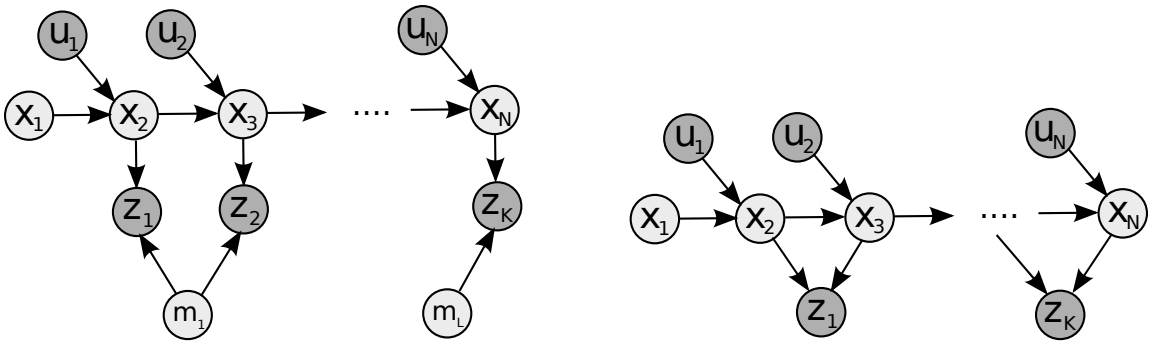


Figure 2-1: Graphical model for SLAM (a) With the map $M = \{m_j\}$ and the robot poses $X = \{x_i\}$ (a) With only robot poses

Solutions ranging from Extended Kalman Filters [47, 13], Rao-Blackwellized Particle Filters [61, 62] and Pose Graph optimizations [51, 90, 38] have been successfully applied to construct large scale metric maps of both indoor and outdoor environments. Many challenges in SLAM have been addressed over the last decade, resulting in very accurate, robust, large-scale mapping frameworks that operate in real-time. Maps resulting from these approaches are useful for a number of tasks, mostly geared toward autonomous navigation by enabling robots to localize themselves, plan feasible paths to goal locations and avoid obstacles.

Our semantic mapping algorithm also maintains the metric properties of the environment using a pose graph formulation, where each pose represents a place in the world, and constraints denote the connectivity between these poses. Unlike EKF based solutions, the pose graph optimization solves the full SLAM problem (2.5), which infers the maximum-likelihood the map as well as all the robot poses up to the present time. Formulating the SLAM problem as a pose graph optimization was first proposed by Lu and Milios [51], who model the problem as a set of constraints between the robot's poses at discrete time steps. Over the years a number of approaches have proposed efficient solutions to this problem in both batch and incremental form.

In pose graph SLAM [66], the measurement function is often encoded as a measurement between two robot poses by scan-matching the robot's laser observations at these

poses:

$$z_k = h_k(x_{i_k}, x_{j_k}) + v_k. \tag{2.6}$$

This results in the joint probability:

$$p(X, u^N, z^K) \propto p(x_0) \prod_{i=1}^{N} p(x_i|x_{i-1}, u_i) \prod_{k=1}^{K} p(z_k|x_{i_k}, x_{j_k}) \tag{2.7}$$

Figure 2-1b shows the graphical model for this formulation. Equation 2.7 is posed as a least squares optimization problem to infer the maximum *a posteriori* (MAP) estimate over the robot's full trajectory X given the robot's odometry U and observations Z. The MAP estimate X^* is obtained by minimizing the negative log likelihood of the joint probability of $p(X, U, Z)$ outlined in Equation 2.1.

$$X^* = \arg\min_X \; -\log p(X, U, Z) \tag{2.8}$$

Using the motion and measurement models this can be expanded to,

$$X^* = \arg\min_X \left\{ \sum_{i=1}^{N} ||g_i(x_{i-1}, u_i) - x_i||^2_{G_i} + \sum_{k=1}^{K} ||h_k(x_{i_k}, x_{j_k}) - z_k||^2_{Q_k} \right\}, \tag{2.9}$$

where $||e||^2_\Sigma = e^T \Sigma^{-1} e$ denotes the Mahalanobis distance for a covariance Σ. For non-linear motion and measurement models, calculating the solution involves linearizing the above equation, thus converting the problem to a standard least-squared minimization problem of the form,

$$\theta^* = \arg\min_\Theta ||A\theta - b||^2. \tag{2.10}$$

This is then solved efficiently using approaches, such as Cholesky or QR factorization, either in batch or incremental form. In our algorithm we make use of iSAM [38], which solves this incrementally to derive the MAP estimate over the robot's poses.

Topological Mapping

However, for robots geared towards operating in human-occupied environments, metric representations can prove limiting due to their lack of high level semantic information, and their incompatibility with the way that humans reason about spaces. Depending on the task at hand, robots operating in these domains require knowledge of semantic properties of their environments, such as the types of region, the colloquial names used to refer to places, the layout of unvisited places, and knowledge about the presence of objects. Topological mapping frameworks address this situation to some degree, by representing the environment as a graph consisting of nodes that denote salient places in the environment and edges that denote their connectivity. The focus of such frameworks is to learn accurate maps of the world that the robot can use to localize and navigate [9, 2, 73]. While humans also reason about spaces using topological representations, the spatial decompositions employed by humans are often incompatible with topologies that are learned from these frameworks. This is due to the fact that locations that are easily distinguishable to the robot's sensors might not correspond to human concepts of rooms or landmarks.

Distributions over Spatial Representations

Some frameworks [45, 94, 22, 73] maintain multiple hypothesis about the spatial layout of the environment. Compared to typical SLAM approaches, these frameworks maintain multiple hypothesis about potential loop-closures in the environment. As such, they are robust to incorrect loop-closures that can occur in highly aliased environments.

Many mapping algorithms build local laser scan patches for each region and correlate these patches to identify loop closures [25]. However, these techniques are prone to perceptual aliasing when the local geometry is not distinctive, such as in the case of hallways. More recent methods consider a region's visual appearance as a more discriminative means of performing place recognition [81, 11, 73]. Of particular note, Cummins and Newman [11] learn a generative model of region appearance using a bag-of-words representation that expresses the commonality of certain features. By effectively modeling this perceptual ambiguity, the authors are able to reject invalid loop closures despite significant aliasing, while

correctly recognizing valid loop closures. This and related approaches in effect choose the maximum likelihood loop closure, relying on the assumption that the place model is sufficiently descriptive that the resulting distribution over the space of loop closures is peaked around the true correspondence. Our approach in Chapter 3 differs in that it uses semantic information to maintain a distribution over the space of loop closures rather than only that which is most likely.

Our formulation is somewhat inspired by Ranganathan and Dellaert [73], who propose Probabilistic Topological Mapping (PTM), a probabilistic representation for constructing topological maps. They make use of a Rao-Blackwellized particle filter formulation that incrementally updates the posterior based on new measurements. However, their focus is on constructing accurate spatial representations, and as such do not maintain any semantic properties of the environment nor integrate any natural language descriptions. Compared to Ranganathan and Dellaert, we are focused on modeling the semantic properties of the environment as well as the spatial layout, and make use of semantic information derived from natural language and robot's sensors to propagate sample topologies. Additionally, they only add new nodes into the topology when the robot visits a new region in the environment, while our approach in Chapter 6 actually extends the topology by adding new spatial entities based on information extracted from natural language descriptions.

2.1.2 Semantic Mapping

Unlike the SLAM problem, semantic mapping [44, 24, 102, 40, 69] addresses the problem of learning a human-compatible spatial-semantic representation of a robot's environment. The spatial representation consists of human salient *places* or regions, their connectivity and associated spatial properties, such as their metric locations and spatial extent. For each spatial entity in the environment, the robot also maintains semantic attributes that are relevant to and defined by humans who inhabit the environment. These semantic attributes can range from the type of each region (e.g., "hallway," or "kitchen"), their colloquial names (e.g., "Mark's office"), the objects that they contain and types of activities that can be carried out at these places (e.g., "eat lunch"). This information is useful for localization and

navigation, but also facilitates human-robot interaction, including more efficient command and control mechanisms, such as natural language understanding.

Early work in semantic mapping includes the Spatial Semantic Hierarchy (SSH) proposed by Kuipers [44] that represents a robot's spatial knowledge as a coupled hierarchy. At the lowest level, the local environment is modeled as a collection of control laws, each expressing the relationship between sensory input and motor output, that facilitate localization and generating local geometric maps. Above the control level is the causal level, which provides a discrete model of the actions that transition between each of the control laws. The topological level represents the environment as a collection of regions, places, and paths that abstract states and actions from the causal level. While the topology serves as the primary global map of the environment, the local geometric maps from the control level can be merged via the topology to formulate a global metric map.

Kuipers et al. [45] describe an extension to the SSH that employs a hybrid metric and topological representation to better represent environments at both small and large scales. The Hybrid SSH treats the environment as a collection of interconnected locations, each being small in scale. The method employs metric maps to model the local geometry of distinct regions from which they use local paths to induce a symbolic global topology that describes the large-scale environment. By decoupling the map in this manner, this approach more efficiently models ambiguities in large-scale loop closures with multiple compact topologies, without requiring that the set of local metric maps be registered consistently in a single global reference frame. This is a distinct benefit over submap approaches to SLAM [46, 5] that similarly employ local metric maps but also seek to ensure that these submaps are consistent in a global reference frame. The authors have shown [60, 3] that the representation allows uncertainties to be handled more effectively by factoring them into individual components that capture local metrical, global topological, and globally metrical uncertainties.

The semantic mapping algorithms outlined in this thesis also consists of a hybrid metric and topological representation and factors the joint distribution into separate metrical and topological terms, employing different hypotheses over the topological map to represent the distribution over the space of loop closures. However, we maintain a globally metric

map of the environment with respect to a single frame of reference, which can make our algorithm sensitive to global inconsistencies within large environments. Unlike our approach, however, the Hybrid SSH does not model the semantic labels or colloquial names associated with different regions of the environment.

More recent efforts similarly take a hierarchical approach to representing semantic and spatial properties of a robot's environment. Many existing solutions [24, 55, 59, 96, 42, 102, 30, 69] build on the effectiveness of SLAM by augmenting a low-level metric map with layers that encode the topological and semantic properties of the environment. Typically, an off-the-shelf SLAM implementation is used to build the metric layer. One level up in the hierarchy is the topological map, taking the form of a graph, where vertices denote different *places* in the environment and edges model their connectivity. Layered above the topology is the semantic map that represents abstract properties associated with each place, such as their type or the objects that they contain. In prior work, the flow of information is strictly bottom-up, where the metric layer is used to infer a fixed topology, which is then augmented with semantic information obtained from the robot's sensors. Our formulation, on the other hand, has a tight coupling between the spatial and semantic layers whereby new information about semantic properties of the world leads to an improved spatial representation (through new updates to the topology) as well as the other way around.

Galindo et al. [24] presents a multi-hierarchical approach to semantic mapping, where they maintain two hierarchies, namely spatial and semantic. Relationships between the spatial and semantic information are represented using anchoring, which connects the symbolic representations with their corresponding spatial entities. The spatial hierarchy is composed of local metric maps and camera images and the topology of the environment. The semantic layer is constructed using standard AI languages, allowing the robot to perform symbolic reasoning. However, there is no probabilistic representation of either hierarchy, and the presence of objects remains the sole source of semantic information.

Zender et al. [102] introduce a similar framework that maintains a multi-layered representation, consisting of a metric map that maintains the geometric structure of the environment using line features, a navigation map that denotes free space and connectivity, a topological map that clusters the free space into areas separated by doorways, and a concep-

tual map that maintains an ontology of abstract concepts and their relationships and their corresponding instances in the topology. Semantic information is inferred using object detection, region type classification (room vs. hallway), and situated linguistic assertions provided by a human during a guided tour phase. However, their framework is limited to handling linguistic descriptions about the immediate space around the robot. They also do not maintain a probabilistic representation of the spatial or semantic layers.

Pronobis and Jensfelt [69] propose an approach that combines a multi-layered spatial representation with a probabilistic semantic representation using factor graphs. They utilize a set of appearance-based classifiers and object detectors coupled with learned models of relationships between types of regions and their appearance and the presence of objects. Their approach is also limited to handle language descriptions of the robot's immediate environment. Their framework infers the topology from the metric layer and as such does not maintain a distribution over the topology.

The semantic mapping algorithms outlined in this thesis differ from the existing state-of-the-art in several fundamental ways. We employ a learned model of free-form utterances to reason over expressions that are less constrained than those handled by other methods. To be precise, we assume that these descriptions involve labels for and spatial relations between one or two locations, though the structure of these expressions is only limited by rules of grammar and the amount of training data. Existing methods that use natural language information are mostly limited to learning the labels associated with the robot's immediate location. Our algorithms handle descriptions that refer to regions outside the robot's immediate location that may even be unobserved by the robot at the time of the description. In Chapters 3 and 4, we use these descriptions to learn the labels of these regions, while in Chapter 6 we use these to learn the spatial layout of as yet unobserved regions. Existing methods allow updates to the metric layer to influence the topological and semantic layers, but do not use information in the semantic layer improve the rest of the hierarchy. By maintaining a joint distribution over the metric, topological, and semantic properties of the environment, our algorithms use updates to any one layer to improve the other layers in the hierarchy. For example, we show how the semantic layer can be used to recognize loop closures, a fundamental problem in SLAM, and thereby add edges to the

topology that, in turn, correct errors in the metric map.

Spatial Segmentation Strategies

Semantic mapping frameworks, and topological mapping in general employ a number of strategies to segment the environment into different spatial entities. Thrun et al. [91], for example, rely on a user to push a button each time the robot transitions to a new region. However, such methods might not be feasible in some scenarios and would prove cumbersome to any human partner. A straightforward automatic strategy is to segment regions based upon distance, placing vertices at a fixed spacing as the robot travels in the environment, which is the method that we take in the work described in Chapter 3. However, this method is not reflective of how humans model spaces. An alternative is to use heuristics, such as door detections, to separate regions [102, 69], which yields segmentations that can be more semantically meaningful within indoor environments. However, while doorways provide a robust signal that the two regions should be separate, door detectors can have false positives and negatives, and additionally different regions are not always separated by doorways. Meanwhile others have segmented regions based upon geometric [6, 4], visual [71] or semantic similarity [55, 96]. Of particular relevance to this work, Ranganathan and Dellaert [73] explore multiple methods to define regions, including manual segmentation at the location of gateways (e.g., doorways, junctions) and automatic segmentation based upon changes in visual appearance [72]. Our approach outlined in Chapter 4 makes use of both door-detection and a laser-based spectral clustering method [4] to infer spatial segmentations.

Sources of Semantic Information

The properties contained within the semantic map are most often inferred from the robot's sensor data (e.g., lidar scans and camera images), using appearance-based classifiers [55, 68] and object detectors [95, 40]. For example, Martínez Mozos et al. [55] use a combination of boosted laser range features and image-based object detections to classify the robot's surround as it navigates, and show how this can be used to induce a topology for the environment. Similarly, Meger et al. [59] layer a visual attention system and image-based

Figure 2-2: A salient landmark, such as this question mark, which indicates the location of the information desk in the Stata Center at MIT, would be difficult to detect using general-purpose object detectors.

object recognition on top of a SLAM occupancy grid map to build semantic maps that encode the locations of objects of interest within the environment. Vasudevan and Siegwart [96] describe a probabilistic framework that uses clustered object detections to learn conceptual models of space that express their hierarchical structure (e.g., that an "office" may include a "workspace" and "meeting area") and the objects that they contain. They argue that this model is amenable to a hierarchical metric-topological-semantic SLAM framework, though they leave that for future work.

These solutions rely upon scene classifiers and object detectors to infer the properties that make up the semantic map. Some use non-probabilistic AI reasoning methods [24, 102], that capture relationships between objects and their presence in certain types

of regions (e.g., finding a microwave in a kitchen). Others [40, 68] use probabilistic methods that make use of learned relationships between objects and spatial regions. The effectiveness of these approaches is a function of the richness of the training data. As such, they perform best when the environments have similar appearance and regular geometry, and when the objects are drawn from a common set. Even in structured settings, it is not uncommon for the regions to be irregular and for the objects to be difficult to recognize, either because they are out of context or are singletons (Figure 2-2). Furthermore, scene classification does not provide a means to infer the specific labels that humans use for a location, such as "Mark's office" or the "Kiva conference room."

Several semantic mapping frameworks are able to learn from natural language descriptions of the environment provided by a human. Kruijff et al. [43] allows for assertions about the immediate location as well as the presence of doorways that they use to aid segmentation. Zender et al. [102] and Pronobis and Jensfelt [69] also integrate descriptions of the robot's immediate surround into the semantic representation. However, use of language is limited to inferring semantic properties of the robot's immediate surround. Williams et al. [100] propose a framework that handles more complicated language descriptions, where they extend the robot's representation, by adding new (unknown) places based on language. However, they do not maintain multiple hypotheses about the space nor do they have any probabilistic reasoning regarding the grounding of language to the map. Their evaluation is also limited to simulation environments.

Our approach outlined in Chapter 3 relies on natural language descriptions as the only source of semantic information. However, unlike prior approaches that use natural language, we are able to integrate semantic information from *allocentric* descriptions that reference areas outside the robot's immediate surroundings as well as *egocentric* descriptions that refer to the robot's immediate surround. We extend this in Chapter 4 to make use of appearance-based classifiers trained with geometric features [55] and image features [68] in addition to language descriptions. While the algorithms in Chapters 3 and 4 only integrate this semantic information once the robot observes the relevant region, in Chapter 6 we use language to reason about the presence and locations of regions and objects that are yet to be observed.

Acquiring Representations of the Environment

There is a spectrum of methods for robots to acquire representations of their environments. On one side of the spectrum lies purely passive approaches where the robot is manually driven around the environment, and a map is constructed either online or offline, and possibly annotated with high level information by an expert. While effective at acquiring maps and even integrating high level information, such approaches are hard to scale to large deployments especially with novice users. Additionally, certain platforms, such as a quad-rotor or a robotic forklift, might require an expert operator. On the other end of the spectrum lies fully autonomous approaches [101, 92, 86], where the robot uses an exploration strategy (e.g., frontier-based exploration) to fully explore its environment. However, such strategies can fail if parts of the environment are inaccessible without human intervention (e.g., separated by a closed door) and any high-level information that is integrated to the representation have to be inferred by the robot using its own sensors.

Semi-supervised methods [43, 102, 30] that fall in between these approaches combine the advantages of both approaches. In such methods, a human conducts a narrated guided tour of the new environment, describing salient locations and objects verbally as shown in Figure 1-4b. The robot follows the guide through an environment, interpreting his spoken utterances and the shared spatio-temporal context. These methods allow even non-expert users to assist robots in building better models of the environments. This also allows the human to impart knowledge about the environment using natural language, which can be integrated to the robot's representation. Additionally, the robot can also carry out dialog interaction with the guide to improve its representations. Our approaches in Chapters 3, 4 and 5 are modeled on this guided tour concept, which allows the robot to receive a stream of sensor observations and language descriptions. We use the algorithm outlined in Hemachandra et al. [30] to enable the tour. This uses a laser-based person tracker coupled with a socially-acceptable person following method to follow the guide through the environment.

2.2 Understanding Natural Language Utterances

There have been a number of efforts aimed at solving what Harnad [26] refers to as the symbol grounding problem, the problem of mapping linguistic elements to their corresponding manifestation in the external world, such as objects, spatial entities and actions. In the robotics domain, the grounding problem has primarily been addressed in the context of following route directions and other natural language commands.

Inferring a behavior such as a desired robot trajectory from natural language commands can be formally defined as follows, where $\boldsymbol{x(t)}$ is the required robot trajectory, Λ is the language command and M is the world model:

$$\arg\max_{\boldsymbol{x(t)}} p(\boldsymbol{x(t)}|\Lambda, M) \tag{2.11}$$

One class of solutions [82, 53, 19, 8, 58, 57] treats grounding as one of parsing free-form commands into a formal control language equivalent, which a planner takes as input. One such algorithm for following natural language directions by Matuszek et al. [57] solves the grounding problem by learning a semantic parsing model that is used to define a distribution over possible control sequences defined using a robot control language.

Another class of solutions [41, 88, 89, 34] function by mapping free-form utterances into their corresponding object and action referents in the robot's world model. These approaches infer the maximum likelihood groundings (Γ) given language and the world model using probabilistic models.

$$\arg\max_{\Gamma} p(\Gamma|\Lambda, M) \tag{2.12}$$

Groundings can be actions, paths, objects or locations in the robot's world model. One such work by Tellex et al. [88] introduces the Generalized Grounding Graph (G^3), which is a probabilistic graphical model with random variables representing linguistic components and groundings in the world. Our work uses G^3 to evaluate the groundings given language descriptions to infer semantic properties. Howard et al. [34] introduce the Distributed Correspondence Graph (DCG), which is a more efficient formulation that is used to infer the

most likely set of planning constraints given the language and the world model. Unlike the G^3 formulation which requires searching over sampled paths in the environment, DCG searches over the planning constraints that satisfy the command. In Chapter 6, we use a hierarchical formulation of DCG [33] to infer spatial-semantic information contained in natural language instructions and infer behaviors based on the distribution over the maps and language.

2.2.1 Following Natural Language Commands in Unknown Environments

With few exceptions, most techniques for understanding natural language commands require *a priori* knowledge of location, geometry, colloquial name, and type of all objects and regions within the environment [41, 34, 88]. Without known world models, however, interpreting free-form commands becomes much more difficult. Existing methods have dealt with this by learning a parser that maps the natural language command directly to plans [53, 8, 57]. Alternatively, Duvallet et al. [18] use imitation learning to train a policy that reasons about uncertainty in the grounding and that is able to backtrack as necessary. However, none of these approaches explicitly utilize the knowledge that the instruction conveys to influence their models of the environment, nor do they reason about its uncertainty.

In contrast, our approach outlined in Chapter 6 uses natural language information to generate a prior over the possible configurations of landmarks by exploiting the information implicitly contained in a given instruction. Information inferred from natural language is used to create new spatial entities in the topology but also infer weak metric properties based on spatial relationships. Unlike Williams et al. [100] who also use natural language information to add new places to their map, our approach reasons about multiple possible configurations of the world and also reasons about metric information inferred by the descriptions.

As we reason in the space of distributions over possible environments, we draw from strategies in the belief-space planning literature. Most importantly, we represent our belief

using samples from the distribution, similar to work by Platt et al. [67]. The problem of learning a policy in partially observable environments can be modeled as a Partially-Observable Markov Decision Process (POMDP) [85]. Approximate solutions to POMDP problems include QMDP [48, 78], which assume the ability to fully observe the state after one time step.

2.3 Taking Actions to Improving Representations

A number of efforts have endowed robots with the ability to take actions to improve their representations or to better respond to human commands, either by taking physical actions by exploring unknown regions in the environment during exploration [54, 86] or using dialog to interact with human partners [43, 12]. Chapter 5 introduces an algorithm for asking questions to improve the robot's spatial-semantic representation during a guided tour.

Several works have address the case where robots autonomously learn maps of their environment by taking physical exploration actions. Makarenko et al. [54] decide on the best action by defining utility values for making new observations, navigation and localizability for each potential exploration action, to build feature based maps. Stachniss et al. [86] make use of the expected information gain and the cost of each action to decide the best exploration-based motion action to reduce the expected entropy over the map. While we do not consider physical exploration actions, we use somewhat similar expected information gain based reasoning to decide on the best question to ask.

For robot systems that interact with human partners, question asking is another form of interactions that can help improve their environment representations [43], aid in learning new activities [76], better interpret a user's commands [12, 14]. Kruijff et al. [43] address this in the context of giving the robot a tour of its environment, where the robot asks clarification questions from its human partner. These questions seek to learn about room segmentation by asking about the existence of doorways, resolve inconsistencies between the robot's understanding of its surroundings and human assertions, and localization failures. However, they do not maintain a probabilistic spatial-semantic representation, nor do

they carry out any probabilistic reasoning about receiving a particular answer.

More recently, Deits et al. [12] have looked at question asking from an information-theoretic perspective for following natural language manipulation commands. They use an information gain-based reasoning method to evaluate the best questions to ask in order to reduce the entropy over the grounding for a given command. Their approach considers a larger space of questions, ranging from yes/no questions seeking to confirm a correspondence between a referent in the command and an entity in the environment (e.g., "Is this the tire pallet?"), targeted questions that expect an open ended answer (e.g. "What does 'the box' refer to?") and reset questions which asks the user to restate the command. Our approach in Chapter 5 only deals with yes/no type questions but questions may not necessarily reflect only a particular grounding. Deits et al. [12] deal with resolving ambiguity in a given natural language instruction for a known map, while our approach deals with reducing ambiguity of natural language instructions as the robot learns the map of the world. They do not need to reason over when to ask the questions, since they immediately follow the corresponding command. However, in our approach, as the robot needs to ask questions during the guided tour, it also needs to reason about when it is useful to asks a question.

While our approach outlined in Chapter 5 also uses an information gain metric similar to Stachniss et al. [86], and Deits et al. [12], we formulate the problem as a decision problem, where the robot has to decide between continuing the tour or interrupting the tour to ask a question. In our case, a question can simultaneously refer to areas that the user described at distant points in time. This necessitates that we consider when it is most meaningful to ask the question and that it be phrased in a manner that provides sufficient context. As the robot maintains multiple hypothesis over the world, we use a QMDP formulation [48, 78] to solve for the best one-step action.

y

Chapter 3

Learning Semantic Maps from Natural Language Descriptions

This chapter outlines our semantic mapping formulation described in [98, 99] that enables robots to efficiently learn human-centric models of the environment from a narrated, guided tour (Figure 3-1) by fusing knowledge inferred from natural language descriptions with conventional low-level sensor data. Our method allows people to convey meaningful concepts, including semantic labels and relations for both local and distant regions of the environment, simply by speaking to the robot. The advantage is that the robot can learn concepts that people are arguably better-able to convey from its opportunistic interaction with humans. The challenge lies in effectively combining these noisy, disparate sources of information. A user's descriptions convey concepts (e.g., "the second room on the right") that are ambiguous with regard to their metric associations: they may refer to the region that the robot currently occupies, to more distant parts of the environment, or even to aspects of the environment that the robot will never observe. In contrast, the sensors that robots commonly employ for mapping, such as cameras and lidars, yield metric observations arising only from the robot's immediate surroundings.

To handle ambiguity, we propose a representation referred to as the *semantic graph* that combines metric, topological, and semantic models of the environment. The topological layer consists of a graph in which vertices correspond to reachable regions of the environment, and edges denote pairwise spatial relations. The metric layer takes the form of a

Figure 3-1: A user gives a tour to a robotic wheelchair designed to assist residents in a long-term care facility.

vector of poses for each region in the environment together with the resulting occupancy-grid map that captures the perceived structure. The semantic layer contains the labels with which people refer to these regions. This knowledge representation is well-suited to fusing concepts from a user's descriptions with the robot's metric observations of its surroundings.

We estimate a joint distribution over the semantic, topological and metric maps, conditioned on the language and the metric observations from the robot's proprioceptive and exteroceptive sensors. The space of semantic graphs, however, increases combinatorially with the size of the environment. We use a Rao-Blackwellized particle filter [15] to efficiently maintain the factored form of the joint distribution over semantic graphs. Specifically, we approximate the marginal over the space of topologies with a set of particles, and analytically model conditional distributions over metric and semantic maps as Gaussian and Dirichlet, respectively. The algorithm updates these distributions iteratively over time using descriptions and sensor measurements as they arrive. We model the likelihood of natural language utterances using the Generalized Grounding Graph (G^3) framework [88]. Given a description, the G^3 model induces a learned distribution over semantic labels for the vertices in the semantic graph that we then use to update the Dirichlet distribution. The algorithm uses the resulting semantic distribution to propose modifications to the graph,

allowing semantic information to influence the metric and topological layers.

The approach outlined in this chapter was the result of joint work with Matthew R. Walter and Stefanie Tellex.

3.1 The Semantic Graph Algorithm

This section presents our approach to maintaining a distribution over semantic graphs, our environment representation that consists jointly of metric, topological, and semantic maps. The metric map models information contained in the robot's low-level sensor readings. The topological map models the connectivity between regions that can be inferred from navigation as well as natural language descriptions. The semantic map represents categories that the user conveys.

3.1.1 Semantic Graph Representation

We model the environment as a set of *places*, regions in the environment a fixed distance apart[1] that the robot has visited. We represent each place by its pose $\boldsymbol{x_i}$ in a global reference frame and a label l_i (e.g., "gym," "hallway"). More formally, we represent the environment by the tuple $\{G_t, X_t, L_t\}$ that constitutes the semantic graph S_t. The graph $G_t = (V_t, E_t)$ denotes the environment topology with a vertex $V_t = \{v_1, v_2, \ldots, v_t\}$ for each place that the robot has visited, and undirected edges E_t that signify observed relations between vertices, based on metric or semantic information. The vector $X_t = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_t}]$ encodes the pose associated with each vertex. The set $L_t = \{l_1, l_2, \ldots, l_t\}$ includes the semantic label l_i associated with each vertex. The semantic graph grows as the robot moves through the environment. Our method adds a new vertex v_{t+1} to the topology after the robot travels a specified distance, and augments the vector of poses and collection of labels with the corresponding pose $\boldsymbol{x_{t+1}}$ and labels l_{t+1}, respectively. This model resembles the pose graph representation commonly employed by SLAM solutions [38]. Figure 3-2 shows an example semantic graph.

[1] We use 5 m spacing for the results presented in this section.

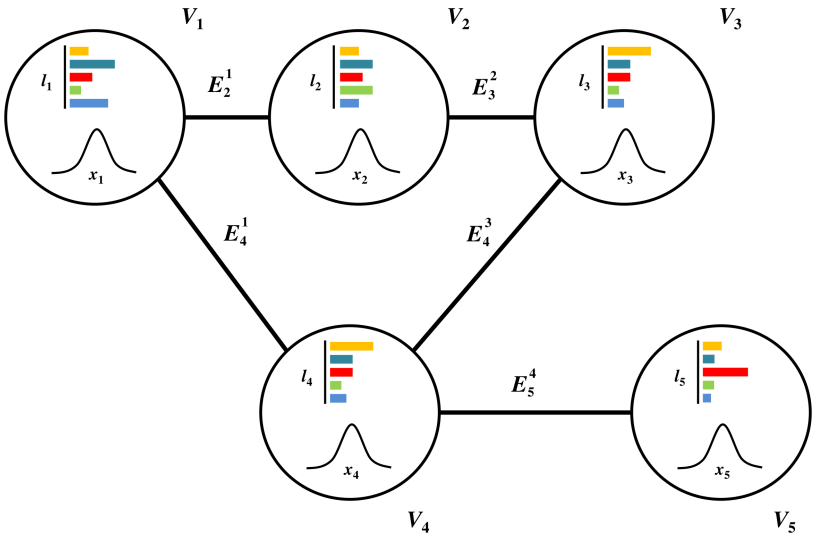


Figure 3-2: Example semantic graph particle: circles denote vertices V_i's and lines denote edges E_i's in the topology, x_i's denote the metric location and l_i's denote the label distribution.

Our goal is to induce a distribution over the semantic graph, including the locations, topology, and semantic labels given information about an environment obtained from a robot's range sensors, odometry readings, and the user's descriptions of the environment.

3.1.2 Distribution Over Semantic Graphs

We estimate a joint distribution over the topology G_t, the vector of locations X_t, and the set of labels L_t. Formally, we maintain this distribution over semantic graphs $\{G_t, X_t, L_t\}$ at time t conditioned upon the history of metric exteroceptive sensor data $z^t = \{z_1, z_2, \ldots, z_t\}$, odometry $u^t = \{u_1, u_2, \ldots, u_t\}$, and natural language descriptions $\Lambda^t = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_t\}$:

$$p(G_t, X_t, L_t | z^t, u^t, \Lambda^t). \tag{3.1}$$

Each language variable Λ_i denotes a (possibly null) utterance, such as "This is the kitchen," or "The gym is down the hall." Table 3.1 outlines our notation. We factor the joint posterior into a distribution over the graphs and a conditional distribution over the

Table 3.1: Semantic Graph Notation

Symbol	Description
$S_t = \{G_t, X_t, L_t\}$	Semantic Graph that combines the topological G_t, metric X_t, and semantic L_t representations.
$G_t = (V_t, E_t)$	Graph representation of the topology at time t that consists of a set of vertices $V_t = \{v_1, v_2, \ldots, v_t\}$ connected by undirected edges E_t.
L_t	Semantic information in the form of labels $l_{t,j}$ associated with each place v_j at time t.
$l_{t,j}^{(i)}$	Label distribution for vertex j in particle i at time t.
Λ_t	Parsed natural language description of the environment at time t.
X_t	Vector of landmark poses $[\boldsymbol{x_1}, ..., \boldsymbol{x_t}]$ at time t
z_t	Observations made at time t by sensors onboard the robot.
u_t	Odometry reading at time t.

vertex poses and labels,

$$p(G_t, X_t, L_t | z^t, u^t, \Lambda^t) = p(L_t | X_t, G_t, z^t, u^t, \Lambda^t) \times p(X_t | G_t, z^t, u^t, \Lambda^t) \times p(G_t | z^t, u^t, \Lambda^t). \quad (3.2)$$

The left-most expression in this factorization explicitly models the dependence of the labels on the topology and the location of each region. The middle term encodes the conditional distribution over the metric map given the topology and, in this way, mimics pose graph formulations to SLAM, given the loop closure (i.e., the topology). The right-most expression denotes the distribution over the graph conditioned upon the sensor history and language.

3.1.3 Space of Potential Topologies

In our representation of the topology, when the robot revisits a region v_i, a new vertex v_j is still created, and an edge e_{ij} is added to denote that it is back in the same space. As such when considering the distribution over the space of topologies, the number of vertices

are deterministic. Therefore the space of possible graphs for that environment is spanned by the potential allocation of edges between the vertices. The space of potential edges, however, is exponential in the number of vertices. Hence, maintaining the full distribution over graphs is intractable for all but trivially small environments. To overcome this complexity, we assume as in Ranganathan and Dellaert [73] that the distribution over graphs is dominated by a small subset of topologies consistent with the robot's observations while the likelihood associated with the majority of topologies is nearly zero. In general, this assumption holds when the environment structure (e.g., indoor, man-made) or the robot motion (e.g., exploration) limits connectivity [73]. In addition, conditioning the graph on language descriptions results in a more peaked distribution, further increasing the validity of this assumption. This is due to the low-likelihood of topologies that contain edges between nodes with inconsistent semantic information.

3.1.4 Maintaining the Posterior over the Semantic Graph

The assumption that the distribution is concentrated around a limited set of topologies suggests the use of particle-based methods to represent the posterior over graphs, $p(G_t|z^t, u^t, \Lambda^t)$. Inspired by the derivation of Ranganathan and Dellaert [73] for topological SLAM, we employ Rao-Blackwellization to model the factored formulation (3.2), whereby we accompany the sample-based distribution over graphs with analytic representations for the conditional posteriors over the vertex locations and labels. Specifically, we represent the posterior over the vertex poses $p(X_t|G_t, z^t, u^t, \Lambda^t)$ by a Gaussian, which we parametrize in the canonical form. We maintain a Dirichlet distribution that models the posterior distribution over the set of vertex labels $p(L_t|X_t, G_t, z^t, u^t, \Lambda^t)$.

We represent the distribution over the semantic graph as a set of particles

$$S_t = \{S_t^{(1)}, S_t^{(2)}, \ldots, S_t^{(n)}\}. \tag{3.3}$$

Each particle $S_t^{(i)} \in S_t$ consists of the set

$$S_t^{(i)} = \left\{G_t^{(i)}, X_t^{(i)}, L_t^{(i)}, w_t^{(i)}\right\}, \tag{3.4}$$

Algorithm 1: Semantic Mapping Algorithm

Input: $S_{t-1} = \left\{ S_{t-1}^{(i)} \right\}$, and (u_t, z_t, Λ_t), where $S_{t-1}^{(i)} = \left\{ G_{t-1}^{(i)}, X_{t-1}^{(i)}, L_{t-1}^{(i)}, w_{t-1}^{(i)} \right\}$

Output: $S_t = \left\{ S_t^{(i)} \right\}$

for $i = 1$ *to* n **do**

1. Employ proposal distribution $p(G_t | G_{t-1}^{(i)}, z^{t-1}, u^t, \Lambda^t)$ to propagate the graph sample $G_{t-1}^{(i)}$ according to u_t and current distributions over $L_{t-1}^{(i)}$ and $X_{t-1}^{(i)}$.
2. Update the Gaussian distribution over the vertex poses $X_t^{(i)}$ according to the constraints induced by the newly-added graph edges.
3. Update the Dirichlet distribution over the current and adjacent vertices $L_t^{(i)}$ according to the language Λ_t.
4. Compute the new particle weight $w_t^{(i)}$ based upon the previous weight $w_{t-1}^{(i)}$ and the metric data z_t.

end

Normalize weights and resample if needed.

where $G_t^{(i)}$ denotes the i'th sample in the space of graphs; $X_t^{(i)}$ is the analytic distribution over locations; $L_t^{(i)}$ is the analytic distribution over labels; and $w_t^{(i)}$ is the weight of particle i.

3.2 Building Semantic Maps with Language

Algorithm 1 outlines the process by which we recursively update the distribution over semantic graphs (3.2) to reflect the latest robot motion, metric sensor data, and utterances. In the first step, we propagate each sample $G_{t-1}^{(i)}$, which represents the posterior $p(G_{t-1} | z^{t-1}, u^{t-1}, \Lambda^{t-1})$ at time $t-1$, by adding a vertex for the robot's new pose (connected by an edge to the previous vertex) and sampling modifications to the topology in the form of additional edges according to the current metric and label distributions. This results in a sample-based estimate for the prior at time t, $p(G_t | z^{t-1}, u^t, \Lambda^t)$. Next, we update the Gaussian distribution over the vertex poses by incorporating the constraints induced by the new loop-closure edges. We then proceed to update the Dirichlet distributions based

upon the structure of the graph and parsed language Λ_t, if available. Finally, we update the weight $w_t^{(i)}$ according to the likelihood of new metric measurements z_t and resample if needed. We repeat these steps for each particle, yielding the particle set representation S_t of the new posterior distribution at time t, $p(G_t, X_t, L_t | z^t, u^t, \Lambda^t)$. The following subsections explain each step in detail.

3.2.1 Graph Augmentation using the Proposal Distribution

Given the posterior distribution over the semantic graph at time $t-1$, we first compute the prior distribution over the graph G_t. We do so by sampling from a proposal distribution that is the predictive prior of the current graph given the previous graph and sensor data, and the recent odometry and language:

$$G_t^{(i)} \sim p(G_t | G_{t-1}^{(i)}, z^{t-1}, u^t, \Lambda^t) \tag{3.5}$$

We formulate the proposal distribution by first augmenting the graph to reflect the robot's motion. Specifically, we add a vertex v_t to the graph that corresponds to the robot's current pose with an edge to the previous vertex v_{t-1} that represents the temporal constraint between the two poses based on the robot's motion. We denote this intermediate graph as $G_t^{-(i)}$. Similarly, we add the new pose as predicted by the robot's motion model to the vector of poses $X_t^{-(i)}$ (according to the process outlined in Subsection 3.2.2) and the vertex's label to the label vector $L_t^{-(i)}$ (according to the process described in Subsection 3.2.3[2]). Thus the new proposal distribution, which is conditioned on $G_t^{-(i)}$ is,

$$p(G_t | G_t^{-(i)}, z^{t-1}, u^t, \Lambda^t). \tag{3.6}$$

Sampling Graph Modifications

We sample the new graph particle $G_t^{(i)}$ by sampling a set of modifications ΔG_t to the graph conditioned on the intermediate graph $G_t^{-(i)}$. In this work the modifications to the graph are in the form of spatial constraints in the form of edges to the graph and corresponding

[2]The label update explains the presence of the latest language Λ_t.

metric constraints.

$$p(G_t|G_t^{-(i)}, z^{t-1}, u^t, \Lambda^t) = p(\Delta G_t|G_t^{-(i)}, z^{t-1}, u^t, \Lambda^t) \tag{3.7}$$

We formulate the proposal distribution (3.7) in terms of the likelihood of adding edges between vertices in this modified graph $G_t^{-(i)}$. The system considers two forms of edges: first, those suggested by the spatial distribution of vertices and second, by the semantic distribution for each vertex.

Spatial Distribution-based Constraints

We first sample edges between the robot's current vertex v_t and others in the graph based on a spatially biased proposal distribution as shown in Equation 3.8, where G_t^{tj} denotes a graph edge between the vertex v_t and v_j. Equation 3.8a reflects the assumption that additional edges expressing constraints involving the current vertex $e_{tj} \notin E^-$ are conditionally independent. This proposal distribution reflects the fact that vertices close in metric space are more likely to have an edge between them. We achieve this by marginalizing over the distances d_{tj} between vertex pairs, as shown in Equation 3.8c, where we omit the history of language observations Λ^t, metric measurements z^{t-1}, and odometry u^t for brevity. Equation 3.8c approximates the marginal in terms of the distance between the two vertices associated with the additional edge.

$$p_a(G_t|G_t^{-(i)}, z^{t-1}, u^t, \Lambda^t) = \prod_{j:e_{tj}\notin E^-} p(G_t^{tj}|G_t^{-(i)}) \tag{3.8a}$$

$$= \prod_{j:e_{tj}\notin E^-} \int_{X_t^-} p(G_t^{tj}|X_t^{-(i)}, G_t^{-(i)}) p(X_t^{-(i)}|G_t^{-(i)}) \tag{3.8b}$$

$$\approx \prod_{j:e_{tj}\notin E^-} \int_{d_{tj}^{(i)}} p(G_t^{tj}|d_{tj}^{(i)}, G_t^{-(i)}) p(d_{tj}^{(i)}|G_t^{-(i)}), \tag{3.8c}$$

The conditional distribution $p(G_t^{tj}|d_{tj}^{(i)}, G_t^{-(i)}, z^{t-1}, u^t, \Lambda^t)$ expresses the likelihood of having an edge between vertices v_t and v_j based upon their spatial location. We represent the distribution for a particular edge between vertices v_i and v_j a distance $d_{ij} = |\boldsymbol{x_i} - \boldsymbol{x_j}|_2$

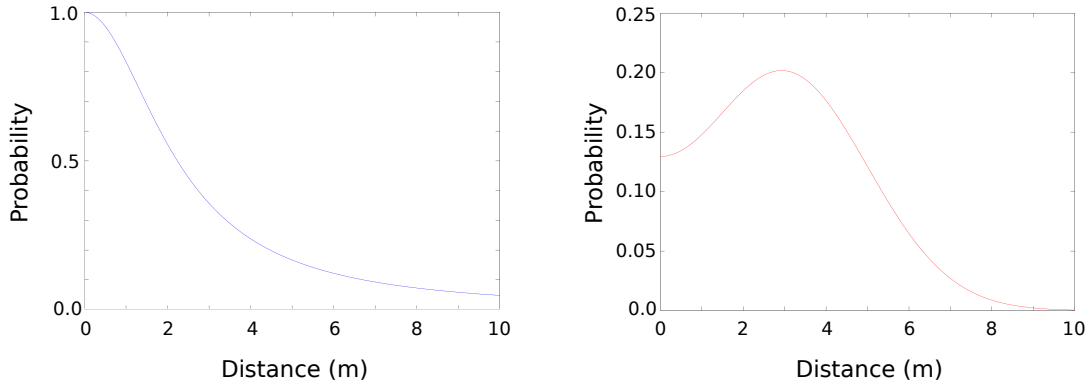


Figure 3-3: (a) Edge likelihood vs. distance between two vertices (b) Folded Gaussian distribution for distance d_{ij} for a mean 3.0 m and standard deviation 2.0 m.

apart as

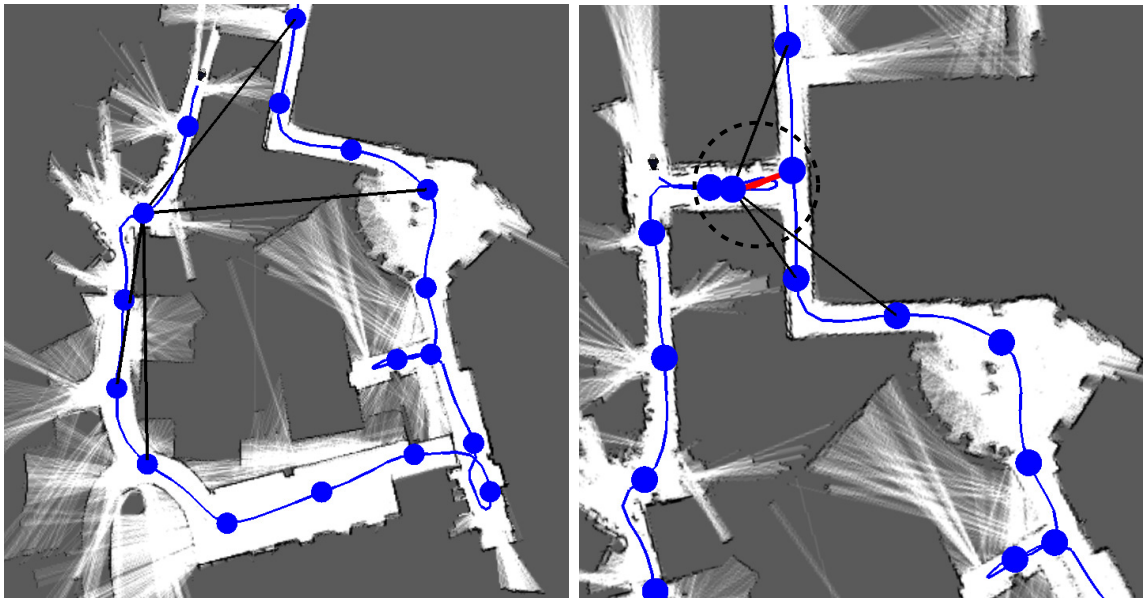
$$p(G_t^{ij}|d_{ij}^{(i)}, G_t^{-(i)}, z^{t-1}, u^t, \Lambda^t) \propto \frac{1}{1+\gamma {d_{ij}^{(i)}}^2}, \tag{3.9}$$

where γ specifies distance bias. For the evaluations in this chapter, we use $\gamma = 0.2$. Figure 3-3a shows how the likelihood of an edge changes based on the distance between two vertices. We approximate the distance prior $p(d_{tj}^{(i)}|G_t^{-(i)}, z^{t-1}, u^t, \Lambda^t)$ with a folded Gaussian distribution,

$$p(d_{ij}^{(i)}; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(-d_{ij}^{(i)} - \mu)^2}{2\sigma^2}\right) + \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(d_{ij}^{(i)} - \mu)^2}{2\sigma^2}\right) \quad \left(d_{ij}^{(i)} \geq 0\right) \tag{3.10}$$

where μ is the the mean and σ is the standard deviation, approximated based upon a linearized model for the distance between the normally distributed positions $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$. The probability is 0 for $d_{ij}^{(i)} < 0$. Figure 3-3b shows an example folded Gaussian distribution based on the distance.

The algorithm samples from the proposal distribution (3.8) to identify candidate edges. Before adding these to the graph, we use laser scans to build local maps around each vertex and compare the maps associated with the two vertices using scan-matching (Figure 3-4). The algorithm rejects edges that fail the scan-matching procedure, in order to eliminate



(a) Rejected Edges (b) Rejected and Accepted Edges

Figure 3-4: In the proposal step, the algorithm hypothesizes the addition of new edges in the graph based upon the estimated distance between vertices. Candidate edges are (a) rejected (black) or (b) accepted (red) based upon scan-matching.

topology samples that have low-likelihood of being correct. Even when the scan-matching is successful, it can still yield false positives for areas with ambiguous local geometry (such as in long corridors). In order to reduce the effects of this perceptual aliasing, we evaluate the likelihood of the scan-matched estimates of the inter-region transformations under our distribution over the metric map. The algorithm retains edges according to their Mahalanobis distance and adds edges deemed to be valid along with their estimated transformations.

Semantic Map-based Constraints

A fundamental contribution of our method is the ability for the semantic map to influence the metric and topological maps. This capability results from the use of the label distributions to perform place recognition. The algorithm identifies loop closures by sampling from a proposal distribution that expresses the semantic similarity between vertices. At each time step, we only sample edges from a subset of possible edges for the graph. We outline this process below.

We define the subset $\mathbb{S}_t$ that consists of the vertices that had semantic information integrated to them at time t. For each such vertex $v_k \in \mathbb{S}_t$ we define the set $\mathbb{S}_k^\Lambda$, containing the indices of each language event that contributed to its label distribution. For each vertex $v_k \in \mathbb{S}_t$, we sample semantic edges from the valid set $\mathbb{E}_k$, which is defined as:

$$\mathbb{E}_k = \left\{ e_{k,j} | \mathbb{S}_k^\Lambda \cap \mathbb{S}_j^\Lambda = \emptyset,\ \mathbb{S}_j^\Lambda \neq \emptyset,\ e_{kj} \notin E^- \right\}. \tag{3.11}$$

Effectively we only sample semantic edges between a node that had semantic information integrated at time t from language Λ_k and any nodes that already contain semantic information from language descriptions other than Λ_k.

For each vertex $v_k \in \mathbb{S}_t$, we sample semantic edges based on the proposal distribution defined below.

$$p_s(G_t | G_t^{-(i)}, z^{t-1}, u^t, \Lambda^t) = \prod_{e_{kj} \in \mathbb{E}_k} p(G_t^{kj} | G_t^{-(i)}, \Lambda_t) \tag{3.12a}$$

$$= \prod_{e_{tj} \in \mathbb{E}_k} \sum_{L_t^{-(i)}} p(G_t^{kj} | L_t^{-(i)}, G_t^{-(i)}, \Lambda_t) p(L_t^{-(i)} | G_t^{-(i)}) \tag{3.12b}$$

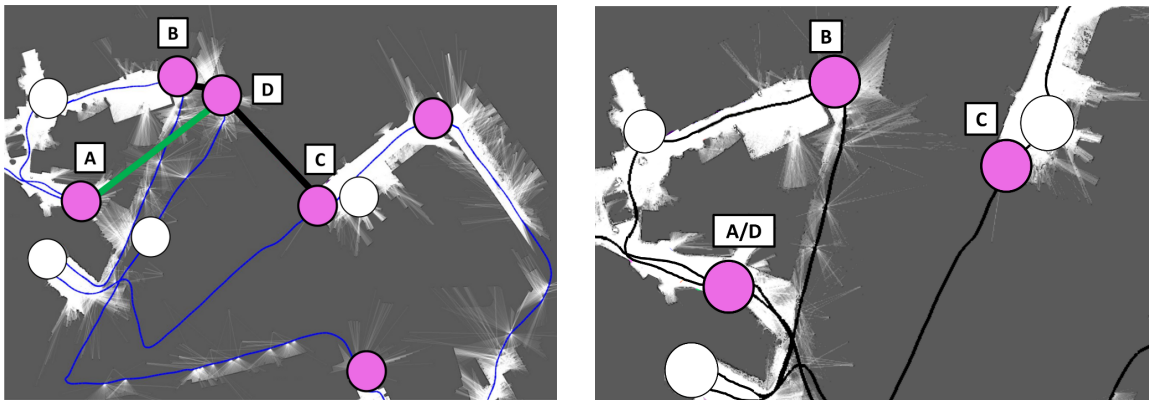
$$\approx \prod_{e_{kj} \in \mathbb{E}_k} \sum_{l_t^-, l_j^-} p(G_t^{kj} | l_k^-, l_j^-, G_t^{-(i)}) p(l_k^-, l_j^- | G_t^{-(i)}), \tag{3.12c}$$

where we have omitted the metric, odometry, and language inputs for clarity. The first line follows from the assumption that additional edges $e_{kj} \in \mathbb{E}_k$ that express constraints to the vertex v_k are conditionally independent. The second line represents the marginalization over the space of labels, while the last line results from the assumption that the semantic edge likelihoods depend only on the labels for the vertex pair.

We model the likelihood of edges between two vertices as non-zero for the same label

$$p(G_t^{kj} | l_k, l_j) = \begin{cases} \theta_l & \text{if } l_k = l_j \\ 0 & \text{if } l_k \neq l_j \end{cases} \tag{3.13}$$

where θ_l denotes the label-dependent likelihood that edges exist between vertices with the same label. Equation 3.12c then measures the cosine similarity between the label distri-



(a) (b)

Figure 3-5: Semantic map-based constraint sampling (for a single particle): (a) When the robot is at node D (described by the guide as an entrance) the algorithm samples edges to three other nodes (A,B, and C) that were labeled as entrances (all nodes with a high-likelihood for the label "entrance" are shown in pink). It rejects invalid edges that result from ambiguous labels (black) and adds the edge (green) that denotes a valid loop closure. (b) The resulting map after the accepted edge (between A and D) is added to the topology, and their metric locations are updated based on the new constraint.

butions. This parameter expresses the fact that certain labels are more commonplace and less likely to suggest that two regions are the same. For example, regions such as hallways are pretty common in the environment, and as such the likelihood of an edge between two hallways is less likely. In practice, we use a value of 0.2 for the label "hallway," and 1.0 for all other labels.

We sample from the proposal distribution (3.12) to hypothesize new semantic map-based edges. As with distance-based edges, we validate proposed edges by building local maps for each region and performing scan-matching between these maps. Figure 3-5 shows several different edges sampled from the proposal distribution for a single particle at one stage of a tour. Here, the algorithm identifies candidate loop closures between different "entrances" in the environment and accepts those (shown in green) whose local laser scans result in a valid scan-match. Note that some particles may add invalid edges (e.g., due to perceptual or semantic aliasing), but their weights will decrease as subsequent measurements become inconsistent with the hypothesis.

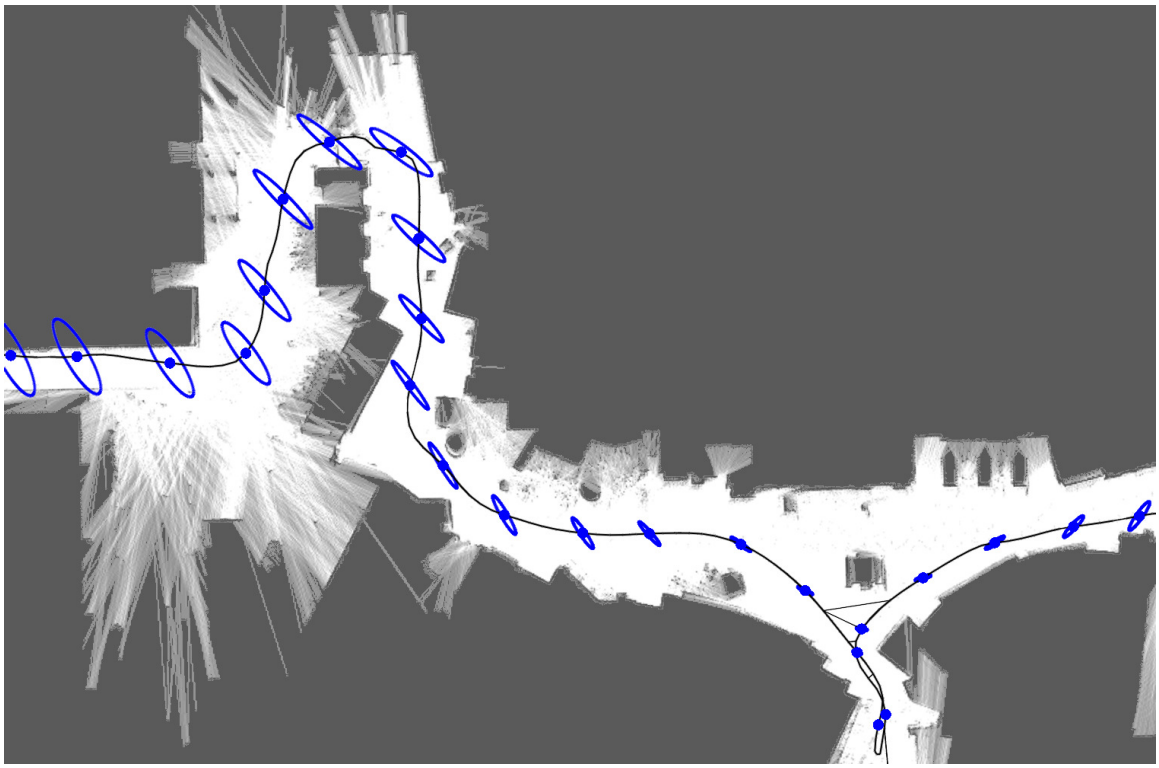


Figure 3-6: The mean position and 1σ uncertainty ellipse for each vertex, along with the resulting occupancy grid map.

3.2.2 Updating the Metric Map Based on New Edges

The proposal step results in the addition, to each particle, of a new vertex at the current robot pose, along with an edge representing its temporal relationship to the previous vertex. The proposal step might also add additional loop-closure edges. Next, the algorithm incorporates these relative pose constraints into the Gaussian representation for the marginal distribution over the map

$$p(X_t^{(i)}|G_t^{(i)}, z^t, u^t, \Lambda^t) = \mathcal{N}^{-1}(X_t^{(i)}; \Sigma_t^{-1}, \eta_t), \tag{3.14}$$

where Σ_t^{-1} and η_t are the information (inverse covariance) matrix and information vector that parametrize the canonical form of the Gaussian. We utilize the iSAM algorithm [38] to update the canonical form by iteratively solving for the QR factorization of the information matrix. Figure 3-6 shows the resulting metric poses and their uncertainties.

3.2.3 Updating the Semantic Map Based on Natural Language

Next, the algorithm updates the distribution over the current labels $L_t^{(i)} = \{l_{t,1}, l_{t,2}, \ldots, l_{t,t}\}$ associated with each particle. This update reflects information regarding labels and spatial relations conveyed by spoken descriptions, as well as semantic concepts that are suggested by the addition of edges to the graph. In maintaining the label distribution, we make the assumption that the vertex labels are conditionally independent given the topology and vertex poses

$$p(L_t^{(i)}|X_t^{(i)}, G_t^{(i)}, z^t, u^t, \Lambda^t) = \prod_{i=1}^{t} p(l_{t,i}|X_t^{(i)}, G_t^{(i)}, z^t, u^t, \Lambda^t). \tag{3.15}$$

This assumption ignores dependencies between labels associated with nearby vertices, but simplifies the form for the distribution over labels associated with a single vertex. We model each vertex's label distribution as a Dirichlet distribution of the form

$$\begin{aligned} p(l_{t,i}|\Lambda_1 \ldots \Lambda_t) &= Dir(l_{t,i}; \alpha_1 \ldots \alpha_K) \\ &= \frac{\Gamma(\sum_1^K \alpha_i)}{\Gamma(\alpha_1) \times \ldots \times \Gamma(\alpha_K)} \prod_{k=1}^{K} l_{t,i,k}^{\alpha_k - 1}, \end{aligned} \tag{3.16}$$

where $l_{t,i,k}$ for $k \in \{1, \ldots, K\}$ is the k^{th} label associated with vertex i at time t. We initialize the parameters $\alpha_1 \ldots \alpha_K$ to 0.2, which results in a prior that is uniform over the different labels. Given subsequent language input, this favors distributions that are peaked around a single label.

We consider user-provided expressions that use spatial relations to describe one or two locations in the environment. The first type are *egocentric* utterances (e.g., "This is the gym") describe the robot's current location. A contribution of our work is the ability to incorporate information from *allocentric* spatial language (e.g., "The kitchen is through the cafeteria") that expresses spatial relations and labels associated with non-local, potentially distant regions in the environment. By interpreting these expressions, our framework enables robots to learn rich semantic maps of their environment more efficiently.

Learning from allocentric expressions is challenging because their groundings can be

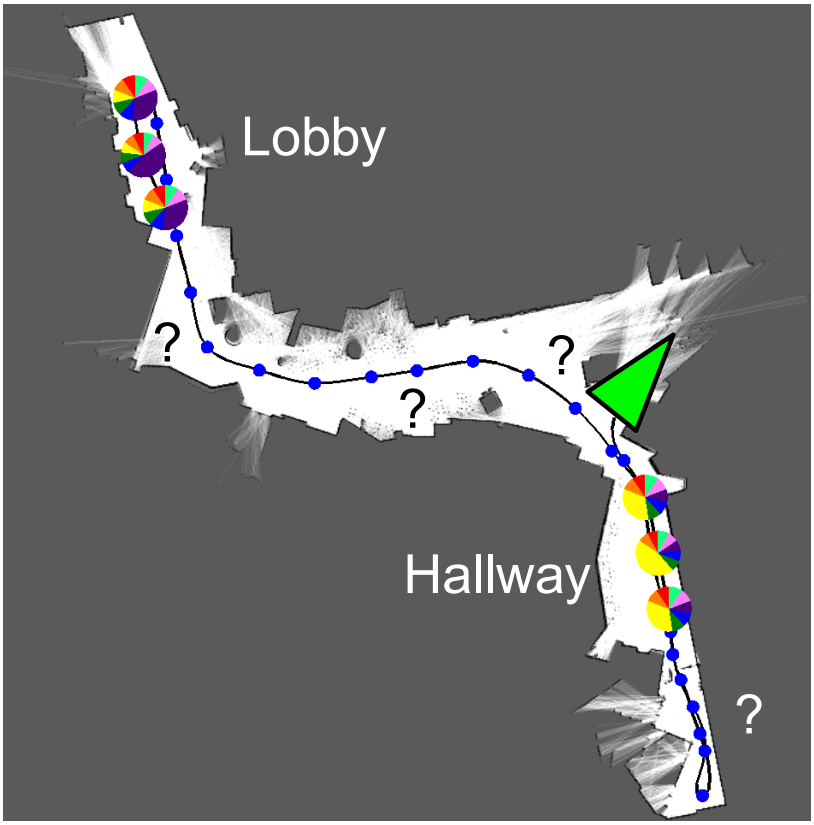


Figure 3-7: The user utters the description "The gym is down the hall" when the robot is at the location indicated by the triangle. (only nodes with non-uniform label distributions are visualized)

ambiguous—the places to which the user refers are often not obvious. Consider the scenario outlined in Figure 3-7. The semantic map includes an area that has a high likelihood of being a "lobby" and a second believed to be a "hallway." As the robot (triangle) continues to explore the environment, the user utters the description "The gym is down the hall." Descriptions like these are often ambiguous. For example, there may be multiple "hall" regions in the map or it may be that the robot has yet to visit the region referred to by the user, or if it has, it is not aware of its label. Similarly, several regions in the map are candidates for being the "gym," but the user may also be identifying a region that is not yet in the map.

Grounding Natural Language Descriptions with G^3

In order to understand allocentric language in our framework, we make use of the G^3 algorithm, proposed by Tellex et al. [88]. We provide an overview of its functionality below and how we apply it in our framework. Given natural language text Λ, G^3 provides a distribution over the space of possible mappings between each word in the parsed description and the corresponding groundings in the robot's model of the world. This distribution takes the general form

$$p(\Phi|\Gamma, \Lambda, M), \tag{3.17}$$

where $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ denotes the set of possible groundings and M represents the robot's world model, which includes the robot's pose and a map of the environment. The correspondence variable Φ contains boolean-valued variables ϕ for each linguistic element $\lambda \in \Lambda$ and grounding $\gamma \in \Gamma$, such that $\phi = \texttt{True}$ iff γ corresponds to λ. In our application, the groundings are the locations of the nodes in the semantic graph the paths between nodes according to the metric map.

Taking advantage of the compositional, hierarchical structure of natural language [37], G^3 parses the utterance into a set of Spatial Description Clauses (SDCs). Each SDC is assigned a type (event, object, place, or path) and consists of landmark λ_i^l, figure λ_i^f, and relation λ_i^r phrases. For descriptions of the environment, G^3 parses descriptions into place and path SDCs using a learned grammar that includes possible labels and spatial relations. G^3 then factors the distribution (3.17) into individual terms, one for each linguistic element

$$p(\Phi|\Gamma, \Lambda, M) = \prod_i p(\phi_i|\lambda_i, \Gamma, M). \tag{3.18}$$

This factored distribution is represented as a graphical model using a factor graph, such as the one shown in Figure 3-8 for the "the gym is down the hall" utterance. The G^3 algorithm uses a log-linear model for each of the factors

$$p(\phi_i|\lambda_i, \Gamma, M) \propto \exp\left(\sum_j \mu_j s_j(\phi_i, \lambda_i, \Gamma, M)\right), \tag{3.19}$$

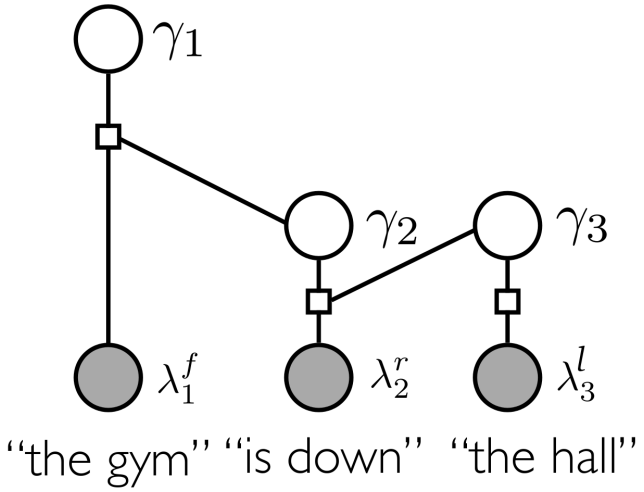


Figure 3-8: The factor graph model for the utterance "The gym is down the hall" that is used by the G^3 algorithm.

where μ_j are weights and s_j are features that encode the relationship between the linguistic element λ_i and the groundings Γ. For example, one such feature relates the length of the path through the map from the landmark grounding γ_i^l and figure grounding γ_i^f when the relation λ_i^r is "down from"

$$s(\gamma_i^l, \gamma_i^f, \lambda_i^r) \triangleq |x_{\gamma_i^l} - x_{\gamma_i^f}| \wedge (\text{"down from"} \in \lambda_i^r). \tag{3.20}$$

Similarly, features for other relations express the consistency of the path between pairs of nodes with the uttered relation. Additional features include the likelihood of the landmark label λ_i^l under the multinomial associated with the node's γ_i^l label distribution.

The G^3 model learns the weights μ_j associated with each feature by training on a corpus of SDCs from natural language descriptions and the known groundings Γ and correspondences Φ. In particular, for this work, we trained the G^3 model using a route directions corpus [41] that includes a set spoken directions through an office building paired with positive and negative examples of paths through the environment.

Given a particular spoken description, we use G^3 to infer groundings for the different parts of the utterance. In the case of the current example, the framework induces a probability distribution over vertices whose location is consistent with being "down from" each of the conditioned landmark vertices, based upon the robot's pose at the time the user offers

the communication. In this manner, we make the assumption that the person is describing the environment in the robot's frame of reference. For egocentric language, the grounding likelihood is simplified since the figure is implicitly the robot's current location. In order to understand an expression like "The gym is down the hall," the system must first ground the landmark phrase "the hall" (λ^l) to a specific entity in the environment. It must then infer the figure entity in the environment that corresponds to the phrase "the gym" (λ^f), given the spatial relation "down" (λ^r). One can no longer assume that the user is referring to the current location as "the gym" (the *figure*[3]) or that the location of the "hall" (the *landmark*) is known (e.g., there are likely many "halls" in the environment). When considering the implications of the natural language description, we make the assumption that both the landmark and the figure region described by the user are already in the map (represented in the topology as vertices). We relax this assumption shortly (as outlined in the next section) to deal descriptions that may describe regions that the robot has not encountered yet. We use the label distribution to reason over the possible vertices that denote the landmark. To arrive at the distribution over the potential landmark groundings we normalize the likelihoods for candidate "hall" vertices as follows, where V_l is the set of vertices within a distance threshold from the robot with a $p(l_{t,k} = \lambda^l) > \beta$

$$p(\phi^l_{v_j} = \mathtt{T}) = \frac{p(l_{t,j} = \lambda^l)}{\sum_{k:v_k \in V_l} p(l_{t,k} = \lambda^l)} \tag{3.21}$$

denotes the likelihood of vertex v_j being the described landmark.

We account for the uncertainty in the figure by formulating a distribution over the vertices in the topology that expresses their likelihood of being the figure. Formally, we model the likelihood that each vertex v_i is the figure by marginalizing over the space of candidate landmarks

$$p(\phi^f_{v_i} = \mathtt{T}) = \sum_{v_j} p(\phi^f_{v_i} = \mathtt{T} | \phi^l_{v_j} = \mathtt{T}, \lambda^r)\, p(\phi^l_{v_j} = \mathtt{T}), \tag{3.22}$$

where λ^r is the described spatial relation between the landmark and the figure region, $\phi^l_{v_j}$ is the binary-valued random variable that indicate that vertex v_j is the landmark and $\phi^f_{v_i}$ is the

[3]In spatial linguistic theory, this is often referred to as the *trajector*.

binary-valued random variable that indicate that vertex v_i is the figure. We only consider vertices within a certain distance threshold from the robot as valid potential figure regions. We arrive at the conditional distribution $p(\phi^f_{v_i} = \mathtt{T} | \phi^l_{v_j} = \mathtt{T}, \lambda^r)$ using the G^3 framework to infer groundings.

$$p(\phi^f_{v_i} = \mathtt{T} | \phi^l_{v_j} = \mathtt{T}, \lambda^r) = p(\phi^f_{v_i} = \mathtt{T} | \gamma^f_i, \gamma^l_j, \gamma_{p_i}, \lambda^r) \tag{3.23}$$

We ground relational utterances λ^r by considering the shortest path p_i that travels from the robot's pose at the time of the description through the pair of landmark γ^l_j and figure γ^f_i vertex groundings. We use the A* algorithm [79] to solve for the shortest path through the semantic graph topology. We then use features over these paths (3.20) to evaluate their consistency with the uttered relation (e.g., "down from," "near," and "through").

For both types of expressions, the algorithm updates the semantic distribution according to the rule

$$p(l_{t,i} | \Lambda_t = (k, i), l_{t-1,i}) = \frac{\Gamma(\sum_1^K \alpha_i^{t-1} + \Delta\alpha)}{\Gamma(\alpha_1^{t-1}) \times \ldots \times \Gamma(\alpha_k^{t-1} + \Delta\alpha) \times \ldots \times \Gamma(\alpha_K)} \prod_{k=1}^{K} l_{t,i,k}^{\alpha_k - 1}, \tag{3.24}$$

where $\Delta\alpha$ is the likelihood of the figure grounding. For allocentric descriptions, we set $\Delta\alpha$ to the landmark likelihood computed via Equation 3.22.

In the case of egocentric language, when the robot's position is implicitly the figure, we set this likelihood to $\Delta\alpha = 1$ for the current vertex in the graph. Additionally we also update the label distribution for a vertex when the proposal step adds an edge to another vertex in the graph. These edges may correspond to temporal constraints that exist between consecutive vertices, or they may denote loop closures based upon the spatial distance between vertices that we infer from the metric map. Upon adding an edge to a vertex for which we have previously incorporated a direct language observation, we propagate the observed label to the newly connected vertex using a value of $\Delta\alpha = 0.5$.

Handling Anticipatory Descriptions

An advantage of having a probabilistic model over the space of groundings is that it provides a means of recognizing when there is not enough information contained in the semantic graph to ground the language. This allows us to recognize many of the situations in which the user describes areas that either the robot has not yet visited or they reference landmarks whose labels were never added to the map. For example, it is not uncommon for the user to mention regions that are within sight but they have yet to reach (e.g., the user may say "The lab is across the lobby," but the robot has never been to the region being referred to as "the lab."). We refer to descriptions of this form as *anticipatory*.

We identify instances of anticipatory descriptions by using our distributions over the landmark and figure locations to evaluate the likelihood that the landmark matches a labeled region in the graph and that there are one or mode candidate figure regions consistent with the language. When the method is sufficiently confident in the ability to ground the language (we use a threshold of 0.2), we update the label distributions as described above. However, when the grounding likelihoods suggest an anticipatory description, the algorithm adds the expression along with its timestamp to a per-particle queue of anticipatory descriptions. As the robot proceeds through the environment and new vertices and semantic information are added to the map, the algorithm periodically evaluates the grounding likelihood (3.22) for the queued descriptions. The logic is that the description is most useful when the robot has visited the regions to which the user refers and, thereby, the map has regions whose labels and inter-region paths that are consistent with the expression. The algorithm performs this process separately for each particle, which may result in some particles incorporating the description sooner than others depending on the topological and metric information associated with each particle.

3.2.4 Updating the Particle Weights

Having proposed a new set of graphs $\{G_t^{(i)}\}$ and updated the analytic distributions over the metric and semantic maps for each particle, we update their weights. The update follows from the ratio between the target distribution over the graph and the proposal distribution,

and can be shown to be

$$w_t^{(i)} = \frac{\text{Target distribution}}{\text{Proposal distribution}} \tag{3.25a}$$

$$= \frac{p(G_t^{(i)}|z^t, u^t, \Lambda^t)}{p(G_t^{(i)}|G_{t-1}^{(i)}, z^{t-1}, u^t, \Lambda^t)} w_{t-1}^{(i)} \tag{3.25b}$$

$$= \frac{p(z_t|G_t^{(i)}, z^{t-1}, u^t, \Lambda^t)}{p(z_t|z^{t-1})} \cdot p(G_{t-1}^{(i)}|z^{t-1}, u^t, \Lambda^t) \tag{3.25c}$$

$$\propto p(z_t|G_t^{(i)}, z^{t-1}, u^t, \Lambda^t) \cdot p(G_{t-1}^{(i)}|z^{t-1}, u^t, \Lambda^t) \tag{3.25d}$$

$$\tilde{w}_t^{(i)} = p(z_t|G_t^{(i)}, z^{t-1}, u^t, \Lambda^t) \cdot w_{t-1}^{(i)}, \tag{3.25e}$$

where $w_{t-1}^{(i)}$ is the weight of particle i at time $t-1$ and $\tilde{w}_t^{(i)}$ denotes the unnormalized weight at time t. We evaluate the measurement likelihood (e.g., of lidar) by marginalizing over the vertex poses

$$p(z_t|G_t^{(i)}, z^{t-1}, u^t, \Lambda^t) = \int_{X_t} p(z_t|X_t^{(i)}, G_t^{(i)}, z^{t-1}, u^t, \Lambda^t) \times p(X_t^{(i)}|G_t^{(i)}, z^{t-1}, u^t, \Lambda^t) dX_t, \tag{3.26}$$

which allows us to utilize the conditional measurement model. In the experiments presented next, we model the measurement as an observed transformation between poses, which we compute via scan-matching. We model this distribution (first term in the integral) as Gaussian, which we have empirically found to be accurate.

After calculating and normalizing the new importance weights, we periodically perform resampling based upon the effective number of particles, as proposed by Liu [49],

$$N_{eff} = \frac{1}{\sum_{i=0}^{N-1} w_i^2}. \tag{3.27}$$

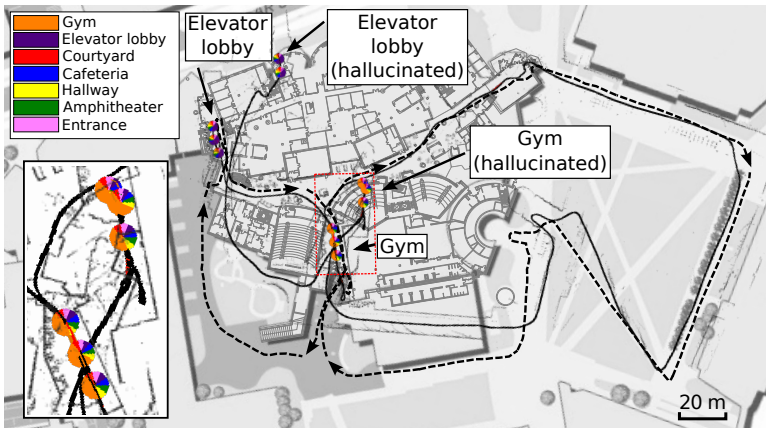
When the effective number of particles N_{eff} falls below the threshold $N/2$, where N is the number of particles, we resample using the algorithm described by Doucet et al. [15].

3.3 Experimental Evaluation

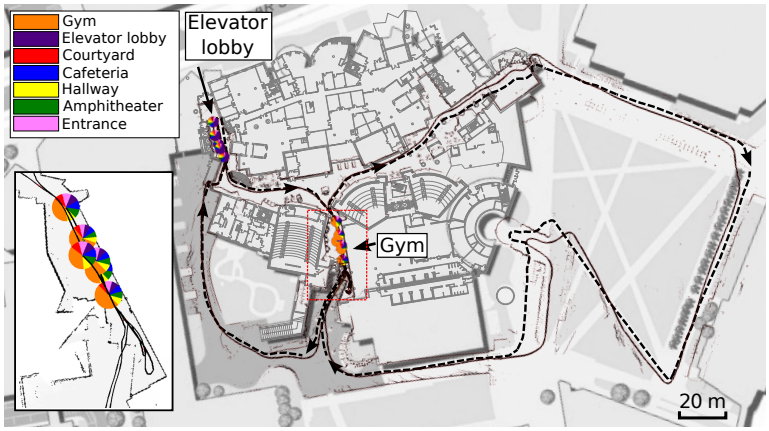
This section evaluates the utility of the semantic mapping framework, specifically its ability to learn semantic properties from natural language descriptions, and its ability to use this semantic information to improve the spatial representation. We also evaluate the effectiveness of the algorithm's natural language grounding capabilities as well as its timing performance. We evaluate our algorithm through six experiments that involve a human giving a robotic wheelchair a narrated tour (Figure 1-4b) [30] of several buildings and courtyards on the MIT campus. The robot was equipped with forward- and rearward-facing lidars, wheel encoders, and an IMU. Speech was recorded using a wireless microphone worn by the user. In the third experiment, the robot autonomously followed the human who provided spoken descriptions, while in the others, the robot was manually driven while the user interjected textual descriptions of the environment. Speech recognition was performed manually. Throughout this chapter, we only visualize the semantic distribution for vertices whose distribution is not uniform.

3.3.1 Indoor/Outdoor: Small Tour

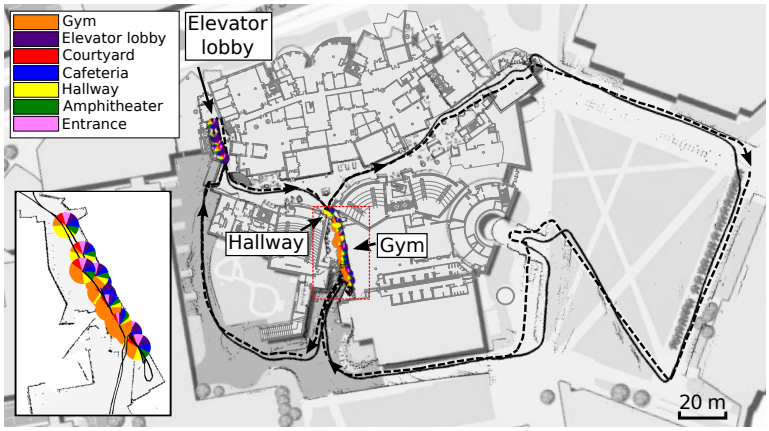
The first experiment (Figure 3-9) took place on the first floor of the Stata Center at MIT, which includes lecture halls, elevator lobbies, a gym, and a cafeteria, as well as the adjacent courtyard. Starting at one of the elevator lobbies, the user proceeded to visit the gym, exited the building and, after navigating the courtyard, returned to the gym and finished at the elevator lobby. The user provided textual descriptions of the environment, twice each for the elevator lobby and gym regions. We compare the performance of our method based upon different forms of language input against a baseline algorithm that emulates the current state-of-the-art in language-augmented semantic mapping. In all cases, the algorithms were run with 10 particles to approximate the distribution over the space of topologies. The final topology contained 137 vertices.



(a) No language constraints



(b) Egocentric language



(c) Allocentric language

Figure 3-9: Maximum likelihood semantic graphs for the small tour. In contrast to (a) the baseline algorithm, our method incorporates key loop closures based upon (b) egocentric and (c) allocentric descriptions that result in metric, topological, and semantic maps that are noticeably more accurate. The dashed line denotes the approximate ground truth trajectory. The inset presents a view of the semantic and topological maps near the gym region.

No Language Constraints

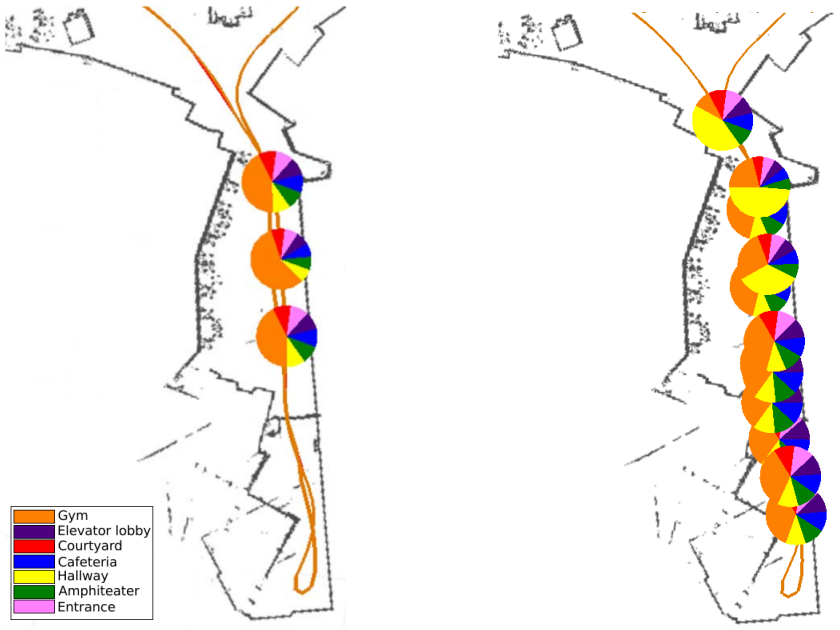
We consider a baseline approach that directly labels vertices based upon egocentric language, but does not propose edges based upon label distributions. It does, however, propose loop closures based upon the distribution over the metric map (Section 3.2.1). The baseline emulates typical solutions by augmenting a state-of-the-art iSAM metric map with a semantic layer without allowing semantic information to influence the other layers.

Figure 3-9a presents the resulting metric, topological, and semantic maps that constitute the semantic graph for the highest-weighted particle. The accumulation of odometry drift results in significant errors in the estimate for the robot's pose when revisiting the gym and elevator lobby. Without using semantic information to propose new edges, the algorithm is unable to detect valid loop closures. This results in significant errors in the metric and topological maps as well as the semantic map, which hallucinates two separate elevator lobbies (purple) and gyms (orange).

Egocentric Language

We evaluate our algorithm when the user provides descriptions in the form of egocentric language, in which case there is no ambiguity in the landmark and figure that are implicitly the robot's current location.

Figure 3-9b presents the semantic graph corresponding to the highest-weighted particle that our algorithm estimates. By considering the semantic map when proposing loop closures, the algorithm recognizes that the second region that the user labeled as "the gym" is the same place that was labeled earlier in the tour. At the time of receiving the second "gym" label, drift in the odometry has led to significant error in the gym's location much like the baseline result (Figure 3-9a). By proposing loop closure edges between the two sets of regions with high likelihood of being a gym, the algorithm is able to correct this error. Without the use of semantic information, it would require searching a combinatorially large space to infer the correct loop closures. The resulting maximum likelihood map is topologically and semantically consistent throughout and metrically consistent for most of the environment. The exception is the courtyard, where only odometry measurements were



(a) Egocentric language (b) Allocentric language

Figure 3-10: Pie charts that compare the semantic map label distributions that result from (a) the egocentric language description "This is the gym" with that of (b) the allocentric language description "The gym is down the hall."

available, causing drift in the pose estimate. Attesting to the model's validity, topologies consistent with the ground truth received 92.7% of the probability mass and, furthermore, the top four particles were each consistent with the ground truth.

Allocentric Language

Next, we consider the algorithm's performance when the figure and landmark regions that the user's descriptions reference can no longer be assumed to be the robot's current position. Specifically, we replaced the initial labeling of the gym with an indirect reference of the form "The gym is down the hallway," with the hallway labeled through egocentric language. The language inputs are otherwise identical to those employed for the egocentric language scenario and the baseline evaluation.

The algorithm incorporates allocentric language into the semantic map using the G^3 framework as described in Section 3.2.3 to infer the vertices in the graph that constitute the figure (i.e., the "gym") and the landmark (i.e., the "hallway"). This grounding attributes

a non-zero likelihood to all vertices that exhibit the relation of being "down" from the vertices identified as being the "hallway." Figure 3-10 compares the label distributions that result from this grounding with those from egocentric language. The algorithm attributes the "gym" label to multiple vertices in the semantic graph as a result of the ambiguity in the figure's location as well as the G^3 model, which yields high likelihoods for several paths as being "down from" the landmark vertices. When the user later labels the region after returning from the courtyard, the algorithm proposes a loop closure despite significant drift in the estimate for the robot's pose. As with the egocentric language scenario, this results in a semantic graph for the environment that is accurate topologically, semantically, and metrically (Figure 3-9c).

3.3.2 Indoor/Outdoor: Large Tour

The second experiment considers an extended tour of MIT's Stata Center as well as two neighboring buildings and their shared courtyard. In order to evaluate the algorithm's ability to deal with ambiguity in the labels, the robot visited several places with the same semantic attributes (e.g., elevator lobbies, entrances, and cafeterias) and visited some places more than once (e.g., one cafeteria and the amphitheater). We accompanied the tour with 20 descriptions of the environment that took the form of both egocentric and allocentric language. Figure 3-11 shows the ground truth path taken by the robot during the experiment as well as the locations of all the described regions.

As with the smaller tour, we compare our method against the baseline semantic mapping algorithm. Figure 3-12a presents the baseline estimate for the environment's semantic graph. Without incorporating allocentric language or allowing semantic information to influence the topological and metric layers, the resulting semantic graph exhibits significant errors in the metric map, an incorrect topology, and aliasing of the labeled places that the robot revisited. In contrast, Figure 3-12b demonstrates that, by using semantic information to propose constraints in the topology, our algorithm yields correct topological and semantic maps, and metric maps with notably less error. Figure 3-13 presents the inset views for the lobby and second cafeteria portion of the map that were labeled with allocentric de-

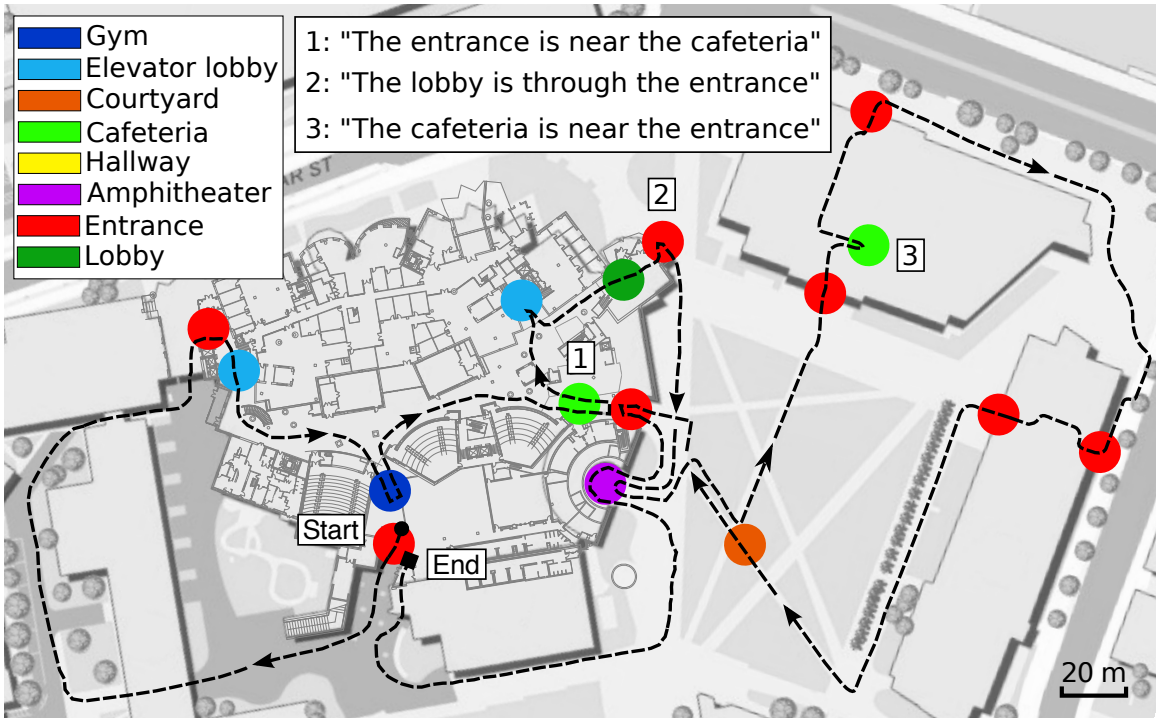
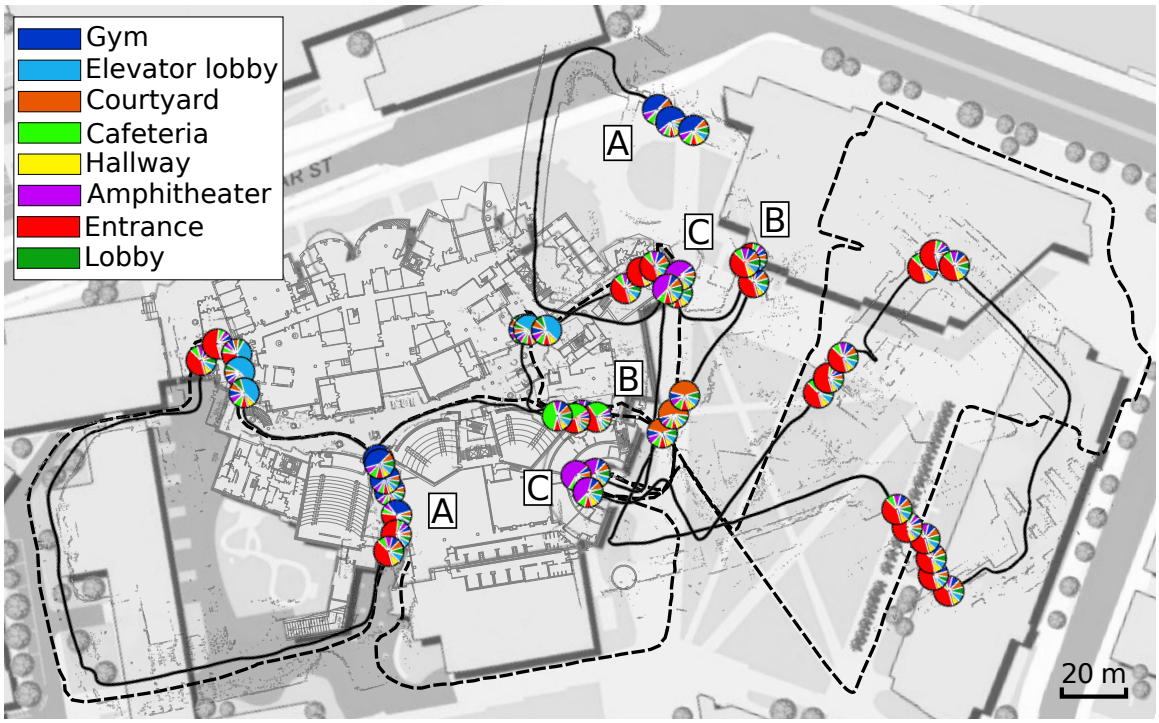


Figure 3-11: The ground truth route taken during the large tour experiment and the labeled regions.

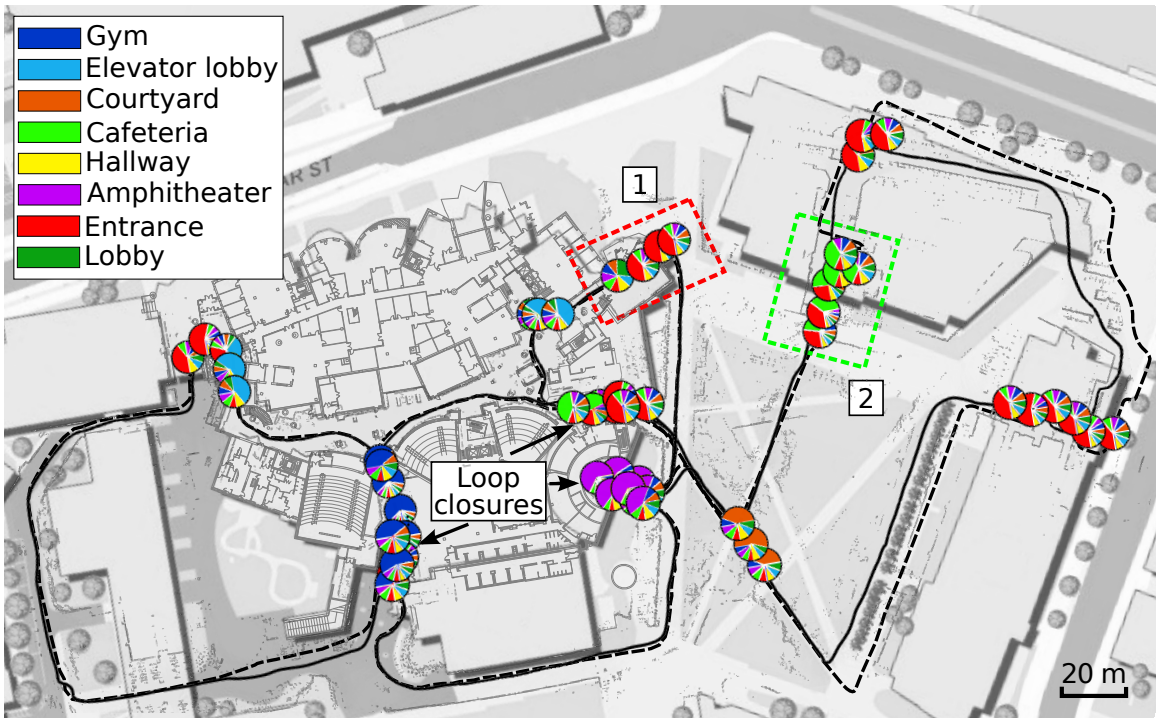
scriptions. The resulting model assigns 93.5% of the probability mass to the ground truth topology, with each of the top five particles being consistent with ground truth.

The results highlight the ability of our method to tolerate ambiguities in the labels assigned to different regions of the environment. This is a direct consequence of the use of semantic information, which allows the algorithm to significantly reduce the number of candidate loop closures that is otherwise combinatorial in the size of the map. This enables the particle filter to efficiently model the distribution over graphs. While some particles may propose invalid loop closures due to ambiguity in the labels, the algorithm is able to recover with a manageable number of particles. In this experiment, the algorithm employed 10 particles to approximate the distribution over topologies. The final topology contained 213 vertices.

For utterances with allocentric language, our algorithm was able to generate reasonable groundings for the figure and landmark locations. However, due to the simplistic way in which we define regions, groundings for "the lobby" were not entirely accurate due to the sensitivity to the local metric structure of the environment when grounding paths that go

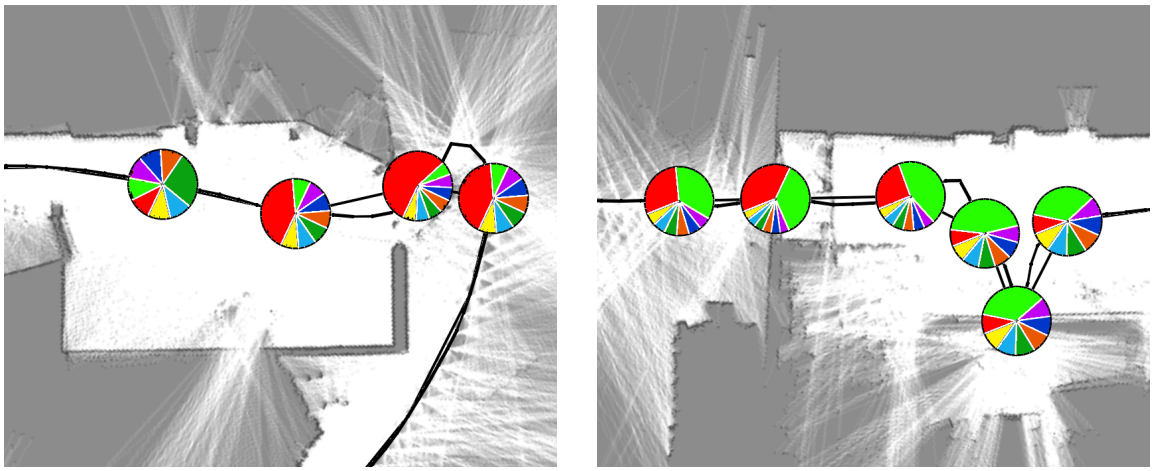


(a) No language constraints



(b) Allocentric language

Figure 3-12: Maximum likelihood semantic graphs for (a) The result of the baseline algorithm with letter pairs that indicate map components that correspond to the same environment region. (b) The result produced by our method based upon language descriptions, with an indication of the loop closures recognized based upon the semantic map.



(a) Lobby (b) Cafeteria

Figure 3-13: Inset views of the (a) lobby and (b) cafeteria portions of the semantic graph for the large tour experiment (Figure 3-12b).

"through the entrance." We discuss this in more detail in Section 3.3.10.

3.3.3 Indoor/Outdoor: Autonomous Tour

In the third experiment [31], the robot autonomously followed a user during a narrated tour along a route similar to that of the first experiment. Using a headset microphone, the user provided spoken descriptions of the environment that included ambiguous references to regions with the same label (e.g., elevator lobbies, entrances). The utterances included both egocentric and allocentric descriptions of the environment. The speech was recorded as it was uttered in synchronization with the lidar and odometry data. The audio was later manually transcribed into text that was inserted alongside the sensor observations according to the time that the audio was initially recorded. In this manner, the algorithm handled the text, lidar, and odometry data as they were received, emulating a scenario in which a speech recognizer was used to parse the user's utterances during the tour.

The algorithm operated in this fashion using 10 particles to approximate the distribution over the space of topologies. The final topology contained 135 vertices. Figure 3-14 presents the maximum likelihood semantic graph that our algorithm estimates. By incorporating information that the descriptions convey, the algorithm recognizes key loop closures that result in accurate semantic maps. The resulting model assigns 82.9% of the probability

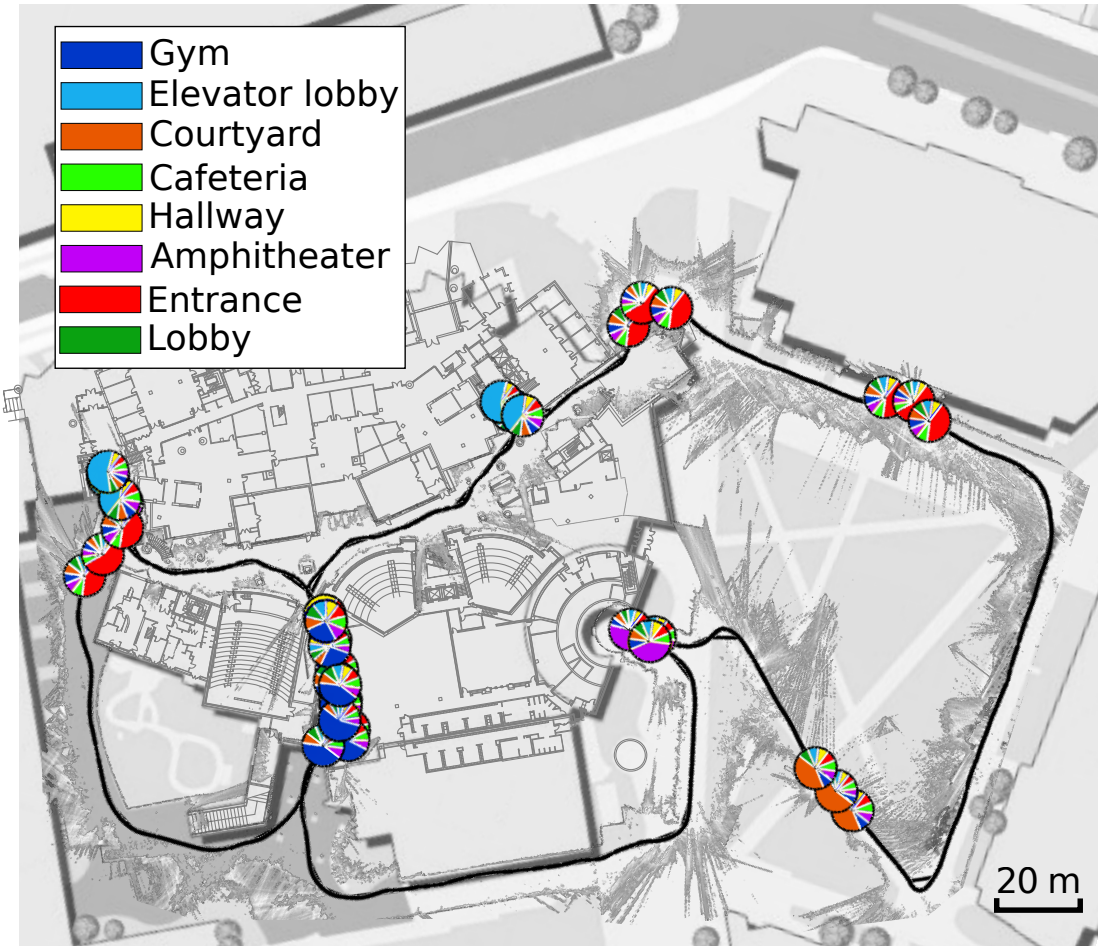


Figure 3-14: Maximum likelihood semantic graph for the autonomous tour.

mass to the ground truth topology, with each of the top nine particles being consistent with ground truth.

3.3.4 Stata Center Lab Tour

We consider an additional experiment in which the robot was driven throughout different labs on the third floor of MIT's Stata Center. The narrated tour involved both egocentric and allocentric descriptions of the environment, the latter of which were anticipatory in nature with the user referencing locations in the environment that the robot had not yet visited. Figure 3-15 presents the maximum likelihood semantic map that our framework learned from the narrated tour using a total of 10 particles. The final topology contained 71 vertices. The system correctly grounds each of the allocentric descriptions despite the

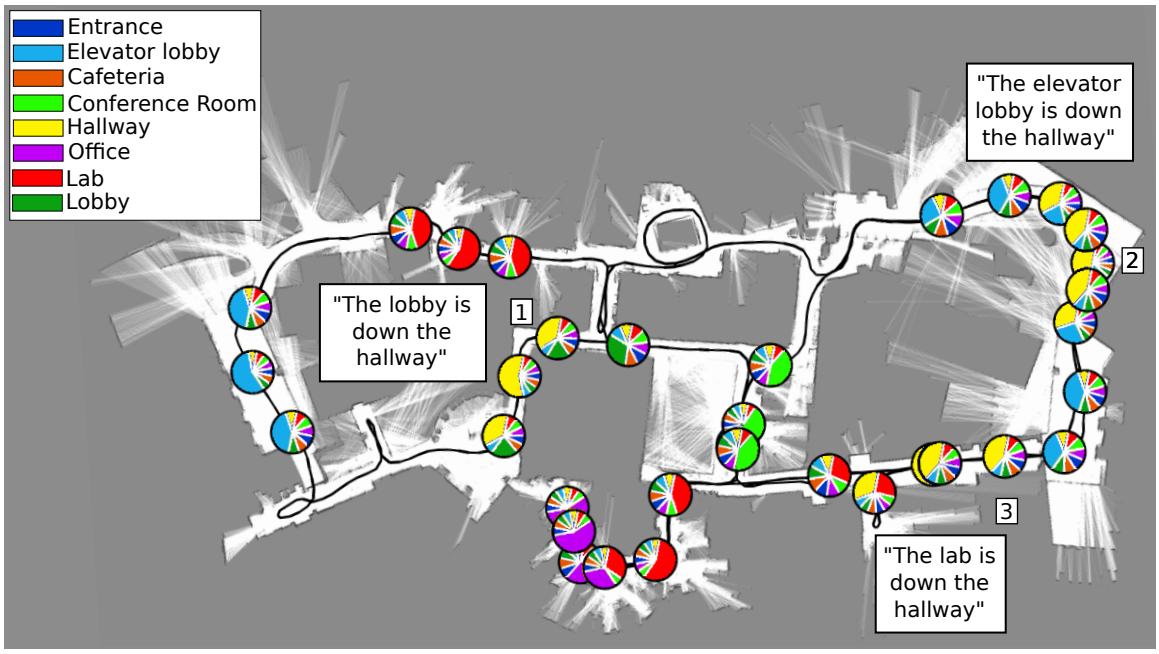


Figure 3-15: Maximum likelihood semantic graph inferred from the narrated tour of the Stata Center lab.

ambiguity that exists in the landmark and figure locations, as we discuss in more detail shortly.

3.3.5 MIT 32-36-38 Tour

In order to verify the validity of the algorithm in different environments, we consider an extended tour of three connected buildings on the MIT campus (buildings 32, 36, and 38). The robotic wheelchair was manually driven throughout the office-like environment, visiting offices, elevator lobbies, conference rooms, and lab spaces whose appearance and structure varied between each building. Text was added at several points throughout the tour to emulate recognized natural language descriptions. We provided both egocentric and allocentric utterances, including several instances of anticipatory descriptions when the robot had not yet visited the referenced portions of the environment (both the figure and the referent). We ran our framework with 10 particles to model the distribution over topologies. The final topology contained 148 vertices. Figure 3-16 denotes the maximum likelihood semantic graph that resulted from our algorithm. The text indicates the allocentric descriptions that were given to the system in the numbered order.

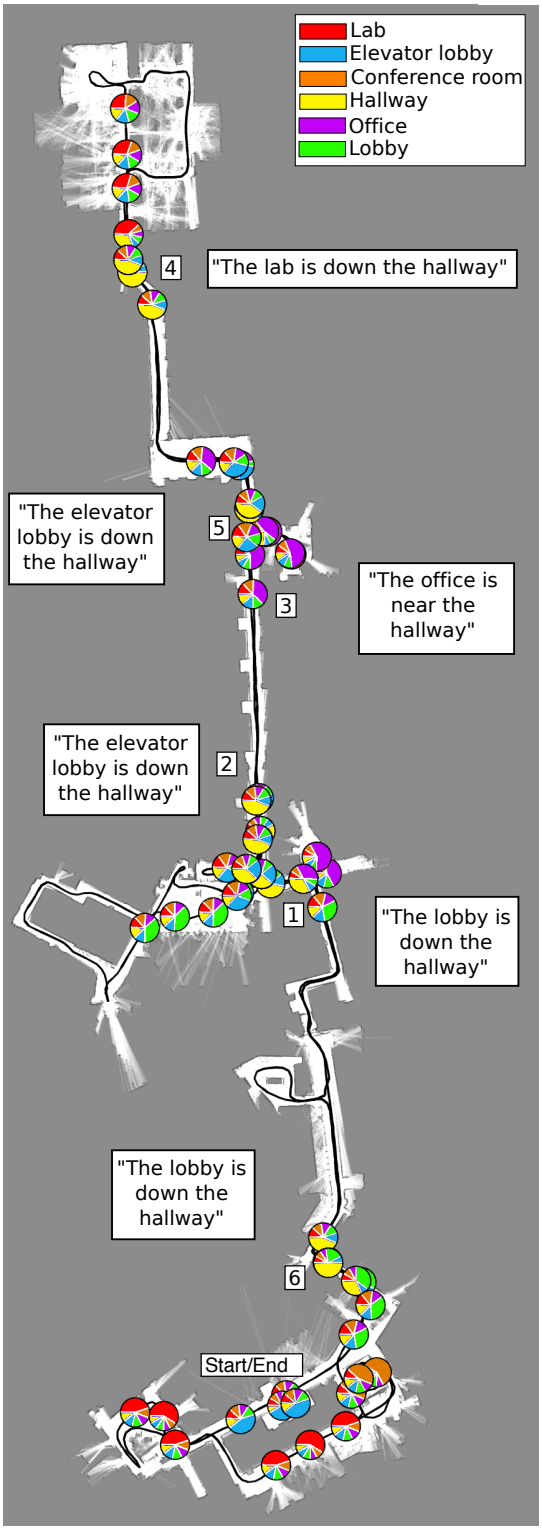


Figure 3-16: Maximum likelihood semantic graph for the MIT 32-36-38 tour. The allocentric descriptions are shown with numbers indicating their order.

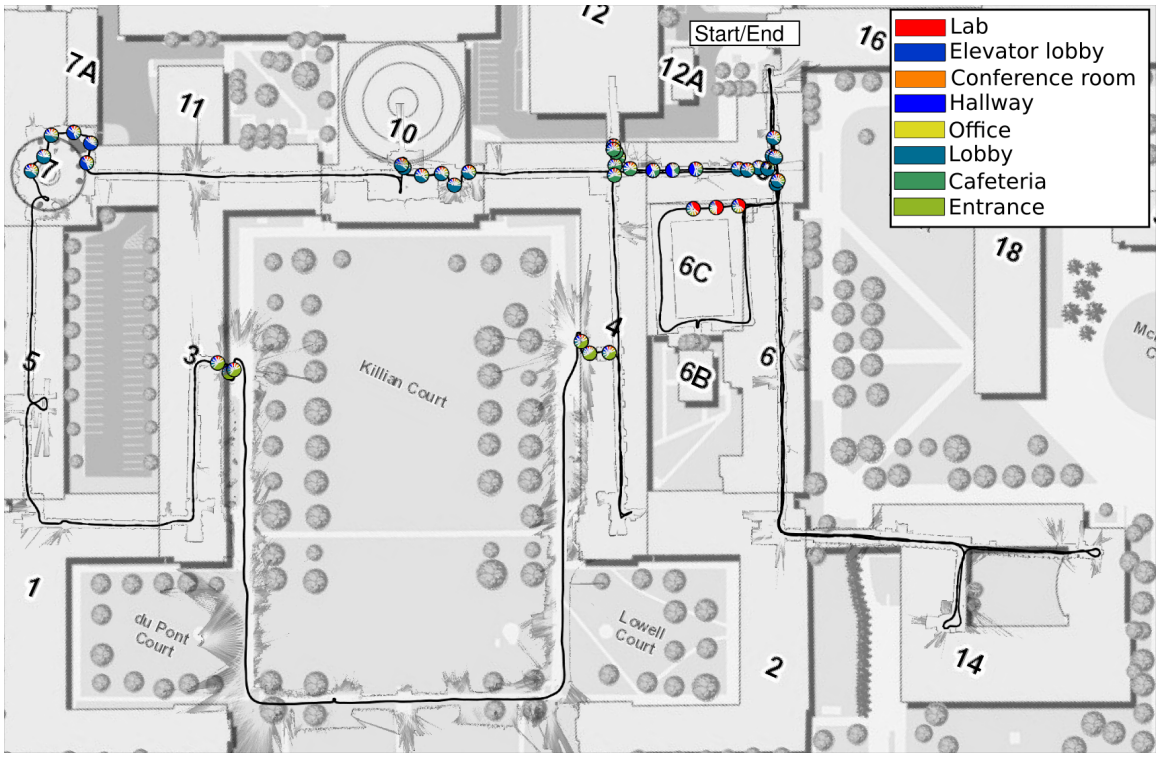


Figure 3-17: The maximum likelihood semantic graph that results from a tour of MIT's Killian Court.

3.3.6 Killian Court Tour

The final experiment considers a tour of Killian Court, a set of interconnected buildings on MIT's campus, which has served as a benchmark environment for previous mapping algorithms. We consider this environment in an effort to see how the algorithm performs when tasked with mapping larger spaces that involve significant geometric and semantic aliasing. Specifically, this part of the MIT campus consists primarily of several long hallways with nearly identical structure, including the so-called "infinite corridor" that serves as one of the main hallways at MIT.

Starting in the north-east corner (Figure 3-17), we gave the robot a tour along the infinite corridor that spans from left to right in the Figure. After entering one of the main lobbies (upper-left), we proceeded through buildings 5 and 3 and then exited into the courtyard. We took a U-shaped path outside, entered building 4, and then traveled through buildings 6, 6C, and 14 before returning to the start. We provided both egocentric and allocentric language descriptions at different points during the tour to assign labels to and spatial relations

between different regions. These descriptions took the form of text that was interjected in synchronization with the lidar and odometry streams as the data was post-processed.

The algorithm learned a distribution over semantic maps from the stream of descriptions, odometry, and lidar data, using 10 particles to hypothesize the different topologies. The final topology contained 276 vertices. Figure 3-17 shows the resulting maximum likelihood semantic graph overlaid on an approximately aligned map of the MIT campus. Qualitatively, the map is metrically, topologically, and semantically accurate with the exception of the map of building 14 where a glass hallway between buildings 2 and 14 forced the algorithm to use odometry for the inter-pose constraints. As with the previous evaluations, we ran our framework without language-based constraints to emulate the current state-of-the-art in language-augmented semantic mapping. While we omit the figure for space, we note that the resulting map is significantly warped.

3.3.7 Computational Requirements

We analyze the computational cost of the algorithm by considering the delay between when a vertex is first proposed (i.e., based on distance traveled) and the time at which it is added to the map. This measure reflects the overall time required of the algorithm, since it will not add vertices until it has finished incorporating the most recent description and proposed loop closures. We consider the delay for the three longest datasets, namely the indoor/outdoor large tour, the MIT 32-36-38 tour and the Killian Court tour. Table 3.2 summarizes the performance for each of these datasets. Note that the implementation has not been optimized to run in real-time, and each particle is currently processed sequentially

Table 3.2: Average Delay in Adding Vertex

Dataset	Average Delay (s)	Standard Deviation (s)
MIT 32-36-38	0.532	3.138
Killian Court Tour	0.682	2.726
Indoor/Outdoor Large Tour	2.186	4.670

(i.e., particle updates are not parallelized). The variance in the delays is due to periods of increased computation that correspond to instances when language annotations are processed. This delay is dominated by two components of the algorithm. The first is the time required to ground allocentric descriptions using the G^3 framework for all particles. The second is the time taken to scan-match the semantic-based loop closures that are subsequently proposed between vertices with updated label distributions. Allocentric language grounding requires computational effort that is linear in the number of unique particles. Similarly, the scan-match verification is linear in the number of vertices that are updated with new label information, which is independent of the size of the map. The computational requirements for verification are dominated by a scan-match procedure that is exhaustive in its search due to the potentially large error in the prior pose-to-pose transform.

3.3.8 Semantic Accuracy

Table 3.3 outlines the accuracy of the resulting semantic maps for four datasets, where we calculate the accuracy as follows. First, we select the regions for which language contributed to their label distributions. We compute the ground truth label for each of these regions and compute the cosine similarity between the ground truth multinomial (assumed to have a likelihood of 1.0 for the true label) and that of the label distribution.

For the indoor/outdoor large tour and the Killian Court tour, we also compared the results for the maps that did not propose language edges. Since large segments of these maps were metrically and topologically inaccurate, we assigned a minimum score for regions that were significantly inaccurate. In effect, this corresponds to assigning these regions a uniform multinomial over labels. As can be seen for the first two datasets, the use of our approach improves the semantic accuracy of a number of regions. This improvement stems both from the metric and topological accuracy of the learned maps as well as the algorithm's ability to integrate allocentric language. In the MIT 32-36-38 and the Stata Center tours, we also achieve reasonable accuracy for most categories. We do note that in case of allocentric language, some expressions can be ambiguous, either due to the presence of multiple potential landmarks or due to the ambiguity in the expression. For example, given

Table 3.3: Semantic Map Accuracy

	Indoor/Outdoor Large Tour		Killian Court Tour		MIT 32-36-38 Tour	Stata Center Tour
Type	Baseline	SG	Baseline	SG	SG	SG
Cafeteria	20%	36%	23%	45%	-	-
Entrance	43%	46%	12%	47%	-	-
Elevator Lobby	46%	46%	49%	49%	34%	40%
Hallway	8%	8%	18%	19%	36%	30%
Lobby	8%	13%	34%	47%	29%	21%
Lab	-	-	0%	47%	42%	37%
Amphitheater	25%	53%	-	-	-	-
Courtyard	12%	47%	-	-	-	-
Office	-	-	-	-	53%	56%
Conference Room	-	-	-	-	51%	56%
Gym	33%	48%	-	-	-	-

the description "The lobby is down the hallway," there may be multiple regions whose location is consistent with being "down" the hallway, of which only one is the lobby. In these situations, each of these regions will receive high likelihood of being the figure and the label distributions for each will be updated accordingly. Additionally, we find that the accuracy of the semantic maps is sensitive to our choice for region decomposition. For example, hallways score fairly low under our fixed-size segmentation, which can significantly underestimate their spatial extent. We see these issues as inherent to the definition of regions used in this chapter that would be alleviated with a more sophisticated segmentation strategy that takes into account local appearance [6, 72, 4], the presence of doorways [69] and semantic [55] properties of the environment. Our approach outlined in the next chapter addresses this by segmenting the regions based on the local appearance.

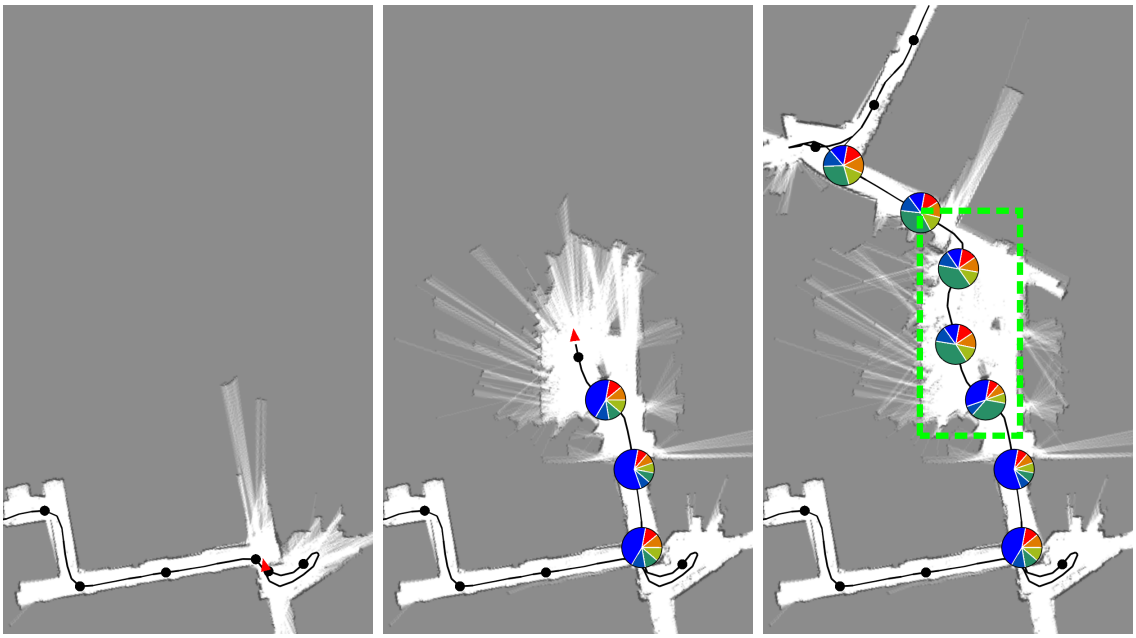
3.3.9 Navigation Efficiency

A consequence of maintaining a joint distribution over each layer of the semantic graph is that the framework is able to use knowledge of the semantic properties of the environment to update the topology and metric map. This improves the accuracy of the resulting semantic graph and, in turn, facilitates navigation. To better understand the effects on navigation efficiency, we consider the task of finding the optimal path between two vertices in the topology, as if the robot were asked to use the semantic graph to navigate from its current location to a named region in the environment.

We examine the semantic graphs that we learned with and without language-based constraints for the two indoor/outdoor scenarios, the autonomous tour, and the Killian Court dataset. For each, we randomly picked 1000 pairs of start and goal vertices in the graph and used a graph search algorithm to find the shortest path through the topology, with equal cost for each edge in the graph. The same vertex pairs were used for each of the semantic graphs for a given environment. Table 3.4 compares the average optimal path length through the graphs that result from our method and the baseline, which does not infer constraints from the descriptions. The graphs that we estimate when language influences only the semantic layer give rise to optimal paths that are noticeably longer than the paths reflected in the graphs that we learn by jointly estimating the semantic graph. This difference stems from the fact that our representation provides semantic-based edges that allow the planner to identify shortcuts in the topology that are otherwise not suggested by the baseline map, which mimics the current state-of-the-art in language-augmented semantic mapping.

Table 3.4: Average Length of the Optimal Path

Experiment	Baseline	SG
Small Indoor/Outdoor	41.59 m	23.50 m
Large Indoor/Outdoor	68.14 m	35.52 m
Autonomous	43.49 m	25.70 m
Killian Court	63.08 m	40.76 m



(a) Description given (b) Landmark is added (c) Figure is grounded

Figure 3-18: A depiction of the process of learning from an anticipatory description. (a) The user describes the "lobby" as being "down the hallway," yet the hallway has not been labeled and there is no vertex for the elevator lobby in the topology. (b) The user labels the current region as the "hallway," providing the landmark location. (c) Once vertices are added that are consistent with the description, the algorithm updates the labels. The green box indicates the actual location of the lobby.

3.3.10 Learning from Allocentric, Anticipatory Language

A contribution of our work is the use of natural language descriptions to produce consistent semantic maps from spatial relations and labels inferred from language. The advantage of this capability is that it allows robots to more efficiently acquire human-centric maps of their environment. The challenge to learning from these expressions is that their groundings are ambiguous—the user may refer to regions that may be distant from the robot and outside the field-of-view of its sensors. Additionally, it may be that the descriptions are anticipatory, when the robot has yet to visit the figure that the user is describing or the landmark that they are referencing. Figure 3-18 depicts the process of learning from an anticipatory description as part of the MIT 32-36-38 tour (Figure 3-16). Figure 3-18a shows the robot traversing a hallway when the user states that "The elevator lobby is down the hallway." At this point, the semantic graph includes several vertices with a high likelihood

of having the label "hallway." However, the robot has yet to visit the specific hallway that the person is using as the landmark and, as a result, the semantic graph does not include vertices for this region. The graph also lacks vertices for the region that the user refers to as the "elevator lobby." The algorithm attempts to ground the description using the language model as described in Section 3.2.3, which yields a likelihood for each pair of vertices as being the landmark and the figure.

This algorithm performs this grounding process for each particle, and updates those for which the likelihood of the top pair is sufficiently high (0.2). In this example, the likelihood of the candidate groundings for most of the particles is low and the algorithm postpones language integration. As the tour proceeds (Figure 3-18b), the guide labels the robot's position as being the "hallway," which updates the label distribution for the adjacent vertex. The algorithm again attempts to ground the language, this time using the newly added hallway vertices as the landmark. However, paths that start at the pose from which the description was first given and pass through the landmark to other vertices do not resemble the learned model for the "down" relation. After the robot and user continue and more vertices are added to the topology (Figure 3-18c), the framework again attempts to ground the description, this time returning highly-confident estimates for the locations of the landmark and the figure, per the induced path. However, not all of the inferred locations are correct, which is consistent with what we see with other allocentric expressions. In this case, the system assigns "elevator lobby" labels to vertices that preceded the hallway as well as several vertices beyond the true location of the lobby (green box). We attribute this to the difficulty in dealing with frame-of-reference when grounding language as well as to using features for the "down" relation that attempt to accommodate a wide range of scales (i.e. the length of hallways differs significantly across the environments that we consider).

In an effort to better understand the accuracy with which the algorithm learns from environment descriptions, we consider regions whose semantic properties were inferred from allocentric utterances. Figure 3-19 presents close-up views of the regions that were labeled as part of the multi-building tour (Figure 3-16). The portion of the semantic graph shown in Figure 3-19a results from two descriptions, "The lobby is down the hallway" and "The elevator lobby is down the hallway," which were uttered at the locations indicated by the

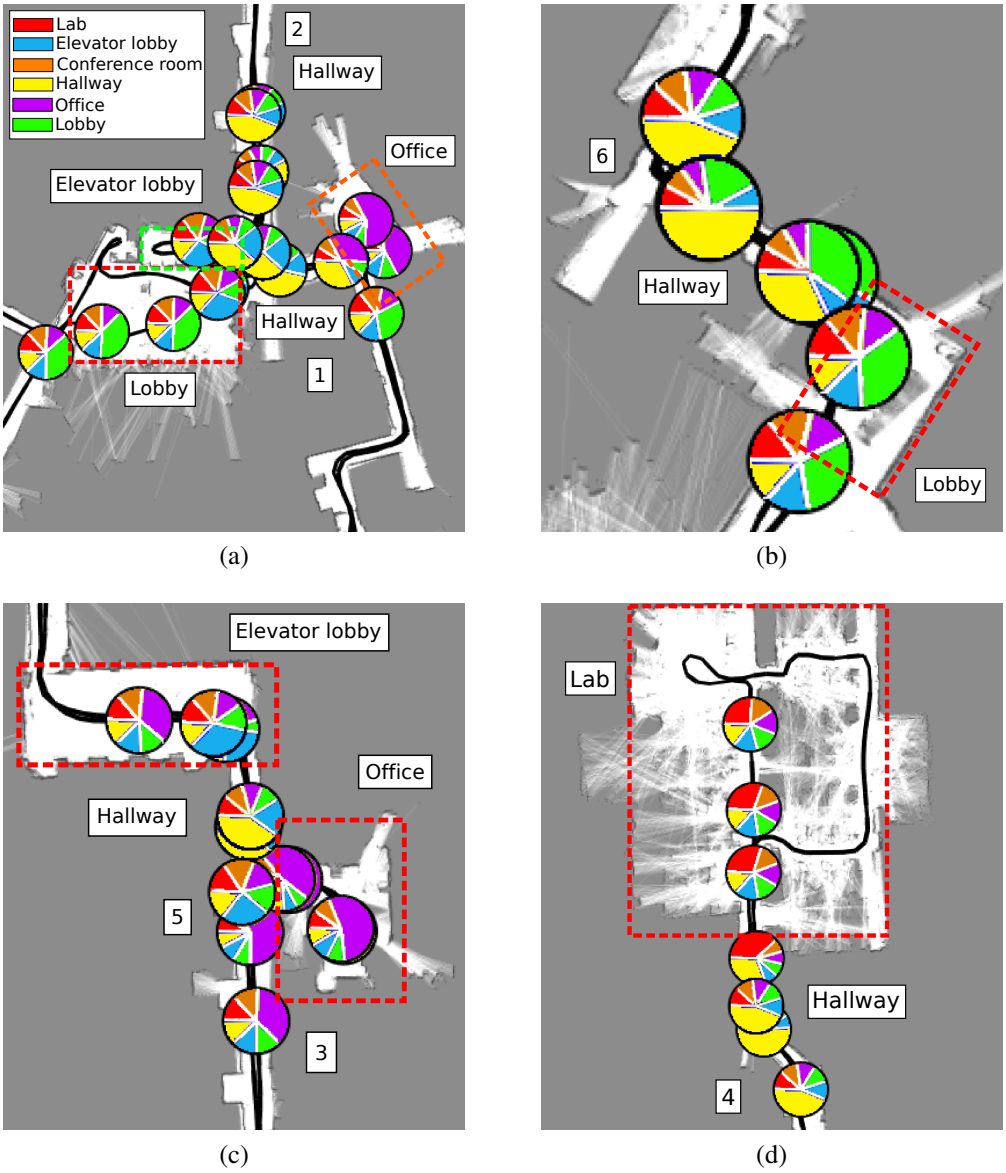


Figure 3-19: Inset views for the MIT 32-36-38 tour (Figure 3-16) that demonstrate the way in which the algorithm learns from allocentric descriptions (a) "The lobby is down the hallway" (anticipatory, location 1) and "The elevator lobby is down the hallway" (location 2), (b) "The lobby is down the hallway" (anticipatory), (c) "The office is near the hallway" (anticipatory, location 3) and "The elevator lobby is down the hallway" (location 5), and (d) "The lab is down the hallway" (anticipatory). The dashed boxes denote the ground-truth boundaries for the regions.

numbers "1" and "2," respectively. The former utterance was anticipatory as the robot had not yet visited the lobby area when the description was given. Nonetheless, the framework successfully labels that region of the environment when the robot later visits it, without

any aliasing effects. However, grounding the second utterance results in high likelihoods associated with some vertices that are not actually in the elevator lobby, causing the label to "bleed" into other areas. We attribute this to the ambiguity that results from not reasoning over frame-of-reference without which the vertices are consistent with being "down" the hallway. The performance improves for the anticipatory utterance in Figure 3-19d where the algorithm waits to infer the location of the lab until it is visited. We see similar effects for the descriptions in Figure 3-19c where the system correctly infers the location of another elevator lobby but attributes the "office" label to vertices that are actually in a hallway. This results from a simple set of features that encode the "near" relation based upon distance. Additionally, our algorithm uses a fixed separation to define regions and does not reason over their geometry (e.g., the shape of hallways is typically distinct from that of offices.) Meanwhile, Figure 3-13a depicts the semantic information inferred for the utterance "The lobby is through the entrance" from the large indoor/outdoor tour where we see that the algorithm correctly grounds the location of the lobby without any aliasing.

3.3.11 Robustness to Semantic Aliasing

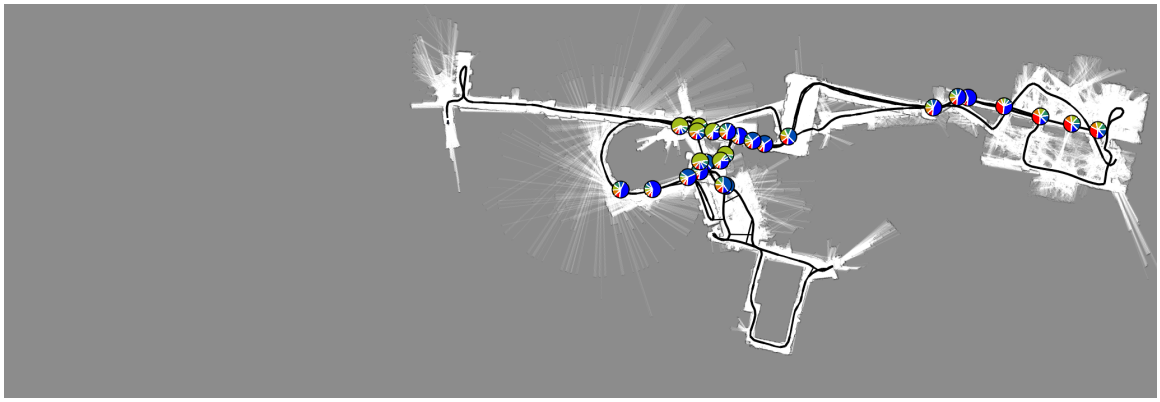
When proposing edges to the topology based upon the label distributions, we perform exhaustive scan-matching to check the validity of each proposed loop closure. While this helps to filter out the large majority of erroneous edges, the matching may yield false positives in regions that are perceptually aliased (Figure 3-20). However, since the hypothesis space of potential language edges is large, the likelihood that all particles sample invalid edges is low, confining such occurrences to a small subset of particles. Empirically, we have found that the weight of these particles is quickly reduced as their metric maps are inconsistent with subsequent sensor measurements. These particles then tend to be removed during resampling.

3.4 Discussion

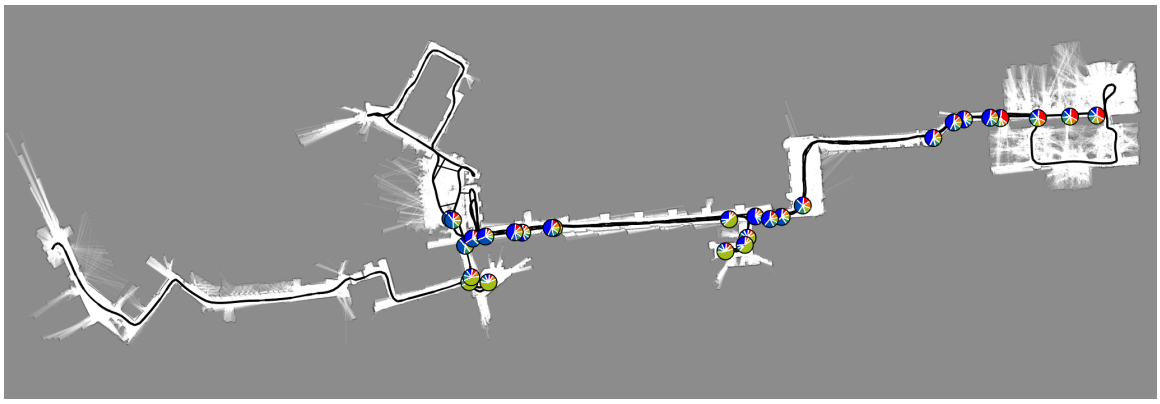
In this chapter we introduced our semantic mapping algorithm that estimates metrically accurate spatial-semantic maps from a user's natural language descriptions. The novelty lies

in learning the joint distribution over the metric, topological, and semantic properties of the environment, which enables the method to fuse the robot's sensor stream with knowledge inferred from the descriptions. We have presented results from several experimental evaluations that demonstrate the algorithm's ability to infer accurate metric, topological, and semantic maps. However, there are several limitations to our current approach.

A known issue with sample-based methods such as ours is the problem of particle depletion [15] whereby a majority of samples evolve to support regions of the distribution with negligible likelihood. This results in a poor approximation to the target distribution and can cause the filter to diverge. Resampling the particles based upon a measure of the



(a) Incorrect loop closure added



(b) No incorrect loop closures

Figure 3-20: A demonstration of the effects of perceptual aliasing for the MIT 32-36-38 tour (Figure 3-16) in which (a) the algorithm accepts an invalid edge between different regions that have similar geometry for one particle. However, the majority of the particles did not propose erroneous edges and the weight of this map soon decreases to $1/10^{th}$ of that of the correct particle and is removed upon resampling.

variance in their weights, as we do, reduces the likelihood of particle depletion. In practice, we have not found depletion to occur, as suggested by the results. We partially attribute this to using the distribution over the semantic map as part of the proposal, which reduces the frequency of erroneous samples. Nonetheless, particle depletion may occur and can be mitigated by adding additional particles to hypothesize new topologies in the event that the distribution appears to misrepresent the target distribution, for example, as suggested by the particle weights [23].

The performance of the algorithm is mainly dependent on two factors, namely the number of particles and the number of nodes in the topology. Increasing the number of particles will increase the computation linearly in the worst case. This is due to the fact that adding a new particle will increase the total number of edges sampled across all particles. However, because our implementation reuses the results of scan-matching procedures carried out for each sampled edge, if an edge between two nodes had already been sampled by another particle (using the same metric prior), we are able to reuse the previous result, saving significant computation. Because we create a new node every time the robot visits a region, the size of the graph grows with time, even if the robot revisits previous regions. This impacts the performance of the metric update that we make using the pose graph. It also increases the computation required to ground language, because revisited regions are represented with more than one node in the graph, leading to duplicated computation.

The algorithm described in this chapter relies exclusively on descriptions from the user to learn semantic information. This means that the algorithm can only model a region's label if it was specifically referenced by the user. Further, it precludes the method from incorporating allocentric descriptions for which the user never labels the landmark. For example, the algorithm can not learn from the description "The gym is down the hall" unless the user identifies the location of the hallway. The next chapter outlines our approach that alleviates this requirement by also using geometric- and appearance-based scene classifiers to infer semantic information from lidar and vision. This new algorithm allows us to use the relationship between a region's appearance and labels to induce a prior over the label distribution, allowing for better integration of language.

In this work, we instantiate regions in the environment at fixed distance intervals along

the robot's trajectory. This can result in regions that are not semantically meaningful, with multiple regions being used to model the same area. In this topology, when the robot revisits a region, the algorithm creates a new vertex and adds an edge to the old vertex. This can result in more than one vertex created to represent the same physical region. This can result in added complexity to the topology and additional ambiguity to the natural language grounding process. We address this in the next chapter by proposing an improved spatial representation.

In summary, we described an approach to learning human-centric maps of an environment from user-provided natural language descriptions. The novelty lies in fusing high-level information conveyed by a user's speech with low-level observations from traditional sensors. By jointly estimating the environment's metric, topological, and semantic structure, we demonstrated that the algorithm yields accurate representations of its environment.

Chapter 4

Semantic Maps from Natural Language and Scene Classification

In the previous chapter we described an algorithm that allows the robot to learn a spatial-semantic representation from natural language and its sensor observations. It uses natural language descriptions provided by a human to learn about semantic properties of the environment, and exploits this knowledge about the semantics of its environment to also improve its spatial representation.

In this chapter we outline an enhanced algorithm described in [27] that addresses several shortcomings in our previous approach. We introduce a more compact spatial representation better reflective of the local decomposition of the world. We also integrate additional sources of semantic information and improve the semantic representation to maintain a richer model. We use these improvements in the spatial and semantic representations to learn more effectively with natural language descriptions.

Improved Spatial Representation

Our previous algorithm decomposed the environment into a collection of fixed, uniformly-sized regions. This has the potential to result in a topology that is inconsistent with human concepts of space. Consequently, the representation may not model the spatial extent of regions referred to by the user's descriptions, leading to incorrect language groundings.

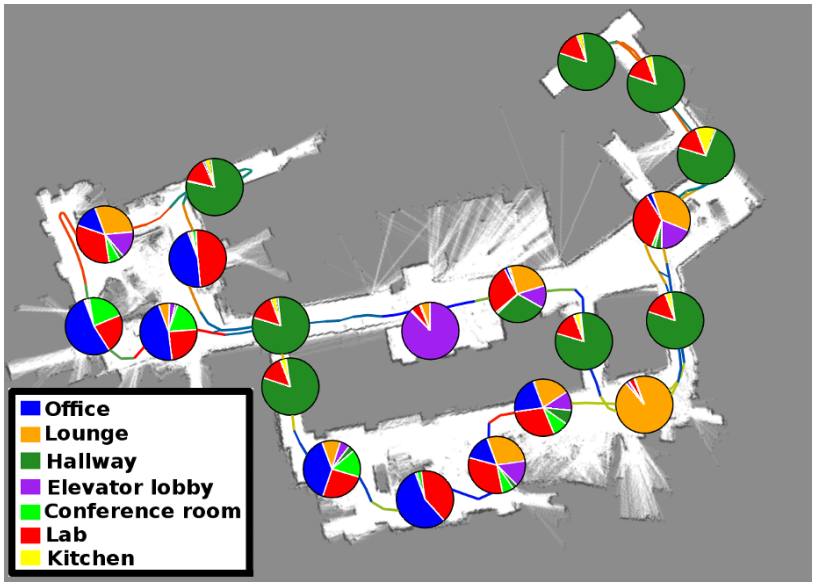


Figure 4-1: Maximum likelihood semantic graph particle of the 6th floor of Stata building (pie charts denote the likelihood of different region categories).

To overcome this, we integrate a laser-based spectral clustering approach to provide more accurate decomposition of spatial regions. We use this method to probabilistically reason over possible segmentations of regions in our environment in addition to its connectivity. More accurate spatial decompositions results in a more human-compatible representation suitable for learning from natural language.

Additionally, our prior approach models revisiting a region by creating a new vertex and adding an edge to the node created during the earlier visit. This leads to a topology that maintains multiple vertices to represent the same spatial region in the environment, where ideally it should use one vertex. This can result in unnecessarily large graphs leading to added complexity. It can also lead to suboptimal grounding of natural language due to the presence of aliased landmark regions that might be described by the user. The algorithm outlined in this chapter will merge the robot's current region with a previous region if it detects that they are the same region. This results in a more compact topology that only grows as the robot encounters previously unvisited parts of the environment.

Semantic Reasoning over Natural Language and Scene Classification

The semantic information in our prior approach was limited to user-provided colloquial names and did not provide a means to reason over properties such as region type that can be inferred from lidars, cameras, or other onboard sensors. Our new formulation integrates additional sources of semantic information, specifically, region appearance observations made using laser and image based appearance classifiers. By modeling the relation between an area's type and its colloquial name, the algorithm can reason over both region type and region label, even in the absence of speech. We also introduce a factor graph mechanism to maintain the semantic information, allowing us to reason about multiple semantic properties of each spatial entity, resulting in a richer learned model.

Enhanced Natural Language Integration

With the integration of additional sources of semantic information, the algorithm is capable of better grounding natural language descriptions by using the prior over spatial entities, unlike the earlier formulation which required even common landmarks to be described before hand. As our new spatial representation is more compact (no duplication of spatial entities to represent the same region in the environment) and more reflective of the spatial layout of the environment (due to our use of spectral clustering), the resulting natural language grounding is also more accurate as it arguably reasons over the correct set of spatial entities in the world.

4.1 Semantic Graph Representation

In the previous chapter, we introduced the semantic graph S_t, which contained topological T_t, metric X_t and semantic L_t representations of the environment. In this chapter, we introduce several enhancements to the semantic graph that allows us to learn a more accurate spatial and semantic representations. The following paragraphs outline our new topological and semantic representations.

The topology G_t is composed of nodes n_i that denote the robot's trajectory through the

environment (sampled at 1 m distances), node connectivity, and node region assignments. We associate with each node a set of observations that include laser scans z_i, semantic appearance observations a_i based on laser l_i and camera i_i models, and available language observations Λ_i. We assign nodes to regions $R_\alpha = \{n_1, .., n_m\}$ that represent spatially coherent areas in the environment compatible with human concepts (e.g., rooms and hallways). Undirected edges exist between node pairs in this graph, denoting traversability. Edges between regions are inferred based on the edges between nodes in the graph. A region edge exists between two regions if at least one graph edge connects a node from one region to a node in the other. The topological layer consists of the nodes, edges, and the region assignments for the nodes. A region R_i in this topology is equivalent to a vertex v_i in our previous definition in Chapter 3. However, now we reason about the extent of these regions by explicitly modeling their constituent nodes.

The pose $\boldsymbol{x_i}$ of each node n_i is represented in a global reference frame. The metric layer is induced by the topology, where edges in the topology also include metric constraints between the corresponding node poses. Metric constraints are calculated by scan-matching the corresponding laser observations of each region. A pose graph representation is employed to maintain the distribution over the pose of each node, conditioned on these constraints. Occupancy maps can be constructed based on the node poses and their corresponding laser observations. Figure 4-2 shows an example semantic graph particle for a trivial environment. As shown in the figure, semantic information is also conditioned on the topology. The semantic layer consists of a factor graph with variables that represent the type C_{R_i} and labels l_{R_i} for each region R_i, properties that can be observed at each node (in each region), and factors that denote the joint likelihood of these variables (e.g., the likelihood of observing a label given a particular room type). Observations of these region properties are made using laser- and image-based scene classifiers and by grounding human descriptions of the environment.

4.1.1 Distribution Over Semantic Graphs

We maintain the joint distribution over the topology G_t, the vector of locations X_t, and the set of semantic properties L_t. Formally, we maintain this distribution over semantic graphs $\{G_t, X_t, L_t\}$ at time t conditioned upon the history of metric exteroceptive sensor data $z^t = \{z_1, z_2, \ldots, z_t\}$, odometry $u^t = \{u_1, u_2, \ldots, u_t\}$, scene appearance observations $a^t = \{a_1, a_2, \ldots, a_t\}$, and natural language descriptions $\Lambda^t = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_t\}$,

$$p(G_t, X_t, L_t | z^t, u^t, a^t, \Lambda^t). \tag{4.1}$$

Each variable Λ_i denotes a (possibly null) utterance, such as "This is the kitchen," or "The gym is down the hall." In the work outlined in this chapter, the scene appearance observation $a_t = \{a_t^l, a_t^i\}$ is made up of image appearance a_t^i and laser appearance a_t^l. We describe this in detail in Section 4.2.3.

We factor the joint posterior into a distribution over the graphs and a conditional distri-

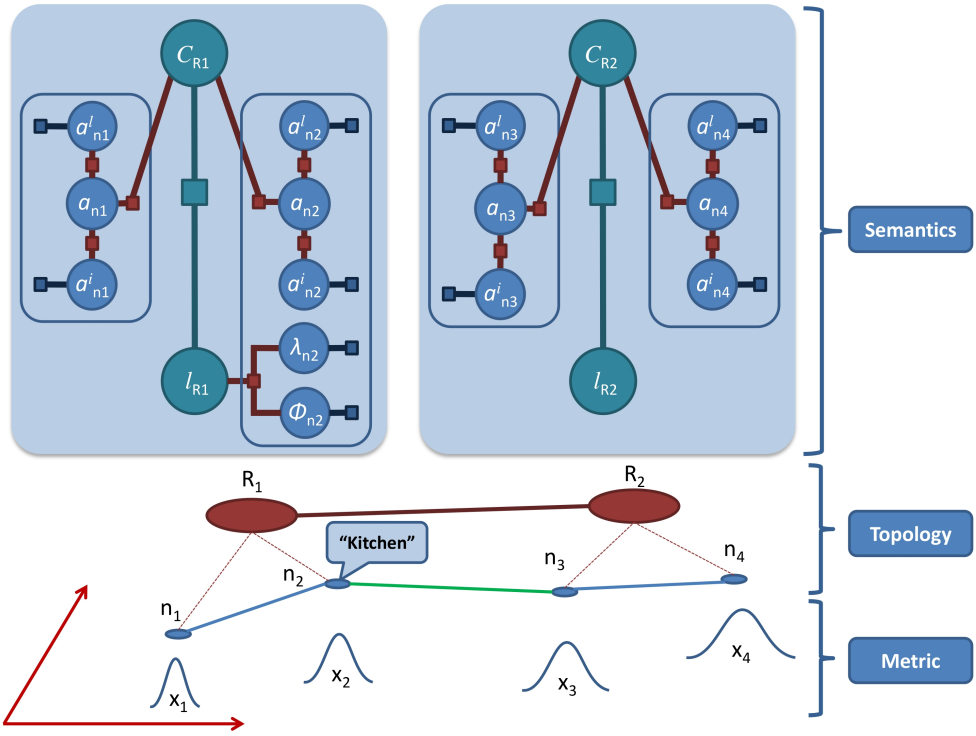


Figure 4-2: Example of a semantic graph particle: Two regions R_1 and R_2 and their constituent nodes n_i's; distributions over node poses $\boldsymbol{x_i}$; and the corresponding factor graph.

bution over the node poses and labels,

$$p(G_t, X_t, L_t | z^t, a^t, u^t, \Lambda^t) = p(L_t | X_t, G_t, z^t, a^t, u^t, \Lambda^t) \times p(X_t | G_t, z^t, a^t, u^t, \Lambda^t) \times p(G_t | z^t, a^t, u^t, \Lambda^t) \quad (4.2)$$

As with our original framework outlined in the previous chapter, we maintain this factored distribution using a Rao-Blackwellized particle filter. However, this distribution is now conditioned on additional types of robot observations.

We represent the joint distribution over the topology, node locations, and labels as a set of particles.

$$S_t = \{S_t^{(1)}, S_t^{(2)}, \ldots, S_t^{(n)}\}. \quad (4.3)$$

Each particle $S_t^{(i)} \in S_t$ consists of the set

$$S_t^{(i)} = \left\{ G_t^{(i)}, X_t^{(i)}, L_t^{(i)}, w_t^{(i)} \right\}, \quad (4.4)$$

where $G_t^{(i)}$ denotes a sample from the space of topologies, $X_t^{(i)}$ is the analytic distribution over locations, $L_t^{(i)}$ is the distribution over semantic properties, and $w_t^{(i)}$ is its weight.

4.2 Semantic Mapping Algorithm

Algorithm 2 outlines the process by which the method recursively updates the distribution over semantic graphs (4.2) to reflect the latest robot motion, metric sensor data, laser- and image-based scene classifications, and the natural language utterances. The following sections explain each step in detail.

4.2.1 The Proposal Distribution

We define the prior distribution over the topology G_t, given the posterior distribution at time step $t-1$. Each new topology particle $G_t^{(i)}$ is sampled from this proposal distribution, which is a predictive prior over the current graph given the previous graph particle, sensor

Algorithm 2: Semantic Mapping Algorithm

Input: $S_{t-1} = \left\{ S_{t-1}^{(i)} \right\}$, and $(u_t, z_t, a_t, \Lambda_t)$, where $S_{t-1}^{(i)} = \left\{ G_{t-1}^{(i)}, X_{t-1}^{(i)}, L_{t-1}^{(i)}, w_{t-1}^{(i)} \right\}$

Output: $S_t = \left\{ S_t^{(i)} \right\}$

for $i = 1$ *to* n **do**

1. Propagate the graph sample based on u_t, Λ_t and a_t using the proposal distribution.
   (a) Sample region allocation
   (b) Sample region edges
   (c) Merge newly connected regions
2. Update the Gaussian distribution over the node poses $X_t^{(i)}$ conditioned on topology.
3. Update the factor graph representing semantic properties for the topology based on appearance observations (a_t^l and a_t^i) and language Λ_t.
4. Compute the new particle weight $w_t^{(i)}$ based upon the previous weight $w_{t-1}^{(i)}$ and the metric data z_t.

end

Normalize weights and resample if needed.

data (excluding the current time step), appearance data, odometry, and language,

$$G_t^{(i)} \sim p(G_t | G_{t-1}^{(i)}, z^{t-1}, a^t, u^t, \Lambda^t). \tag{4.5}$$

First we augment the topology $G_{t-1}^{(i)}$ to reflect the robot's motion by adding a node n_t to to the current region R_c in the topology and an edge to the previous node n_{t-1}, resulting in an intermediate graph $G_t^{-(i)}$. This represents the robot's current pose and the connectivity to its previous pose. This yields an updated vector of poses $X_t^{-(i)}$ and semantic properties $L_t^{-(i)}$.

Then we sample modifications to each topology sample in three steps. Firstly, the algorithm samples a segmentation to the current region. If the current region was segmented, it then samples new edges between the latest segmented region and previously created re-

gions. Finally based on the newly created edges, the algorithm considers merging the last segmented region with the best matching connected region.

Creation of New Regions

We probabilistically bisect the current region R_c using the spectral clustering method proposed by Blanco et al. [4]. We construct the similarity matrix using the laser point overlap between each pair of nodes in the region. Equation 4.6 defines the likelihood of bisecting the region, which is based on the normalized cut value N_c of the graph involving the proposed segments. The likelihood of accepting a proposed segmentation rises as the N_c value decreases, i.e., as the separation of the two segments improves (minimizing the inter-region similarity),

$$P(s/N_{cut}) = \frac{1}{(1 + \alpha N_c^3)}. \tag{4.6}$$

This results in more spatially distinct areas in the world having a higher likelihood of being distinct regions, leading to more particles modeling these areas as separate regions. If a particle segments the current region, a new region R_i is created that does not include the newly added node. This method can however result in over-segmenting an environment when the local spatial layout is significantly cluttered.

Edge Proposals

When a new region R_i is created, the algorithm proposes edges between this region, and other regions in the topology, excluding the current region R_c.

The algorithm samples inter-region edges from a spatial-semantic proposal distribution that incorporates the semantic similarity of regions, as well as the spatial distributions of its constituent nodes. This reflects the notion that regions that are nearby and semantically similar are more likely to be connected. We measure semantic similarity based upon the

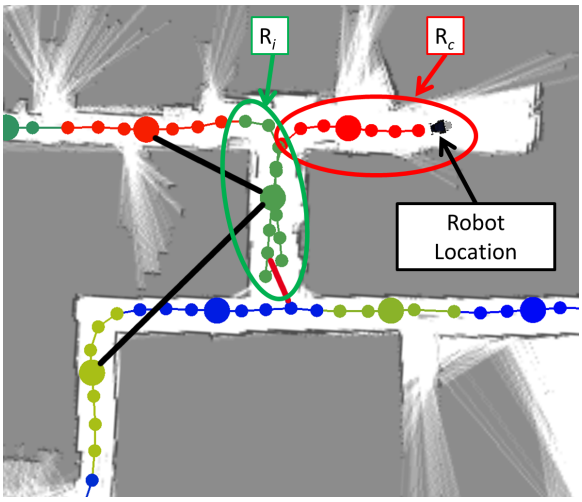


Figure 4-3: Example of region edges being proposed (black lines represents rejected edge proposals; the red line represents an accepted edge).

label distribution associated with each region. The resulting likelihood has the form

$$p_a(G_t|G_t^{-(i)}, z^{t-1}, u^t, a^t, \Lambda^t) = \prod_{j:e_{ij}\notin E^-} p(G_t^{ij}|G_t^{-(i)}) \tag{4.7a}$$

$$\propto \prod_{j:e_{ij}\notin E^-} p_x(G_t^{ij}|G_t^{-(i)}) p_s(G_t^{ij}|G_t^{-(i)}), \tag{4.7b}$$

where we have omitted the history of language observations Λ^t, metric measurements z^{t-1}, appearance measurements a^t, and odometry u^t for brevity. Equation 4.7a reflects the assumption that additional edges that express constraints involving the current node $e_{ij} \notin E^-$ are conditionally independent. While $p_x(G_t^{ij}|G_t^-)$ encodes the likelihood of the edge based on the spatial properties of the two regions, $p_s(G_t^{ij}|G_t^-)$ describes the edge likelihood based on the regions' semantic similarity. Equation 4.7b reflects the assumed conditional independence between the spatial- and the semantic-based edges.

For the spatial distribution prior, we consider the distance d_{ij} between the mean nodes of the two regions, where the mean node is that with its pose closest to the region's average

pose

$$p_x(G_t^{ij}|G_t^{-(i)}) = \int_{X_t^{-(i)}} p(G_t^{ij}|X_t^{-(i)}, G_t^{-(i)}, u_t)p(X_t^{-(i)}|G_t^{-(i)}) \tag{4.8a}$$

$$= \int_{d_{ij}} p(G_t^{ij}|d_{ij}, G_t^{-(i)})p(d_{ij}|G_t^{-(i)}). \tag{4.8b}$$

The conditional distribution $p(G_t^{ij}|d_{ij}, G_{t-1}, z^{t-1}, u^t)$ expresses the likelihood of adding an edge between regions R_t and R_j based upon the location of their mean nodes. We represent the distribution for a particular edge between regions R_i and R_j with distance $d_{ij} = |\bar{X}_{R_i} - \bar{X}_{R_j}|_2$ as

$$p(G_t^{ij}|d_{ij}, G_t^-, z^{t-1}, u^t) \propto \frac{1}{1 + \gamma d_{ij}^2}, \tag{4.9}$$

where γ specifies a distance bias. For the evaluations in this chapter, we use $\gamma = 0.3$. We approximate the distance prior $p(d_{ij}|G_t^-, z^{t-1}, u^t)$ with a folded Gaussian distribution.

The semantic prior expresses the increased likelihood that edges exist between regions with similar distributions over labels l. The label distributions for the regions are modeled in the semantic layer,

$$p_s(G_t^{ij}|G_t^-) = \sum_{L_t^-} p(G_t^{ij}|L_t^-, G_t^-)p(L_t^-|G_t^-) \tag{4.10a}$$

$$= \sum_{l_i^-, l_j^-} p(G_t^{ij}|l_i^-, l_j^-, G_t^-)p(l_i^-, l_j^-|G_t^-). \tag{4.10b}$$

Equation 4.11 expresses the likelihood of an edge existing between two regions, given the value of the regions' respective label values

$$p(G_t^{ij}|l_i, l_j) = \begin{cases} \theta_{l_i} & \text{if } l_i = l_j \\ 0 & \text{if } l_i \neq l_j \end{cases}, \tag{4.11}$$

where θ_{l_i} denotes the likelihood that edges exist between nodes with the same label. In practice, we assume a uniform saliency prior for each label. Equation 4.10b measures the

cosine similarity between the label distributions.

After a region edge is sampled from the spatial-semantic prior, a scan-match procedure attempts to find the best alignment between the two regions. Upon convergence of the scan-match routine, the edge is accepted and is used to update the topology.

Region Merges

After a new region R_i has been created and edges to other regions have been checked and added, the algorithm determines whether it is possible to merge with one of the newly connected regions. The newly-created region is merged with an existing (connected) region if the observations associated with the smaller of the two regions can be adequately explained by the larger region. This results in regions being merged when the robot visits a location already represented in the graph.

The merge process is designed to ensure that each particle maintains a single region entity to represent each spatial region, thus ensuring that the complexity of the topology increases only when the robot explores new areas, leading to more efficient region edge proposals as well as more accurate language groundings.

4.2.2 Updating the Metric Map Based on New Edges

The algorithm then updates the spatial distribution over the node poses X_t conditioned on the proposed topology,

$$p(X_t^{(i)}|G_t^{(i)}, z^t, a^t, u^t, \Lambda^t) = \mathcal{N}^{-1}(X_t^{(i)}; \Sigma_t^{-1}, \eta_t), \tag{4.12}$$

where we parametrize the Gaussian in the canonical form in terms of the information matrix Σ_t^{-1} and information vector η_t. As in the previous chapter, we make use of the iSAM algorithm [38] to incrementally update this distribution for each particle.

4.2.3 Updating the Semantic Layer

Compared with our original formulation in the previous chapter, our updated representation maintains a distribution over a larger set of semantic properties associated with the environment. The distribution over the semantic layer is maintained using a factor graph [63] that is conditioned on the topology for each particle. As Figure 4-4 shows, the semantic layer maintains two variables associated with each region R_i, namely the region category (or type) C_{R_i} (e.g., *hallway*, *conference room*) and the label that can be used to describe the region l_{R_i}.

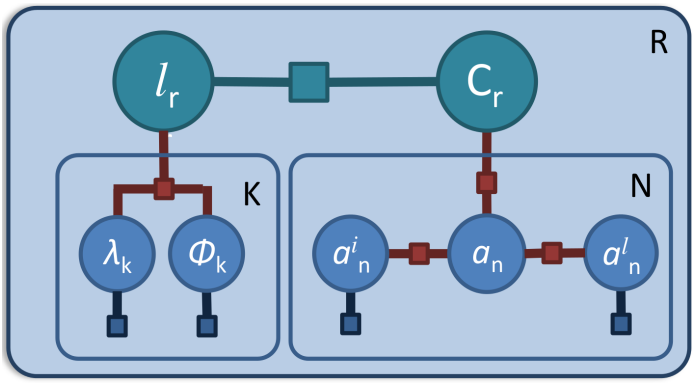


Figure 4-4: Semantic Layer (plate representation) for a region

We model the relation between a region's category C_{R_i} and its label l_{R_i} to account for the fact that people describe certain region categories more often with particular labels. For example, while a person might describe a region of type *conference room* with a label "meeting room" or "conference room," it is unlikely that they will describe it as a "kitchen." The factor that joins these two variables represents the likelihood of each room category generating a particular label. In this implementation, we identified a limited subset of labels associated with each region category in our representation (e.g., the *hallway* category can generate "hall," "hallway," or "corridor"). When building the factor between the label and room category variables, we assigned higher likelihoods to labels associated with each category and lower likelihoods to the other labels (capturing the likelihood of generating these labels given a particular room category). Ideally this factor should be learned using data regarding labels people use to describe different types of regions.

At each node n within a region, the robot can observe one or more of these semantic

properties, either directly or indirectly. We make the assumption that these observations are conditionally independent. The robot learns about the region label l_{R_i} when it grounds a phrase λ_k that the person used to describe that region (Φ_k is a correspondence variable that denotes whether λ_k actually references that region). The robot learns about the region category C_{R_i} indirectly, by observing the region appearance a_n at each node in a region. The region appearance can be one of three broad appearance classes, *room*, *hallway*, and *open area*. These are observed using the robot's lidar a_n^l and camera a_n^i (with the use of trained appearance models) at each node in the region. For each particle, we update the distributions over the region label and category variables by running belief propagation at each time step as new variables and factors are added.

Scene Appearance Observations

Each node has an appearance variable a_n that is related to its region category. We consider three general appearance classes, *room*, *hallway*, or *open area*. The factor that connects a region category variable C_{R_i} to an appearance variable a_n encodes the fact that a region of a particular type often has a distinctive appearance. For example, a region of type *office* has a high-likelihood of having a *room* appearance. The category-to-appearance factor was learned from annotated data from several other floors of the Stata building.

The robot makes two observations of the region's appearance a_n at node n using its lidar and rgb-camera combined with pre-trained appearance models.

1. Laser Appearance (a_n^l): The laser appearance model uses geometric features derived from the laser observations similar to those outlined in Mozos et al. [64].

2. Image Appearance (a_n^i): The image appearance model uses Composed Receptive Field Histograms (CRFH) [70].

We train these appearance models using Support Vector Machines [7] to carry out multi-class classification with probability estimates. Laser and camera appearance variables a_n^l and a_n^i are connected to the node's appearance a_n using factors built from the confusion matrix for the two trained models. The classification results for the two sensors provide a distribution representing the likelihood of the observations being generated from each

appearance category. The classifier outputs are integrated to the factor graph as factors attached to variables a_n^l and a_n^i.

Natural Language Descriptions

Each region in the topology has a label variable l_{R_i} that represent how people refer to that region (e.g., "meeting room"). We learn the distribution over the labels by integrating the user's descriptions about the environment. These are either ego-centric descriptions that describe the robot's immediate location (e.g., "I am at the kitchen") or allocentric descriptions that refer to spatially distant regions (e.g., "The kitchen is down the hall").

A region can have zero, one, or multiple label observations depending on the number of descriptions made by the user about that region. We represent each relevant observation of the region's label with λ_k and a correspondence variable Φ_k. The variable λ_k denotes the label used by the user in the description when referring to the region. The variable Φ_k is a binary-valued variable specifying whether or not λ_k describes the region. If the label does not correspond to that region ($\Phi_k = 0$), the observation λ_k is uninformative about the region's label, and will have equal likelihood for each label value. However, when the correspondence holds ($\Phi_k = 1$), the factor encodes the likely co-occurrences between different labels. For example, if the robot heard the label "conference room" with a high likelihood of $\Phi_k = 1$, it will result in other labels that often co-occur with "conference room" (e.g., "meeting room") as having high likelihoods as well. Currently, high co-occurrence is added for words that are synonyms (e.g., "hallway" and "corridor"). In this way, we use the correspondence variable to handle the ambiguity inherent in grounding natural language descriptions. When a label is grounded to a region, we create a label observation λ_k and correspondence variable Φ_k, and connect it to the associated region's label variable l_{R_i} using a co-occurrence factor. We integrate the correspondence observation Φ_k by attaching a factor encoding this likelihood. We treat the observed label as having no uncertainty, as our current model does not model errors arising from speech recognition.

We derive the factor denoting the observation of Φ_k based on the type of description given by the user. If the user describes the current location (e.g., "I am at the living room"), we have higher correspondence with spatially local nodes. For such descriptions, we al-

locate a high likelihood of correspondence with the current region , i.e., $p(\Phi_k = 1) = 0.8$. For allocentric descriptions (e.g., "the kitchen is down the hallway", where the user describes the location of the referent "kitchen" with relation to the landmark "hallway"), we use the G^3 framework [88] to calculate the correspondence likelihood given the potential landmarks. We marginalize the landmarks to arrive at correspondence likelihood of each referent region in a manner similar to our previous approach.

$$p(\phi^{\mathcal{F}}_{R_i} = \mathtt{T}) = \eta \sum_{R_j} p(\phi^{\mathcal{F}}_{R_i} = \mathtt{T} | \gamma_{\mathcal{L}} = R_j, \gamma_{p_i}, \lambda^r)\, p(\gamma_{\mathcal{L}} = R_j), \tag{4.13}$$

where $\phi^{\mathcal{F}}_{R_i}$ is the correspondence variable for the figure, η is the normalization factor and γ_{p_i} represents the shortest path p_i that the robot can take from the location of the description through the pair of landmark γ^l_j and figure γ^f_i vertex groundings. As before, we use the A* algorithm [79] to solve for the shortest path through the topology.

However, unlike our approach outlined in the previous chapter (see Section 3.2.3), we normalize all valid figure groundings to arrive at the observation of the correspondence variable $\Phi_k = 1$. This reflects the fact that there should only be one correct spatial entity that is referred to by the description on the assumption that the topology represents an accurate decomposition of the world. We did not enforce this normalization in our prior approach due to two reasons. Firstly, due to the fixed spatial segmentation that we employed in the prior approach, there was a higher likelihood that the correct spatial segmentation might not be reflected in the topology. Secondly, due to the fact that we created new nodes when the robot revisits regions, if we applied the normalization, only one of the valid region nodes would receive the probability of the figure, even though they both are the correct figure region. But with our new more compact and semantically accurate spatial representation, we feel is fair to assume that there should ideally be one valid figure region referred to by the user when providing the description. Admittedly, it is possible that in cluttered environments for a region to be over-segmented resulting in more than one spatial entity being created.

Compared to our approach outlined in the previous chapter, we make several improvements in integrating semantic information from allocentric language to our representation.

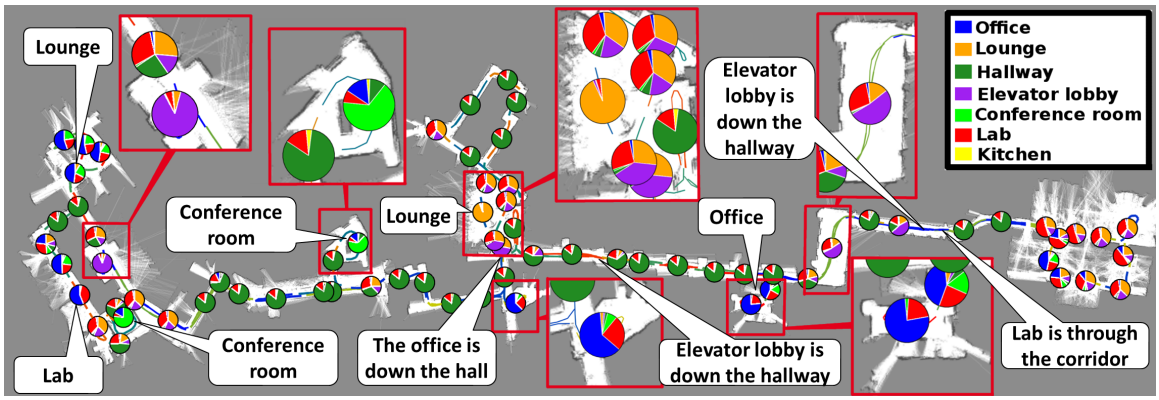


Figure 4-5: Maximum likelihood semantic graph of a multi-building environment on the MIT campus.

The grounding is more efficient due to the more compact topology and more accurate because the spatial regions are arguably closer to what the human thinks of as regions. By using the richer semantic representation, we are also able to induce a prior over the set of landmarks and figures even in the absence of language. For example, we can identify landmark regions such as hallways which provide visually distinct cues for laser and image classifiers, even in the absence of explicit descriptions. The presence of visual appearance observations also acts as a prior over the label distributions.

4.2.4 Updating the Particle Weights and Resampling

We update the particle weights and resample in the same manner as in Section 3.2.4.

4.3 Results

We evaluate our algorithm through four experiments in which a human gives a robotic wheelchair [30] narrated guided tours of different floors in the Stata Center (S3, S4, S6) as well as a multi-building indoor tour (MF) on the MIT campus. The robot was equipped with a forward-facing lidar, a camera, wheel encoders, and a microphone. In these experiments we drove the robot using a joystick, and provided it with textual natural language descriptions at specific salient locations.

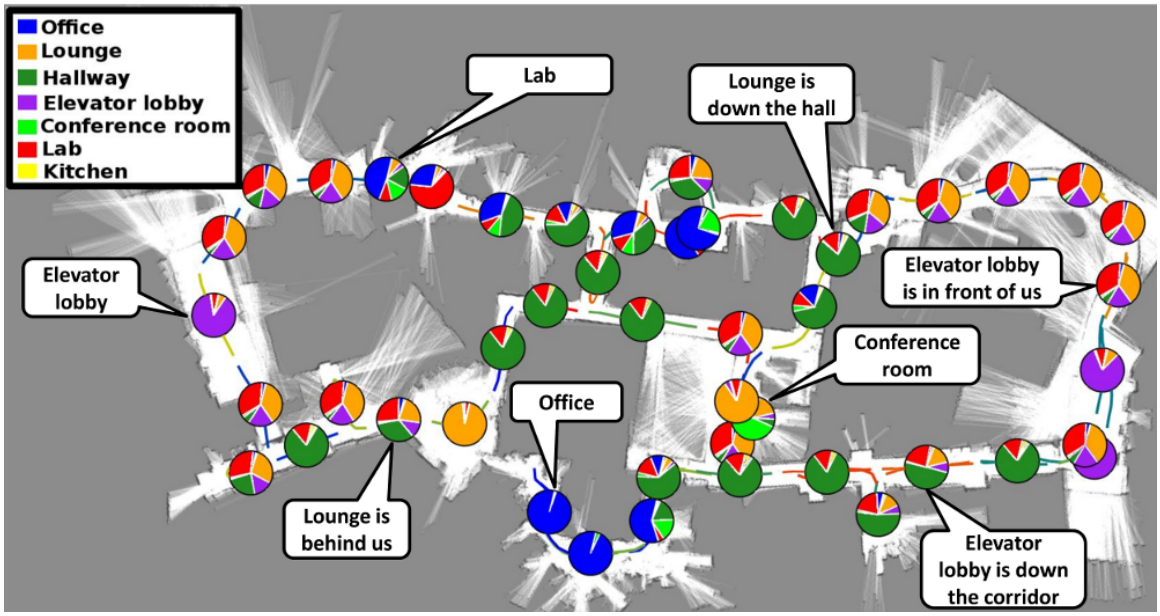


Figure 4-6: Maximum likelihood semantic map of the 3rd floor of the Stata building (pie charts denote the likelihood of different region categories).

We evaluate the resulting semantic maps with regards to their topological accuracy, compactness, segmentation accuracy, and semantic accuracy. All experiments were run with 10 particles. The results show that our framework produces more compact and more accurate semantic graphs than our previous approach. They also demonstrate the improvement in semantic accuracy due to language descriptions. We also show the ability of our framework to ground allocentric language even in the absence of previous labels for the referent (e.g., it handles the expression "the lobby is down the hall" even when the hall has not been labeled).

4.3.1 Topological Accuracy

We compare the topological accuracy, conditioned upon the resulting segmentation, by comparing the maximum likelihood map with the ground truth topology. We define a topology as matching ground truth if node pairs that are spatially close (1 m) in a metrically accurate alignment are at most one region hop away. This avoids penalizing occasional regions that do not contain valid edges as long as a nearby region was accurately connected

(and possibly merged with the nodes from a prior visit). This can happen when an edge was not sampled or when scan-matching failed to converge.

The percentage of close node pairs that were more than one region hop away from each other for the third, fourth and sixth floor were 2.8%, 3.7%, and 3.8%, respectively. Most region-matching errors occurred in areas with significant clutter which caused over-segmented spatial regions, which do not look similar enough to each other. Metric maps derived from the maximum likelihood particles were accurate for all three floors.

4.3.2 Topological Compactness

We compare the allocation of nodes to regions in the current framework to the previous method. In the previous approach, the topology update did not merge regions even when the robot revisited a region; it simply created an edge between the regions. The redundancy of the regions has several negative implications. Firstly, it unnecessarily increases the hypothesis space of possible region edges, reducing the likelihood of a sample proposing valid region edges. Secondly, it increases the hypothesis space for grounding language, forcing the framework to consider more region pairs as possible groundings for user descriptions.

We measure the duplicity of the region allocation as

$$\mathbb{C} = N_s/N_t, \tag{4.14}$$

where N_s is the number of close node pairs ($< 1\,m$) assigned to the same region and N_t is the total number of close node pairs. If the topology is efficient at allocating regions, this ratio should be high, as only nodes near region boundaries should belong to different regions. Table 4.1 compares these scores for three different floors. The new method scores significantly higher in all three experiments. The difference is more pronounced when the robot revisits more regions. Since the sixth floor dataset did not have too many revisited regions, the scores for the two approaches are closer.

Table 4.1: Region allocation efficiency ($\mathbb{C}$)

Floor	New Framework	Old Framework
Stata Floor 3 (S3)	.67	.29
Stata Floor 4 (S4)	.77	.37
Stata Floor 6 (S6)	.69	.52

Table 4.2: Region Segmentation and Semantic Accuracy

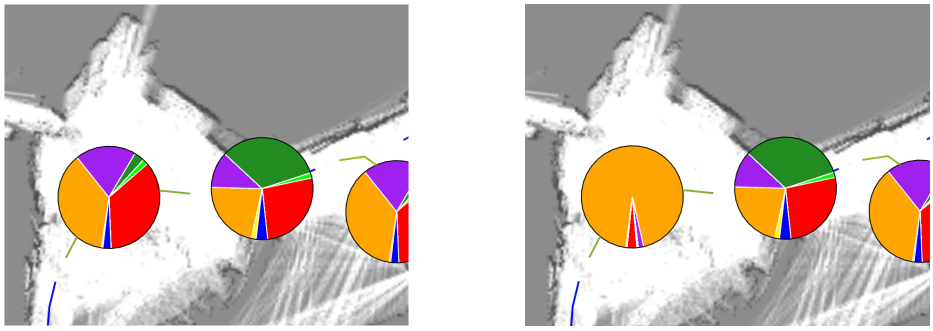
Region Type	Segmentation Accuracy		Semantic Accuracy			
			Without Lang		With Lang	
	S3	MF	S3	MF	S3	MF
Conference room	80.0	81.7	8.8	15.1	48.5	58.7
Elevator lobby	59.7	72.8	18.8	12.8	64.1	46.4
Hallway	49.4	55.7	44.5	58.5	44.4	58.0
Lab	52.8	30.1	11.8	27.2	14.2	30.6
Lounge	42.9	39.4	28.6	36.6	62.0	40.5
Office	62.5	76.1	78.1	45.6	98.6	60.2

4.3.3 Segmentation Accuracy

Table 4.2 outlines the segmentation accuracy for the maximum likelihood particle for two datasets, outlined according to region type. We picked the best matches based on the Jaccard index (number of intersecting nodes divided by the number of union nodes) for each ground truth annotated region and the resulting segmented region. Since our segmentation method depends on the similarity of laser observations, large cluttered region types, such as lab spaces and lounges, tend to be over-segmented. Additionally long hallways tend to be over-segmented by our method, which is reflected in the lower scores for hallways.

4.3.4 Inference of Semantic Properties

Table 4.2 also outlines the semantic accuracy for the maximum likelihood particle for two datasets. Semantic accuracy was calculated for each ground truth region by assigning each constituent node with its parent region's category distribution and taking the cosine similarity. We observe that the semantic accuracy with language is higher for most region types, with the exception of hallways that show minimal improvement since they were



(a) Appearance only (b) With language

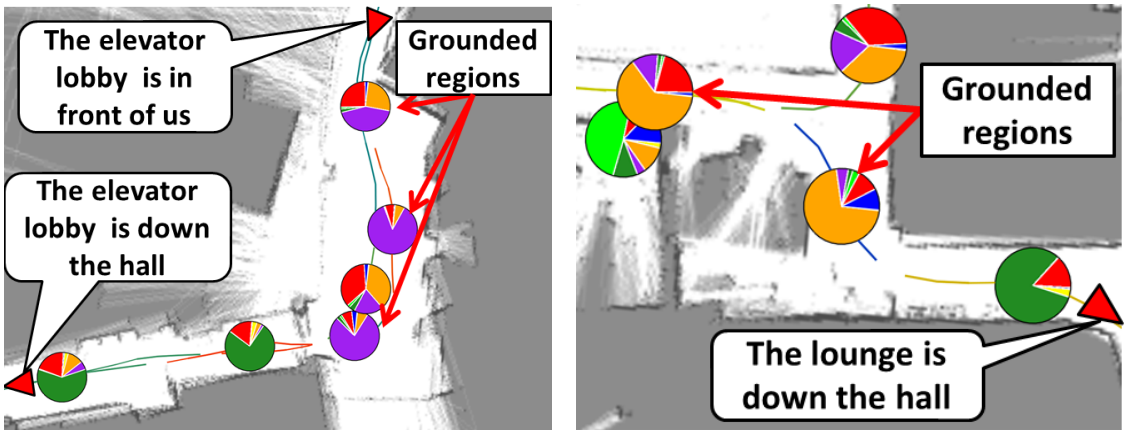
Figure 4-7: Region category distribution (a) for a region with only appearance information and (b) and with both appearance and language "the lounge is behind us". (category "lounge": yellow).

rarely described by users. Some regions, such as labs, which were labeled with egocentric descriptions, have low scores because the regions are over-segmented and the language is attributed only to the current region. In these experiments the hallway regions were not labeled through language but were inferred based on scene appearance observations from the robot's sensors. Figure 4-7 compares the region category properties with and without language. In the absence of language (Figure 4-7a), the appearance of the region gives equal likelihood for both "elevator lobby" and "lounge." In Figure 4-7b, the region was grounded with the label "lounge" and the framework inferred a higher likelihood of the region category being a lounge.

4.3.5 Grounding Allocentric Language Descriptions

We also tested our framework with allocentric language descriptions. When handling phrases that include a landmark and a referent (e.g., "the gym is down the hall"), our earlier framework required the landmark to have already been labeled before describing the referent location. With our new framework, the robot is able to ground language when the landmark corresponds to locations that may not have been labeled, but can be inferred from other semantic cues (e.g., appearance classification). We tested this situation using several instances in our dataset.

Figure 4-8 shows instances in which allocentric language utterances were grounded into



(a) (b)

Figure 4-8: Resulting category distribution for allocentric language phrases (a) "the elevator lobby is down the hall" and (b) "the lounge is down the hall" ("elevator lobby": purple, "lounge": yellow, "hall": green). The "hall" regions were inferred using scene appearance to learn the location of the "elevator lobby" and the "lounge" respectively.

the semantic graph. As the label distributions for the surrounding regions demonstrate, the framework is able to ground the referent with a high degree of accuracy, even though the landmark was never explicitly labeled. However, since there is more uncertainty about the landmark region, the information derived from the allocentric language has less influence on the semantic properties on the region (since we marginalize the landmark likelihood when calculating the grounding likelihood Φ).

4.4 Discussion

In this chapter, we described an enhanced semantic mapping algorithm that learns spatial-semantic representations of environments from natural language descriptions and scene classifications. This results in more compact spatial representations that are closer to how humans perceive environments and richer semantic maps that are more conducive to learning semantic properties from natural language.

Because the algorithm outlined in this Chapter samples different possible segmentations in each particle, its performance increase is somewhat poorer than the approach in Chapter 3. This is due to the fact that different particles in this approach can have differing

numbers of regions as well as edges, where the regions across particles are no longer the same. Thus we are no longer able to share the results of scan-matching across different particles, leading to somewhat worse scaling.

The algorithms outlined in this Chapter and Chapter 3 have made use of natural language to learn labels by reasoning about the information contained in natural language. However, both these methods require the robot to visit the referred regions before it can learn from language. In Chapter 6 we use natural language to directly extend the robot's spatial representation by creating regions (as yet unvisited by the robot) based on natural language descriptions and inferring weak metric constraints based on spatial relations.

These algorithms model the scenario of a guided tour, where a human guide provides a tour of the environment. However, up to now the robot has been a passive participant in how it learns about the world. In the next chapter, we introduce a mechanism with which the robot reasons about the ambiguity of the natural language descriptions provided by the user and its current semantic map, and then ask questions from the user to improve its representation.

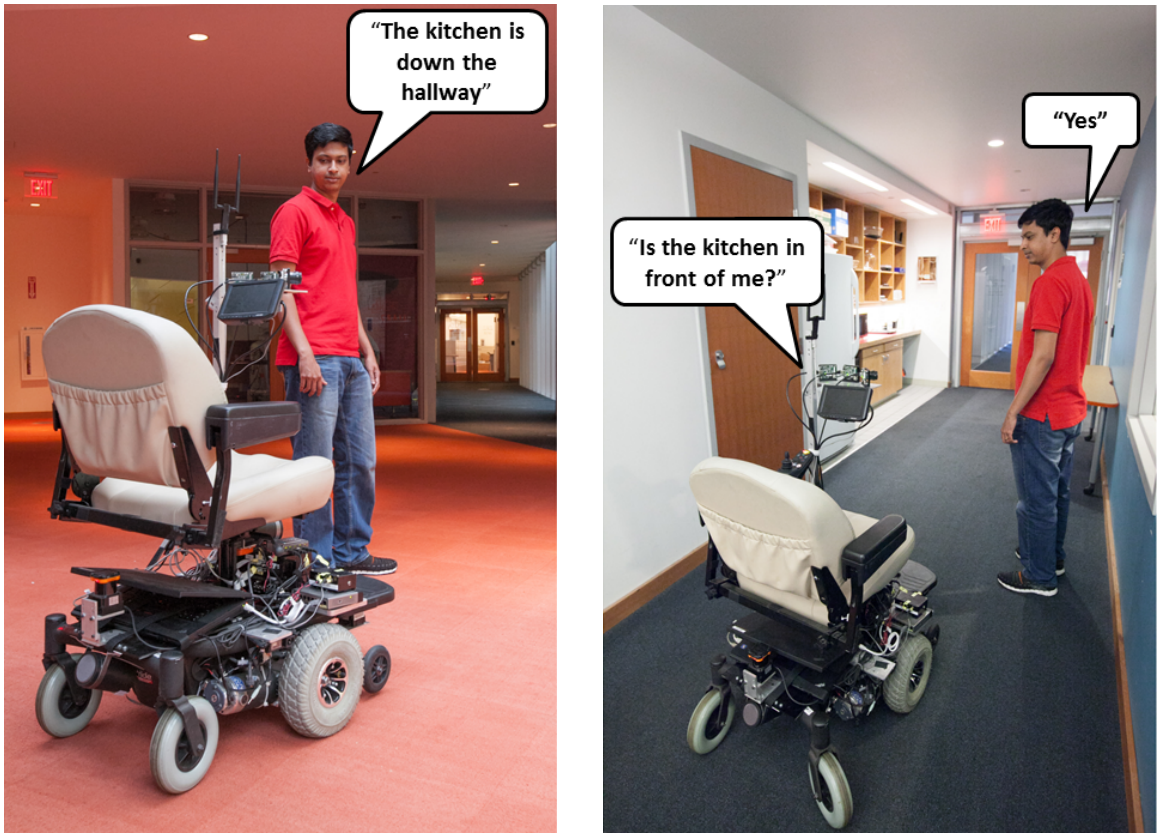
Chapter 5

Information Theoretic Question Asking to Improve Semantic Maps

In the previous chapters we outlined our algorithms that allowed a robot to learn a spatial-semantic representation from natural language and its sensors during a guided tour provided by a human. While the robot would autonomously follow the human while exploring the environment, the higher-level decision of which areas would be explored was decided by the human. The robot was a passive partner when any natural language description was provided, only confirming the description before integrating it to the representation.

One challenge to learning from natural language descriptions is the higher level of ambiguity that such descriptions presents. The user's descriptions can often be ambiguous, with several possible interpretations for a particular environment. For example, the user may describe the location of the kitchen as being "down the hall," yet there may be several hallways nearby, each leading to a number of different rooms.

Rather than try to passively resolve the ambiguity in the inferred map, the robot can actively take information-gathering action that can improve its representation. These could be in the form of physical exploration of the environment to validate the presence and location of spatial entities described by the human, or by asking targeted questions from the user. In this chapter we take the latter approach, by enabling the robot to ask questions that disambiguate its uncertainty over the implications of natural language descriptions provided by the guide. Figure 5-1 shows such a situation where the guide provides an



(a) (b)

Figure 5-1: A user gives a tour to a robotic wheelchair designed to assist residents in a long-term care facility. (a) The guide provides an ambiguous description of the kitchen's location. (b) When the robot is near one of the likely locations, it asks the guide a question to resolve ambiguity.

ambiguous description, which the robot clarifies by asking a question at a later time.

Challenges

In order for this approach to be useful it needs to balance the robot's need to follow the tour without unnecessary interruptions, and the need to improve the robot's representation by asking questions. To achieve this, the algorithm needs to decide what is the best question to ask and when is the best time to ask a question. It needs to overcome several challenges in order to reason about questions successfully.

The robot needs ask questions that provide enough context to the guide. Questions that query the user about the robot's current location can provide a lot of context but are

of limited use. Firstly, there is limited opportunity to ask these questions about relevant regions. Secondly asking the question is most useful if the robot expects to receive a yes answer, because a no answer often provides minimal information about any other region. Thus it is useful to ask questions about temporally and spatially distant regions. However, asking such questions requires the robot to provide sufficient context for the human to comprehend and provide a meaningful answer. This context can be provided with the use of spatial relations that refer to non-immediate part of the environment (e.g., "Is the kitchen in front of me?") , or by referring to salient landmarks in the environment (e.g., "Is the lounge near the conference room?").

The robot also needs to balance the utility of asking a questions with following the tour with minimal interruptions. However, the robot is forced to evaluate the utility of asking a question based on a possibly incomplete map of its environment. To calculate the utility of asking a question would require reasoning over how the user might respond to the question, and the likelihood of receiving a particular response. This also needs to model the cost of asking questions that can be burdensome to the human, for example by frequently asking questions.

To address this, in this chapter we formulate a guided tour model as a decision process which decides between following the human and asking questions to improve its representation [32, 29]. During the tour, the robot maintains a distribution over the semantic graph representation that we introduced in the previous chapter. The algorithm reasons over the natural language descriptions and the current learned map to identify potential questions that best reduces ambiguity in the map. The algorithm considers egocentric and allocentric binary (yes/no) questions that consist of spatial relations between pairs of regions. These regions may be local to the robot in the case of situated dialog (e.g., "Are we in the kitchen?", "Is the lab on my right?") or distant in the case of non-situated dialog (e.g., "Is the lounge next to the conference room?"). We evaluate this approach over its ability to resolve ambiguity using two experiments and demonstrate that the resulting semantic maps are more accurate than the current state-of-the-art.

5.1 Semantic Mapping Algorithm

During the guided tour the robot constructs a spatial-semantic representation based on the algorithm we outlined in the previous chapter. The representation we employ is identical to the one outlined in the previous chapter. However, unlike in the previous chapter where each particle sampled a segmentation using spectral clustering (as outlined in Section 4.2.1), in this work we deterministically infer region transitions using a threshold on the spectral clustering method, resulting in all particles having the same region boundaries.

In the following section we reiterate the mechanism that we use to integrate natural language descriptions to our representation, and highlight how ambiguity in the descriptions can affect the utility of the information inferred from natural language. Next, we outline an improved natural language evaluation process that efficiently reevaluates the implications of each utterance as relevant parts of the environment are visited during the tour.

5.1.1 Grounding Natural Language Descriptions

We consider two broad types of natural language descriptions provided by the guide. Egocentric descriptions that involve the robot's immediate surround are directly grounded to the robot's current region when the description was provided. Allocentric descriptions that provide information about distant regions require more careful handling.

We parse each natural language command into its corresponding Spatial Description Clauses (SDCs), a structured language representation that includes a figure, a spatial relation and possibly a landmark [88]. For example, the allocentric description "the lounge is down the hallway," results in an SDC in which the figure is the "lounge," the spatial relation is "down from," and the landmark is the "hallway". With egocentric descriptions, the landmark is implicitly the robot's current position. Thus we treat each language description as containing a figure region $\gamma_{\mathcal{F}}$, spatial relation and a potential landmark region $\gamma_{\mathcal{L}}$.

In order to ground each expression, the algorithm first identifies regions in the map that may correspond to the grounding based upon their semantic label likelihood. We normalize

these likelihoods to compute the landmark grounding probability for each of these regions

$$p(\gamma_{\mathcal{L}} = R_j) = \frac{p(\phi^{\mathcal{L}}_{R_j} = \mathtt{T})}{\sum_{R_j} p(\phi^{\mathcal{L}}_{R_j} = \mathtt{T})}, \tag{5.1}$$

where $\gamma_{\mathcal{L}}$ is the landmark region grounding and $\phi^{l}_{R_j}$ denotes the binary correspondence variable that specifies whether region R_j is the landmark. For each potential landmark region, the algorithm then calculates the likelihood that each region in the map corresponds to the figure based on a model for the spatial relation r. We arrive at the overall figure grounding likelihood by marginalizing over the landmarks

$$p(\phi^{\mathcal{F}}_{R_i} = \mathtt{T}) = \sum_{R_j} p(\phi^{\mathcal{F}}_{R_i} = \mathtt{T} | \gamma_{\mathcal{L}} = R_j, r_k)\, p(\gamma_{\mathcal{L}} = R_j), \tag{5.2}$$

where $\phi^{\mathcal{F}}_{R_i}$ is the correspondence variable for the figure. We normalize these likelihoods for each potential figure region

$$p(\gamma_{\mathcal{F}} = R_i) = \frac{p(\phi^{\mathcal{F}}_{R_i} = \mathtt{T})}{\sum_{R_i} p(\phi^{\mathcal{F}}_{R_i} = \mathtt{T})}. \tag{5.3}$$

This expresses the likelihood of the correspondence variable being true for each figure region R_j in the factor graph in the semantic layer.

However, when there is uncertainty over the landmark or figure grounding, the likelihood of the label associated with the figure region can become diluted. Thus, the utility of information contained in each natural language description is dependent on the level of ambiguity in the expression.

5.1.2 Continuous Evaluation of Natural Language Descriptions

The challenge with fusing information inferred from natural language descriptions with a representation built using the robot's own sensors is the potential spatial and temporal disconnect between the two sources of information. In the previous chapters, we handled the temporal disconnect between natural language descriptions and robot's sensor observations

by attempting to ground the language immediately, and if this fails, deferring the language grounding to a later time (where it reevaluates the groundings periodically as more parts of the environment are explored until a valid grounding is found).

However, due to the potential ambiguity associated with such descriptions, there could be multiple valid potential groundings for a particular description. If the algorithm makes a hard decision to conclude evaluating a particular description, there is no guarantee that the correct grounding has been found. However, a naive implementation where the language is evaluated on the entire map as it is updated will be too inefficient. To address this, we enhance the natural language understanding process to reevaluate the descriptions efficiently as the robot encounters new relevant regions in the environment. We make use of the semantic prior provided by the robot's own sensors to identify potential new landmarks and any relevant extensions to the spatial model, and reevaluate the descriptions on the relevant modifications.

For each particle when the topology is extended by adding a new region or edge, for each language description received by the robot, we identify any differences in the space of grounding variables. These differences can be:

- A different shortest path (due to an update in the metric/topological information, such as an addition of a loop closure edge).
- The addition of a new potential landmark or figure region.
- The removal of an existing figure or landmark region (due to the region merging with another region).

Depending on the type of change, we evaluate the relevant grounding pairs, and recalculate the correspondence variables for the figure regions. We reconstruct the factor graph with the relevant information, and rerun belief propagation to update the semantic information.

5.2 Action Selection Algorithm

The approach that we present in this chapter is modeled on a robot that is following a narrated guided tour [30]. Our approach models the problem as a decision process, where at

each time step the robot reasons over two sets of high-level actions, to follow the guide and continue with the tour or ask a question that reduces its uncertainty over its representation.

The algorithm reasons over the natural language descriptions and the current learned map to identify potential questions that best reduces ambiguity in the map. The algorithm considers egocentric and allocentric binary (yes/no) questions that consist of spatial relations between pairs of regions. These regions may be local to the robot in the case of situated dialog (e.g., "Are we in the kitchen?", "Is the lab on my right?") or distant in the case of non-situated dialog (e.g., "Is the lounge next to the conference room?").

Algorithm 3: Semantic Mapping and Action Selection

Input: $S_{t-1} = \left\{ S_{t-1}^{(i)} \right\}$, and $\left(u_t, z_t, a_t, \mathcal{A}_{t-1}, z_t^{\mathcal{A}}, \Lambda_t \right)$, where

$S_{t-1}^{(i)} = \left\{ G_{t-1}^{(i)}, X_{t-1}^{(i)}, L_{t-1}^{(i)}, w_{t-1}^{(i)} \right\}$

Output: $\left\{ \mathcal{A}_t, S_t = \left\{ S_t^{(i)} \right\} \right\}$

1) Update Distribution with odometry and sensor data and language.

for $i = 1$ *to* n **do**

1. Employ proposal distribution to propagate the graph sample $G_t^{(i)}$ based on u_t, Λ_t and a_t.
2. Update the Gaussian distribution over the node poses $X_t^{(i)}$ conditioned on the topology.
3. If A_{t-1} was a question, add $\left\{ A_{t-1}, z_t^{\mathcal{A}} \right\}$ to the corresponding description.
4. Reevaluate language descriptions and update the semantic layer $L_t^{(i)}$.
5. Update particle weights.

end

Normalize weights and resample if needed.

2.) Evaluate action costs and select minimum cost action $\mathcal{A}_t$.

$$\mathcal{A}_t^* = \arg\max_{\mathcal{A}_t} \sum_{S_t^{(i)}} p(S_t^{(i)}) Q(S_t^{(i)}, \mathcal{A}_t)$$

Algorithm 3 outlines the process by which the robot updates its representation and chooses the optimal action A_t^*. At each time step, the method integrates new odometry

and sensor information to update the distribution over the semantic graph. This includes reevaluating the language descriptions and the guide's answers to questions. Then, the algorithm evaluates the value and cost of each valid dialog action, and executes the best action. Next we elaborate on the action selection procedure.

We formulate the guided tour as a decision process where the robot selects an action at every time step. These actions include following the person, and asking a particular question. For each allocentric description provided by the guide, we define a set of actions that ask a question from the guide. We model the value of the robot's next state using an information gain heuristic, such that the robot values asking useful questions. The information gain is defined as the reduction in entropy in the groundings for the natural language description based on the asked question and a received answer. We introduce a cost function for these question-asking actions to model the social cost of asking a question.

We define the current state as the robot's current representation over the world, i.e., the semantic graph S_t. If the robot asks a question and receives an answer, the semantic graph S_t will be updated based on the $\left\{\mathcal{A}_t, z^{\mathcal{A}}_{t+1}\right\}$ tuple, which we represent as S_{t+1}. When the robot selects a question-asking action, it will stop before asking the question and only resume following once the answer is received (or after a timeout). As such, the next state S_{t+1} is only dependent on S_t, the robot's question, and the guide's response. Since each question that the robot asks refers to the location of the figure region described in a given natural language description, the update to the semantic graph will only modify the distribution over this grounding. If the robot takes the following action, the semantic graph would be modified due to new observations and possible descriptions. In this work we do not model this change in state for following the person as this requires us to reason over the unobserved part of the world.

As we maintain the distribution over the semantic graph using a Rao-Blackwellized particle filter, the robot's state $S_t = \left\{S_t^{(i)}\right\}$ is a collection of weighted particles. Thus, we solve this for each particle $S_t^{(i)}$ and infer the optimal action given the distribution over the semantic graph using the QMDP heuristic [48].

For a single particle $S_t^{(i)}$, we define the Q value as,

$$Q(S_t^{(i)}, \mathcal{A}_t) = \sum_{S_{t+1}^{(i)}} \gamma V(S_{t+1}^{(i)}) \times p(S_{t+1}^{(i)} | S_t^{(i)}, \mathcal{A}_t) - \mathcal{C}(\mathcal{A}_t). \tag{5.4}$$

We define the value of each possible next state $S_{t+1}^{(i)}$ for the particle $S_t^{(i)}$ as a function of the information gain associated with improving the grounding of a particular language description by taking a question-asking action and receiving a given answer. Specifically, we model this by calculating the reduction in entropy for the distribution over the figure grounding $\gamma_{\mathcal{F}}$ for a given natural language description, by asking a question about the location of that figure region and receiving a particular answer. The information gain heuristic biases the decision process to value actions that on expectation reduce the uncertainty over the language groundings in the semantic graph. We also model the cost of each action $\mathcal{A}_t$. We use a discounting factor $\gamma = 1$.

At each time step, the robot takes the best action $\mathcal{A}_t^*$ from the available set of actions using the QMDP heuristic, as the robot maintains a distribution over the state of the world.

$$\mathcal{A}_t^* = \arg\max_{\mathcal{A}_t} \sum_{S_t^{(i)}} p(S_t^{(i)}) Q(S_t^{(i)}, \mathcal{A}_t), \tag{5.5}$$

where $p(S_t)$ is the particle weight $w_t^{(i)}$.

The following paragraphs explain this process in detail.

5.2.1 Action Set

The action set consists of the "Follow Person" action $A_{\mathcal{F}}$, and the valid set of question-asking actions. The "Follow Person" action $A_{\mathcal{F}}$ is available at all times except when the robot is waiting for an answer to a question, when the robot stops and waits for an answer (or the question to timeout). We derive our questions from a templated set for each grounding entity in a natural language description. These templates can be categorized into two basic types.

I *Immediate Questions*: This template takes a spatial relation from the set of spatial

relations (at, near, away, in front, behind, left of, right of) and a grounding variable to create a question of the type "Is the kitchen in front of me?". For such questions, the possible answers are "yes," "no," and "invalid" (for questions that do not make sense given a spatial entity). For this work, we define questions about the figure groundings.

II *Landmark Questions*: This template defines questions in terms of spatial relations between non-local locations in the environment. If the robot is highly confident of the semantic label of a particular location, it could generate a question about regions close to that entity to resolve uncertainty. For example, when the robot is uncertain about the location of the "lounge," but thinks one possibility is the space next to the "conference room," while several are not, it could ask "Is the lounge next to the conference room?". This allows the robot to ask questions about regions outside the robot's immediate area.

The robot can only use questions of the first type to ask about spatial regions in its immediate vicinity. As such, the ability to receive useful information is limited to instances when the robot is near a potential hypothesized location. Questions of the second type allow the robot to reduce its uncertainty even when a hypothesized location is not within its immediate vicinity. However, this may place a higher mental burden on the user who must then reason about spatial entities outside their immediate perception range.

5.2.2 Value Function

As we define the current state as the semantic graph at time t, the next state S_{t+1} is dependent on S_t coupled with the question and answer pair. For a single particle, the next state is defined as:

$$S_{t+1}^{(i)} = \left\{ S_t^{(i)}, \mathcal{A}, z^{\mathcal{A}} \right\} \tag{5.6}$$

Since the robot will be stopped from the time the question is asked to the time the answer is received, there will be no change to the spatial representation. Thus the only change will be to the groundings of the natural language description for which the question

was asked. Receiving the answer would ideally reduce the entropy over the groundings. Thus, for a semantic graph particle $S_t^{(i)}$, the set of next potential $S_{t+1}^{(i)}$ are limited by the set of question and answer pairs.

We define the value of the next state using an information gain based heuristic. The information gain is calculated for the language grounding from which each question is defined. Since the addition of a question and answer pair can only effect the language grounding for which it is defined, we only calculate the information gain for the relevant language grounding.

$$V(S_{t+1}^{(i)}) = \mathcal{F}(I(\mathcal{A}_t, z_{t+1}^{A})). \tag{5.7}$$

Information Gain

The information gain $I(\mathcal{A}, z^{\mathcal{A}})$ for action $\mathcal{A}$, as shown in Equation 5.8 is defined as the reduction in entropy by taking action $\mathcal{A}$ and receiving observation $z^{\mathcal{A}}$. In our framework, the entropy is over a grounding variable $\gamma_{\mathcal{F}}$ created for a natural language description provided by the guide. Calculating the exact entropy is infeasible since the map might not yet be complete and also because it is inefficient to calculate the likelihood of spatial regions that are too far outside the local area. Therefore, we approximate the distribution based on the spatial regions considered during the language grounding step for the language description Λ_k.

$$I(\mathcal{A}, z^{\mathcal{A}}) = H(\gamma_{\mathcal{F}}|\Lambda_k) - H(\gamma_{\mathcal{F}}|\Lambda_k, \mathcal{A}, z^{\mathcal{A}}) \tag{5.8}$$

In this work, our focus is on questions that have a predefined set of answers, which allows us to model the information gain for each question and answer pair, as well as reason over the likelihood of receiving each answer given the question. This allows us to calculate the Q value for each state $S_t^{(i)}$ and action $\mathcal{A}_t$ as outlined in equation 5.4.

Given the answer, we evaluate the change it has on the distribution over the particular grounding variable. For most spatial relations (excluding *near* and *at*), we define a *valid range* from the robot, over which a particular question can be applied in a meaningful manner. In our experiments, we used a valid range of 20 m when evaluating a question. As such, we limit the entropy calculation to the regions for which the question is expected to

be meaningful.

$$p(\gamma_{\mathcal{F}} = R_j|\Lambda_k, \mathcal{A}, z^{\mathcal{A}}) = \frac{p(z^{\mathcal{A}}|\mathcal{A}, R_j) \times p(\gamma_{\mathcal{F}} = R_j|\Lambda_k)}{\sum_{R_j} p(z^{\mathcal{A}}|\mathcal{A}, R_j) \times p(\gamma_{\mathcal{F}} = R_j|\Lambda_k)} \tag{5.9}$$

For the action $A_{\mathcal{F}}$, we assume that there is no change in the entropy as we are not modeling the expected change in the language groundings based on spatial exploration. Thus, the Q value for $A_{\mathcal{F}}$ is only the cost of the action.

5.2.3 Transition Likelihood

The transition function for a single particle $p(S_{t+1}^{(i)}|S_t^{(i)}, \mathcal{A}_t)$ is equivalent to the likelihood of receiving each answer given the state and the question-asking action. Using our spatial relation models we are able to calculate $p(z^{\mathcal{A}}|S_t^{(i)}, R_j, \mathcal{A})$, which is the likelihood of receiving a particular answer given the question and the correct grounding is R_j. We calculate the overall likelihood of receiving each answer by marginalizing out the grounding variable.

$$p(z^{\mathcal{A}}|S_t^{(i)}, \mathcal{A}) = \sum_{R_j} p(z^{\mathcal{A}}|S_t^{(i)}, R_j, \mathcal{A}) \times p(R_j|\Lambda_k) \tag{5.10}$$

This results in a higher expected likelihood of receiving a particular answer if there were spatial regions that had a high *a priori* likelihood of being the grounding and also fit the spatial relation in the question.

5.2.4 Cost Function Definition

We define a hand-crafted cost function that encodes the desirability of each robot action.

$$\mathcal{C}(\mathcal{A}_t) = \mathcal{F}(f(\mathcal{A}_t)) \tag{5.11}$$

For question-asking actions, this is a function of several relevant features. For this implementation, we have used the following:

i Time since last question asked

ii Time since last question asked about grounding

iii Number of questions asked about entity

In our current implementation, we use a linear combination of these features to arrive at a reasonable cost function. The weights have been set such that they result in negligible burden on the user and do not impeded the conducting of the tour. Ideally, these weights would be learned from user preferences based upon human trials.

For the person following action $A_{\mathcal{F}}$, we assign a fixed (negative) cost such that only a reasonably high expected information gain will result in a question being asked. The value was set empirically to achieve a reasonable level of questions.

5.2.5 Integrating Answers to the Representation

We couple each of the user's answers with the original question defined for a language description Λ_k to arrive at an equivalent natural language description of the environment. As such, each new answer modifies the distribution over that grounding variable, and any informative answer improves the robot's representation. However, since the question is tied to a particular spatial entity, we treat the question and answer pair together with the original description, according to Equation 5.9. However, unlike in the entropy calculations, we consider all potential groundings, including regions outside the meaningful area for the question.

To arrive at the $p(z^{\mathcal{A}}|\mathcal{A}, R_i)$, we factor the likelihood as,

$$p(z^{\mathcal{A}}|\mathcal{A}, R_i) = \sum_{v=F}^{T} p(z^{\mathcal{A}}|v, a, R_i) \times p(v|a, R_i), \tag{5.12}$$

where $v = t$ implies the question is valid for R_i and $v = f$ implies the question is invalid, and marginalize over v. The $p(v = t|\mathcal{A}, R_i)$ is high only for R_i within the valid range of the robot's location (when the question was asked). When the question is invalid for R_i, the $p(z^{\mathcal{A}} =$ "no"$|v, \mathcal{A}, R_i)$ has a high likelihood. $z^{\mathcal{A}} =$ "no".

When new valid grounding regions are added, we reevaluate both the original description as well as the likelihood of generating the received answer for each new region, and

update the language grounding. Figure 5-2 shows the grounding likelihoods before and after asking three questions.

5.3 Experimental Evaluations

We evaluate our algorithm on two indoor datasets modeled on a human giving a robotic wheelchair a narrated guided tour (Figure 5-1). The datasets are from two different floors of MIT's Stata Center building. For this experiment, we injected natural language descriptions at locations where the descriptions are ambiguous. We ran the algorithm on the datasets (played back at real time) and a human provided answers to the questions asked by the system. We ran two experiments, where in experiment I the robot was allowed to ask both immediate and landmark-based questions, while in experiment II we only allowed the robot to ask landmark-based questions. All experiments were run with two particles.

We quantify the results using two metrics, the reduction of entropy over the figure groundings for each language utterance, and the improvement in the accuracy of the language grounding. Table 5.1 outlines the entropy over the figure groundings with and without questions. As can be seen in all cases, the entropy over the groundings decreases significantly with question asking.

In order to calculate the accuracy of each grounding k, we annotated the ground truth region referred to by each annotation, and then calculated the overlap of each grounded figure region ($\mathbb{O}_{R_i}$) with the ground truth annotation. The overlap ratios were weighted with the grounding likelihood and summed to arrive at the accuracy score $\mathbb{S}_k$.

$$\mathbb{S}_k = \sum_{R_i} \mathbb{O}_{R_i} \times p(\gamma_{\mathcal{F}} = R_i | \Lambda_k, \{\mathcal{A}, z^{\mathcal{A}}\}) \tag{5.13}$$

The accuracy score penalizes situations where the groundings are assigned to regions outside the ground truth region, and also when some grounded regions only contain a part of the ground truth region (due to improper segmentation).

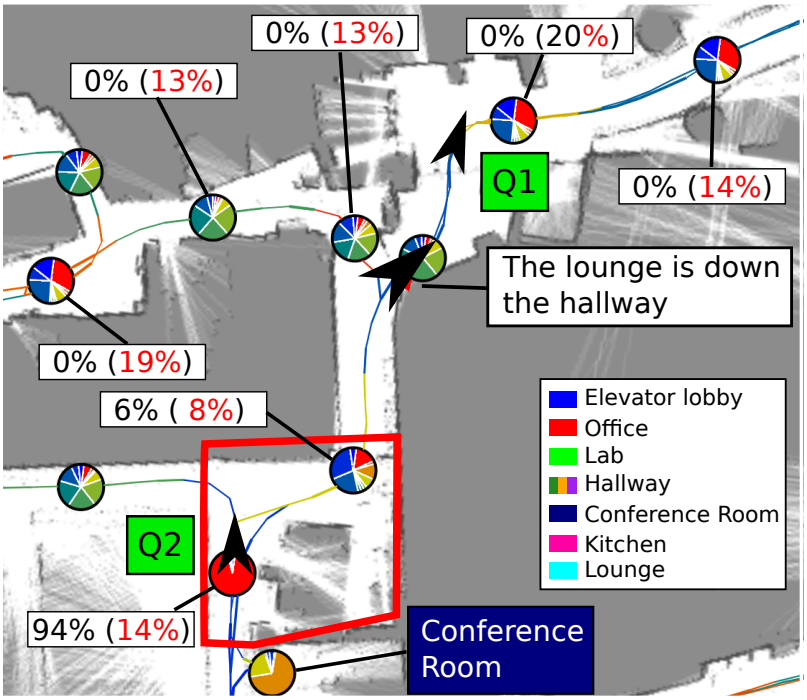


Figure 5-2: Experiment I: Language groundings for the expression "The lounge is down the hall". Grounding likelihood with questions is in black and without questions in red. Questions asked (and answers), Q1: "Is the lounge near me?" ("No"); Q2: "Am I at the lounge?" ("Yes"). The ground truth region boundary is in red. Pie charts centered in each region denote its type while path color denotes different regions.

5.3.1 Experiment I: Immediate and Landmark-Based Questions

As can be seen in Table 5.1, the semantic maps that result from integrating the answers received from the guide have much less uncertainty (and lower entropy) over the figure groundings. For all descriptions, the robot was able to significantly reduce the entropy over the figure groundings by asking questions. They also have significantly improved accuracy over the figure groundings compared to the approach without questions. On average there was a 78% reduction in the entropy over the figures and a 347% improvement in the accuracy over the grounding. The impact of question asking is dependent on the ambiguity of each description as well as the available set of questions along the path taken by the guide. In this experiment, the robot never asked any landmark based questions, owning to the fact that the path taken involved travel close to the ambiguous figure regions, allowing it to ask immediate questions that better resolved the ambiguity.

Dataset I contains seven descriptions of the robot's location that the algorithm grounds

Table 5.1: Experiment I: Entropy over figure groundings with and without questions

	Entropy		Accuracy		
Utterance	NQ	Q	NQ	Q	# Q
(A) "The lounge is down the hallway" (Figure 5-2)	1.911	0.237	17.3%	90.6%	2
(B) "The elevator lobby is down the hallway"	1.574	0.566	35.8%	70.9%	2
(C) "The lounge is behind you"	0.403	0.095	87.2%	98.4%	1
(D) "The lab is down the hall" (Figure 5-3)	2.041	0.310	14.6%	91.6%	3
(E) "The conference room is down the hallway"	2.061	0.664	6.5%	65.5%	8
(F) "The lounge is in front of us"	1.053	0.107	20.6%	43.8%	2

Dataset I has utterances A, B and C, and Dataset II has utterances D, E and F. NQ: Method without Questions, Q: Method with Questions, # Q: Number of questions

to the current region, and three allocentric expressions that describe regions with relation to either landmarks in the environment (e.g., "the elevator lobby is down the hall") or to the robot (e.g., "the lounge is behind you"). The robot asked a total of five questions of the guide, all in relation to itself.

Dataset II contains one descriptions of the robot's location that the algorithm grounds to the current region, and three allocentric expressions that describe regions with relation to either landmarks in the environment or to the robot. The robot asked a total of 13 questions of the guide, all of which were in relation to itself. In this dataset, one of the descriptions ("the conference room is down the hallway") was highly ambiguous due to the presence of multiple landmarks close to the described location, resulting in multiple potential valid groundings. Therefore the robot required 8 questions to reduce the ambiguity in the figure grounding. The large number of questions were due to the fact that there were no questions that could immediately disambiguate between the two most likely regions, given the route taken by the guide.

5.3.2 Experiment II: Landmark-Based Questions Only

We also reran the first dataset while only allowing the system to ask questions based on landmarks. Due to the path taken in the dataset, the robot traveled close to the potential fig-

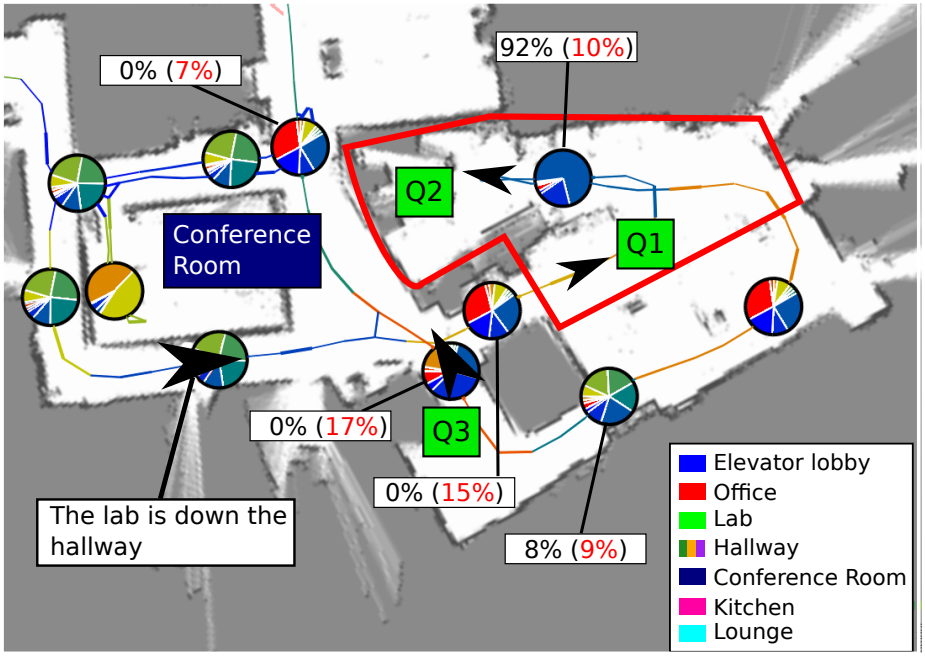


Figure 5-3: Experiment I: Language groundings for the expression "The lab is down the hall". Grounding likelihood with questions is shown in black and without questions in red. Question asked (and answer), Q1: "Is the lab near me?" ("No"); Q2: "Am I at the lab?" ("Yes"); Q3: "Am I at the lab?" ("No"). The ground truth region is outlined in red.

ure regions, and as such was able to resolve its ambiguity better by asking questions relative to itself. By only allowing the robot to ask landmark questions, we demonstrate the potential to reduce the ambiguity even when the robot does not revisit the locations it is uncertain about. However, the reduction in the entropy is conditioned on the presence of salient land-

Table 5.2: Entropy over figure groundings with immediate and landmark questions

	Entropy		Accuracy	
Utterance	Immediate Questions	Landmark Questions	Immediate Questions	Landmark Questions
(A)	0.237 (2)	2.001 (1)	90.6%	27.0%
(B)	0.566 (2)	0.000 (1)	70.9%	96.6%
(C)	0.095 (1)	0.441 (0)	98.4%	79.8%

Number of questions asked in each method are shown in brackets in their respective entropy columns.

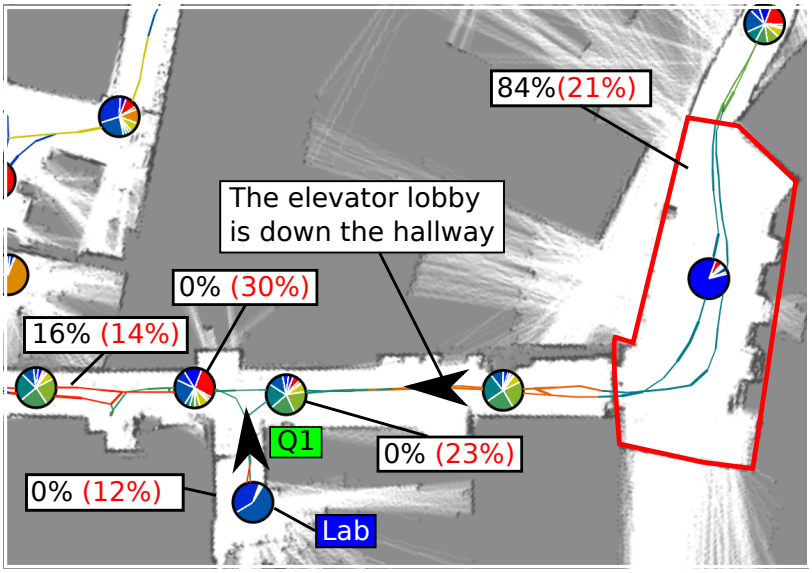


Figure 5-4: Experiment II: Language groundings for the expression "The elevator lobby is down the hallway". Grounding likelihood with questions is shown in black and without questions in red. Question asked (and answer), Q1: "Is the elevator lobby near the lab?" ("No"). The ground truth region is outlined in red.

marks (which in this experiment came from language annotations). Table 5.2 compares the performance of asking landmark-based questions with asking immediate questions. The robot is still able to reduce the ambiguity but not to the level that it achieves with asking immediate questions. The reasons are two fold; firstly, there are only a few salient landmarks that could be used to generate valid questions; secondly, landmark questions can only be asked using the "near" spatial relation. Thus the available set of useful questions are limited. Figure 5-4 shows the resulting grounding likelihoods for the utterance "the elevator lobby is down the hallway".

5.4 Discussion

We outlined a framework that enables a robot to engage a human in dialog in order to improve its learned semantic map during a guided tour. We enabled this behavior by continuously regrounding natural language descriptions as new parts of the environment are encountered by the robot. We also outlined two experiments conducted to evaluate the benefits of asking questions to improve the robot's semantic map.

The current approach is capable of both reducing the entropy over the figure groundings and increasing the accuracy of the grounded region. This entropy is calculated based on the distribution over the potential set of groundings for the figure region that is described in a natural language description. However, if the correct figure region is over-segmented in the robot's learned map, this will cause the robot over-estimating the level of ambiguity for that description, resulting in incorrect questions being asked.

As our algorithm does not reason over possibly unvisited parts of the environment, the measure of ambiguity is based on a partially known map. This can result in an incorrectly low measure of ambiguity at a location (and time) where asking a question would result in the most improvement to the learned map. A more comprehensive approach would be to model the likelihood that figure references ground to unvisited regions in the environment, and evaluate the affect of the questions on these regions as well.

Additionally, because we depend on spectral clustering to segment the environment, there is a delay to when the robot learns that it has transitioned in to a new region. This can also lead to an incorrectly low measure of ambiguity at a time when a useful question could be asked from the user. Thus, better techniques for reasoning about spatial decomposition, especially ones that allow for the robot to learn of a region transition with less delay would lead to improved reasoning about the utility of asking questions allowing for more accurate learned maps.

Chapter 6

Inferring Maps and Behaviors from Natural Language Instructions

Natural language instructions offer an effective means for untrained users to control complex robots, without requiring specialized interfaces or extensive user training. However, with few exceptions, existing techniques in language understanding require the robot to posses *a priori* knowledge of location, geometry, colloquial name, and type of all objects and regions within the environment [41, 34, 88]. Without known world models, interpreting free-form commands becomes much more difficult. Existing methods have dealt with this by learning a parser that maps the natural language commands into a formal control language equivalent [53, 8, 57]. Alternatively, Duvallet et al. [18] use imitation learning to train a policy that reasons about uncertainty in the grounding and that is able to backtrack as necessary.

Oftentimes, the command itself provides information about the environment that can be used to hypothesize suitable world models, which can then be used to generate the correct robot actions. For example, Figure 6-1 shows a user in a robotic wheelchair instructs the robot to "navigate to the kitchen that is down the hallway," where the hallway and the kitchen are outside the robot's field-of-view. While the robot has no *a priori* information about the environment, the instruction conveys the knowledge that there is a "kitchen" that is "down" a "hallway." A robot capable of reasoning about this information will be able better respond to the command (e.g., reason over the presence and location of a kitchen

Figure 6-1: A user commanding a robotic wheelchair using natural language that contains information about the environment.

when it observes a hallway).

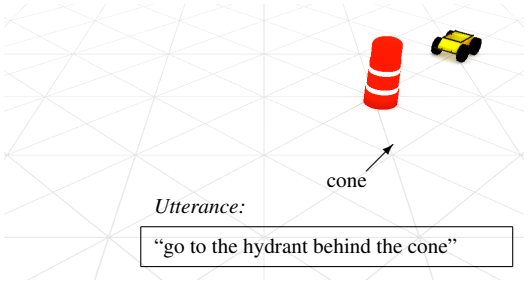
Joint Inference over Maps and Behaviors

In this chapter, we use our semantic mapping algorithm to induce a distribution over the world based on information contained in natural language, explicitly reasoning over unobserved parts of the environment that have been specified in the command. We then use this distribution over the world to solve for a policy that is consistent with the command, which is then executed by the robot. The robot updates its internal representation of the world as it makes new metric observations (such as the location of perceived landmarks) and updates its policy appropriately. This approach enables robots to interpret and execute natural language commands that refer to unknown regions and objects in the robot's environment. By reasoning and planning in the space of beliefs over the presence and location of objects and regions that are not initially observed, we are able to robustly follow natural language navigation instructions given by a human operator.

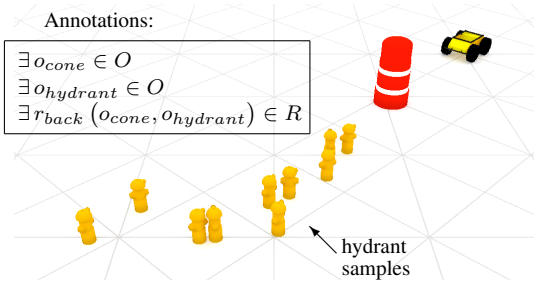
The semantic mapping algorithm introduced in this chapter uses information contained

in a natural language command to reason about the spatial structure of the environment, possibly unknown to the robot at the moment. For example, in Figure 6-2a the robot infers the presence of a hydrant (unobserved) behind the cone (observed) based on the command "go to the hydrant behind the cone". The mapping algorithm hypothesizes the presence of a hydrant and also reasons about its metric location based on the spatial relation "behind". This is in contrast to our mapping algorithms outlined in the previous chapters, where we used language to inform us of semantic information about entities only once they are observed by the robot using its own sensors.

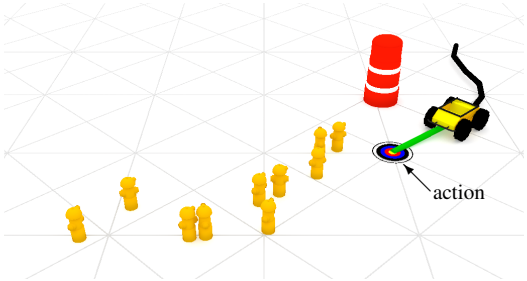
In the following section, we describe our probabilistic framework that first extracts annotations from a natural language instruction (Figure 6-2a). These annotations describe facts about robot's environment implicitly contained in natural language instructions that specify the existence of and spatial relations between objects and regions relevant to executing the command. It then treats these annotations as noisy sensor observations in a



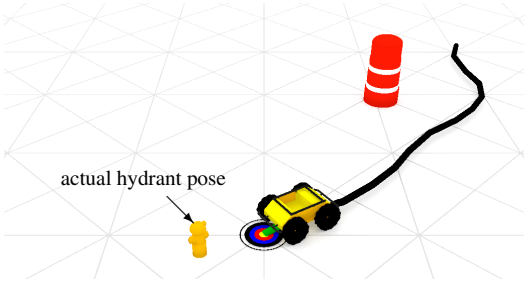
(a) First, we receive a verbal instruction from the operator.



(b) Next, we infer the map distribution from the utterance and prior observations.



(c) We then take an action (green), using the map and behavior distributions.



(d) This process repeats as the robot acquires new observations, refining its belief.

Figure 6-2: Visualization of one run for the command "go to the hydrant behind the cone," showing the evolution of the robot's beliefs (over the possible locations of the hydrant). The robot begins with the cone in its field of view, but does not know the hydrant's location.

mapping framework, and uses them to generate a distribution over a semantic model of the environment that also incorporates observations from the robot's sensor streams (Figure 6-2b). This prior is used to ground the actions and goals within the command, resulting in a distribution over desired behaviors. This is then used to solve for a policy that yields an action that is most consistent with the command, under the map distribution so far (Figure 6-2c). As the robot travels and observes new parts of the environment, it updates its map prior and inferred behavior distribution, and continues to plan until it reaches its destination (Figure 6-2d).

The approach outlined in this chapter is the result of joint work with Felix Duvallet, Thomas M. Howard and Matthew R. Walter. We apply the joint behavior and map inference framework to following natural language navigation instructions in the two following scenarios.

Following Object-Relative Navigation Commands

We apply this approach to allow a robot to execute free-form instructions that direct it to unknown objects [17]. In these experiments, a human operator issues natural language commands in the form of text that directs the robot to navigate to an object in the environment. These commands also contain information about the environment, describing the presence of objects (e.g., "go to the hydrant") and spatial relations between objects (e.g., "go to the hydrant behind the cone"). We evaluate our framework through a series of simulation-based and physical experiments on two mobile robots that demonstrate its effectiveness at carrying out navigation commands, as well as highlight the conditions under which it fails.

Following Natural Language Directions in Indoor Environments

We use this approach to enable a robot to follow natural language route directions in unknown indoor environments [28]. We consider directions that reference regions in the environment. We evaluate this both in simulation and physical experiments on a robotic wheelchair platform.

6.1 Technical Approach Overview

We define the problem of following natural language commands as one of inferring the robot's trajectory $x_{t+1:T}$ up to time horizon T that is most likely, given the history of natural language commands Λ^t,

$$\underset{x_{t+1:T} \in \Re^n}{\arg\max}\ p\left(x_{t+1:T} | \Lambda^t, z^t, u^t\right). \tag{6.1}$$

where z^t and u^t are the history of sensor observations and odometry data, respectively. Inferring the maximum *a posteriori* trajectory (6.1) for a given natural language utterance is challenging without knowledge of the environment for all but trivial applications. We address this by introducing a latent random variable S_t that represents the world model as a semantic map. The semantic map encodes the location, geometry, topology and type of the spatial entities, such as objects or regions, within the environment. This representation is derived from the work outlined in the previous chapters, although we introduce several key modifications to the representation and the inference process to allow us to reason about unobserved spatial entities described in language. We then interpret the natural language command in terms of the latent world model, which results in a distribution over behaviors β_t. We then solve the inference problem (6.1) by marginalizing over the latent world model and behaviors:

$$\underset{x_{t+1:T} \in \Re^n}{\arg\max} \int_{\beta_t} \int_{S_t} p(x_{t+1:T} | \beta_t, S_t, \Lambda^t) \cdot p(\beta_t | S_t, \Lambda^t) \cdot p(S_t | \Lambda^t, z^t, u^t)\, dS_t\, d\beta_t. \tag{6.2}$$

We maintain the distribution over the semantic maps using numerical sampling with particles (similar to the previous chapters) and infer a discrete set of behavior groundings for each semantic map particle, resulting in:

$$\underset{x_{t+1:T} \in \Re^n}{\arg\max} \sum_{\beta_t^{(j)}} \sum_{S_t^{(i)}} p(x_{t+1:T} | \beta_t^{(j)}, S_t^{(i)}, \Lambda^t) \cdot p(\beta_t^{(j)} | S_t^{(i)}, \Lambda^t) \cdot p(S_t^{(i)} | \Lambda^t, z^t, u^t), \tag{6.3}$$

where each semantic map particle $S_t^{(i)}$ maintains a possible configuration over the environment by modeling the objects, the regions and their locations, and each behavior $\beta_t^{(j)}$ specifies a set of actions defined for a semantic map particle.

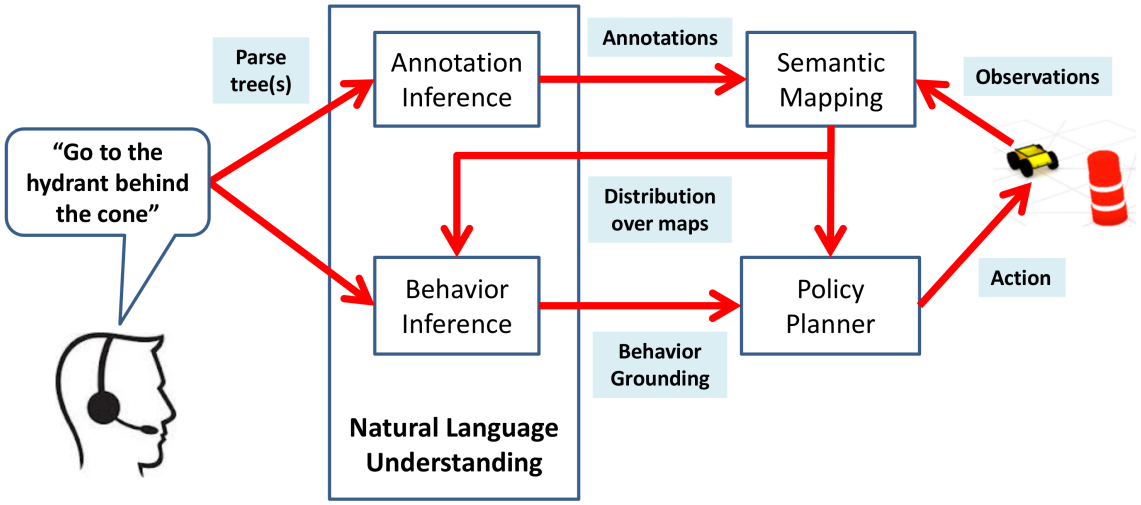


Figure 6-3: Framework outline.

By structuring the problem in this way, we are able to treat inference as three coupled learning problems. Figure 6-3 shows the overall framework that we use to follow natural language commands. The framework first converts the natural language instructions into a set of environment annotations using learned language grounding models (Annotation Inference). It then treats these annotations as observations of the environment (i.e., the existence, name, and relative location of rooms and objects) that it uses together with data from the robot's onboard sensors to learn a distribution over possible semantic maps (Semantic Mapping). Our framework then infers a distribution over behaviors conditioned upon the world model and the command (Behavior Inference). We then solve for the navigation actions that are consistent with this behavior distribution using a learned belief space policy (Policy Planner). As the robot executes this action, we update the world model distribution based upon new utterances and sensor observations, and subsequently select an updated action according to the policy. This process repeats until the policy selects the stop action. The following sections outline each component in detail.

6.2 Natural Language Understanding

Our framework relies on learned models to identify the existence of annotations and behaviors conveyed by free-form language and to convert these into a form suitable for semantic mapping and the belief space planner. This is a challenge because of the diversity of natural language instructions, annotations, and behaviors. We perform this translation using the Hierarchical Distributed Correspondence Graph (HDCG) model [33], which is an extension of the Distributed Correspondence Graph (DCG) [34] that offers improved efficiency. The DCG exploits the grammatical structure of language to formulate a probabilistic graphical model that expresses the correspondence $\phi \in \Phi$ between linguistic elements from the command and their corresponding constituents (*groundings*) $\gamma \in \Gamma$. The factors f in the DCG are represented by log-linear models with feature weights that are learned from a training corpus. The task of grounding a given expression then becomes a problem of inference on the DCG model. Since no world model is assumed when inferring linguistic annotations from an utterance, the DCG considers the grounding space of symbols to be the possible object and region types in the world. When inferring behaviors for each map particle, the space of groundings is limited to the set of (observed or hypothesized) objects and regions each particle.

The HDCG model employs DCG models in a hierarchical fashion, by inferring rules $\mathtt{R}$ to construct the space of groundings for lower levels in the hierarchy. At any one level, the algorithm constructs the space of groundings based upon a distribution over the rules from the previous level:

$$\Gamma \rightarrow \Gamma\left(\mathtt{R}\right). \tag{6.4}$$

The HDCG model treats these rules and, in turn, the structure of the graph, as latent variables. Language understanding then proceeds by performing inference on the marginalized models:

$$\arg\max_{\Phi} \int_{\mathtt{R}} p\left(\Phi|\mathtt{R}, \Gamma\left(\mathtt{R}\right), \Lambda, \Psi\right) \times p\left(\mathtt{R}|\Gamma\left(\mathtt{R}\right), \Lambda, \Psi\right) \tag{6.5a}$$

$$\arg\max_{\Phi} \int_{\mathtt{R}} \prod_{i} \prod_{j} f\left(\Phi_{i_j}, \Gamma_{i_j}\left(\mathtt{R}\right), \Lambda_i, \Psi, \mathtt{R}\right) \times \prod_{i} \prod_{j} f\left(\mathtt{R}, \Lambda_i, \Psi, \Gamma_{i_j}\left(\mathtt{R}\right)\right). \tag{6.5b}$$

6.2.1 Annotation Inference

We define an annotation as a statement of the existence of and relationships between one or more spatial entities. A spatial entity is either a region (e.g., "kitchen" or "lounge") or an object (e.g., "trashcan", "fire hydrant"). An area is a portion of state-space that is typically associated with a relationship to some spatial entity (e.g., "in front of the fire hydrant", "down the hallway"). Relations are a particular type of association between a pair of objects or regions (e.g., front, back, near, far). Since any set of spatial entities, areas, and relations may be inferred as part of symbol grounding, the size of the space of groundings for map inference grows as the power set of the sum of these symbols. We use the trained HDCG model to infer a set of annotations α_t from the positively expressed groundings at the root of the parse tree. Figure 6-4 illustrates the model for the direction "go to the kitchen that is down the hall". At the root of the sentence the symbols for a "down" spatial relation between a "kitchen" and "hallway" region are sent to the semantic map to fuse with other observations. The semantic mapping method, which is outlined in Section 6.3, utilizes these annotations and observations to learn a distribution over the environment map.

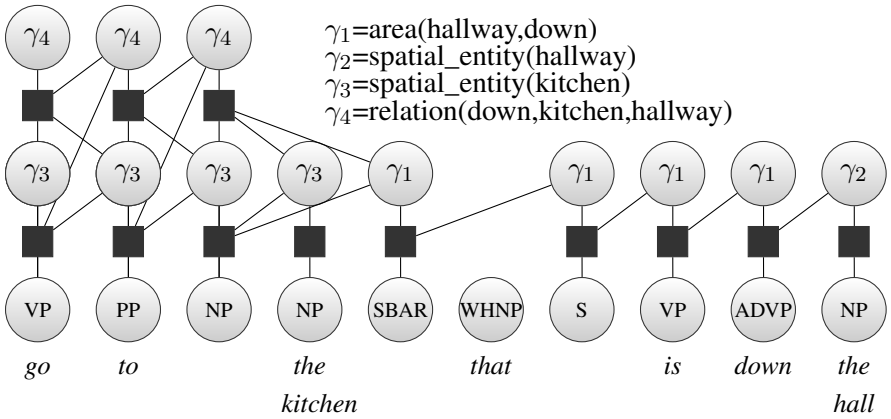


Figure 6-4: The active groundings in annotation inference for the direction "go to the kitchen that is down the hall". The two symbols at the root of the sentence (γ_3,γ_4) are sent to the semantic map to fuse with other observations.

6.2.2 Behavior Inference

Given the utterance and the semantic map distribution, we now infer a distribution over robot behaviors. The space of symbols used to represent the meaning of phrases in behavior inference is composed of spatial entities, areas, actions, and goals. Spatial entities and areas are defined in the same manner as in annotation inference, though the presence of a spatial entity is a function of the inferred map. Actions and goals specify to the planner how the robot should perform a behavior. Since any set of actions and goals can be expressed to the planner, the space of groundings also grows as the power set of the sum of these symbols. For the experiments discussed in Section 6.5, we assume a number of spatial entities, areas, actions, and goals that are proportional to the number of objects in the hypothesized world model. We use the trained HDCG model to infer a distribution of behaviors $\beta_t^{(j)}$ from the positively expressed groundings at the root of the parse tree. As we maintain a distribution

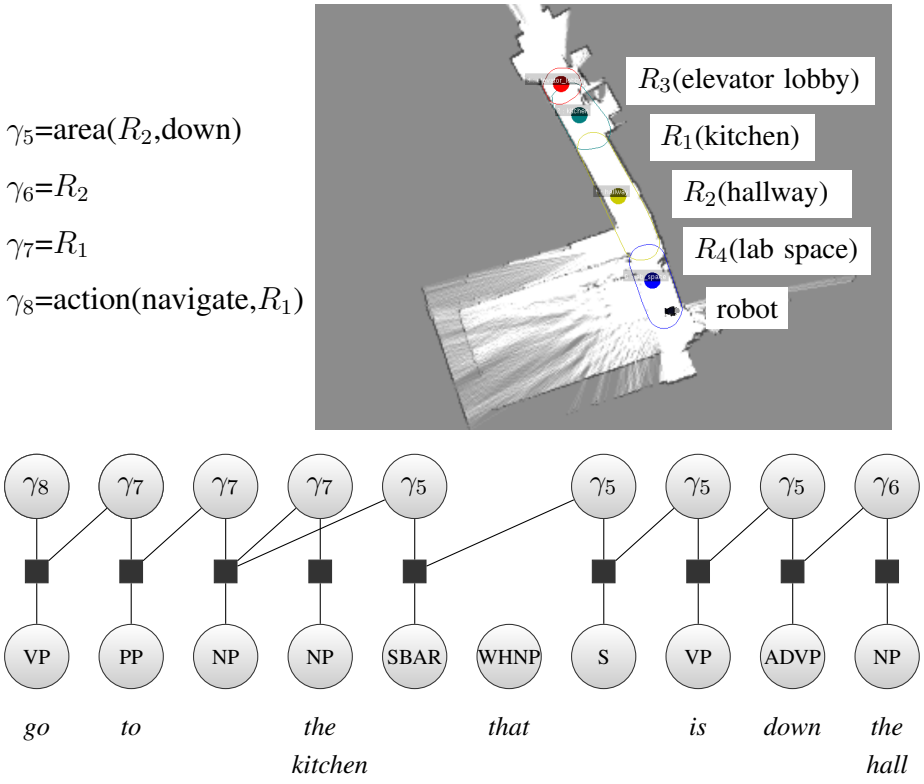
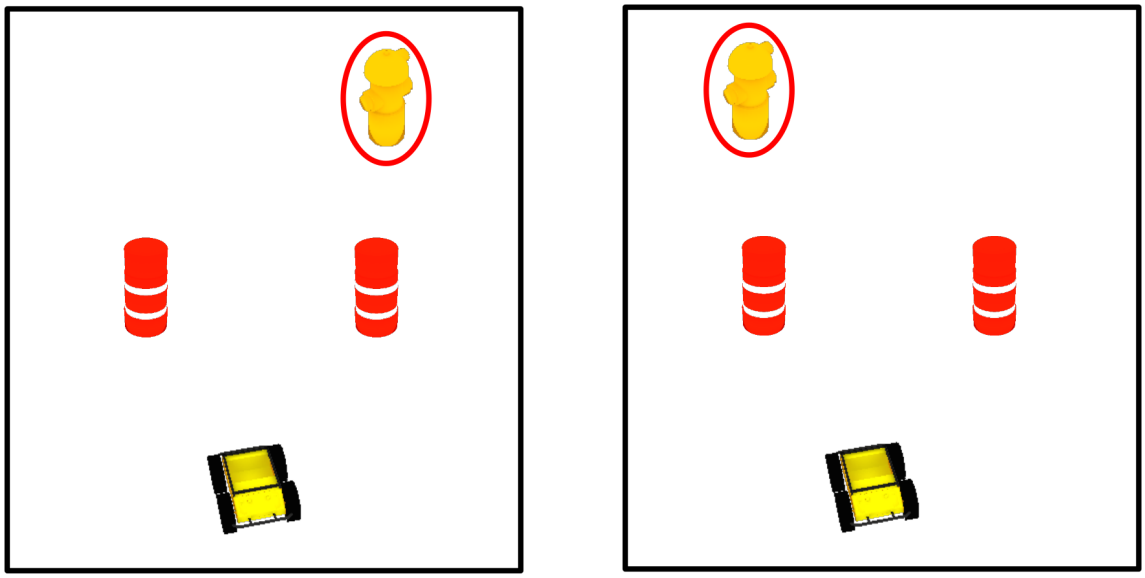


Figure 6-5: The active groundings in behavior inference for the direction "go to the kitchen that is down the hall" in the context of a inferred map with 4 objects. In this example a *navigate* action with a goal relative to R_1 would be sent to the policy planner.



(a) (b)

Figure 6-6: Behavior groundings for the command "go to the hydrant behind the cone" for two semantic map samples, where the destination object is circled in red.

over the semantic map in sample form, we infer behaviors for each sampled semantic graph $S_t^{(i)}$. The joint likelihood of each behavior and semantic map sample is defined as:

$$p(\beta_t^{(j)}, S_t^{(i)}) = p(\beta_t^{(j)}|S_t^{(i)})p(S_t^{(i)}). \tag{6.6}$$

Figure 6-6 shows the behavior groundings for two different semantic map samples. The belief space policy planner outlined in Section 6.4 uses this set of behaviors $\{\beta_t^{(j)}\}$ and semantic map samples $\{S_t^{(i)}\}$ to infer a policy consistent with the command.

6.3 Semantic Mapping Algorithm

In this section, we introduce our algorithm to learn the distribution over the semantic map $S_t = \{G_t, X_t\}$ from the set of annotations α^t inferred from the language command, the

robot's odometry u^t, and sensor observations z^t.

$$p(S_t|\Lambda^t, z^t, u^t) \approx p(S_t|\alpha^t, z^t, u^t) \tag{6.7a}$$

$$= p(G_t, X_t|\alpha^t, z^t, u^t) \tag{6.7b}$$

$$= p(X_t|G_t, \alpha^t, z^t, u^t)p(G_t|\alpha^t, z^t, u^t), \tag{6.7c}$$

where the last line expresses the factorization into a distribution over the environment topology (graph G_t) and a conditional distribution over the metric map (X_t). Unlike our prior approaches, we do not maintain a distribution over the semantic properties, as we assume the ability to directly observe the semantic labels of spatial entities (the type of object or region). We employ a sample-based approximation to maintain the distribution over the topology similar to the approaches outlined in the previous chapters. In this manner, each particle $S_t^{(i)} = \{G_t^{(i)}, X_t^{(i)}, w_t^{(i)}\}$ consists of a sampled topology $G_t^{(i)}$, a Gaussian distribution over the poses $X_t^{(i)}$, and a weight $w_t^{(i)}$.

The semantic mapping algorithm outlined in this section has several differences from the algorithms outlined in Chapters 3 and 4 in the way we represent the world, and the way we learn from natural language information. We explain them in detail in the following sections.

Representation

The topology G_t consists of two layers. The higher-level topology consists of spatial entities $\mathcal{E}_i$ that are either regions R_i or objects O_i. These entities are either observed by the robot or hypothesized based on language. The location of an object is observed by the robot using its cameras and the location and spatial extent of a region is observed when the robot drives through the area. Each entity also has an associated semantic label denoting the type of object (e.g., "cone", "trash can") or region (e.g., "kitchen", "office"). Since we assume that the robot can directly observe this semantic label, we do not maintain the semantic information as a distribution (unlike our representations in the previous chapters). In physical experiments we enable this by augmenting the environment using AprilTag fiducials [65], which are added to the relevant objects and regions. We also assume the ability to detect

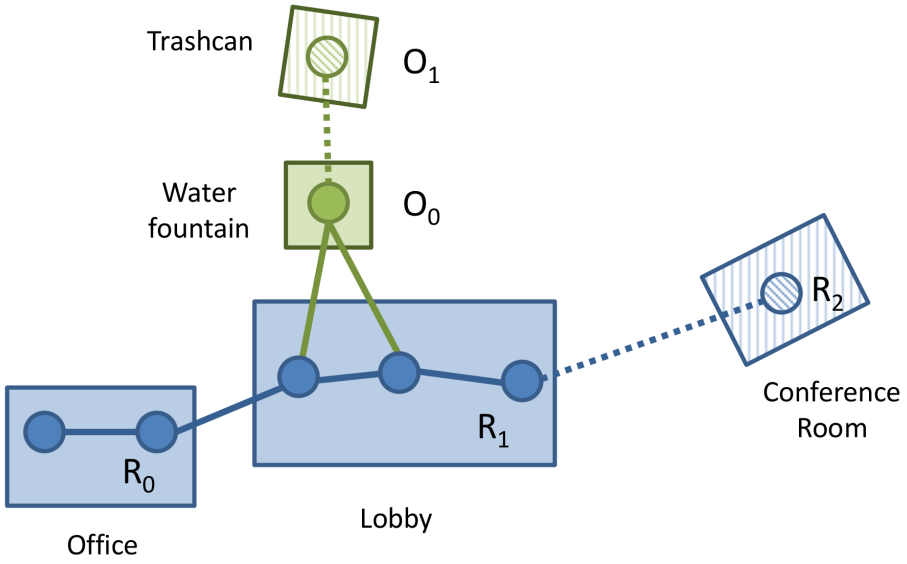


Figure 6-7: Example of a semantic map sample: Nodes are represented using circles while regions (in blue) and objects (green) are shown with rectangles. Hypothesized entities (O_1 and R_2) are shaded and sampled edges are shown with dashed lines.

when the robot makes a transition from one region to another.

The locations of these spatial entities are represented using nodes n_i belonging to the lower-level topology. We associate a pose x_i with each node n_i, the vector of which constitutes the metric map X_t. An observed region is composed of a set of nodes denoting the area traversed by the robot, while a region hypothesized based on language is represented by a single placeholder node. The location of an object that is either observed or hypothesized based on language is represented by a single node. An edge in the lower-level topology denotes a spatial relationship between two nodes and contains an associated metric constraint. An edge is added either when the robot navigates from one location to another (based on odometry), when it revisits an old region (by scan-matching laser observations), when it observes an object or when hypothesizing a region or object based on language (based on spatial relations). Figure 6-7 shows an example semantic map sample.

Compared to our representation outlined in Chapter 4, the key differences in this representation are the inclusion of objects and the presence of hypothesized spatial entities and the direct observability of semantic labels.

Algorithm Overview

To efficiently maintain the semantic map distribution over time as the robot receives new annotations and observations during execution, we use a Rao-Blackwellized particle filter similar to our prior approaches. This filtering process has three key steps: First, the algorithm propagates the topology of each particle by sampling modifications to the graph upon receiving new sensor observations z_t or annotations α_t inferred from the utterance. The algorithm uses these observations to infer the location of regions and objects. Second, the algorithm uses the proposed topology to perform a Bayesian update to the Gaussian distribution over the node poses. Third, we update the weight of each particle based on the likelihood of generating the given observations, and resample as needed to avoid particle depletion. We perform this process at each time step.

The key difference from our prior approaches is how we use information inferred from natural language to inform the robot's representation. In our previous works, we used language that often described parts of the world distant from the robot to learn about semantic properties, such as region labels. We utilized learned models of spatial relations to identify these regions referred to in the descriptions. However, we relied on the robot to visit these regions before we added them to the topology and only then did we infer any relevant semantic properties about them from language. In this work, we use language directly to infer the presence and layout of regions outside the robot's immediate location by using language in a very similar manner to other sensor observations. This key insight allows us to induce a distribution over the world even in the absence of concrete observations from the robot, and thus plan useful actions to accomplish the user's command. This is achieved by the introduction of a new proposal step that samples modifications to each topology based on natural language. The following sections provide details of this process.

Sampling Graph Modifications from a Proposal Distribution

During the proposal step, we first augment each sample topology with an additional node n_t and edge that model the robot's motion u_t, resulting in a new topology $G_t^{(i)-}$. We then sample modifications to the graph $\Delta_t^{(i)} = \{\Delta_{\alpha_t}^{(i)}, \Delta_{z_t}^{(i)}\}$ based upon the most recent

annotations α_t and sensor observations z_t:

$$p(G_t^{(i)}|G_{t-1}^{(i)}, \alpha_t, z_t, u_t) = p(\Delta_{\alpha_t}^{(i)}|G_t^{(i)-}, \alpha_t)\, p(\Delta_{z_t}^{(i)}|G_t^{(i)-}, z_t)\, p(G_t^{(i)-}|G_{t-1}^{(i)}, u_t), \tag{6.8}$$

where $\Delta_{\alpha_t}^{(i)}$ are the modifications based on natural language and $\Delta_{z_t}^{(i)}$ are the modifications based on the robot's sensor observations. This updates the proposed graph topology $G_t^{(i)-}$ with the graph modifications $\Delta_t^{(i)}$ to yield the new semantic map $G_t^{(i)}$.

These modifications can result in the addition or removal of regions, objects, and new edges into the topology. We sample the graph modifications by first sampling a grounding for each sensor observation or annotation, and depending on the resultant grounding, deterministically selecting the required graph modification. Sampling a grounding involves selecting a spatial entity that could explain the observation or annotation. For example, for a language annotation that describes the presence of a kitchen, if we were unable to sample a grounding to an existing region, we would create a hypothesized kitchen region, and also sample a spatial constraint based on information contained in the annotation.

We make use of a Dirichlet Process model to sample these groundings. This allows us to model the likelihood of a new region being responsible for a language annotation by taking into account the current layout of the world. Where we define the valid set of existing grounding entities γ_i and their weights w_i is defined as $\mathbb{S}_{\gamma,w} = \{\gamma_i, w_i\}$, and θ_{new} is a parameter that controls grounding to a new entity γ_{new}, the likelihood of sampling a grounding entity is:

$$p(\gamma^* = \gamma_i) = \begin{cases} \frac{w_i}{\sum_{w_i \in \mathbb{S}_{\gamma,w}} w_i + \theta_{\text{new}}}, & \text{if } \gamma_i \in \mathbb{S}_{\gamma,w} \\ \frac{\theta_{\text{new}}}{\sum_{w_i \in \mathbb{S}_{\gamma,w}} w_i + \theta_{\text{new}}}, & \text{if } \gamma_{\text{new}} \end{cases} \tag{6.9}$$

In our algorithm, the valid set of groundings are either spatial entities (regions R_i or objects O_i) or pairs of entities with an associated spatial relation. Depending on the scenario, the value used for the weights w_i will also change.

Next we explain how these graph modifications are sampled from natural language annotations and robot observations.

6.3.1 Graph Modification Based on Natural Language

When the algorithm receives a set of annotations $\alpha_t = \{\alpha_{t,j}\}$, it samples a modification to the graph for each particle:

$$p(\Delta_{\alpha_t}^{(i)} | G_t^{(i)-}, \alpha_t) \tag{6.10}$$

Each annotation $\alpha_{t,j}$ can imply the presence of spatial entities and the presence of spatial relationships between these entities. Conceptually, the algorithm handles these annotations by grounding the specified spatial entities in to its representation. The grounding process is different depending on the information contained in the annotation. This treatment of language is different from the approaches we outlined in the previous chapters. Previously, we maintained the distribution over the space of groundings for an annotation in each particle. In this, by sampling a grounding for each annotation in each particle, we are selecting one possible configuration of the world that could account for the annotation. The following paragraphs explain this process in detail, for descriptions that involve regions and their relationships. The process is similar for descriptions that reference objects. This process is carried out for each annotation $\alpha_{t,j} \in \alpha_t$.

An annotation $\alpha_{t,j}$ can describe the existence of a figure region or object in the environment. Specifically, for a figure region $R_\mathcal{F}$ with a label $l_\mathcal{F}$, the annotation implies:

$$\exists R_\mathcal{F} : l_{R_\mathcal{F}} = l_\mathcal{F}. \tag{6.11}$$

For example, the command "go to the kitchen" describes the presence of a region $R_\mathcal{F}$, where the label $l_\mathcal{F}$ is the "kitchen". For such an annotation, we sample a grounding for the described figure region $R_\mathcal{F}$ using a Dirichlet Process as follows:

$$p(R_\mathcal{F} = R_i) = \begin{cases} \frac{1}{n_{l_\mathcal{F}} + \theta_{l_\mathcal{F}}}, & \text{if } R_i \in \mathbb{S}_{l_\mathcal{F}} \\ \frac{\theta_{l_\mathcal{F}}}{n_{l_\mathcal{F}} + \theta_{l_\mathcal{F}}} & R_i = \textit{new} \end{cases} \tag{6.12}$$

where l_{R_i} is the label associated with region R_i, $\mathbb{S}_{l_\mathcal{F}} = \{R_i | l_{R_i} = l_\mathcal{F}\}$, $n_l = |\mathbb{S}_{l_\mathcal{F}}|$, and $\theta_{l_\mathcal{F}}$ biases the likelihood of the grounding towards a new region. This is an instantiation of the Equation 6.9, where the groundings are regions, and the weights are 1 for any valid region.

With this distribution, as the number of existing regions of the same type increases, the likelihood of creating a new region decreases. If this grounds to an existing region, there would be no modification to the graph particle. If it grounds to a new region, we add a new region to the topology, as well as an accompanying placeholder node and an edge from the current node n_t to this node. We also sample a metric constraint associated with this edge. The sampling method is explained at the end of this section. We sample the location of this new hypothesized region to be near the frontier locations in the environment. We follow the same procedure for objects, except we bias the sampling of the object's location towards areas unobserved by the robot (by taking account the robot's field-of-view and places already visited).

Alternatively, an annotation can express a relationship between two spatial entities. Specifically, the existence of a landmark region $R_{\mathcal{L}}$ with a label $l_{\mathcal{L}}$, a figure region $R_{\mathcal{F}}$ with a label $l_{\mathcal{F}}$, and a spatial relation r:

$$\exists R_{\mathcal{F}} : l_{R_{\mathcal{F}}} = l_{\mathcal{F}}, \tag{6.13a}$$

$$\exists R_{\mathcal{L}} : l_{R_{\mathcal{L}}} = l_{\mathcal{L}}, \tag{6.13b}$$

$$\exists r(R_{\mathcal{F}}, R_{\mathcal{L}}). \tag{6.13c}$$

For example, the command "go to the lobby through the hallway" describes the presence of a figure region ($R_{\mathcal{F}}$) with label "lobby" ($l_{\mathcal{F}}$), a landmark region ($R_{\mathcal{L}}$) with label "hallway" ($l_{\mathcal{L}}$), and the spatial relation "through" ($r(R_{\mathcal{F}}, R_{\mathcal{L}})$) that exists between them. For annotations of this type, we employ a two-stage sampling process to find the groundings. We attempt to ground the figure and landmark regions specified in the annotation to an existing pair of regions (with correct labels) in each particle's topology. We define $\mathbb{S}_{\mathcal{F},\mathcal{L}} = \{R_i, R_j | l_{R_i} = l_{\mathcal{F}}, l_{R_j} = l_{\mathcal{L}}\}$, where $p_{r(R_i,R_j)}$ is the likelihood of a pair of regions $\{R_i, R_j\} \in \mathbb{S}_{\mathcal{F},\mathcal{L}}$ conforming to the spatial relation r, and $\theta_{\mathcal{F},\mathcal{L}}$ as a parameter that biases the grounding to at least one new region. The probability of a pair of groundings is given

by the distribution:

$$p(R_{\mathcal{F}} = R_i, R_{\mathcal{L}} = R_j) = \begin{cases} \frac{p_{r(R_i,R_j)}}{\sum_{\{R_i,R_j\}\in\mathbb{S}_{\mathcal{F},\mathcal{L}}} p_{r(R_i,R_j)}+\theta_{\mathcal{F},\mathcal{L}}}, & \text{if } \{R_i, R_j\} \in \mathbb{S}_{\mathcal{F},\mathcal{L}} \\ \frac{\theta_{\mathcal{F},\mathcal{L}}}{\sum_{\{R_i,R_j\}\in\mathbb{S}_{\mathcal{F},\mathcal{L}}} p_{r(R_i,R_j)}+\theta_{\mathcal{F},\mathcal{L}}} & R_i = \textit{new} \text{ or } R_j = \textit{new} \end{cases} \tag{6.14}$$

This is an instantiation of Equation 6.9, where the groundings are pairs of regions, and the weights w_i's are the likelihood of these region pairs conforming to the given spatial relation. If sampling using the above equation results in two existing regions, there will be no modification to the sample topology. Otherwise, we apply the following procedure;

1. First, sample a grounding for the landmark region $R_{\mathcal{L}}$ using Equation 6.12. If this results in an existing region, we create a new (hypothesized) region with label $l_{\mathcal{F}}$ to represent the figure, sample a constraint consistent with r and add these to the topology.

2. If the above step results in a new region for $R_{\mathcal{L}}$, we sample a grounding for the figure region $R_{\mathcal{F}}$ using Equation 6.12. If this results in an existing region, we create a new (hypothesized) region with label $l_{\mathcal{L}}$ to represent the landmark, sample a constraint consistent with r and add these to the topology.

3. If both of the above steps result in new region groundings for $R_{\mathcal{L}}$ and $R_{\mathcal{F}}$, then we create a new (hypothesized) landmark region and a (hypothesized) figure region with labels $l_{\mathcal{L}}$ and $l_{\mathcal{F}}$ respectively.

When the above process results in at least one hypothesized region, we add the pair of regions and the corresponding relation r to an outstanding set of annotations $\mathbb{S}_{\alpha}$. We follow the same procedure for annotations describing objects.

In this section we outlined how we sample modifications to the graph based on Next we outline how the algorithm samples the spatial constraints described above.

Sampling Spatial Relation Constraints

When sampling a constraint based on a specified spatial relation, we use a set of models trained from a natural language corpus [88]. These models employ features that describe the locations of the regions or objects, the region boundaries and the location of the robot at the time of the utterance. We make use of the set of features outlined in Tellex et al. [88] to learn this distribution. When sampling the constraint given the spatial relation, we use the location of one region (observed or hypothesized) and the robot's location, and sample a set of possible locations for the second region (within a area that bound the maximum size of the environment) and evaluate the likelihoods of the resulting constraints. We then select the sampled location which resulted in the maximum likelihood for the spatial relation, and use the corresponding metric constraint.

Sampling Object Locations without a Spatial Relation

We keep track of the area that has been observed based on the locations visited by the robot and the field-of-view of its sensors. When we sample a new object based on natural language, we need to ensure that the new sampled object is not in an area already visited by the robot (as this should already have been observed assuming perfect sensing). Thus we limit the sampling of the constraint to sample only from unobserved areas in the environment.

Sampling Region Locations without a Spatial Relation

We keep track of frontier areas in the environment that could be traversable by the robot using the robot's lidar observations. When sampling new regions based on language, we sample locations near these frontier regions.

In this section we outlined how the algorithm samples modifications to each semantic map sample based on a sequence of annotations inferred from the natural language command. Each annotation contains information about the presence of regions or objects and specify any spatial relations between them. We use these to sample new regions and objects and also infer weak metric constraints. Next we outline how graph modifications are sampled based on observations made by the robot using its onboard sensors.

6.3.2 Graph Modification Based on Robot Observations

This section details how we sample modifications to the topology based on the robot's observations of new objects and regions.

$$p(\Delta_{z_t}^{(i)}|G_t^{(i)-}, z_t) = \prod_j p(\Delta_{z_{t,j}}^{(i)}|G_t^{(i)-}, z_{t,j}). \tag{6.15}$$

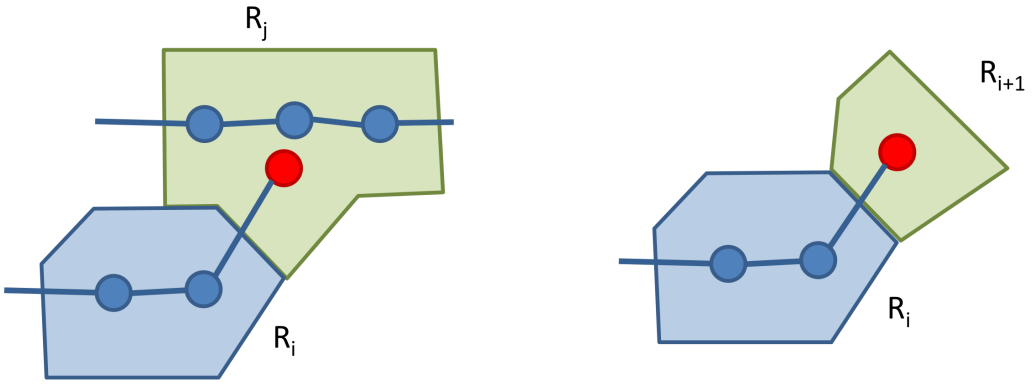
We make use of two types of observations made by the robot: observations of region transitions (z_t^R) made at node n_t and the location and types of objects in the environment (z_t^o). The algorithm uses a region transition to infer that the robot has moved to a different region, and attempts to associate this newly observed region with one of the existing regions, or with a new region. Conceptually this is similar to what we do with language annotations, but the robot's observations of a region provide more precise information especially about its connectivity, metric location and spatial extent. While a similar process is carried out for object observations, unlike a new region observation, the robot can observe the presence and location of multiple objects at a single instant.

Next two sections outline how this sampling is carried out based on observations of the regions and objects respectively.

Graph Modification Based on Region Transitions

As the robot traverses the environment attempting to satisfy the language command, it travels through new regions in the environment. In this chapter, we assume that the robot is immediately aware when it travels from one region to another. This is in contrast to the spectral clustering method we used in Chapter 4, which will only detect a transition sometime after entering a new region. We achieve this in the physical experiments using AprilTag fiducials [65] to mark the entrance of each region. We trigger a region transition when the robot observes a new AprilTag that conflicts with the current region's label. We also use the fiducials as an observation of the region label

If the robot does not observe a region transition, the algorithm adds the new node n_t to the current region, modifying its spatial extent. If there are any hypothesized regions that have a constraint from the current region based on a spatial relation, the algorithm



(a) (b)

Figure 6-8: Assignment of the current node (shown in red) (a) to a previously visited region R_j (b) to a new region R_{i+1}.

evaluates the sampled constraint and re-samples it if the likelihood has changed. This is done because some of the features that are used to calculate the likelihood of a constraint are dependent on the region's spatial extent, which grows as the robot observes more of the region.

If however, a region transition is observed, the new node n_t is assigned to a new or existing region using the following steps.

- **Grounding to an Existing Observed Region:**

  First we check if this new region is one that was previously visited by the robot as shown in Figure 6-8a. In this implementation, we deterministically assign this to an existing (visited) region of the same type if the distance from the new node to any node in this region is below a distance threshold. Otherwise we create a new region to represent visiting a new part of the environment as shown in Figure 6-8b. This new region is given the observed label and associated with the current node n_t as its only node.

- **Grounding to an Existing Hypothesized Region:**

  If the above step resulted in the creation of a new region R_n, we then attempt to ground this region to any existing hypothesized regions in the particle (of the same type). This is done using a similar process to how language grounding is carried out.

However, unlike annotation groundings, we select a grounding for the new region R_n from the set of hypothesized regions. This is done in a two-step process.

1. We select from the ungrounded set $\mathbb{S}_\alpha$, the set of region pairs where the only hypothesized region is the one that has the same label as l_{R_n}. If this set is not empty we sample a grounding to one of the hypothesized regions using a similar Dirichlet distribution as outlined Equation 6.9. In this scenario, the weight w_i's are the likelihood of the spatial relation when replacing the hypothesized region with the current region. This biases us to sample the hypothesized region that when replaced will best conform to the associated spatial relation. If this results in a valid grounding to a hypothesized region, we remove the associated set of regions and relation from the ungrounded set $\mathbb{S}_\alpha$, remove the hypothesized region and the constraint.
2. If the above procedure did not result in a hypothesized region being grounded to the newly observed region, we consider the wider set of hypothesized regions that match the labels (except for the ones considered above). We sample a grounding to one of these hypothesized regions using a similar distribution to Equation 6.12. This set includes regions in the ungrounded set $\mathbb{S}_\alpha$ where both regions are hypothesized and regions created based on annotations that described the existence of a region (equation 6.11).

Next we outline how the algorithm samples modifications to the graph based on new object observations.

Graph Modification Based on Object Observations

When the robot makes observations of objects in the environment at node n_t, it also results in several possible modifications to the graph. The observation includes the location and the type of object. We sample modifications to the topology based on each observation using the following steps.

- **Grounding to an Existing Observed Object:**

  First, the algorithm samples a grounding to a previously observed object (of the same

type). We use an observation model defined based on the sensor's range and field-of-view to calculate the likelihood of each existing object generating the given observation. When calculating this likelihood, we account for the uncertainty associated with the object's relative location to the robot. If this sampling results in a grounding to an existing object, we create a new edge between n_t and the existing object's node. This edge encodes the metric constraint observed using the robot's camera.

- **Grounding to an Existing Hypothesized Object:**
  If the above step did not result in grounding to an existing observed object, the algorithm samples a grounding to an object hypothesized based on language. This follows the same two step process outlined above for grounding the new region to an existing hypothesized region.

- **Ground to a New Object:**
  If both steps fail to ground this observation to an existing (observed or hypothesized) object, we create a new object, and a node to represent its location and an edge between n_t and this node that express the metric constraint.

This section described how the algorithm uses sensor observations to sample modifications each semantic map particle. We treat these sensor observations to infer the presence of regions and objects in the environment, and then sample modifications based on whether they were previously observed by the robot or described by language.

6.3.3 Update the Metric Information

After proposing modifications to each particle, we perform a Bayesian update to its Gaussian distribution similar to the approach in Chapter 4.2.2. The nodes and edges in the lower-level topology of each particle are used to induce a pose graph, which allows us to maintain the distribution over the metric locations.

6.3.4 Re-weighting Particles

We then re-weight each particle by taking into account the likelihood of generating language annotations, and robot observations.

$$w_t^{(i)} = p(z_t, \alpha_t | G_{t-1}^{(i)})\, w_{t-1}^{(i)} = p(\alpha_t | G_{t-1}^{(i)})\, p(z_t | G_{t-1}^{(i)})\, w_{t-1}^{(i)}. \tag{6.16}$$

The re-weighting step accounts for the differences between the proposal distributions that we employed to sample the modifications to the graph and the distribution which resulted in the observations.

Language Observation Model

For annotations, we use the natural language grounding likelihood under the map at the previous time step. As such a particle with an existing pair of regions conforming to a specified language constraint will be weighted higher than one without. This likelihood is calculated by evaluating the probability of the matching object pair for the spatial relation specified in the annotation.

Region Observation Model

For region observations, we model the observation likelihood such that it down weights particles that have hypothesized regions on top of areas that the robot has already observed as being of a different region type (by traversing through these areas). Since the robot is able to observe the region type of its current location (which is part of our assumptions that is enabled with the placement of AprilTags in the experiments) we evaluate the likelihood of each hypothesized regions generating this region type observation.

This function is modeled such that only hypothesized regions that are spatially very close to the robot's current location can influence the region type observation. For each hypothesized region R_i, we model its ability to influence the observed region type at the current location using a binary latent variable v. The likelihood $p(v|R_i)$ is high when the distance to the region from the current location is below a distance threshold. We then define a likelihood function $p(z_t|v, R_i)$ for generating a given region type observation given

v and its type. This function encodes the high correlation between the region's type and the observed type when v is True.

Then for each particle, the likelihood generating the current observation is:

$$p(z_t|G_{t-1}^{(i)}) = \prod_{R_i \in R_u} \left(\sum_{v \in True, False} p(z_t|v, R_i) \times p(v|R_i) \right), \tag{6.17}$$

where where R_u is the set of unobserved regions in particle $S_{t-1}^{(i)}$.

Object Observation Model

For each particle $S_{t-1}^{(i)}$, we use both the positive and negative information in the observations.

For positive observations of objects, we evaluate the likelihood of generating each observed object given each topological sample $G_{t-1}^{(i)}$. This assigns the observation to an existing object in the topology (that provides the maximum likelihood) and calculates the likelihood of observing the object at the given location. This likelihood is defined using the observed relative constraint of the object compared to the relative constraint of the object's current location in the map to the robot. Since the location of objects observed by the robot's sensor are the same in each particle, the difference between the particles are based on having a hypothesized region near where an actual object (of the same type) is observed.

For each existing object that did not generate a matched observation, we calculated the likelihood of not generating an observation given the relative location of the object to the robot. This also makes use of the sensor's field-of-view and the uncertainty over the pose of the object relative to the robot.

We define this negative likelihood function as:

$$p(z_n^i|o_i) = \int_{x_{n_t}} \int_{x_{o_i}} p(z_n|x_{n_t}, x_{o_i}) dx_{n_t} dx_{o_i} \tag{6.18a}$$

$$= \int_{r_{o_i}} \int_{\theta_{o_i}} p(z_n|r_{o_i}, \theta_{o_i}) dr_{o_i} d\theta_{o_i}, \tag{6.18b}$$

where z_n^i denotes not observing object i in the topology, x_{n_t} is the robot's current pose, x_{o_i}

is the object's current pose, and r_{o_i} and θ_{o_i} denote the relative range and heading of object o_i from the robot's pose.

We employ a simplified observation model for our sensor that assigns the likelihood of observing an object to be 1.0 if the relative range and heading of the object to the robot falls within the sensor's modeled range and field-of-view. The complement is used to calculate the likelihood of a negative observation. This results in objects that fall within the sensor's sensing range and field-of-view with high confidence to have a low-likelihood of generating a negative observation. Since the only variability of objects between each particle is the hypothesized objects, this results in sampled topologies with objects in locations inconsistent with the actual layout world having lower weight as the robot observes the relevant areas.

6.3.5 Resampling

When the particle weights fall below a threshold, we resample particles to avoid particle depletion [15]. When an existing particle is duplicated in this sampling process, any edges that were created based a spatial constraint specified by a language annotation is re-sampled.

The semantic mapping algorithm outlined here is used to infer a distribution over the robot's environment. By treating information inferred by language on par with sensor observations, we are able to learn about regions and objects in the world that is relevant to accomplishing the user's command. By inferring spatial layout information from these annotations, we induce a prior over the semantic map that is refined as the robot observes the world with its own sensors as it performs actions. This allows the robot to refine its behavior over time as its distribution over the environment improves.

6.4 Learning Belief Space Policies

Searching for the complete trajectory that is optimal in the distribution of maps would be intractable. Instead, we treat the task of following natural language commands as sequential decision making under uncertainty, where a policy π minimizes a single step of the cost

function c over the available actions $a \in A_t$ from state x:

$$\pi\left(x, S_t, \beta_t\right) = \arg\min_{a \in A_t} c\left(x, a, S_t, \beta_t\right), \tag{6.19}$$

where β_t is the behavior provided by the Behavior Inference 6.2.2. After taking the action and updating the map distribution, we repeat this process until the policy declares it has completed following the direction.

As the robot travels in the environment, it keeps track of the nodes it has visited $\mathcal{V}$ and frontiers $\mathcal{F}$ which lie at the edge of explored space. The action set A_t consists of paths to nodes in the graph. We only consider paths that terminate at a node in $\mathcal{F}$ in order to bias exploration towards unknown areas and prevent repeatedly visiting areas. An additional action a_{stop} declares that the policy has completed following the direction. Intuitively, an action represents a single step along the path that takes the robot to its destination. Each action may explore new parts of the environment (for example continuing to travel down a hallway) or backtrack if the policy has made a mistake (for example, traveling to a room in a different part of the environment).

The following sections explain how the policy reasons in belief space, and the novel imitation learning formulation to train the policy from demonstrations of correct behavior.

6.4.1 Belief Space Reasoning using Distribution Embedding

The semantic map S_t provides a distribution over the possible locations of landmarks in the world, while β_t specifies which landmarks are relevant to the command currently being followed, as inferred via Behavior Inference described in Section 6.2.2. The policy π must reason about the *distribution* of relevant landmarks when computing the cost of any action a. We accomplish this through a kernel embedding of the semantic map distribution [84], using the first K moments of the features computed with respect to each map sample $S_t^{(i)}$

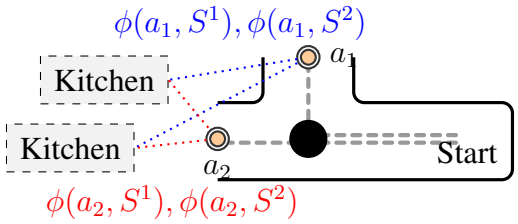


Figure 6-9: Simplified illustration of computing feature moments in the space of hypothesized landmarks (two kitchens in this case). For each action, we aggregate the features across all hypothesized kitchens. These are then used to compute moment statistics.

and behavior $\beta_t^{(j)}$:

$$\hat{\Phi}_1\left(x, a, S_t, \beta_t\right) = \sum_{S_t^{(i)}, \beta_t^{(j)}} p(\beta_t^{(j)}, S_t^{(i)})\ \phi\left(x, a, S_t^{(i)}, \beta_t^{(j)}\right) \tag{6.20}$$

$$\hat{\Phi}_2\left(x, a, S_t, \beta_t\right) = \sum_{S_t^{(i)}, \beta_t^{(j)}} p(\beta_t^{(j)}, S_t^{(i)})\ \left(\phi\left(x, a, S_t^{(i)}, \beta_t^{(j)}\right) - \hat{\Phi}_1\right)^2 \tag{6.21}$$

$$\ldots$$

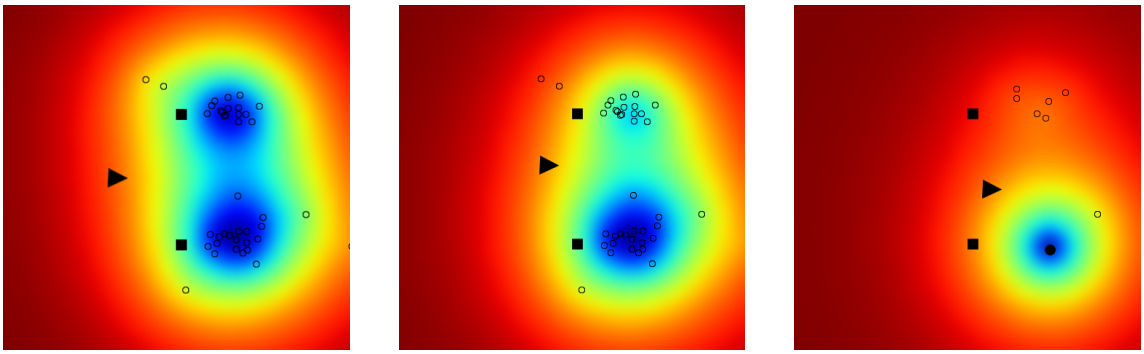
$$\hat{\Phi}_k\left(x, a, S_t, \beta_t\right) = \sum_{S_t^{(i)}, \beta_t^{(j)}} p(\beta_t^{(j)}, S_t^{(i)})\ \left(\phi\left(x, a, S_t^{(i)}, \beta_t^{(j)}\right) - \hat{\Phi}_1\right)^k \tag{6.22}$$

Intuitively, this computes features for the action and all relevant (observed or hypothesized) landmarks individually, aggregates these feature vectors, and then computes moments of the feature vector distribution (mean, variance, and higher order statistics). Each inferred behavior $\beta_t^{(j)}$ specifies the relevant landmarks for a given $S_t^{(i)}$. A simplified illustration is shown in Figure 6-9, for a command that goes to an unknown kitchen (with two possible hypothesized locations).

The cost function in (6.19) is modeled as a weighted sum of the first K moments of the feature distribution:

$$c\left(x, a, S_t\right) = \sum_{i=1}^{K} w_i^T\ \hat{\Phi}_i\left(x, a, S_t, \beta_t\right). \tag{6.23}$$

By concatenating the weights and moments into respective column vectors $W := [w_1; \ldots; w_k]$ and $F := [\hat{\Phi}_1; \ldots; \hat{\Phi}_k]$, we can rewrite the policy in (6.19) as minimizing a weighted sum



(a) $t = 0$ (b) $t = 4$ (c) $t = 8$

Figure 6-10: Visualization of the value function over time for the command "go to the hydrant behind the cone," where the triangle denotes the robot, squares denote observed cones, and circles denote hydrants that are sampled (empty) and observed (filled). The robot starts off having observed the two cones, and hypothesizes possible hydrants that are consistent with the command (a). The robot first moves towards the left cluster, but after not observing the hydrant, the map distribution peaks at the right cluster (b). The robot then moves right and observes the actual hydrant (c).

of the feature moments F_a for action a:

$$\pi\left(x, S_t\right) = \arg\min_{a \in A_t} W^T F_a. \tag{6.24}$$

The vector $\phi(x, a, S_t^{(i)}, \beta_t^{(j)})$ computes features of the action and a *single* landmark in $S_t^{(i)}$ as specified in $\beta_t^{(j)}$. It contains geometric features describing the shape of the action (e.g., the cumulative change in angle), the geometry of the landmark (e.g., the area of the landmark), and the relationship between the action and landmark (e.g., the difference between the ending and starting distances to the landmark). See Duvallet et al. [18] for more details. Figure 6-10 shows the evolution of this value function in one of our experiments.

6.4.2 Imitation Learning Formulation

The policy is trained using imitation learning, by treating action prediction as a multi-class classification problem. Given an expert demonstration, we wish to correctly predict the expert's action out of all possible actions from the same state. Although prior work introduced imitation learning for training a policy to follow directions, it operated in partially known

environments [18]. In this work, we train the policy using a distribution of hypothesized maps to learn a belief space policy.

We assume the expert's policy π^* minimizes the unknown immediate cost $C(x, a^*, S_t)$ of performing the demonstrated action $a*$ from state x, under the current belief distribution S_t. However, since we cannot directly observe the true costs of the expert's policy, we must instead minimize a surrogate loss that penalizes disagreements between the expert's action a^* and the policy's action a, using the multi-class hinge loss [10]:

$$\ell\left(x, a^*, c, S_t\right) = \max\left(0, 1 + c\left(x, a^*, S_t\right) - \min_{a \neq a^*}\left[c\left(x, a, S_t\right)\right]\right). \tag{6.25}$$

The minimum of this loss occurs when the cost of the expert's action is lower than the cost of all other actions, with a margin of one. This loss can be re-written and combined with equation 6.24 to yield:

$$\ell\left(x, a^*, W, S_t\right) = W^T F_{a^*} - \min_a\left[W^T F_a - l_{xa}\right], \tag{6.26}$$

where $l_{xa} = 0$ if $a = a^*$ and 1 otherwise. This ensures that the expert's action is better than all other actions by a margin [74]. Adding a regularization term λ to (6.26) yields our complete optimization loss:

$$\ell\left(x, a^*, W, S_t\right) = \frac{\lambda}{2}\|W\|^2 + W^T F_{a^*} - \min_a\left[W^T F_a - l_{xa}\right]. \tag{6.27}$$

Although this loss function is convex, it is not differentiable. However, we can optimize it efficiently by taking the subgradient of (6.27) and computing action predictions for the loss-augmented policy [74]:

$$\frac{\partial \ell}{\partial W} = \lambda W + F_{a^*} - F_{a'}, \tag{6.28}$$

for the best loss-augmented action a' at state s:

$$a' = \arg\min_a\ \left[W^T F_a - l_{xa}\right]. \tag{6.29}$$

Note that a' is simply the solution to our policy using a loss-augmented cost. This leads to the update rule for W:

$$W_{t+1} \leftarrow W_t - \alpha \frac{\partial \ell}{\partial W} \tag{6.30}$$

with a learning rate $\alpha \propto 1/t^{\gamma}$. Intuitively, if the current policy disagrees with the expert's demonstration, (6.30) decreases the weight (and thus the cost) for the features of the demonstrated action F_{a^*}, and increases the weight for the features of the planned action F_a. If the policy produces actions which agree with the expert's demonstration, the update will only be for the regularization term.

We train the policy using the DAGGER (Dataset Aggregation) algorithm [77], which learns a policy by iterating between collecting data (using the current policy) and applying expert corrections to the decisions that were made (using the expert's demonstrated policy). Key to this approach is that we collect training information from all states visited by the policy, not just states that were in the demonstration [18]. This enables us to learn a policy that does well on the distribution of states induced by the learned policy, instead of only the distribution of states that were visited by the expert.

Treating direction following in the space of possible semantic maps as a problem of sequential decision making under uncertainty provides an efficient approximate solution to the belief space planning problem. By using a kernel embedding of the distribution of features for a given action, we still reason about the distribution of landmarks in the semantic map. Using imitation learning for training the policy is simple, elegant, and requires no complex engineering of components or tuning of parameters.

6.5 Experimental Evaluation

In this section we outline the experimental evaluation of our framework to follow object relative navigation commands and natural language route directions.

6.5.1 Following Object-Relative Navigation Commands

This section outlines the application of our approach to follow directions to objects in unknown environments. Since the directions considered for this application were limited to references to objects, our semantic maps do not reason about different regions in the environment. As such, the trajectory traversed by the robot belong to a single region.

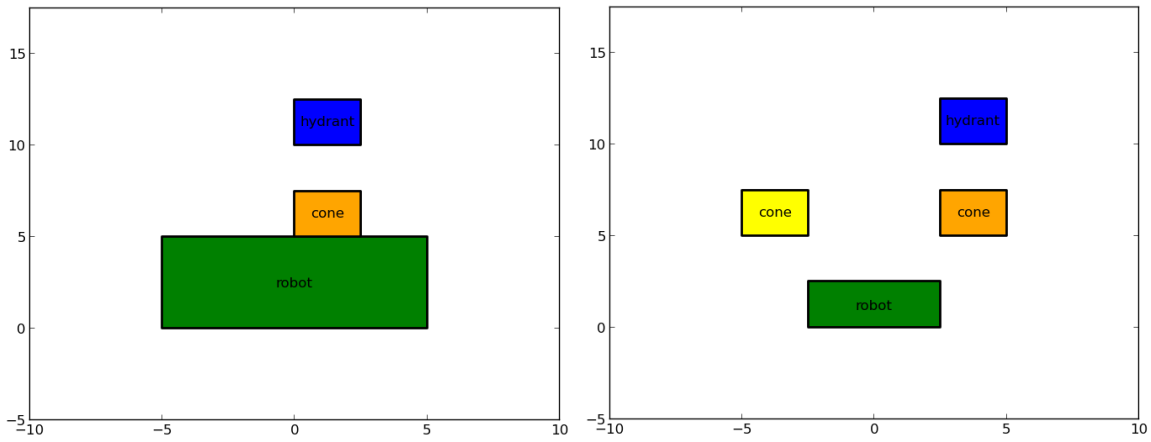
We analyze the effectiveness of our end-to-end framework through simulations that consider environments and commands of varying complexity, and different amounts of prior knowledge. We then demonstrate the utility of our approach in practice using experiments run on two mobile robot platforms. These experiments provide insights into our algorithm's ability to infer the correct behavior in the presence of unknown and ambiguous environments.

Monte Carlo Simulations

First, we evaluate the entire framework through an extended set of simulations in order to understand how the performance varies with the environment configuration and the command. We consider four environment templates, with different numbers of figures (hydrants) and landmarks (cones). Figure 6-11 shows two of the templates used for these experiments. For each configuration, we sample ten environments, each with different object poses. For these environments, we issued three natural language instructions "go to the hydrant," "go to the hydrant behind the cone," and "go to the hydrant nearest to the cone," which were not part of the corpus used to train the HDCG model. For each sampled environment, we ran 10 trials for each language command, resulting in 100 trials for each combination of environment template and language instruction. We considered a trial to be successful if the planner stops within 1.5 m of the intended goal.

Table 6.1 presents the success rate and distance traveled by the robot for these trials. We also provide the results for each command where the planner used a completely known world model as a ground-truth baseline for these environments.

We compare the performance of our method for commands that contained useful information about the environment ("go to the hydrant behind the cone" and "go to the hydrant



(a) 1C, 1H (b) 2C, 1H

Figure 6-11: Templates used to sample environments for the instruction "go to the hydrant behind the cone": Locations for the objects and robot were sampled to be within the specified region boundaries

nearest to the cone") against the commands that did not contain information about the environment ("go to the hydrant"). The results demonstrate that our algorithm achieves greater success compared to the uninformed language commands in all the environments. In two of the environments, it also results in more efficient paths with shorter distances on average. However, for environments with one hydrant and two cones (1H, 2C) our method performs worse on average when using the information from language. But it should also be noted that this also results in a higher standard deviation for the distance traveled. This result is due to the ambiguity in the environment, where there are two cones in the environment. In

Table 6.1: Monte Carlo simulation results with 1σ confidence intervals (Hydrant, Cone).

			Success Rate (%)		Distance (m)	
World	Range (m)	Relation	Known	Ours	Known	Ours
1H, 1C	3.0	null	100.0	93.9	8.75 (1.69)	16.78 (7.90)
1H, 1C	3.0	"behind"	100.0	98.3	8.75 (1.69)	13.43 (7.02)
1H, 2C	3.0	null	100.0	100.0	11.18 (1.38)	32.54 (18.50)
1H, 2C	3.0	"behind"	100.0	99.5	11.18 (1.38)	40.02 (29.66)
2H, 1C	3.0	null	100.0	54.4	10.49 (1.81)	21.56 (10.32)
2H, 1C	3.0	"behind"	100.0	67.4	10.38 (1.86)	18.72 (10.23)
2H, 1C	5.0	"nearest"	100.0	46.2	9.19 (1.54)	12.05 (5.76)

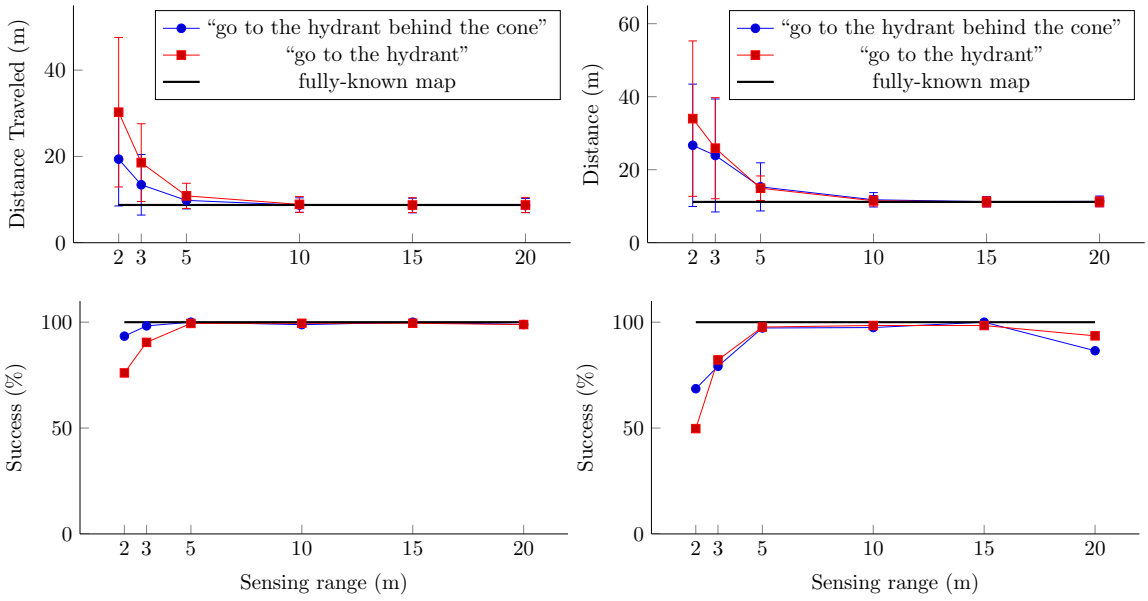


Figure 6-12: Simulation results for distance traveled (top) and success rate (bottom) as a function of the sensor range for the command "go to the hydrant behind the cone" with an unknown map, compared against the performance with a fully-known map and when given an uninformative command "go to the hydrant" (with an unknown map): (a) World contains 1 cone and 1 hydrant (b) World contains 1 cone and 2 hydrants.

instances where the robot first travels behind the cone that actually has the hydrant behind it, the distance traveled is significantly smaller. But, in instances where it selects the other cone, the robot ends up navigating behind the wrong cone until the belief in those particles are down-weighted enough. This results in a longer distance in the second scenario. This increases the average distance value.

One interesting failure case is when the robot is instructed to "go to the hydrant nearest to the cone" in an environment with two hydrants. In instances where the robot sees a hydrant first, it hypothesizes the location of the cone, and then identifies the observed hydrants and hypothesized cones as being consistent with the command. Since the robot never actually confirms the existence of the cone in the real world, this results in the incorrect hydrant being labeled as the goal.

Next, we evaluate the how different sensing ranges affect the performance our framework, both in terms of the distance traveled and the success rate. We ran approximately 100 experiments each, with six different settings for the robot's sensing range (2 m, 3 m, 5 m,

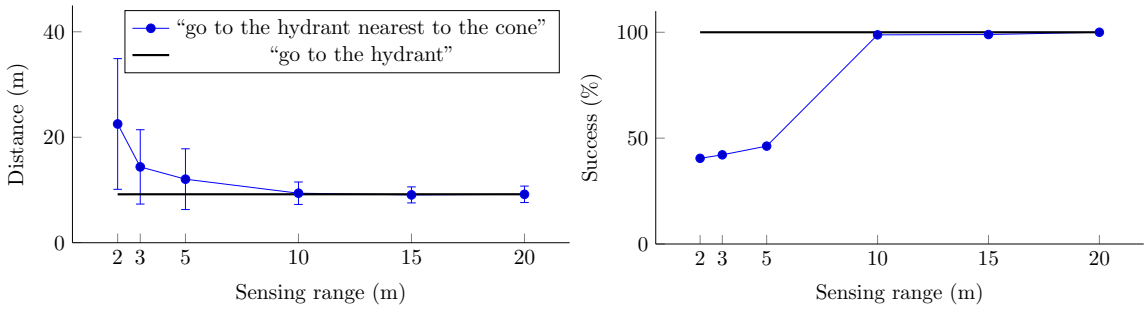


Figure 6-13: Simulation results for distance traveled (top) and success rate (bottom) as a function of the sensor range for the command "go to the hydrant nearest to the cone" with an unknown map, compared against the performance with a fully-known map. (world contains 1 cone and 2 hydrants)

10 m, 15 m, and 20 m) for the language instructions "go to the hydrant behind the cone" and "go to the hydrant nearest to the cone". Figure 6-12 outlines the change in performance for the command "go to the hydrant behind the cone" for two environments. We compare them to the baseline scenario where the command did not include information about the layout of the environment ("go to the hydrant") as well as the ground-truth baseline with a completely known map (ten runs each). Figure 6-12 shows how success rate increases and distance traveled decreases as the robot's sensing range increases, quickly approaching the performance of the system when it begins with a completely known map of the environment. It also illustrates that the robot's performance is poorer when the command does not contain information about the world. As the sensor's range increases, the robot is able to observe the relevant parts of with a smaller number of traversals, resulting in improving performance that approach the ground-truth baseline. We can also see that the gap between the performance of our approach and the baseline approach decreases as the sensing range increases. This reflects the fact that with increasing sensing ranges, there is diminishing benefit to learn about unobserved parts of the environment through language. Figure 6-13 shows the performance of the robot for the instruction "go to the hydrant nearest to the cone". There is significant improvement in the success rate with the increase in the sensing range. This is due to the fact that the robot is more likely to observe both hydrants as well as the cone with a larger sensing range, resulting in improved performance.

Physical Experiments

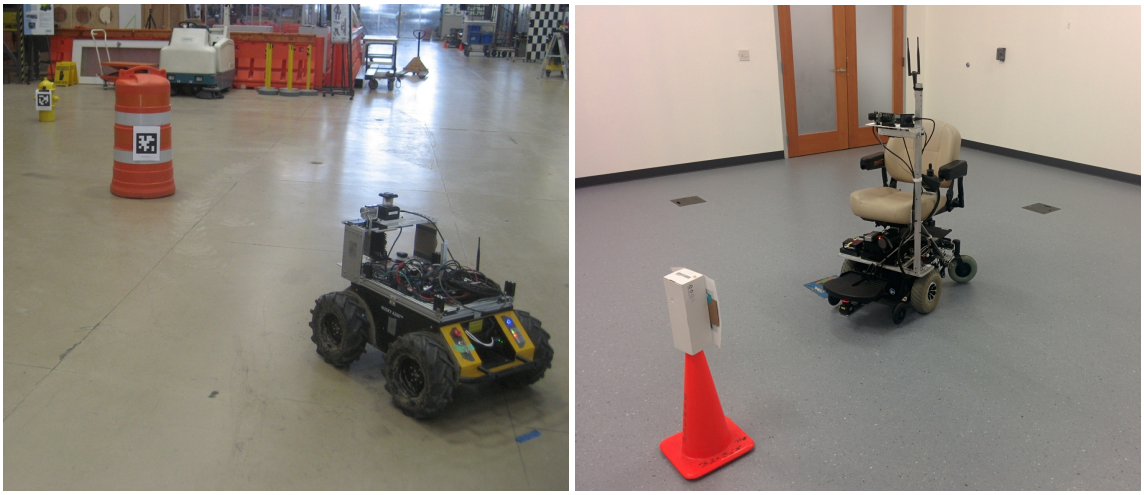
We applied our approach to two mobile robots, a Husky A200 mobile robot (Figure 6-14a) and an autonomous robotic wheelchair [30] (Figure 6-14b). The use of both platforms demonstrates the application of our algorithm to mobile robots with different vehicle configurations, underlying motion planners, and sensor configurations. The actions determined by the planner are translated into lists of waypoints that are handled by each robot's motion planner. We used AprilTag fiducials [65] to detect and estimate the relative pose of objects in the environment, subject to self-imposed angular and range restrictions.

In each experiment, a human operator issues natural language commands in the form of text that expresses (possibly null) spatial relations between one or two objects. The results that follow involve the commands "go to the hydrant," "go to the hydrant behind the cone," and "go to the hydrant nearest to the cone." As with the simulation-based experiments, these instructions did not match those from our training set. For each of these commands, we consider different environments by varying the number and position of the cones and hydrants and by changing the robot's sensing range. For each configuration of the environment, command, and sensing range, we perform ten trials with our algorithm. For a ground-truth baseline, we perform an additional run with a completely known world model. We consider a run to be a success when the robot's final destination is within 1.5 m of the intended goal.

Table 6.2 presents the success rate and distance traveled by the wheelchair for these experiments. Compared to the scenario in which the command does not provide a relation

Table 6.2: Experimental results with 1σ confidence intervals (Hydrant, Cone).

			Success Rate (%)		Distance (m)	
World	Range (m)	Relation	Known	Ours	Known	Ours
1H, 1C	2.5	null	100.0	100.0	4.69	16.56 (7.20)
1H, 1C	2.5	"behind"	100.0	100.0	4.69	9.91 (3.41)
1H, 2C	3.0	"behind"	100.0	100.0	4.58	7.64 (2.08)
2H, 1C	2.5	"behind"	100.0	80.0	5.29	6.00 (1.38)
2H, 1C	4.0	"nearest"	100.0	100.0	4.09	4.95 (0.39)
2H, 1C	3.0	"nearest"	100.0	50.0	6.30	7.05 (0.58)



(a) Husky (b) Wheelchair

Figure 6-14: The setup for the experiments with the (a) Husky and (b) wheelchair platforms.

(i.e., "go to the hydrant"), we find that our algorithm is able to take advantage of available relations ("go to the hydrant behind the cone") to yield behaviors closer to that of ground truth. The results are similar for the Husky platform, which resulted in an 83.3% success rate when commanded to "go to the hydrant behind the cone" in an environment with one cone and one hydrant. These results demonstrate the usefulness of utilizing all of the information contained in the instruction, such as the relation between various landmarks in the environment, that can be helpful during navigation.

The robot trials exhibited a similar failure mode as the simulation experiments: if the environment contains two figures (hydrants) and the robot only detects one, the semantic map distribution then hypothesizes the existence of a landmark (cone) in front of the hydrant, which leads to a behavior distribution peaked around this goal and plans that do not look for the possibility of another hydrant in the environment. As expected, this effect is most pronounced with shorter sensing ranges (e.g., a 3 m sensing range for the command "go to the hydrant nearest to the cone" resulted in the robot reaching the goal in only half of the trials compared to a 4 m sensing range).

6.5.2 Following Natural Language Directions

In this section we outline the application of our framework to following natural language route directions in unknown indoor environments. In our experiments, we only considered directions that reference the presence of regions. As such, the representation did not include any objects.

We evaluate the performance both in simulation and through physical experiments on the robotic wheelchair platform. We compare our framework against two other methods. One emulates the previous state-of-the-art and uses a known map of the environment in order to infer the actions consistent with the route direction. The second method assumes no prior knowledge of the environment (as with ours), but does not use language to modify the map. The language models were trained from a parallel corpus of 54 fully labeled examples.

Monte Carlo Simulations

In simulation, we created a world comprised of an office, hallway and a kitchen, with the robot starting off in the office. We commanded the robot to execute the instruction "go to the kitchen that is down the hallway." Our method achieved comparable results to the known map method while outperforming the method without language (Table 6.3). Each method was run ten times.

Physical Experiments

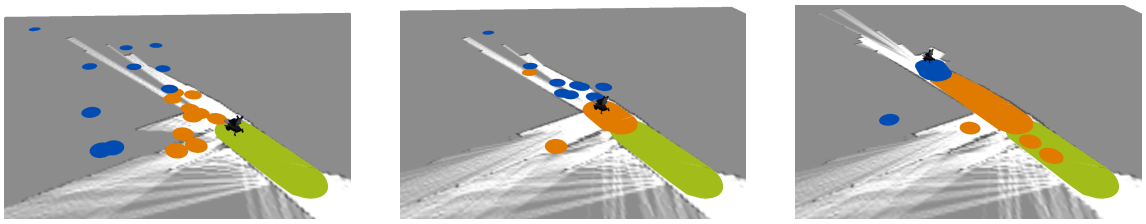
We implemented the algorithm on our voice-commandable wheelchair (Figure 6-1), which is equipped with three forward-facing cameras with a collective field-of-view of 120 de-

Table 6.3: Direction following efficiency in simulation

	Distance (m)		Time (s)	
Algorithm	Mean	Std Dev	Mean	Std Dev
Known Map	12.88	0.06	18.32	3.54
With Language	16.64	6.84	82.78	10.56
Without Language	25.28	12.99	85.57	17.80

Table 6.4: Direction following efficiency on the robot

	Distance (m)		Time (s)	
Algorithm	Mean	Std Dev	Mean	Std Dev
Known Map	13.10	0.67	62.48	16.61
With Language	12.62	0.62	122.14	32.48
No Language	24.91	13.55	210.35	97.73



(a) $t = 3$ (b) $t = 4$ (c) $t = 8$

Figure 6-15: Visualization of one run on the robot, depicting the evolution of the semantic map over time for the command "go to the kitchen that is down the hallway." Sampled regions are drawn as small circles and visited regions are shown with the area filled in (lab: green, hallway: yellow, kitchen: blue). The robot first samples possible locations of the kitchen and moves towards them (a), then observes the hallway and refines its estimate using the "down" relation provided by the user (b). Finally, the robot reaches the actual kitchen (c) and declares it has finished following the direction.

grees, and forward- and rear- facing lidars. We set up an experiment in which the wheelchair was placed in a lobby within MIT's Stata Center, with several hallways, offices, and lab spaces, as well as a kitchen on the same floor. As scene understanding is not the focus of this chapter, we placed AprilTag fiducials [65] that identified the different regions in the environment.

We then directed the wheelchair to execute the instruction "go to the kitchen that is down the hallway." We performed six runs with our algorithm, three runs with the known map method, and five with the method that does not use language, all of which were successful. Table 6.4 compares the total distance traveled and total execution time when using the three different methods. Our algorithm resulted in paths whose lengths were close to those of the known map, and significantly outperformed the method that did not use language. Our framework did require significantly more time to follow the directions than the known map method, due to the fact that it repeats the three steps of the algorithm when new

sensor data arrives. Figure 6-15 shows a visualization of the semantic maps over several time steps in one successful run on the robot.

6.6 Discussion

Enabling robots to reason about parts of the environment that have not yet been visited solely from a natural language description serves as one step towards effective and natural collaboration in human-robot teams. By treating language as a sensor that informs the robot about the spatial structure of areas outside the robot's immediate field-of-view, we are able to paint a rough picture of what the unvisited parts of the environment could look like. We utilize this information during planning, and update our belief with actual sensor information during task execution.

Our approach exploits the information implicitly contained in language to infer the existence of relationship between objects and regions that may not be initially observable. By learning a distribution over the map, we generate a useful prior that enables the robot to sample possible hypotheses, representing different environment possibilities that are consistent with both the language and the available sensor data. Learning a policy that reasons in the belief space of these samples achieves a level of performance that approaches that of an *a priori* known environment.

A key component of usefully reasoning about the environment using natural language is the ability to hypothesize configurations of the world suggested by the currently observed spatial structure. While our approach reasons about observed parts of the environment when sampling hypothesized locations in the world, it makes simplifying assumptions about the observability of the objects due to the use of Apriltags. We also assume a simple observation model for the object detection, with a range and a field-of-view, which would not necessarily translate to actual object detectors. Additionally, when sampling unobserved regions in the world based on language, we make use of frontiers in the world observed using lidar. This method can be noisy and may fail to detect parts of the world behind doorways etc. This can lead to failures where no particle contains valid hypothesis in the correct areas. In the direction following, we also assumed the ability to observe

region type and transitions immediately. In an actual real world scenario, this would not be possible. There is uncertainty around the segmentation of regions, and also uncertainty about the region type that can be inferred using the robot's sensors. This would require us to reason over the distribution of region labels for each region, which would require a larger number of particles.

Also, while we reason over the distribution over the world given natural language, our belief space planning framework does not explicitly reason over potential information gathering actions. More comprehensive approaches that also reason over information gathering actions as well as actions that satisfy the command might result in better performance.

Chapter 7

Conclusion

For robots to effectively interact with humans, they require the ability to learn representations of their environments that are compatible with the conceptual models used by people. Current approaches to constructing such spatial-semantic representations rely predominantly on traditional sensors to acquire knowledge of the environment, which restricts robots to learning limited knowledge of their local surround. In contrast, natural language descriptions allow people to share rich information about their environments with their robot partners in a flexible, efficient manner that allows robots to *observe* spatial and semantic properties that are beyond the range and capabilities of traditional sensors.

This thesis has addressed the problem of fusing information contained in natural language descriptions with the robot's onboard sensors to construct spatial-semantic representations useful for human-robot interaction. The novelty of the thesis lies in its treatment of natural language as another sensor observation that can inform the robot about its environment. Towards this end, we have introduced algorithms that allow the robot to learn from natural language descriptions that describe spatial entities, such as regions and objects that may be unknown to the robot and outside its field-of-view. Our algorithms use information contained in such descriptions to learn hard-to-perceive semantic properties of the world and the spatial structure of unvisited parts of the environment. We then use these learned models to enable robots to interact more effectively with human partners.

7.1 Contributions

We summarize the key contributions made in this thesis towards learning spatial-semantic representations from natural language descriptions.

Learning Representations from Natural Language and Scene Appearance

We introduced the semantic graph, a novel representation that combines metric, topological, and semantic models of the environment, and a probabilistic algorithm (Chapter 3) that efficiently maintains the joint distribution over this representation, conditioned on the language and the metric observations from the robot's proprioceptive and exteroceptive sensors during a narrated guided tour. We demonstrated the algorithm's ability to learn semantic properties of the environment from natural language descriptions about distant parts of the environment, including the ability to handle descriptions that refer to yet-unvisited parts of the world. We showed how this semantic information can be used to improve the spatial representation in a number of large-scale environments.

We also presented an extension to this algorithm (Chapter 4) that introduced a more spatially accurate and compact representation, and the ability to merge natural language with other sources of semantic information inferred from the robot's own sensors to create semantic models that allow for better natural language integration.

The two aforementioned algorithms wait till the robot visits a referenced location before incorporating knowledge conveyed by language. In our final semantic mapping algorithm (Chapter 6) the robot uses information contained in natural language instructions to directly learn about the spatial properties about unvisited regions in the world. The algorithm probabilistically reasons over the presence of and spatial relations between regions and objects specified by language to directly extend its spatial representation.

Improving Spatial-Semantic Representations

We introduced a mechanism that allows the robot to reason over the ambiguity of natural language descriptions and to ask questions from the user during the course of a guided tour. Our mechanism balances expected information gain of asking a question with a cost that

measures the burden on the user. We showed how this approach results in less ambiguity over the descriptions and semantic maps that are more accurate.

Learning Models for Following Natural Language Instructions in Unknown Environments

Next, we outlined a novel approach that enables a robot to follow natural language navigation instructions in completely unknown environments by using our semantic mapping algorithm to learn a prior over the spatial layout of distant (as yet unobserved) parts of the environment using information contained in the instruction. Our algorithm then uses this distribution over the world to solve for a policy consistent with the language instruction and to then take an action. As the robot observes the world while executing its actions, the algorithm improves the semantic map distribution, which leads to more accurate behavior that ultimately satisfies the command. We demonstrated its effectiveness at following directions to unknown objects and following natural language route directions.

7.2 Future Work

One key challenge to learning from natural language descriptions is that they convey human-level concepts, which are often difficult to fuse with observations made from robot sensors, such as lidars and cameras. For example, we rely on spectral clustering to segment the environment into spatially coherent regions in the world. However, a human's model of space is often hierarchical. For example, several rooms could be called "offices" but they could collectively be called a "lab." Our representation, even accounting for accurate segmentation, is still represents the world as a flat spatial structure, which can result in language being incorrectly associated with the robot's spatial model (e.g., only part of the map being labeled "lab"). Learning a hierarchical spatial model would allow for the ability to better integrate language, but would require reasoning about different possible hierarchies.

The semantic properties that we learned from natural language are labels that could be used to describe these locations. Because we assumed a fixed set of possible labels for these regions, the algorithm only learns a distribution over this known set of labels. Approaches

that can extend the set of labels during operation would prove useful. Additionally, the ability to perceive additional social cues provided by users, such as gaze and pointing gestures, can allow for better integration of natural language. For example, learning from a description "the kitchen is down that hallway" accompanied by a pointing gesture towards the "hallway" is less ambiguous than only learning from the description. In addition, information that we are currently able to infer from a user's descriptions is limited to a region's colloquial name and its relation to another region in the environment. Our method does not support a user's ability to convey general properties of the environment, such as "You can find computers in offices," or "nurses' stations tend to be located near elevator lobbies." Learning from such expressions can allow the robot to model a prior over the world.

Additionally, our approaches only relied on language and appearance models to infer semantic properties. The ability to perceive salient objects and to reason about their relationship with region types and labels is a strong source of semantic information. Models that include object detections in addition to language would allow the robot to learn more useful spatial-semantic representations. Another source of useful semantic information is signage (textual and symbolic) present throughout human environments.

Our approach outlined in Chapter 5 focused on asking questions of the user to improve the learned representation. It considered questions about spatial entities described by the user and reasoned over the ambiguity of the statements based on the current representation. Because the robot is only able to reason over the ambiguity given a partial map of the world, the robot can fail to ask useful questions. Calculating ambiguity that takes unvisited regions into account would improve this process. We also used several simple features to model the cost of a question-asking action. A more principled modeling of cost together with training and validation through human dialog experiments would result in a more meaningful cost metric. This approach only considers questions that reduce the entropy over language groundings. Allowing the robot to ask questions even in the absence of language descriptions based on semantic information it observes using onboard sensors would allow the robot more opportunities to improve its learned model. For example, upon observing a computer monitor, it could ask whether it is in an office.

Increasing the space of questions would also prove beneficial. Our approach was lim-

ited to questions that provide a yes or no answer, which limit the potential information content in the answer. A trivial extension would be to ask questions that provide several known options (e.g., "Is the kitchen on my right or my left?"), which would prove more useful. Open ended questions would be another avenue but would require a different mechanism to reason about the best question to ask, as our current method requires an *a priori* distribution over the answers. Another extension would be for the robot to take physical exploration actions (by searching for a location that the user referenced) or by asking the user to visit the location.

Our approach in Chapter 6 demonstrated work that enables a robot to understand and follow natural language navigation instructions in unknown environments by using our semantic mapping algorithm to reason over unobserved parts of the environment. We used natural language to directly extend the robot's representation by reasoning over the presence of and spatial relationships between regions and objects. We made simplifying assumptions about the robot's ability to perceive semantic properties of the environment from its own sensors with the use of AprilTag fiducials. Approaches that exploit vision-based object detection and scene classification would remove this assumption. We also made assumptions about the likelihood of encountering new spatial entities in environments when we used the Dirichlet process prior. Learning this from training data would provide for more robust direction following.

Our approach only considered executing the inferred behaviors, and did not explicitly reason over information gathering actions. Any new observations that allowed for the robot to improve its representation was incidental to following the instructions. Explicit reasoning over exploratory behaviors that take information gathering actions to resolve uncertainty in the map would provide better performance.

A key component of our approach in Chapter 6 was the inference of weak metric information based on spatial relations. In sampling these metric constraints from spatial relations, we made assumptions about the scale of the environment (maximum area from which to sample potential locations), which might reduce the generality of the approach to environments of different scale.

This approach assumed that the human provides an initial natural language instruction

and allows the robot to carry out the instruction without further assistance. If we allow the robot to learn from additional instructions from the user, especially about the validity of the current action, it would lead to more effective behaviors. For example, if a robot responding to the command "go to the hydrant behind the cone" in an environment with multiple cones starts to go behind the wrong hydrant, the user could indicate that this is the wrong action. This would require the algorithm to reason over the validity of its current action as implied by the human and how this is impacted by its current belief. Treating this as an additional observation about the environment would allow it to modify the map such that it reduces the invalid hypotheses about the world that lead to the current invalid action. Such reasoning would require tight coupling between the mapping and policy inference, which does not exist in the current approach.

In conclusion, we believe that using natural language to inform the robot about its environment will result in robots that are better able to interact with human partners. This thesis has outlined several approaches that take robots towards this goal. Better spatial representations and better natural language understanding capabilities, coupled with richer semantic perception capabilities, will result in more accurate and useful ways for robots to learn about their environments.

Bibliography

[1] Aethon TUG robot. URL `http://www.aethon.com/`.

[2] Philipp Althaus and Henrik I. Christensen. Automatic map acquisition for navigation in domestic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[3] P. Beeson, J. Modayil, and B. Kuipers. Factoring the mapping problem: Mobile robot map-building in the Hybrid Spatial Semantic Hierarchy. *International Journal of Robotics Research*, 29(4):428–459, April 2010.

[4] J.-L. Blanco, J. Gonzalez, and J.A. Fernandez-Madrigal. Consistent observation grouping for generating metric-topological maps that improves robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[5] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research*, 23(12):1113–1139, 2004.

[6] E. Brunskill, T. Kollar, and N. Roy. Topological mapping using spectral clustering and classification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3491–3496, 2007.

[7] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[8] David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 859–865, 2011.

[9] Howie Choset and Keiji Nagatani. Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17:125–137, 2001.

[10] Koby Crammer and Yoram Singer. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research*, 2: 265–292, 2002.

[11] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6):647–665, 2008.

[12] Robin Deits, Stefanie Tellex, Pratiksha Thaker, Dimitar Simeonov, Thomas Kollar, and Nicholas Roy. Clarifying commands with information-theoretic human-robot dialog. *Journal of Human-Robot Interaction*, 2(2):58–79, 2013.

[13] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229 –241, June 2001. doi: 10.1109/70.938381.

[14] F. Doshi and N. Roy. Efficient model learning for dialog management. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 65–72, March 2007.

[15] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 176–183, 2000.

[16] H. Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics Automation Magazine, IEEE*, 13(2):99–110, June 2006. doi: 10.1109/MRA.2006.1638022.

[17] F. Duvallet, Matthew R. Walter, T. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz. Inferring maps and behaviors from natural language instructions. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, Marrakech/Essaouira, Morocco, June 2014.

[18] Felix Duvallet, Thomas Kollar, and Anthony Stentz. Imitation learning for natural language direction following through unknown environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[19] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn. What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4163–4168, 2009.

[20] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, June 1987.

[21] M. R. Endsley, S. J. Selcon, T. D. Hardiman, and Croft D. G. A comparative analysis of sagat and sart for evaluations of situation awareness. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 42(1):82–86, October 1998.

[22] Nathaniel Fairfield and David Wettergreen. Active slam and loop prediction with the segmented map using simplified models. In *Field and Service Robotics*, volume 62 of *Springer Tracts in Advanced Robotics*, pages 173–182. Springer Berlin Heidelberg, 2010.

[23] P. Fearnhead and P. Clifford. Online inference for hidden markov models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–889, November 2003.

[24] C. Galindo, A. Saffiotti, S. Coradeschi, and P. Buschka. Multi-hierarchical semantic maps for mobile robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3492–3497, 2005.

[25] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1999.

[26] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[27] S. Hemachandra, Matthew R. Walter, S. Tellex, and S. Teller. Learning spatial-semantic representations from natural language descriptions and scene classifications. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2014.

[28] S. Hemachandra, F. Duvallet, T. Howard, N. Roy, A. Stentz, and Matthew R. Walter. Learning models for following natural language directions in unknown environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015 (Submitted, under review).

[29] Sachithra Hemachandra and Matthew R Walter. Information theoretic question asking to improve spatial semantic representations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015 (Submitted, under review).

[30] Sachithra Hemachandra, Thomas Kollar, Nicholas Roy, and Seth Teller. Following and interpreting narrated guided tours. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2574–2579, 2011.

[31] Sachithra Hemachandra, Matthew R. Walter, Stefanie Tellex, and Seth Teller. Learning semantic maps from natural language descriptions, 2013. URL `http://vimeo.com/67438012`.

[32] Sachithra Hemachandra, Matthew R Walter, and Seth Teller. Information theoretic question asking to improve spatial semantic representations. In *AAAI Fall Symposium Series*, 2014.

[33] Thomas M. Howard, I. Chung, O. Propp, Matthew R. Walter, and N. Roy. Efficient natural language interfaces for assistive robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop on Rehabilitation and Assistive Robotics*, 2014.

[34] T.M. Howard, S. Tellex, and N. Roy. A natural language planner interface for mobile manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[35] Intouch Health RP-Vita. URL `http://www.intouchhealth.com/`.

[36] iRobot Ava 500. URL `http://www.irobot.com`.

[37] R. Jackendoff. *Semantics and Cognition*. The MIT Press, September 1985.

[38] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *Transactions on Robotics*, 24(6):1365–1378, 2008.

[39] Knightscope K5 robot. URL `http://knightscope.com/`.

[40] Thomas Kollar and Nick Roy. Utilizing object-object and object-scene context when planning to find things. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4116–4121, 2009.

[41] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 259–266, 2010.

[42] B. Krieg-Brückher, U. Frese, K. Lüttich, C. Mandel, T. Massakowski, and Robert J. Ross. Specification of an ontology for route graphs. *Spatial Cognition IV: Reasoning, Action, Interaction*, 3343:390–412, 2005.

[43] Geert-Jan M. Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik I. Christensen. Clarification dialogues in human-augmented mapping. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Salt Lake City, UT, 2006.

[44] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119(1):191–233, 2000.

[45] B. Kuipers, J. Modayil, P. Beeson, and M. MacMahon. Local metrical and global topological maps in the Hybrid Spatial Semantic Hierarchy. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4845–4851, April 2004.

[46] J. Leonard and P. Newman. Consistent, convergent, and constant-time SLAM. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1143–1150, Acapulco, Mexico, August 2003.

[47] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, volume 3, pages 1442–1447, Nov 1991.

[48] Michael L Littman, Anthony R Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1995.

[49] J.S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6:113–119, 1996.

[50] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[51] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[52] K. Lynch. *The Image of the City*. MIT Press, 1960.

[53] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1475–1482, 2006.

[54] Alexei A. Makarenko, Stefan B. Williams, Frederic Bourgault, and Hugh F. Durrant-Whyte. An experiment in integrated exploration. In *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS*, pages 534–539, 2002.

[55] O. Martínez Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, 55(5):391–402, May 2007.

[56] C. Matuszek, N. FitzGerald, L. Zettlemoyer, L. Bo, and D. Fox. A joint model of language and perception for grounded attribute learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

[57] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2012.

[58] Cynthia Matuszek, Dieter Fox, and Karl Koscher. Following directions using statistical machine translation. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 251–258, 2010.

[59] D. Meger, Per-Erik Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, James J. Little, and David G. Lowe. Curious George: An attentive semantic robot. *Robotics and Autonomous Systems*, 56(6):503–511, June 2008.

[60] J. Modayil, P. Beeson, and B. Kuipers. Using the topological skeleton for scalable global metrical map-building. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1530–1536, September 2004.

[61] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 593–598, 2002.

[62] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1151–1156, 2003.

[63] Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, Aug. 2010.

[64] O.M. Mozos, C. Stachniss, and W. Burgard. Supervised learning of places from range data using adaboost. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[65] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[66] Edwin Olson, John Leonard, and Seth Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *icra*, pages 2262–2269, 2006.

[67] Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Simultaneous localization and grasping as a belief space control problem. In *Proceedings of the International Symposium of Robotics Research (ISRR)*, 2011.

[68] A. Pronobis, O. Martínez Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *International Journal of Robotics Research*, 2010.

[69] Andrzej Pronobis and Patric Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3515–3522, 2012.

[70] Andrzej Pronobis, Oscar Martinez Mozos, Barbara Caputo, and Patric Jensfelt. Multi-modal semantic place classification. *International Journal of Robotics Research (IJRR)*, 29(2–3):298–320, February-March 2010. ISSN 0278-3649.

[71] A. Ranganathan. Pliss: Detecting and labeling places using online change-point detection. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[72] A. Ranganathan and F. Dellaert. Bayesian suprise and landmark detection. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2017–2023, May 2009.

[73] A. Ranganathan and F. Dellaert. Online probabilistic topological mapping. *International Journal of Robotics Research*, 30(6):755–771, 2011.

[74] Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Maximum Margin Planning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2006.

[75] Rethink Robotics Baxter. URL `http://www.rethinkrobotics.com/`.

[76] S. Rosenthal, A.K. Dey, and M. Veloso. How robots' questions affect the accuracy of the human responses. In *IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, pages 1137–1142, Sept 2009.

[77] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

[78] Nicholas Roy, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[79] Stuart J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*, pages 97–104. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 2003.

[80] Savioke SaviOne robot. URL `http://www.savioke.com/`.

[81] Stephen Se, David G Lowe, and James J Little. Vision-based global localization and mapping for mobile robots. *Transactions on Robotics*, 21(3):364–375, 2005.

[82] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial language for human-robot dialogs. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(2):154–167, 2004. ISSN 1094-6977.

[83] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[84] Alex Smola, Arthur Gretton, Le Song, and Bernhard SchÃűlkopf. A Hilbert space embedding for distributions. In *International Conference In Algorithmic Learning Theory*, 2007.

[85] E.J. Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University, 1971.

[86] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, 2005.

[87] Suitable Technologies BeamPro. URL `https://www.suitabletech.com/`.

[88] S. Tellex, T. Kollar, S. Dickerson, Matthew R. Walter, Ashis G. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1507–1514, 2011.

[89] Stefanie Tellex, Pratiksha Thaker, Robin Deits, Thomas Kollar, and Nicholas Roy. Toward information theoretic human-robot dialog. In *Proceedings of Robotics: Science and Systems (RSS)*, 2012.

[90] S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *International Journal on Robotics Research*, 25(5/6):403–430, 2005.

[91] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and Benjamin J. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 989–995, Madison, WI, July 1998.

[92] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, D. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and Whittaker W. Autonomous exploration and mapping of abandoned mines. *Robotics Automation Magazine, IEEE*, 11(4):79–91, Dec 2004.

[93] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

[94] Nicola Tomatis and Illah Nourbakhsh. Hybrid simultaneous localization and map building: closing the loop with multi-hypothesis tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[95] A. Torralba, Kevin P. Murphy, William T. Freeman, and Mark A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 273–280, Nice, France, October 2003.

[96] S. Vasudevan and R. Siegwart. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, 56 (6):522–537, June 2008.

[97] Vecna QC Bot. URL `http://www.vecna.com/`.

[98] Matthew R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller. Learning semantic maps from natural language descriptions. In *Proceedings of Robotics: Science and Systems (RSS)*, Berlin, Germany, June 2013.

[99] Matthew R. Walter, Sachithra Hemachandra, Bianca Homberg, Stefanie Tellex, and Seth Teller. A framework for learning semantic maps from grounded natural language descriptions. *International Journal of Robotics Research*, 33(9):1167–1190, 2014.

[100] Tom Williams, Rehj Cantrell, Gordon Briggs, Paul Schermerhorn, and Matthias Scheutz. Grounding natural language references to unvisited and hypothetical locations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2013.

[101] B. Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 146–151, July 1997.

[102] H. Zender, O. Martínez Mozos, P. Jensfelt, G.J.M. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.