

PRACTICAL QUALITATIVE RELATIONAL MAPPING AND NAVIGATION FOR ROBOTS

A Dissertation

Presented to the Faculty of the Graduate School
of Cornell University

in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Jennifer J Padgett

August 2017

© 2017 Jennifer J Padgett
ALL RIGHTS RESERVED

PRACTICAL QUALITATIVE RELATIONAL MAPPING AND NAVIGATION
FOR ROBOTS

Jennifer J Padgett, Ph.D.

Cornell University 2017

Qualitative relational maps, which represent space as a set of coarse, relative relationships between landmarks are useful abstractions of space for robots. While qualitative relational maps have been used in several robotic scenarios, the methods for qualitative relational mapping to date are not always practical for modern robotic systems because they cannot incorporate probabilistic information, cannot adapt qualitative spatial representations to low-level data from an environment, and do not have methods for navigation that generalize to the set of all qualitative relational maps. Three approaches that make qualitative relational maps practical for modern robots are presented: 1) the probabilistic qualitative relational mapping algorithm, which adapts traditional qualitative relational maps for the incorporation of uncertain information, 2) an algorithm for adapting a qualitative spatial relationship to best represent underlying data at desired resolution, and 3) a general method for navigating with any type of qualitative relational information. Each method is tested in simulation studies and in vivo experiments to determine performance in a variety of scenarios with different data.

BIOGRAPHICAL SKETCH

Jennifer Padgett completed her undergraduate degree at Rutgers University (BS EE, 2012) and graduated with highest honors. Her focuses in her undergraduate degree included signal processing and communications. She has conducted research on the smart grid, sparse sampling, secrecy capacity of networks, automated CT scan evaluation, and qualitative relational mapping for robotics. Jennifer has been honored in her graduate career to receive an NSF graduate research fellowship award. Her current research focuses on qualitative mapping and long-term learning in robots.

This thesis is dedicated to my wonderful parents, Jay and Joyce Padgett, and to my partner in everything, Nitish Thatte.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the generous support and help provided by my committee. A special thank you to Professor Mark Campbell for investing time in my research and always pushing me to do my best. Thank you to Professor Hadas Kress-Gazit for teaching me the fundamentals of robotics and teaching interesting classes that I have had the privilege to attend. Finally, thank you to Professor Noah Snavely for his unique insights and comments that inspired one chapter of this thesis. Thank you to all for investing time in me and my work.

I would also like to thank all of the people who have supported me through my PhD and had unwavering faith in me. Thank you to my parents, Jay and Joyce Padgett, and my partner, Nitish Thatte, for their unconditional love and support. Another special thank you to my sister in all but name, Sherry Padden, for always reminding me to keep things in perspective. I could not have completed this without the support of any of them.

I would also like to thank all of the PhD students in the Autonomous Systems Laboratory for all of their feedback and support, with a special thank you to Lucas De La Garza, Dr. Daniel Lee, Dr. Rina Tse, and Dr. Kevin Wyffels for many interesting research discussions.

Finally, a special thank you to all of the people who have mentored me along the way and fostered an interest in research: Dr. Christopher Chambers, Professor Sennur Ulukus, Professor Richard Mammone, and Professor Christopher Rose.

Funding agencies: NSF Graduate Research Fellowship Program

TABLE OF CONTENTS

Biographical Sketch	iii
Dedication	iv
Acknowledgements	v
Table of Contents	vi
List of Tables	viii
List of Figures	ix
1 Introduction	1
2 Probabilistic Qualitative Relational Mapping	3
2.1 Introduction	3
2.1.1 Related Work: SLAM and Topological Mapping	4
2.1.2 Qualitative Maps	6
2.2 The Probabilistic Qualitative Relational Mapping Algorithm	9
2.2.1 Constraint-Based Qualitative Maps	9
2.2.2 Computing Probabilities of Qualitative States	12
2.2.3 The Online PQRM Algorithm	16
2.3 Experiments and Results	18
2.3.1 Validation Via Monte Carlo Studies	20
2.3.2 PQRM Evaluation on the New College Dataset	31
2.4 Conclusion	46
3 Automated generation of discrete salient regions from probabilistic environmental information	48
3.1 Introduction	48
3.2 Problem Definition	50
3.2.1 Map Saliency	50
3.2.2 Representations	51
3.2.3 The General Problem of Finding Salient Maps	53
3.3 The SERGE Algorithm	55
3.3.1 Resolution Scaling	57
3.3.2 Using SR maps for Qualitative Reasoning	60
3.4 Experiments and Results	63
3.4.1 The Data	63
3.4.2 Evaluation of SR maps	69
3.4.3 A Use Case for the Multi-Resolution SR Maps	74
3.5 Conclusion	78
4 Q-Link: A General Planning Architecture for Navigation with Qualitative Relational Information	80
4.1 Introduction	80
4.2 Qualitative Spatial Relationships and Qualitative Relational Maps	83

4.3	The Q-Link Planning Architecture	86
4.3.1	Qualitative Links	87
4.3.2	The Q-Link Planning Architecture	88
4.3.3	Incorporation of Probabilistic Information	93
4.3.4	Path Completion Guarantees	96
4.4	Simulation and Experimental Result and Analysis	100
4.4.1	Simulation Studies	102
4.4.2	In Vivo Experiments	106
4.5	Conclusion	113
5	Conclusion and Contributions	114
	Bibliography	118

LIST OF TABLES

4.1	Failure rates for the different simulation scenarios. The parameters for the sensor range trial is the robot sensor range, and the parameter for the occlusion trial is the obstacle size.	105
4.2	Average trajectory lengths in the in vivo study reported as a function of Link Graph path length, $ P $	112

LIST OF FIGURES

2.1	The double cross, a ternary calculus, with 100 simulated measurements of two different landmarks. As seen, when the landmark is far away from the separating constraints (red) it can be well classified as residing in one of the qualitative regions. However, when the landmark approaches the boundaries of the qualitative regions (green) it is not clear in which qualitative region the landmark truly is.	7
2.2	Examples of qualitative relationships that may be used with the PQRM.	10
2.3	Factor graph representation of the distribution for landmark set, where N is the number of landmark sets mapped, c_a represents the value of one of the the k constraint equations for set of landmarks \mathcal{L}^n . \mathcal{Z}^n are the measurements of \mathcal{L}^n and \mathcal{R}^n is the qualitative relationships of the landmark set.	13
2.4	Overview of the PQRM process. The robot gathers measurements from the environment and uses these measurements to update the probability distributions over landmark sets. After all distributions are updated and new sets are added to the map, the robot moves and collects new measurements. The update process follows (2.10).	17
2.5	An EDC with labeled qualitative states and the graph used to compute the neighbor distance for EDC qualitative states.	22
2.6	Mean neighbor distance of the PQRM maps for varying levels of sensor noise. Note that the worst case standard deviation from the mean is a neighbor distance of 0.14.	23
2.7	Mean entropy of the PQRM maps for the Monte Carlo simulations over varying levels of sensor noise. The blue squares represent the maximum possible mean entropy (3.00 nats) for the simulations. The maximum standard deviation from the mean is 0.180 nats.	24
2.8	Triplets estimated incorrectly on the EDC for two different values of σ_N	25
2.9	Confusion matrix for all errors made in the 1000 Monte Carlo trials conducted for the EDC qualitative representation.	27
2.10	Absolute and normalized mean map entropy for Monte Carlo trials computed for the EDC and five reduced representations shown in Figure 2.2.1.	28
2.11	Mean map error rate and probability of maximum likelihood state for six qualitative representations reported as a function of σ_N	30
2.12	Bounding boxes and corresponding centroids computed for three landmarks in the first 35sec of the New College dataset. . .	33

2.13	Evolution of select distributions $p(\mathcal{R} \mathcal{Z})$ for triplets in the New College dataset with time.	36
2.14	Histograms of the entropy for the PQRM maps generated with the three types of measurement processing.	39
2.15	Histograms of the neighbor distance for the PQRM maps generated with three types of measurement processing.	41
2.16	Error rate for all PQRM representations tested. The error rate is reported a function of the number of most likely states included in the PQRM output for the Stereo method, Sampled Stereo method, and Monocular method.	44
3.1	A graphical model showing the relationship between each of the variables in the salient region mapping problem. The shaded nodes represent observed variables, while the remaining nodes represent uncertain variables. This model shows three cliques that are used to factor the potential function in equation (3.4). . .	53
3.2	A graphical model showing the relationship between distributions in two levels of a resolution tree, $l, l - 1$. Two cliques (indicated by the connections shown in red and green) relate the two levels: $\{B^{l-1}, B_i^l, \mathcal{L}^{B,l-1}\}$ and $\{C^{l-1}, C_i^l, \mathcal{L}^{C,l-1}\}$	58
3.3	An example a multi resolution SR map. SRs are shown by the different colors. As shown on the left, at the top level the SR map has one SR; the second layer splits the SR. The third layer of the SR map is formed by splitting each parent SR into two additional SRs. The result is an SR map (at the finest resolution) with four SRs. The resulting full map at each layer of the tree is shown on the right.	60
3.4	A well separated dataset where measurements from the different landmarks are shown in unique colors. The beliefs representing map landmarks have small variance relative to their spacing. . .	64
3.5	Data from a hallway simulation.	65
3.6	The lab data acquired with a Kinect sensor. An example image (a) and a plot of the data (b) from an overhead view are shown.	67
3.7	Results from the 5-fold cross validation study on all datasets. . .	70
3.8	Example SR maps for each dataset at two different parameter values. The unique colors represent the different qualitative regions. The black points are the data points (the entire dataset). .	72
3.9	An example of a navigation scenario using a SR map and a resolution tree.	75
3.10	An example of the compose operator on the lab dataset. The possible relationships for $\{DE : G\}$ are shown given that $\{DE : F\} = 4$ and $\{EF : G\} = 14$	77

4.1	Two different QSR representations, the double cross [53] shown by the black solid lines, and a star QSR [27] shown by the gray solid lines. Qualitative states, q , are the regions of space separated by the lines. The blue landmarks are non-oriented point landmarks, and the red landmarks are oriented point landmarks. The positions of qualitative states are determined by \mathcal{L}^{ref} . For the double cross $\mathcal{L}^{\text{ref}} = \{L_A, L_B\}$ and for the star calculus $\mathcal{L}^{\text{ref}} = \{L_C\}$ in this example. A robot is shown by the green rectangle at three time instances with poses $x_0^{L_A, L_B}$, $x_1^{L_A, L_B}$, and $x_2^{L_C}$. The trajectory traveled is shown by the gray dashed line.	83
4.2	The Q-Link architecture, with three levels: the Link Graph level generates plans over a QRM, the QSR level reasons over qualitative states, and a Trajectory level generates low level controls, u , given commands, U , from the QSR level.	89
4.3	An example of the two scenarios for which path completion is guaranteed. The robot is represented by the black rectangle, and its field of view is represented by the dashed green line. The landmarks for two scenarios are presented (yellow and blue) and are all contained in a circle with radius R (purple dotted line). Landmarks for the general scenario are shown in blue, and landmarks for the more restricted scenario are shown in yellow. . . .	96
4.4	A 200m \times 200m test environment for Q-Link. Landmarks are indicated by the blue circles.	101
4.5	Two QSR representations defined with non-oriented point landmarks that are used for testing. The first, represented by both the solid and dashed lines, partitions space into quadrants. The second, represented by the solid line only, partitions space into left and right halves.	101
4.6	A histogram showing the distribution of Link Graph path lengths for the Monte Carlo simulations.	103
4.7	Average trajectory length (m) reported as a function of Link Graph path length for the occlusion and sensor range trials. . . .	104
4.8	Setup for the in vivo navigation trials.	107
4.9	Navigation trajectories in the in vivo study. The solid lines represent successful paths, and the dashed lines represent paths that did not reach their goal.	110

CHAPTER 1

INTRODUCTION

Qualitative relational maps (QRM), which are maps made of Qualitative Spatial Relationships (QSR), have a long history of use in robotic applications. Starting in the early 1990s with work by Levitt and Lawton [28] and ranging through work in 2016 by McClelland [33], several researchers have shown the utility of qualitative relationships in robotics. The diversity of robotics applications in which QSR and QRM are used cover indirect object search [22], learning by observation [52], mapping space with point objects [33], mapping space with extended objects [7], and in home human-aid robotics [44]. While QSR and QRM have several uses in robotics, to date approaches that use QSR and QRM do not support probabilistic sensor readings, do not adapt QSR representations to better fit incoming data, and do not use general navigation methods. Instead perfect sensing is assumed and a QSR representation is hand-picked by an engineer when QRM are created for use in an environment. Furthermore, navigation methods used with QRM are either specific to one type of QSR or rely on an additional map.

This work presents general, adaptable methods that make QSR practical for in use in robotics. Chapter 2 addresses the incorporation of probabilistic information into a QRM framework. The algorithms developed are general, and can be used with any type of map created from QSR. Studies detail the mapping performance of probabilistic QRM (PQRM) in simulated and experimental settings. Chapter 3 addresses the problem of generating QSR automatically to better fit incoming data from a robot. Given beliefs about several objects in an environment, a robot uses the SaliEnt Regeion GEneration (SERGE) algorithm

to generate a QSR at desired resolution and maximize a quantity called saliency. Outputs of the SERGE algorithm have uses beyond QSR to generate finite representations of space that are useful for navigation. Simulation and experimental studies show that the SERGE algorithm is able to generate finite representations of space with high saliency and desired resolution. Furthermore, compose operators are computed from the outputs of the SERGE algorithms for use with QSR. Chapter 4 addresses the general problem of navigation with qualitative relational information. A three level architecture, called qualitative linking (Q-Link), is introduced that allows a robot to reason over information in a QRM at a high level and generate low-level controls to reach a goal. Q-Link is general and can be used with any type of qualitative map, including PQRMs and QRMs with hybrid QSR representations. The performance of Q-Link is demonstrated in several scenarios in simulation and in an in vivo experiment.

CHAPTER 2

PROBABILISTIC QUALITATIVE RELATIONAL MAPPING

2.1 Introduction

Long-term autonomous robot missions require that robots are equipped to map potentially large, unstructured spaces. Resulting maps must yield information from which the robot can reason about the environment, build low-level mission plans, and localize itself in the environment. Such maps must be robust to sensor noise and divergence that can result from poor sensor data processing, data association mistakes, and incorrect loop closures (viewing a past scene again). In this work, the Probabilistic Qualitative Relational Map (PQRM) is formulated as an approach to qualitative relational maps, based on soft metrical information, in the presence of sensor noise. The benefits of the proposed approach include robustness to extreme sensor noise, map updates that are linear with the number of observations, a mapping process that does not require localization of the robot within a world coordinate frame, and a representation that is a complement to pure metrical map representations.

Popular approaches to mapping in robotics include Simultaneous Localization and Mapping (SLAM) and topological mapping. While SLAM algorithms produce a metrical representation of an environment, this rich and precise representation often comes at the cost of significant computational resources. In contrast, topological mapping algorithms produce a graph-like map representing the structure of the environment, but with little or no metrical information. While topological maps can be inexpensive to build in comparison to SLAM maps, the lack of metrical information prevents the robot from deriving control

laws to drive from place to place. An alternative mapping strategy, qualitative relational mapping, inexpensively produces a relational map of an environment that inherently preserves metrical information about the environment. Qualitative relational maps are very flexible and can scale well, as shown by the varied representations and examples in [31, 32, 33, 7, 28, 44]. Work to date in qualitative relational mapping assumes that the robot’s sensors return noiseless measurements. The goal of this paper is to develop a general probabilistic QRM which allows the use of imperfect sensor measurements and utilizes a representation to enable recursive updates over time.

2.1.1 Related Work: SLAM and Topological Mapping

SLAM algorithms simultaneously estimate the locations of environmental features and the robot’s trajectory through the environment in a fixed coordinate frame [48]. Features are typically landmarks that are explicitly extracted from incoming data or grid cells that indicate occupancy of a volume of space. SLAM is either implemented incrementally as a filter (EKF SLAM) [48] or in a batch optimization (Graph SLAM) [48]. EKF SLAM, while computationally efficient, has been proven to diverge [16]. The issue of divergence in the filtering paradigm has been addressed by taking sampling-based approaches to SLAM. However, the sampling based approaches (typically particle filters) suffer several drawbacks including slow run time with a large number of samples, particle deprivation, and solutions that may not be intuitive to interpret [48]. The batch solution often produces consistent maps but is expensive to implement due to the linearization and iterative optimization over a large number of variables. Considerable work in recent years has focused on efficient solutions for real time

implementation of the batch solution via incremental smoothing [17]. While incremental smoothing makes the batch solution efficient, it can still diverge if data association errors are made upon loop closure. This issue has been addressed in the back-end of SLAM solutions by adding correspondence variables, but at the expense of problem size and computation [46].

The other popular approach to mapping is topological mapping. Unlike SLAM, topological approaches do not yield landmark positions in a fixed coordinate frame, but a graph that represents the structure and connectivity of landmarks or places in an environment [43]. Due to the abstract nature of topological maps, no unified framework exists to solve the topological mapping problem [42], thus several topological mapping algorithms exist. Some approaches, such as FABMap [6], make hard decisions about locations in the topological map and map connectivity at run time. However, approaches that rely on hard decisions typically diverge if an incorrect loop closure is made [42]. Other approaches avoid making hard decisions by maintaining all possible map hypotheses as the robot makes observations. The probability of each hypothesis is tracked in the Probabilistic Topological Mapping approach [42] and used to determine likely maps. An alternative approach by Wallgrun tracks qualitative information about hallway junctions and prunes map hypotheses based on the consistency of that information [51]. However, even if the topological maps are consistent, they lack the metrical information that is necessary for low-level planning and control [43]. One solution that allows for some low level planning over topological maps is the hybrid topological-metric map [20, 14]. These hybrid maps contain local metric maps at each node and transformations that map between coordinate frames along edges, allowing for local low-level planning. Planning is done over the topological map at a high level, and the local maps

are used for low-level planning that allows for traversal between nodes.

2.1.2 Qualitative Maps

QRM produces a map that contains discretized relative spatial information about sets of landmarks in an environment. Unlike the output of SLAM, landmarks positions are not defined in a fixed coordinate frame, thus estimation of the robot trajectory through the environment is unnecessary. QRMs can be constructed without expensive linearizations and optimizations. Additionally, QRMs contain relative spatial information about landmarks that can be used for low-level planning. Several instances of qualitative map have been developed for robots, which can be roughly partitioned into two types: those based on local spatial constraints, and those based on global visual orderings. While the two are closely related, a distinction is drawn based on the navigation capabilities for robots demonstrated with each type of calculus to date. Constraint based maps include maps developed by Rost [44] and McClelland [31, 32, 33]. The constraint based maps represent relationships between a constant number of triplets in local coordinate frames. Since these maps contain rich relative spatial information, low-level control laws may be derived from the spatial information. Examples of QRMs based on global visual orderings include the visual cyclical ordering maps of Levitt [28] and Fogliaroni [7]. These maps represent relationships as the clockwise or counter-clockwise ordering of a variable number of landmarks in a robot's field of view. Most ordering-based maps rely on topological maps to enable robot navigation between the different qualitative states.

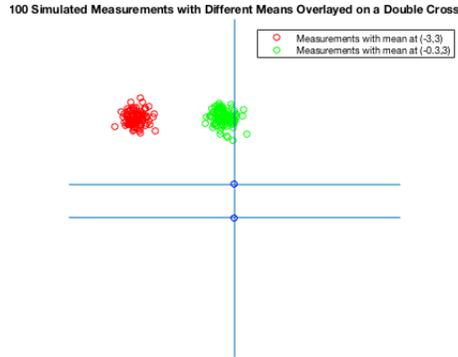


Figure 2.1: The double cross, a ternary calculus, with 100 simulated measurements of two different landmarks. As seen, when the landmark is far away from the separating constraints (red) it can be well classified as residing in one of the qualitative regions. However, when the landmark approaches the boundaries of the qualitative regions (green) it is not clear in which qualitative region the landmark truly is.

Qualitative methods to date avoid consideration of a probabilistic sensor model by assuming that measurements are consistent given a coarse representation of space. However, this assumption is unrealistic in the general mapping scenario. Consider the case shown in Figure 2.1 where a landmark is localized in one of the regions of a Double Cross [53]. The red markers show measurements taken of a landmark that is located far from any of the partitioning lines between qualitative regions. It is highly likely that the landmark is correctly localized in the top left qualitative state, thus validating the assumption that coarse regions result in perfect qualitative measurements. However, the example measurements shown by the green markers illustrate a highly likely scenario in which a landmark is located close to the partitioning lines. Here the landmark is likely to be located in either the top left, top right or top center (represented by the partitioning line) regions. Cases with multiple likely qualitative regions are unavoidable in the general mapping problem. Therefore, an explicit noise model is

essential for accurate qualitative relational mapping in the general scenario. A probabilistic formulation for general constraint-based QRMs is developed here for a wide range of applications and constraint-based qualitative maps when sensor noise is present.

The PQRM algorithm uses probabilistic inference to determine qualitative states from noisy sensor measurements. Given the measurements, a noise model for the robotic sensor, and a data association to the landmarks, noise is propagated through a geometric model of the qualitative relationships. This noise propagation leads to a probability of the observed landmarks being in one qualitative state of the chosen qualitative representation. A recursive Bayes filter is used to update probabilities of the landmark triplets being in a qualitative state after each measurement. Monte Carlo studies are conducted to determine the behavior of the maps under different sensor noise conditions, and resulting maps are evaluated for convergence and correctness. Additionally, the New College dataset [45] is used to validate map performance on experimental data.

This paper is organized as follows. First, an overview of the QRM is given. Then, the PQRM algorithm is developed as a recursive filter implementation. Finally, the developed maps are evaluated for uncertainty and correctness in Monte Carlo simulations and on data from the New College dataset.

2.2 The Probabilistic Qualitative Relational Mapping Algorithm

2.2.1 Constraint-Based Qualitative Maps

The PQRM makes use of qualitative maps that are constructed as sets of constraints describing spatial relationships to some point-like references. The references can include distinct environmental features such as buildings or mountains from which one extracts a centroid; for ease of discussion these references are referred to as landmarks. The constraints are any set of k algebraic equations, $f_a(\mathcal{L}) = c_a$ with $a \in 1 : k$ and \mathcal{L} being a set of landmarks. A single inequality written with respect to sets of landmarks, $f_a(\mathcal{L}) = c_a < 0$ or $f_a(\mathcal{L}) = c_a > 0$, denotes a spatial region, and a set of inequalities define distinct qualitative states. Typical constraints, $f_a(\mathcal{L})$, used in qualitative maps are equations describing relative landmark distances or signed distances from lines joining subsets of landmarks.

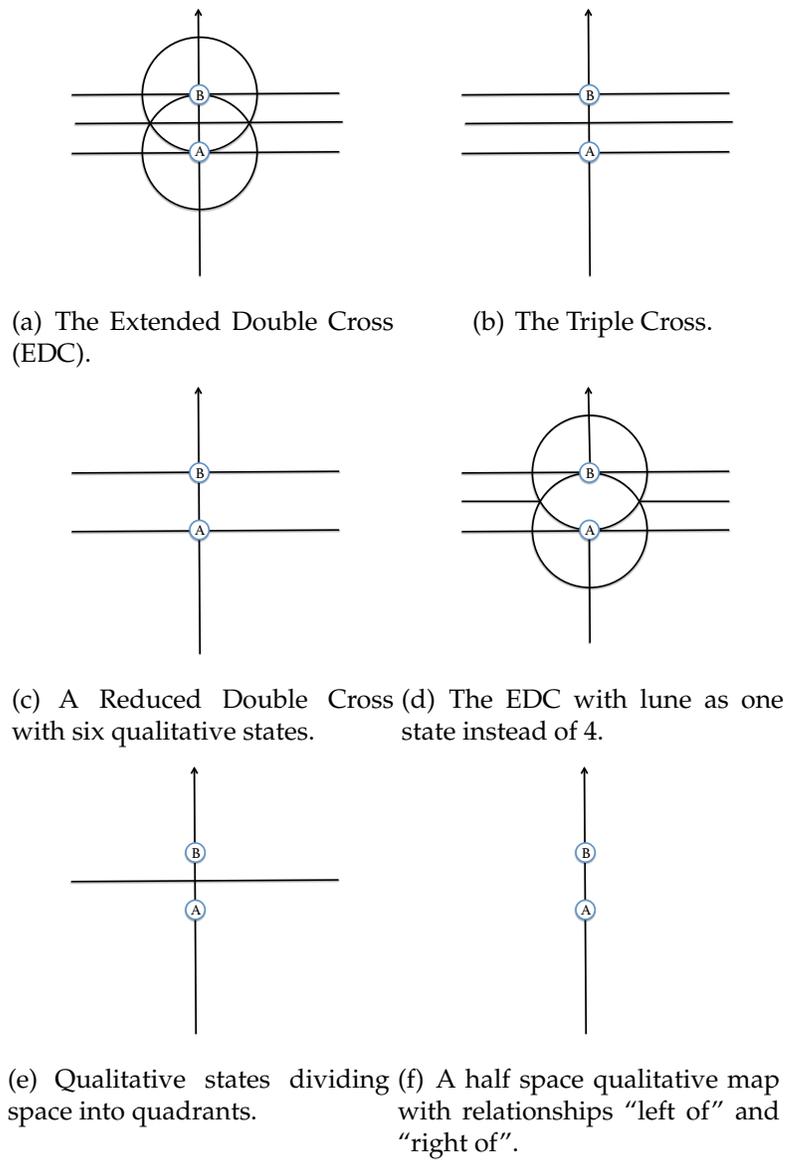


Figure 2.2: Examples of qualitative relationships that may be used with the PQRM.

This paper considers any qualitative map which can be defined using sets of inequality constraints. Some examples of qualitative maps that may be used are shown in Figure 2.2.1. Here, a reduced representation of the well known Double Cross [53] and Extended Double Cross [33] are shown along with variants created by merging qualitative states of these representations. The constraints

in these representations are written with respect to triplets of landmarks. Two landmarks from each triplet are shown, and the qualitative map defines the spatial regions where a third landmark may reside. The lines indicate that the third landmark may be placed such that at least one of the constraints is equal to zero, $c_a = 0$. Lines where $c_a = 0$ are not considered individual qualitative states in Figure 2.2.1, they are merged with adjacent qualitative states. In addition to ternary qualitative relationships shown, binary qualitative relationships in which only two landmarks are involved may also be used provided that the landmarks have an associated orientation. Examples of binary calculi include the OPRA, EPRA, and StarVars calculi [35, 36, 27]. The qualitative representations discussed are for a 2-D environment and triplets or pairs of landmarks; however, qualitative maps may be constructed for higher dimensional environments or different number of landmarks.

The maps developed in this paper do not contain entries for robot pose. However, the qualitative pose of the robot in relation to environmental landmarks at any given time instance can be added into the map. Developing the map exclusively from landmark triplet measurements ensures that the qualitative relationships are viewpoint invariant, and that robot can reason about the environment and relative locations of landmarks after mapping. For the specific task of navigation, the robot can reason at a high level about the position of a goal landmark relative to position of currently observed landmark triplets via the compose operator and intersection of feasible goal regions. The robot can choose a heading to the goal landmark given a set of possible locations. The robot updates its heading and possible goal positions as it gathers more local landmark observations. This is not a complete navigation solution by itself, rather a high level algorithm that provides a feasible region of space in which

the goal can be found. A low-level controller for obstacle avoidance is necessary for this type of navigation since the qualitative map only provides high level information. A safe-search planner can also be used to guarantee that the robot finds the goal landmark within the feasible space.

Any sensor may be used to provide measurements given that there exists a mapping from measurements of landmark sets, \mathcal{Z} , to qualitative states, \mathcal{R} , and appropriate sensor noise characterization. Robots are typically equipped with sensors such as lidar or vision, often producing measurements $\mathcal{Z} \in \mathbb{R}^2$; this case is considered here. However, other sources of information can be incorporated into this framework. For example, soft information such as “left of” or “right of” provided by humans may be used [1].

2.2.2 Computing Probabilities of Qualitative States

The goal of PQRM is to estimate the probability of the relational map, M_{QR} , given sensor data, Z , from the robot, or

$$p(M_{QR}|Z). \quad (2.1)$$

For computational tractability, all qualitative relationships in the map are assumed to be independent. This assumption, as in the case of grid cells in an occupancy map [48], trades a more accurate model of the probability distribution for fast computation. Given this assumption (2.1) can be re-written as

$$p(M_{QR}|Z) = \prod_{n=1:N} p(\mathcal{R}^n|\mathcal{Z}^n) \quad (2.2)$$

where N is the number of landmark sets in the environment that have been mapped, \mathcal{R}^n is the qualitative relationship of the landmarks in the n^{th} set, and \mathcal{Z}^n are all measurements from the robot to the n^{th} set of landmarks.

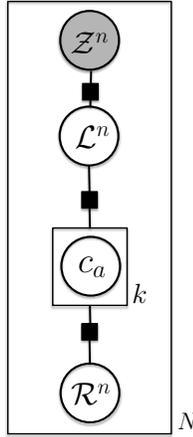


Figure 2.3: Factor graph representation of the distribution for landmark set, where N is the number of landmark sets mapped, c_a represents the value of one of the the k constraint equations for set of landmarks \mathcal{L}^n . \mathcal{Z}^n are the measurements of \mathcal{L}^n and \mathcal{R}^n is the qualitative relationships of the landmark set.

Constraint-based qualitative maps require an intermediate connection in the representation between measurements, constraints, and qualitative states. Typically, the process of inferring a distribution over qualitative states of a landmark set given a set of measurements requires a mapping of random variables from $\mathcal{Z}^n \rightarrow \mathcal{R}^n$, as illustrated by the graphical model in Figure 2.3. Typically the qualitative states, \mathcal{R}^n , are represented by the integers, \mathbb{Z} . The random variables \mathcal{Z}^n are observations of the true but unknown positions in of landmarks in the n^{th} set, \mathcal{L}^n , represented in the local coordinate frame of the robot. The landmark locations spatially define the constraint equations, $f_a(\cdot)$, and their values c_a . The values of the c_a then give the qualitative state $\mathcal{R}^n \in \mathbb{Z}$.

Using the compact representation for the n^{th} detected landmark set and associated qualitative state, the distribution required for mapping, $p(\mathcal{R}^n = q | \mathcal{Z}^n)$,

is written as

$$\begin{aligned}
p(\mathcal{R}^n = q|\mathcal{Z}^n) &= \int_{\tilde{\mathcal{L}}^n} \int_{\tilde{\mathcal{C}}} p(\mathcal{R}^n = q|C = \tilde{\mathcal{C}}) \\
p(C = \tilde{\mathcal{C}}|\mathcal{L}^n = \tilde{\mathcal{L}}^n)p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n)d\tilde{\mathcal{C}}d\tilde{\mathcal{L}}^n
\end{aligned} \tag{2.3}$$

where $C = F(\mathcal{L}^n)$ denotes the values of the k constraint equations that describe the chosen QRM.

In this representation, C and \mathcal{R}^n are functions of random variables since $C = F(\mathcal{L}^n)$ where $F(\cdot)$ is the set of k constraint equations defining the states of the chosen QRM, and $\mathcal{R}^n = g(C)$ where $g(\cdot)$ is a function that maps from the real numbered values of the constraints to the qualitative states. The distribution $p(\mathcal{R}^n = q|C = \tilde{\mathcal{C}})$ defines the probability of a landmark triplet being in qualitative state, $q \in \mathbb{Z}$, given the real numbered values of the constraint equations, $\tilde{\mathcal{C}} \in \mathbb{R}^k$. The distribution $p(\mathcal{R}^n = q|C = \tilde{\mathcal{C}})$ is Bernoulli and evaluates to either 1 or 0 depending on whether q is possible given the values of the constraint equations. The distribution $p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n)$ is the inverse sensor model for the robot.

The distribution $p(C = \tilde{\mathcal{C}}|\mathcal{L}^n = \tilde{\mathcal{L}}^n)$ is a delta function in k dimensions since each of the constraint equations can only evaluate to one real number given the relative positions of landmarks in a set.

$$p(C = \tilde{\mathcal{C}}|\mathcal{L}^n = \tilde{\mathcal{L}}^n) = \prod_{a=1:k} p(c_a = \tilde{c}_a|\mathcal{L}^n). \tag{2.4}$$

where $p(c_a = \tilde{c}_a|\mathcal{L}^n) = \delta(\tilde{c}_a - f_a(\mathcal{L}^n))$. Given the decomposition of the constraint equations in (2.4), the qualitative map distribution in (2.3) can be written as

$$\begin{aligned}
p(\mathcal{R}^n = q|\mathcal{Z}^n) &= \int_{\tilde{\mathcal{L}}^n} \int \cdots \int_{-\infty}^{\infty} p(\mathcal{R}^n = q|C = \tilde{\mathcal{C}}) \\
\prod_{a=1:k} (p(c_a = \tilde{c}_a|\mathcal{L}^n = \tilde{\mathcal{L}}^n)d\tilde{c}_a) p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n)d\tilde{\mathcal{L}}^n.
\end{aligned} \tag{2.5}$$

Rearranging terms, (2.5) is rewritten as

$$\begin{aligned}
p(\mathcal{R}^n = q|\mathcal{Z}^n) &= \int_{\tilde{\mathcal{L}}^n} \prod_{a=1:k} \left(\int_{-\infty}^{\infty} p(c_a = \tilde{c}_a|\mathcal{L}^n) \right) \\
p(\mathcal{R}^n = q|C = \tilde{C}) &= \prod_{a=k:1} (d\tilde{c}_a) p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n) d\tilde{\mathcal{L}}^n
\end{aligned} \tag{2.6}$$

Noting that an integral over each constraint equation can be decomposed into two parts as

$$\begin{aligned}
&\int_{-\infty}^{\infty} p(c_a = \tilde{c}_a|\mathcal{L}^n = \tilde{\mathcal{L}}^n) d\tilde{c}_a \\
&= \int_{-\infty}^{0^-} p(c_a = \tilde{c}_a|\mathcal{L}^n = \tilde{\mathcal{L}}^n) d\tilde{c}_a \\
&+ \int_{0^+}^{\infty} p(c_a = \tilde{c}_a|\mathcal{L}^n = \tilde{\mathcal{L}}^n) d\tilde{c}_a,
\end{aligned} \tag{2.7}$$

equation (2.6) can be written as

$$\begin{aligned}
p(\mathcal{R}^n = q|\mathcal{Z}^n) &= \int_{\tilde{\mathcal{L}}^n} \sum_{C^L, C^U} \int_{C^L}^{C^U} p(C = \tilde{C}|\mathcal{L}^n = \tilde{\mathcal{L}}^n) \\
p(\mathcal{R}^n = q|C = \tilde{C}) &= d\tilde{C} p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n) d\tilde{\mathcal{L}}^n
\end{aligned} \tag{2.8}$$

where C^L and C^U are represent all possible combinations of lower (0^- and $-\infty$) and upper (0^+ and ∞) limits of integration for the constraint equations.

Equation (2.8) can be rewritten as one integral and k summations by noting that $p(\mathcal{R}^n = q|C = \tilde{C})$ evaluates to either 1 or 0, and $p(\mathcal{R}^n = q|C = \tilde{C}) = p(\mathcal{R}^n = q|C = \bar{C})$, where \bar{C} is a Boolean variable indicating whether each constraint equation is greater or less than zero

$$\begin{aligned}
p(\mathcal{R}^n = q|\mathcal{Z}^n) &= \sum_{\{c_1 < 0, c_1 > 0\}} \cdots \sum_{\{c_k < 0, c_k > 0\}} p(\mathcal{R}^n = q|\bar{C}) \\
&\int_{\tilde{\mathcal{L}}^n} p(C|\mathcal{L}^n = \tilde{\mathcal{L}}^n) p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n) d\tilde{\mathcal{L}}^n.
\end{aligned} \tag{2.9}$$

For each qualitative state, the quantity $p(\mathcal{R}^n|\bar{C})$ only evaluates to 1 for one specific combination of the $c_a < 0$ or $c_a > 0$ (note that one of these inequalities will

include equality depending on the state with which the user chooses to merge ($c_a = 0$). Therefore, to evaluate $p(\mathcal{R}^n = q|\mathcal{Z}^n)$ for each qualitative state q , one need only find $p(\mathcal{R}^n = q|\bar{C}) = 1$ and evaluate that portion of the summation. All other additive terms are multiplied by $p(\mathcal{R}^n = q|\bar{C}) = 0$ and hence are irrelevant for evaluation.

The integral $\int_{\tilde{\mathcal{L}}^n} p(C|\mathcal{L}^n = \tilde{\mathcal{L}}^n)p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n)d\tilde{\mathcal{L}}^n$ in (2.9) is intractable to compute exactly. Therefore, a sampling based approach is proposed whereby samples are drawn from $p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n)$. Given the use of point-like landmarks, a Gaussian inverse sensor model is assumed for $p(\mathcal{L}^n = \tilde{\mathcal{L}}^n|\mathcal{Z}^n)$. Alternate measurements and QRM representations may require different inverse sensor models or sampling strategies; the reader is referred to [2] for further information on sampling strategies.

2.2.3 The Online PQRM Algorithm

The robot develops its belief, $p(M_{QR}|Z)$, of the world over time as observations of landmarks are gathered while the robot is exploring the environment. At each time step, t , the robot acquires a measurement Z_t of all landmarks in its field of view. Once measurements are acquired, the robot must then process the measurements into measurements of landmark sets, \mathcal{Z}^n . The landmark sets are evaluated for correspondence to existing sets in the map. The belief, $p(\mathcal{R}^n|\mathcal{Z}^n)$, for sets included in M_{QR} is updated. Landmark sets observed for the first time at time t are added into the map with distribution $p(\mathcal{R}^n|\mathcal{Z}^n)$. Once all sets have been processed the robot moves and records new observations Z_{t+1} . The mapping process is detailed in Figure 2.4.

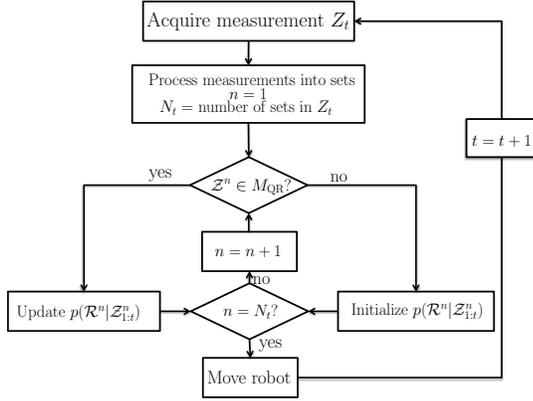


Figure 2.4: Overview of the PQR process. The robot gathers measurements from the environment and uses these measurements to update the probability distributions over landmark sets. After all distributions are updated and new sets are added to the map, the robot moves and collects new measurements. The update process follows (2.10).

A key component of the mapping algorithm is the map update. Landmark sets are updated recursively with a Bayes filter as new observations are made.

$$\begin{aligned}
 p(\mathcal{R}^n = q | \mathcal{Z}_{1:t}^n) &= \\
 \frac{p(\mathcal{R}^n = q | \mathcal{Z}_t^n) p(\mathcal{R}^n = q | \mathcal{Z}_{1:t-1}^n)}{\sum_{q=1:Q} p(\mathcal{R}^n = q | \mathcal{Z}_t^n) p(\mathcal{R}^n = q | \mathcal{Z}_{1:t-1}^n)} & \quad (2.10)
 \end{aligned}$$

where t is the current time, and q takes on one of the Q qualitative states. The distribution $p(\mathcal{R}^n = q | \mathcal{Z}_t^n)$ is the belief of the qualitative state given the current data, as computed in (2.9). The output of the filter from time step $t - 1$ is the distribution $p(\mathcal{R}^n = q | \mathcal{Z}_{1:t-1}^n)$. If there is no map entry for the n^{th} triplet, initialization takes the form of $p(\mathcal{R}^n = q | \mathcal{Z}_{1:t}^n) = p(\mathcal{R}^n = q | \mathcal{Z}_t^n)$. Landmark sets that are not observed at time step t are updated as $p(\mathcal{R}^n = q | \mathcal{Z}_{1:t}^n) = p(\mathcal{R}^n = q | \mathcal{Z}_{1:t-1}^n)$.

There are off the shelf algorithms which give both landmarks and correspondences [9, 25]. While the PQR assumes that correspondences are given, the PQR formulation is inherently robust to many errors including measurement

and correspondence errors. The results section studies the sensitivity of the PQRM algorithm to metrical errors. An off the shelf computer vision algorithm is used to find landmarks and correspondences. Intuitively, if a better correspondence algorithm were used, the mapping results would improve.

Algorithm 1: Map Updates

Input Z_t, M_{QR}

for each landmark set \mathcal{L}^n in Z_t **do**

Compute $p(\mathcal{R}^n | \mathcal{Z}_t^n)$

if $\mathcal{L}^n \in M_{QR}$ **then**

$$p(\mathcal{R}^n | \mathcal{Z}_{1:t}^n) = \frac{p(\mathcal{R}^n = q | \mathcal{Z}_t^n) p(\mathcal{R}^n | \mathcal{Z}_{1:t-1}^n)}{\sum_{q=1:Q} p(\mathcal{R}^n = q | \mathcal{Z}_t^n) p(\mathcal{R}^n | \mathcal{Z}_{1:t-1}^n)}$$

else

Initialize \mathcal{L}^n in M_{QR} with distribution $p(\mathcal{R}^n | \mathcal{Z}_t^n)$

end if

end for

for each landmark set \mathcal{L}^n in M_{QR} and not in Z_t **do**

$$p(\mathcal{R}^n | \mathcal{Z}_{1:t}^n) = p(\mathcal{R}^n | \mathcal{Z}_{1:t-1}^n)$$

end for

2.3 Experiments and Results

The PQRM algorithm is evaluated on both simulated and experimental data. Monte Carlo simulations are used to statistically characterize the PQRM maps shown in Figure 2.2.1 as a function of key system parameters, such as landmark distribution and sensor noise statistics. The Monte Carlo simulations also study the performance as the map representation varies (Figure 2.2.1). The New Col-

lege dataset [45] is used for experimental evaluation of the PQRM algorithm in practical settings. The EDC is used as the primary map representation since it is the finest partition of space shown in Figure 2.2.1, and all other QRMs in Figure 2.2.1 can be created by merging subsets of EDC states.

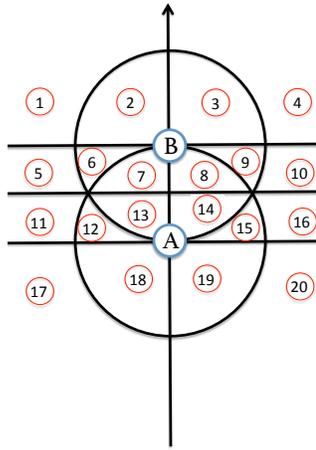
The proposed algorithm runs in $O(nsQ^2)$ time at every update, where n is the number of landmark sets observed, Q is the number of qualitative states in the given representation, and s is the number of samples used to approximate the integral in equation (2.3). The landmark sets are triplets since a ternary calculus is used. The parameter n can range anywhere from m^3 to $\binom{m}{3}$, where m is the number of distinct landmarks observed. The worst case, m^3 , occurs when the relationship of all 6 possible landmark orderings is recorded. Entering $\binom{m}{3}$ possibilities results in a non-exhaustive map that is faster to compute for large number of observations. The unitary operators can be used to provide the missing relationships at a later time. The reported results record all possible orderings of landmark triplets. It is assumed that a sufficiently small number of observations is made at every time step so that computational speed is not an issue when exhaustively mapping all possible triplet orderings. No more than 10 landmarks are observed at a single time instance in either the New College dataset or simulated maps. However, if compute time is a concern and a reliable orientation feature is available one may choose to use the binary calculi which will have better performance in terms of number of landmarks observed at every time step (worst case m^2 in the number of landmarks observed).

2.3.1 Validation Via Monte Carlo Studies

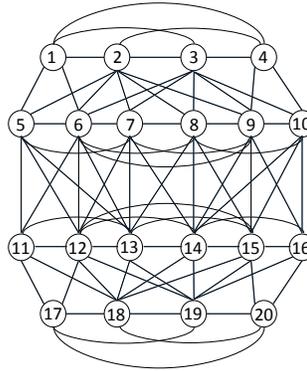
The online PQRM is evaluated in simulation for convergence and correctness of maps resulting from noisy sensor measurements. In each simulation, the locations of environmental landmarks are sampled from a uniform distribution over a $100\text{m} \times 100\text{m}$ environment. For each sample environment, 21 cases of sensor noise standard deviation ranging from 0m to 100m are evaluated 1000 times each. The robot is simulated driving one loop of a circular trajectory around the perimeter of the environment, taking one measurement at intervals of 16 degrees, in each simulation. The sensor range of the robot is set to 60m to ensure that the robot does not observe all of the landmarks at every time step. The number of landmark sets measured in each simulation depends on the distribution of the landmarks in the environment. The resulting maps are evaluated with two criteria: mean map entropy to evaluate map convergence and mean map error rate to evaluate map correctness. Additional statistics, such as the mean probability of the maximum likelihood state are also reported. All results are reported as a function of a normalized sensor noise, $\sigma_N = \frac{\sigma_R}{\sigma_E}$, where σ_R is the standard deviation of the distribution from which observations are drawn and σ_E is the standard deviation of the distribution from which environmental landmarks are drawn. The largest normalized standard deviation ($\sigma_N = 3.5$) corresponds to a sensor noise standard deviation of 100m. The EDC map representation is analyzed first to give insight into the map results, then results from other representations (see Figure 2.2.1) are stated.

Evaluation Metrics

Map correctness in the EDC is evaluated as both the mean map error rate and the distance of the maximum likelihood qualitative state from the true qualitative state. A metric based on the conceptual neighborhood [8] is introduced to evaluate this distance. This metric, termed the *neighbor distance*, is analogous to a chessboard distance over qualitative states. The neighbor distance is the number of hops between adjacent qualitative states that lie between the estimate and ground truth. The neighbor distance is most easily found by forming a graph where every node is a qualitative state (labeled as shown in Figure 2.5(a)) and an edge exists between nodes if the two qualitative states are spatially adjacent, as shown in Figure 2.5(b). Qualitative states that share a corner are considered to be spatially adjacent. For example, there is an edge between the node that represents qualitative state 1 and those that represent qualitative states 2-6. The neighbor distance is the minimum number of edges in the graph that must be traversed to get from the maximum likelihood qualitative state to the true qualitative state. For the EDC, the minimum neighbor distance is 0, meaning the estimate qualitative state is the correct one. The maximum neighbor distance is 4, denoting maximal traversal of the EDC. For coarse qualitative representations, the maximum neighbor distance is 2 or less, yielding less interesting information than in the case of the finer representations of space, like the EDC.



(a) The EDC shown with labeled qualitative states.



(b) Graph used to compute the neighbor distance for the EDC. Qualitative states are shown in circles, and edges represent state connectivity.

Figure 2.5: An EDC with labeled qualitative states and the graph used to compute the neighbor distance for EDC qualitative states.

Map convergence is measured in terms of entropy; both total map entropy and mean map entropy have been used with discrete maps in the past to measure map convergence, as discussed in [3]. The mean PQRm map entropy is used here since it is not sensitive to the number of triplets in the map. The

PQRM mean map entropy is computed as

$$H_{\text{mean}} = \sum_{n=1:N} \frac{H^n}{N} \quad (2.11)$$

where $H^n = -\sum_{q=1:Q} p(\mathcal{R}^n = q|\mathcal{Z}^n) \log [p(\mathcal{R}^n = q|\mathcal{Z}^n)]$ is the entropy of the distribution describing the n^{th} triplet in the map, Q is the number of qualitative states for each triplet in the QRM, and N is the total number of landmark sets in the map. The worst case H_{mean} for the EDC is 3.00 nats, representing that all landmark sets in the map have a uniform distribution over the possible qualitative states. The best case is $H_{\text{mean}} = 0$ nats, representing the case where the qualitative state of every landmark set in the map is known perfectly.

Evaluation on the EDC

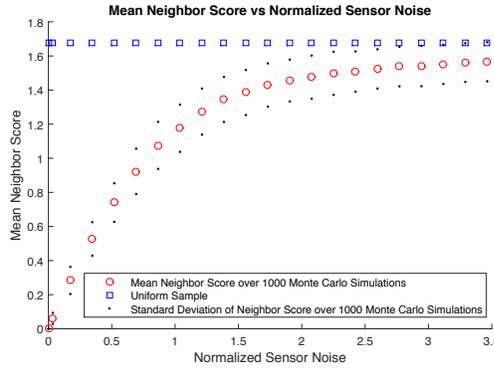


Figure 2.6: Mean neighbor distance of the PQRM maps for varying levels of sensor noise. Note that the worst case standard deviation from the mean is a neighbor distance of 0.14.

The mean neighbor distance for each set of Monte Carlo simulations in each environment is shown in Figure 2.6. Figure 2.6 also shows the mean neighbor distance for the case of randomly sampling estimated states and computing the distance to a truth qualitative state 1000 times per each truth qualitative state.

By comparison, it can be seen that as the sensor noise increases, the average neighbor distance approaches the case where qualitative states are uniformly sampled. This is an intuitive result since as the sensor noise increases the distribution that is sampled from approaches uniform. However, for reasonable levels of normalized sensor noise ($\sigma_N < 1$) the mean neighbor distance is less than one, showing that the map converges either to the correct qualitative state or one close to it.

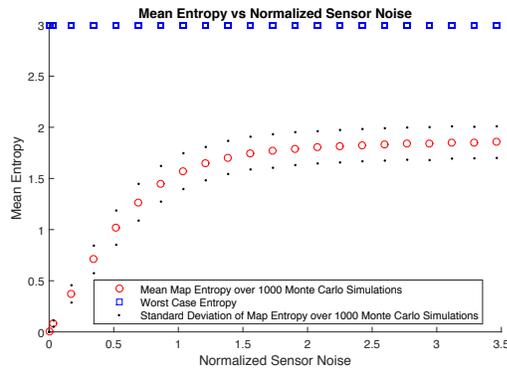
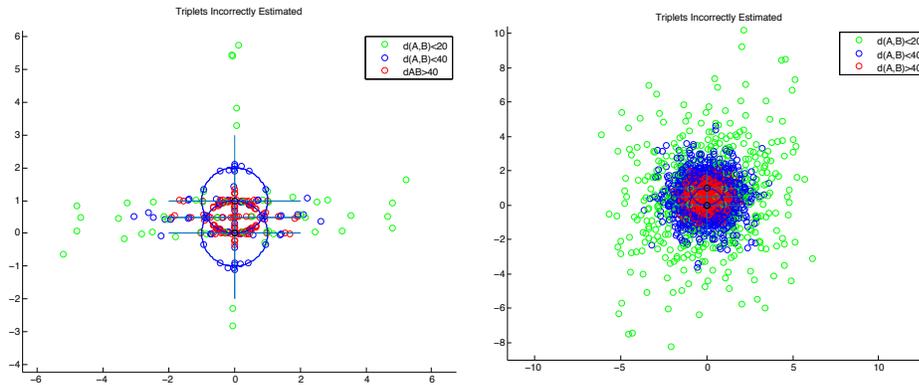


Figure 2.7: Mean entropy of the PQRm maps for the Monte Carlo simulations over varying levels of sensor noise. The blue squares represent the maximum possible mean entropy (3.00 nats) for the simulations. The maximum standard deviation from the mean is 0.180 nats.

Figure 2.7 plots the mean PQRm EDC map entropy for varying sensor noise. As shown, the mean PQRm map entropy is significantly less than the worst case ($H_{\text{mean}}^{\text{EDC}} \ll H_{\text{mean}}^{\text{worst}}$, with a minimum distance of 1.14 nats), even for extreme levels of sensor noise ($\sigma_N > 1$). The worst case standard deviation for each of the Monte Carlo trial runs is 0.18 nats, and it grows for $\sigma_N < 1$ while remaining approximately constant for $\sigma_N > 1$. The mean map entropy asymptotes at 1.85 nats. These results imply that the qualitative map converges even under worst-case noise statistics from the sensors. Empirically, in the EDC PQRm an

entropy for a single map element in the range $1.7 \text{ nats} < H^n < 2.6 \text{ nats}$ indicates that a single $p(\mathcal{R}^n = q|\mathcal{Z}^n)$ has three or four likely qualitative states with similar likelihood values, whereas an entropy of $H^n < 0.7 \text{ nats}$ indicates that the distribution has only one likely state. As seen in Figure 2.7, the PQRM maps are able to converge to four or less qualitative states per triplet of landmarks, on average, for all levels of sensor noise. The true state of the triplet may be among these states, but not the maximum likelihood state. Therefore, for high noise environments, where distribution over triplets have a high entropy, it may be necessary to either gather more measurements until the filter converges to one state or consider all of the high likelihood states when using the PQRM map for subsequent functions.

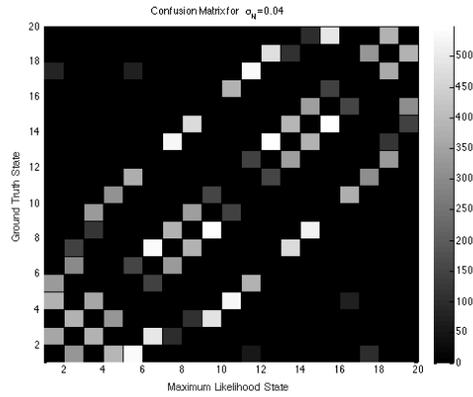


(a) Incorrect triplets for a sensor noise standard deviation of 1m ($\sigma_N = 0.04$) (b) Incorrect triplets for a sensor noise standard deviation of 50m ($\sigma_N = 1.73$)

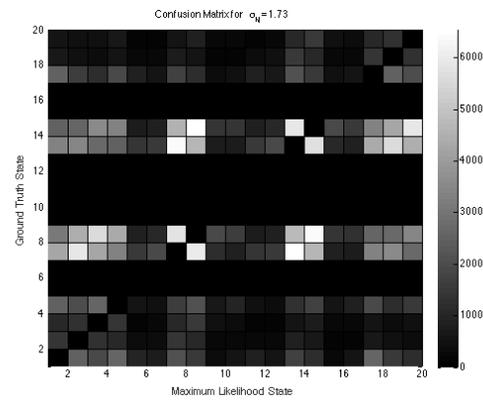
Figure 2.8: Triplets estimated incorrectly on the EDC for two different values of σ_N .

To understand the performance of the EDC representation more clearly, the errors made on the EDC are analyzed. Figure 2.8 shows the location of the third landmark for all trials of the PQRM where the maximum likelihood qualitative state is incorrect. The locations are all normalized to a 0-1 EDC such that they can be overlaid on one plot. The landmarks are color coded by the euclidean

distance between the landmarks A and B , $d(A, B)$. Figure 2.8(a), plots the landmark errors for the low noise scenario, $\sigma_N = 0.04$. Notice that nearly all errors are made exclusively on the boundaries of the qualitative states. However, as the noise increases to $\sigma_N = 1.73$, errors occur more frequently, especially on the central qualitative states. Figure 2.9 plots the confusion matrix created for all relationships over the 1000 Monte Carlo runs for these two scenarios. Results show that most of the errors occur in the inner lune states of the EDC (states 7, 8, 13, and 14). This leads to the conclusion that when the sensor noise increases to a level greater than the size of the qualitative states, then other, coarser, representations of the qualitative map should be considered.



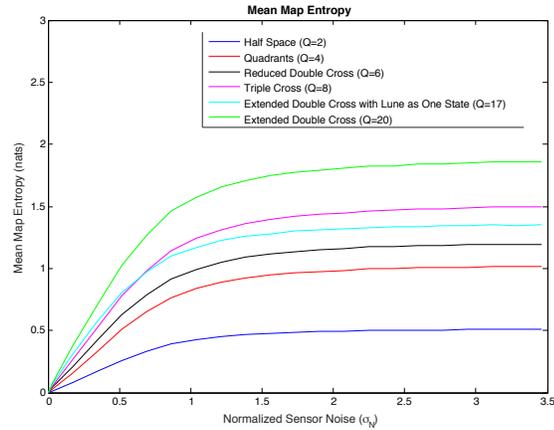
(a) Confusion matrix for $\sigma_N = 0.04$.



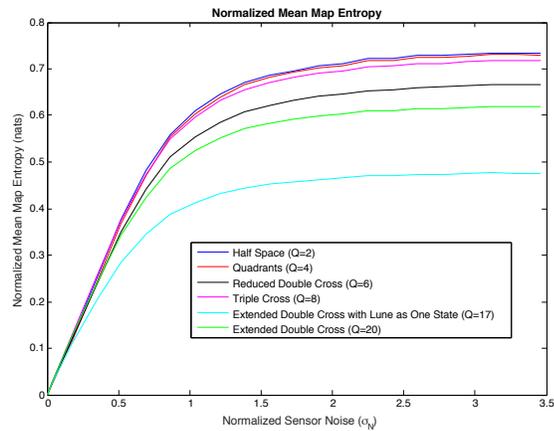
(b) Confusion matrix for $\sigma_N = 1.73$.

Figure 2.9: Confusion matrix for all errors made in the 1000 Monte Carlo trials conducted for the EDC qualitative representation.

Evaluation on Reduced Representations of the EDC



(a) Mean map entropy over 1000 monte carlo simulations for six different qualitative representations of space.

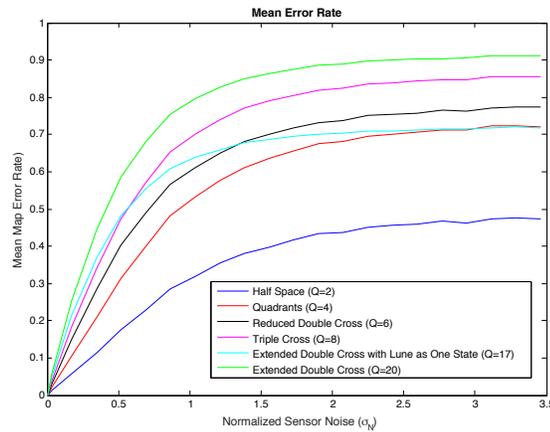


(b) Normalized mean map entropy over 1000 monte carlo simulations for six different qualitative representations of space.

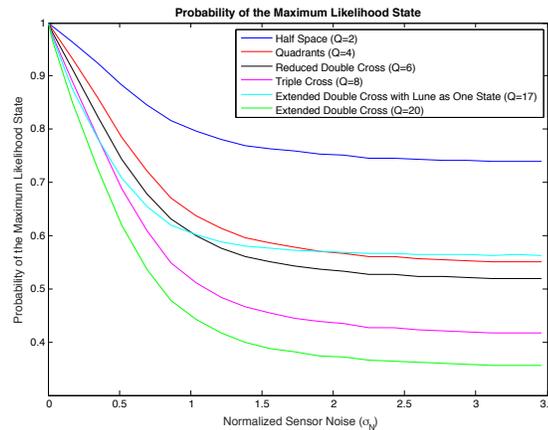
Figure 2.10: Absolute and normalized mean map entropy for Monte Carlo trials computed for the EDC and five reduced representations shown in Figure 2.2.1.

Figure 2.10(a) plots the mean map entropy for the five reduced EDC qualitative representations shown in Figure 2.2.1, each simulated in 1000 Monte Carlo simulations for varying levels of sensor noise. The general trend shows that

the entropy is lower for coarser representations of space for all σ_N with the notable exception of the EDC with the lune as one state. The general trend can be attributed to two causes: the coarser states containing more of the cumulative distribution $p(\mathcal{L}|\mathcal{Z})$ and the fact that the maximum possible entropy for each distribution decreases as the number of possible states decreases. The mean map entropy normalized by the maximum possible entropy for each representation is plotted in Figure 2.10(b) to account for the different levels of maximum entropy. On this normalized scale, it is seen that the EDC with the lune as one state is the best performer with the division of space into left and right performing the worst. The general trend is that a finer resolution of space does better in terms of normalized entropy; however, in some cases performance increases when smaller qualitative states are merged and sensor noise is large. Consider the cases of the EDC with the merged lune states and the Reduced Double Cross. In both cases, small qualitative states are merged and these representations outperform the EDC and Triple Cross, respectively, in terms of normalized mean map entropy. Relating back to the confusion matrices in Figure 2.9 of the Monte Carlo studies, it is seen that both the Reduced Double Cross and the EDC with merged lune states combine states of finer representations that are easily confused. Hence, merging easily confused states leads to a better performance in terms of map convergence.



(a) Mean map error rate for different values of sensor noise



(b) Mean probability of the maximum likelihood state for different values of sensor noise.

Figure 2.11: Mean map error rate and probability of maximum likelihood state for six qualitative representations reported as a function of σ_N .

The mean error rate for each qualitative representation is shown in Figure 2.11(a). As seen, the error rate decreases in general for decreasing number of qualitative map states. The notable exception again is the EDC with lune as one state. For all qualitative representations shown, when the sensor noise is at its highest, $\sigma_N = 3.5$, the error rate is lower than that error rate for randomly guessing a qualitative state. As shown in Figure 2.11(b), the probability of the

maximum likelihood qualitative state decreases on average as the error rate increases. Thus the confidence in the estimate, on average, decreases with increasing error rate and mean map entropy.

2.3.2 PQR Evaluation on the New College Dataset

The New College dataset [45] consists of data taken from a robot driving loops around the Oxford campus. Data include lidar scans, panorama images from a Point Gray Ladybug, calibrated stereo images from a Point Gray Bumblebee, GPS data, and inertial sensing data. The data is divided into epochs consisting of: 1) loops around a small quad, 2) loops around a parkland, and 3) loops around a second quad connecting the parkland to the first quad. The first epoch of the New College dataset is used in the analysis here, during which the robot remains stationary for 35 seconds and then drives several loops around a quad. A total of 5082 image pairs are used to map the quad.

Generation of Point-Landmarks

The PQR algorithm requires the use of point-like landmarks. Neither surveyed landmarks (for truth), or specific landmarks segmented from data (lidar or vision) are provided in the New College Dataset. While the proposed PQR algorithm is designed to be used with any method that generates point-like landmarks from raw data, a common approach is taken here where visually salient regions [10] are extracted from the stereo data. First, VOCUS [9] is used to compute a saliency map from the stereo visual data. Since the stereo images in the New College dataset are black and white, the color portion is omitted from

the saliency map. The most salient regions are found by applying a threshold to the saliency map. Bounding boxes for the contours of the salient regions are computed.

Point-like landmarks must be extracted from the calculated visually salient regions. There are many approaches to do this, and each can create errors during the process of landmark extraction and recall. First, thresholding the saliency map to compute the landmark region can result in bounding boxes of different sizes and positions for the same landmark, as shown in Figure 2.12(a). If the centroid of the bounding box is used, the varying sizes of the boxes over time create varying locations of the centroid position (x, z) . Second, mistakes in landmark association can cause errors. Third, the algorithm itself (e.g. centroid vs closet point, etc.) used to compute the point-like landmark can introduce errors unique to the processing scheme.

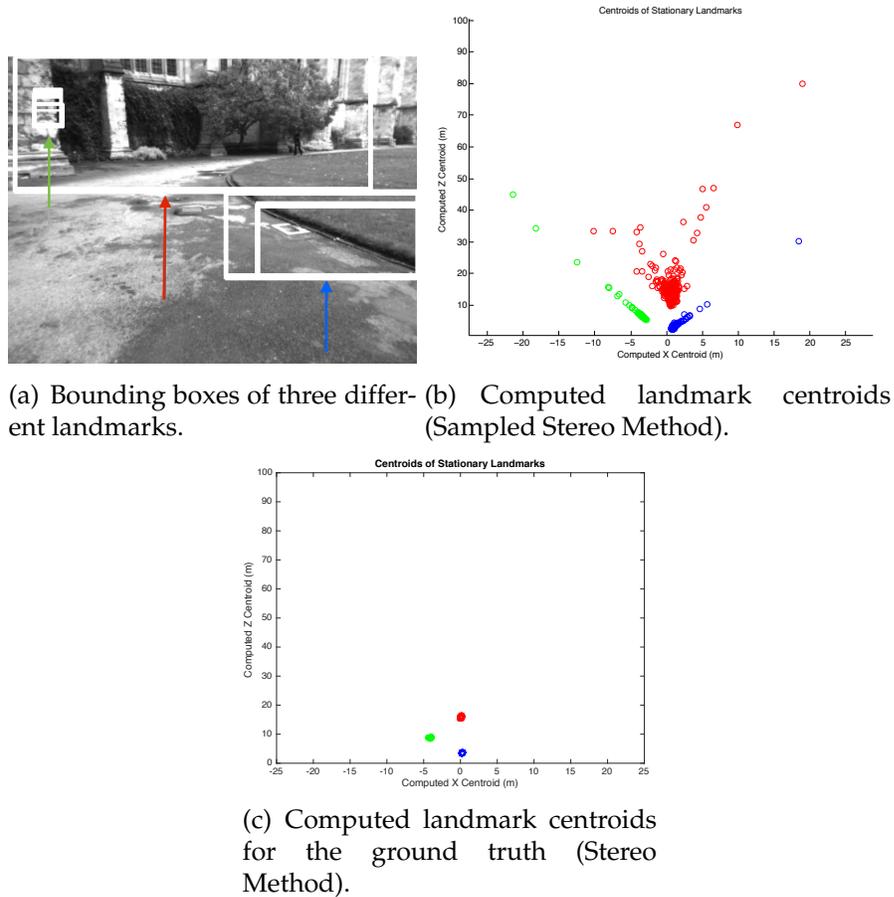


Figure 2.12: Bounding boxes and corresponding centroids computed for three landmarks in the first 35sec of the New College dataset.

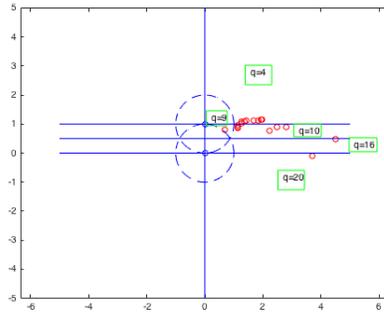
Three different processing methods for point-landmark extraction from the computed salient regions are developed and compared in the PQRM algorithm analysis. The first type of processing is based on correspondences of robust image features. Measurements produced by this type of processing are the mean (x, z) coordinates of sampled SIFT features within a salient region bounding box that have correspondences between the left and right stereo images. Examples of this type of measurement can be seen in Figure 2.12(b). The second type of measurement is computed using a block matching algorithm to calculate the disparity between left and right stereo images. The disparity is used to find

the (x, z) coordinate of every pixel within a salient region bounding box. The centroid of all the pixel coordinates is the point-landmark measurement; an example of the data resulting from this processing method can be seen in Figure 2.12(c). Finally, a third type of processing is tested for cases when a user only has access to a monocular camera. The monocular measurement is calculated by observing the change in landmark position in one camera as the robot moves. These three approaches to collecting point-landmark measurements from the visual data are referred to as the Sampled Stereo method, Stereo method, and Monocular method, respectively. Note that data points from the Monocular method are not shown in Figure 2.12 because the robot is stationary while observing these landmarks. Due to the need for motion, there are fewer data points using the Monocular method than the either Sampled Stereo or Stereo methods.

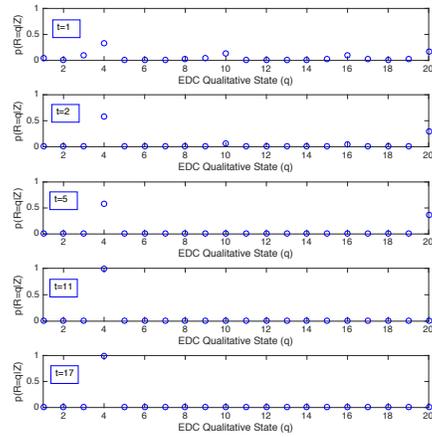
Since the Stereo method yields measurements with low variance (as seen by examining the difference between Figures 2.12(b) and Figures 2.12(c)), these measurements are used to derive a ground truth. Specifically, every measured triplet is transformed into a normalized coordinate frame where the first landmark is always at $(0, 0)$, and the second landmark is always at location $(0, 1)$. The location of the third landmark is transformed into this coordinate system, and the resulting coordinates of the third landmark are referred to as the normalized triplet position. The average point location of the normalized triplet position for all observations of the triplet is computed, and the qualitative state of the mean normalized triplet position is used as the ground truth for the triplet. The performance of the PQRM for each of the three measurement types is evaluated for mean neighbor distance compared to ground truth and entropy. The performance is evaluated on the EDC and the reduced representations presented in

this work for error rate as compared to the ground truth.

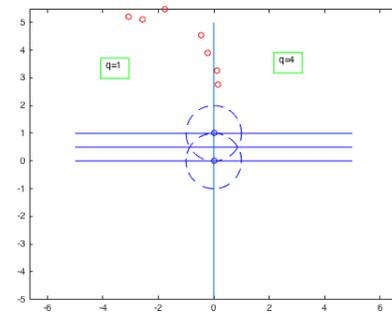
The PQRM and Noisy Data



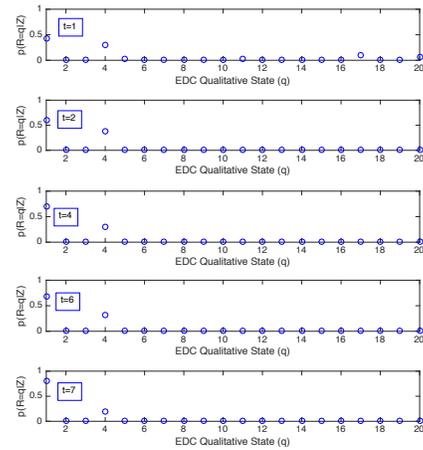
(a) 17 observations of a single landmark triplet in the New College dataset.



(b) Distribution $p(\mathcal{R}|\mathcal{Z})$ for the triplet observations in (a) as time evolves. The final distribution has converged onto qualitative state four. The neighbor distance to the ground truth state is 2.



(c) 7 observations of a single landmark triplet in the New College dataset.



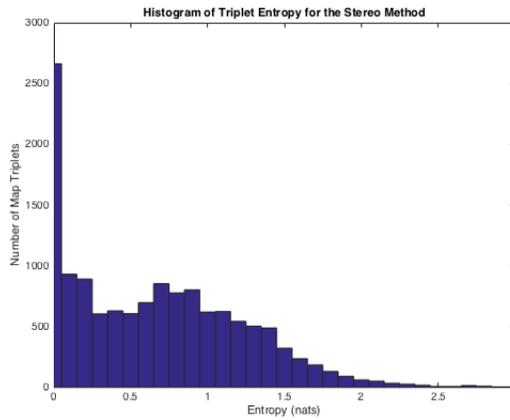
(d) Distribution $p(\mathcal{R}|\mathcal{Z})$ for the triplet observations in (c) as time evolves. Note that the final distribution assigns probabilities to states that is proportional to number of observations made for that state. The neighbor distance to the ground truth state is 0.

Figure 2.13: Evolution of select distributions $p(\mathcal{R}|\mathcal{Z})$ for triplets in the New College dataset with time.

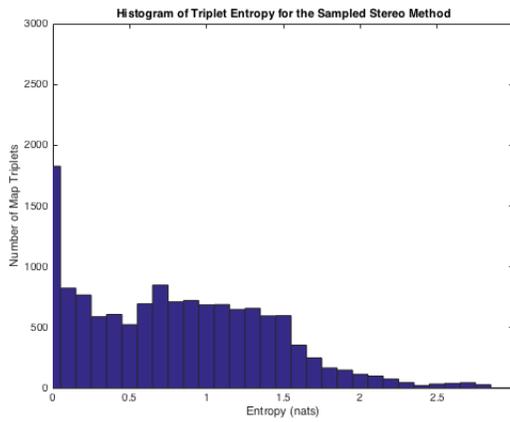
To develop an intuition of how the PQRM works on noisy data, an example of two landmark triplets being mapped is shown in Figure 2.13. Here, the normalized triplet position resulting from noisy data produced by the Sampled Stereo method is plotted with an overlaid EDC for two landmark triplets (shown in Figure 2.13(a,c)). In a metrical sense, the third landmark variance is larger in Figure 2.13(c) than in Figure 2.13(a). In terms of qualitative states, the raw landmark measurements in Figure 2.13(a) appear in five qualitative states, whereas the landmark measurements in Figure 2.13(c) only appear in two qualitative states. Figures 2.13(b,d) plot the estimate of the qualitative states (a total of 20 from left to right) for each of these cases over time (a total of 5 time steps from top to bottom), for the PQRM algorithm outlined in Algorithm 0. In the case of Figure 2.13(a), despite the noisy measurements that appear in multiple qualitative states, the filter converges to the qualitative state with the most evidence, $q = 4$ (7 of the total 17 measurements). Similarly, in Figure 2.13(c) there are two observations in qualitative state $q = 4$ while there are five in $q = 1$. At $t = 1$ and $t = 2$ for this triplet, the data points received are in $q = 1$ but metrically close to $q = 4$. This is reflected in the distribution shown in Figure 2.13(d), where states $q = 1$ and $q = 4$ are almost equally likely at $t = 1$ and $t = 2$. At the remaining time steps shown, the PQRM receives further data points in $q = 1$ and converges to $q = 1$. This shows that the PQRM considers the evidence provided in the form of single point landmark measurements, and renders a verdict proportional to the evidence calculated via a Bayes filter.

Analysis of Different Measurements and Data Processing

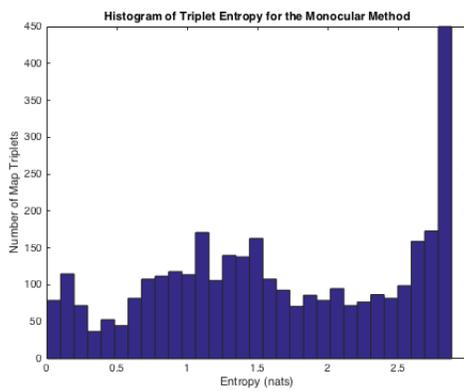
The performance of the EDC PQRM for the New College dataset is tested with the measurements resulting from the Stereo, Sampled Stereo, and Monocular methods, and evaluated in terms of neighbor distance and mean map entropy. Histograms of entropy for map triplets in PQRMs created with the three types of measurement processing are shown in Figure 2.14. For the PQRM created with measurements from the lowest noise Stereo method (Figure 2.14(a)) the majority of the map triplets have low entropy, indicating distributions that are well converged onto a single qualitative state. The mean map entropy is 0.7 nats, reflecting that on average the PQRM has converged onto one qualitative state for each map triplet. The PQRM created with the higher noise Sampled Stereo method (Figure 2.14(b)) shows that the majority of map triplets have low entropy. However, due to the higher noise when using the Sampled Stereo method of data extraction there is a noticeable increase in the number of map triplets with high entropy. The mean map entropy for the PQRM created with the Sampled Stereo method is 1.3 nats, indicating that the map triplets have converged to between one and four qualitative states on average. The PQRM created with the noisiest measurements extracted with the Monocular method has the highest entropy of the three maps created, as seen from the histogram in Figure 2.14(c). Here, more than one third of the map triplets have an entropy of greater than 2.0 nats, and the mean map entropy is 1.7 nats, indicating that the map has converged onto 4 or more likely states at best.



(a) Entropy of the map triplets for the Stereo method.



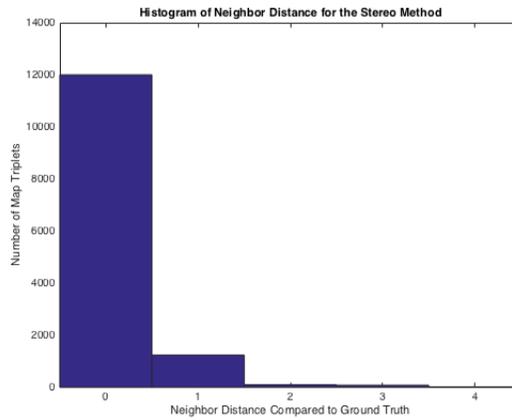
(b) Entropy of the map triplets for the Sampled Stereo method.



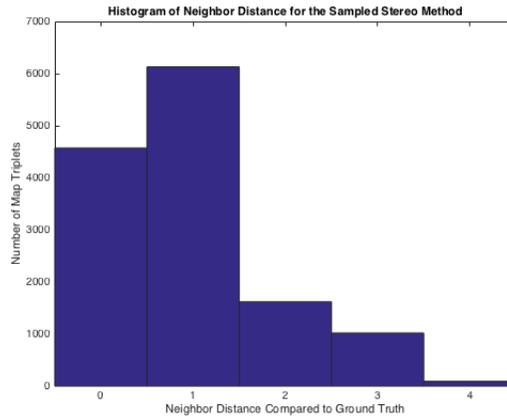
(c) Entropy of the map triplets for the Monocular method.

Figure 2.14: Histograms of the entropy for the PQRMs generated with the three types of measurement processing.

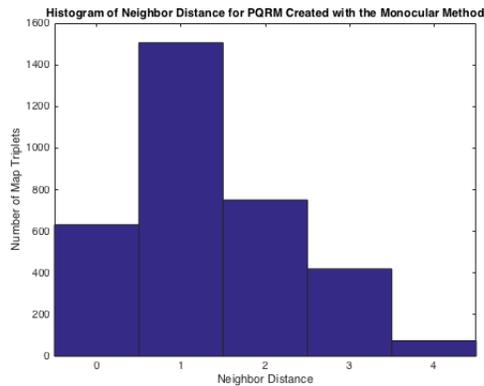
The neighbor distance of the maximum likelihood state compared to the ground truth is used to evaluate the performance of the PQRM s created with different methods of measurement processing. While there may be several highly likely qualitative states for a map triplet, only the maximum likelihood state is used for map evaluation with the neighbor distance since the neighbor distance is a coarse metric and including multiple likely states results in a low neighbor distance. Figure 2.15 shows the histograms of neighbor distance compared to ground truth of the PQRM s created. The mean neighbor distance for the PQRM created with the Stereo method (Figure 2.15) is 0.13 neighbor distance units, and mode neighbor distance is 0. Thus, the PQRM created with the Stereo method most often estimates the correct qualitative state for map triplets. Even though the ground truth map is created from the data extracted with the Stereo method, it is not always the case that the PQRM matches the qualitative state predicted by the normalized mean triplet position, especially when several measurements occur close to the boundaries of the qualitative states. The mean neighbor distance for the PQRM created with the Sampled Stereo method is 0.95 neighbor distance units, and mode neighbor distance is 1, as seen in Figure 2.15(b). These results, along with the results for the map entropy reported in Figure 2.14(b), imply that there may be more than one likely state for most map triplets, but that the most likely state is often one removed from the ground truth. The mean neighbor distance for the PQRM created with the Monocular method is 1.3 neighbor distance units, and mode neighbor distance is 1, as seen in Figure 2.15(c). Similar to the PQRM created with the Sampled Stereo method, these results along with the results reported in Figure 2.14(c) imply that several likely states exist for the majority of map triplets, however the most likely state is most often one removed from the ground truth state.



(a) Neighbor distance of the maximum likelihood qualitative state compared to the ground truth for the Stereo method.



(b) Neighbor distance of the maximum likelihood qualitative state compared to the ground truth for the Sampled Stereo method.



(c) Histograms of the neighbor distance for the PQRMs maps generated with three types of measurement processing.

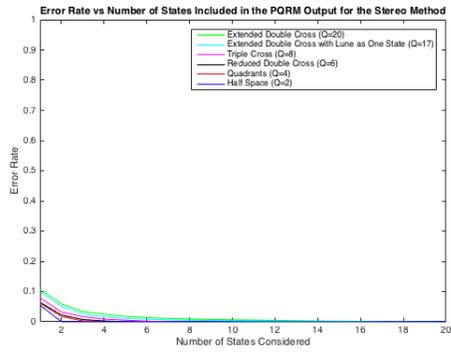
Figure 2.15: Histograms of the neighbor distance for the PQRMs maps generated with three types of measurement processing.

Drawing direct connections between the results reported for the Monte Carlo studies and the results reported for the New College dataset is difficult. The landmarks in the New College dataset have a higher density than the landmarks in the Monte Carlo studies, and the landmarks detected in the New College dataset are located within a limited bearing window at every time step due to the use of a stereo rig with limited viewing angle. Therefore, the inter-landmark distances for mapped triplets are reduced and sensor noise compared to the landmark spacing may be higher than indicated by the Monte Carlo Studies. As an approximation, the values of normalized sensor noise ($\sigma_N = 0.05$ for the Stereo method, $\sigma_N = 0.7$ for the Sampled Stereo method, and $\sigma_N = 1.5$ for the Monocular method) that most closely match the neighbor distances reported in Figure 2.6 are used. Using these values and referring back to Figure 2.7 shows that the mean map entropy reported for the New College PQRM is higher than the values reported for the approximated normalized sensor noise. This indicates that the New College PQRM is conservative compared to the PQRM created in Monte Carlo simulations for the same neighbor distance.

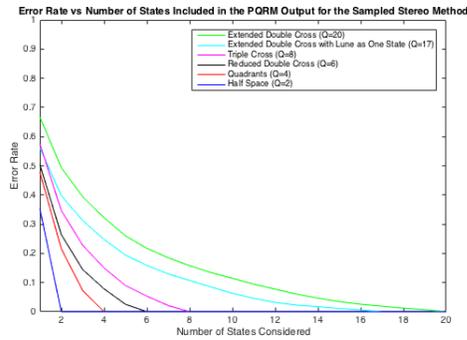
Analysis on Reduced Representations of the EDC

Reduced representations are used to further evaluate the performance of the PQRM algorithm with the three measurement processing methods on the New College data. The mean map error rate compared to ground truth is used as the evaluation metric. The error rate is defined as the fraction of estimated qualitative states that are not the same as the ground truth qualitative state. Given that measurements computed with the three different processing methods may yield relationships where several qualitative states are highly likely, multiple likely

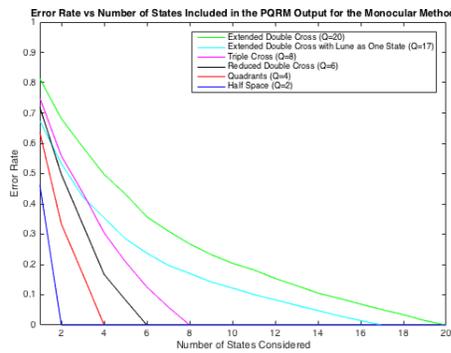
qualitative states are considered when computing the error rate. The results reported in Figure 2.16 show error rates for multiple qualitative state outputs. The vertical axis represents the error rate, and the horizontal axis represents the case where the n most likely qualitative states from the PQRM are used as the output of the algorithm. If the ground truth agrees with one of the n output states then the estimate is said to match the ground truth.



(a) Error rate vs number of most likely states included in the PQRM output for the Stereo method.



(b) Error rate vs number of most likely states included in the PQRM output for the Sampled Stereo method.



(c) Error rate vs number of most likely states included in the PQRM output for the Monocular method.

Figure 2.16: Error rate for all PQRM representations tested. The error rate is reported a function of the number of most likely states included in the PQRM output for the Stereo method, Sampled Stereo method, and Monocular method.

Figure 2.16(a) shows the error rate for the PQRM s created with the Stereo method. The error rates are reduced by half when the number of states output from the PQRM increases from 1 to 2. The error rates for all PQRM s continue to decrease with increasing number of states output. Notice that for only 1 qualitative state output, the error rate reported matches with that reported for $\sigma_N = 0.05$ in Figure 2.11(a). Figure 2.16(b) reports the error rate for the PQRM s created with the Sampled Stereo method. The error rates decrease with increasing number of state outputs, with a reduction by half for the finer qualitative representations by having an output of 4 qualitative states instead of 1. The error rate for all PQRM s created with the Sampled Stereo method and only 1 state output match those reported in Figure 2.11(a) for $\sigma_N = 0.7$. Finally, Figure 2.16(b) reports the error rate for the PQRM s created with the Monocular method. A reduction by half is reached for the finer qualitative representations by considering an output of 5 qualitative states instead of 1. The error rate for all PQRM s created with the Sampled Stereo method and only 1 state output match those reported in Figure 2.11(a) for $\sigma_N = 1.5$. A user of the PQRM algorithm may choose to leverage error rate with number of states considered in the final output of the algorithm. For applications in planning, considering several output qualitative states can be useful in developing contingency plans for robots when observations do not match the robot’s belief of the environment.

In conclusion, experiments on the New College data show that the PQRM algorithm can produce results that closely match results for error rates and neighbor distance predicted from simulation, even in the presence of sensor noise that may not be well modeled by a Gaussian distribution. However, the map entropy is inflated, indicating that conservative estimates of the map certainty are produced, especially in high noise conditions. These results show that the

PQRM algorithm can be successfully used on a field robot with a stereo rig and an off the shelf computer vision algorithm for landmark detection. When multiple qualitative states are considered as the end result, the PQRM created with the Monocular method can be successfully used on a field robot with a monocular camera.

2.4 Conclusion

A probabilistic qualitative relational mapping (PQRM) algorithm is developed for mobile robots that produces maps which contain discretized relative spatial information without the need for a fixed coordinate frame, expensive linearizations, or iterative optimizations. The constraint-based representation allows for a large variety of qualitative spatial relationships to be used in the PQRM framework. Monte Carlo results show that maps are considered converged with low entropy, even for extreme amounts of sensor noise. The PQRM maps constructed on data where ground truth is available show a neighbor distance of less than one for moderate levels of sensor noise and a fine representation of space. Furthermore, map error rates for all representations are less than the error rate of a random guess even for extreme levels of sensor noise. Evidence shows that errors at low sensor noise occur exclusively on the borders of qualitative states while at higher levels of sensor noise errors are likely to occur anywhere. It is shown that by merging easily confused qualitative states, maps with better convergence properties can be achieved without sacrificing significant spatial resolution.

Results using the New College dataset show that PQRM maps can be suc-

cessfully generated using an off the shelf detector for landmarks. Error rates and neighbor distance for PQRM generated from the New College data are well predicted by the results from simulation. The results indicate that maps created with the PQRM have a consistent estimate of the distribution over qualitative states. In situations with extremely noisy measurements, it may be necessary to include multiple qualitative states as the output of the algorithm. A user may leverage number of qualitative states output against error rate depending on the type of sensor and measurement processing used.

The general PQRM algorithm with a given qualitative representation has several uses beyond a single robot mapping an environment. First, given good landmark correspondences this can easily scale to multi-robot scenarios. The PQRM representation can be used for planning and navigation in any environment that contains distinct landmarks. Algorithms for navigation over QRMs are demonstrated in [33]. Such methods can be further expanded to account for map uncertainty in the PQRM representation.

CHAPTER 3

AUTOMATED GENERATION OF DISCRETE SALIENT REGIONS FROM PROBABILISTIC ENVIRONMENTAL INFORMATION

3.1 Introduction

Mapping is one of the fundamental problems in robotic perception, as the creation of consistent maps for planning is essential for ensuring success in many robotic applications. The breadth of map definitions and research allow robots to execute various tasks at different levels of abstraction seamlessly: from high level topological [43] and qualitative maps [53] that allow for reasoning about the spatial arrangement of objects at a high level, to low-level metric maps [48, 17] that allow for accurate navigation and obstacle avoidance. Maps at a higher level of abstraction in a complex environment, in particular, allow for efficient reasoning and planning by abstracting the continuous world into finite number of discrete regions. Topological maps [43], for example, abstract the world into a graph structure for planning, where nodes represent features or places and edges represent connectivity. Such maps work well for coarse navigation, such as in hallways [50] and between viewpoints of an environment [6]. Qualitative maps [53, 36, 27, 33], on the other hand, represent space as a set of large regions relative to each other or specific localtions, such as 'left', 'right', 'closer', or 'farther', and can be accurately described by spatial constraints. The spatial constraints describe regions relative to local coordinate frames; [27] and [33] demonstrate how these regions can be created using one or two landmarks for binary and ternary spatial calculi. Using such constraint information, a robot can reason about the relative positions of objects and nav-

igate to broad regions that position it ‘to the left’ of some objects relative to a reference.

Discrete representations of an environment are typically prescribed by a user before the robot enters its environment; however, the representation used may not be a good fit for the underlying data if the environmental structure is unknown a priori [39]. Therefore, one of the current challenges in robotics is creating high level maps that are consistent with low-level data. This challenge has been addressed in the planning domain with algorithms that automatically learn high-level state or symbolic representations from interactions with an environment [37, 19]. While these methods are useful for automatically learning actions and outcomes, they require detailed models of objects in the environment, classifiers that determine when high-level states are true, and are tightly coupled to the robot that learns the model.

The goal of this paper is to develop a scaleable discrete representation of an environment that both fits the low-level data well and can be used as a map for discrete planning within the environment by any robot. The discrete representation introduced in this paper divides environment into salient regions (SRs), which are characterized as containing interesting information about an environment, such as the location of objects. The SRs are learned from probabilistic beliefs about the environment, specifically defined as distributions over spatial locations of objects. Importantly, the input robot belief is very general; examples include the distribution over landmark locations from a simultaneous localization and mapping (SLAM) algorithm [17], distributions over relative locations learned from querying humans [49], or distributions over relative positions of objects (landmark pairs or triplets) or extended objects that are projected into a

common coordinate frame [39]. Compared to topological [43, 50, 6] and qualitative [53, 36, 27, 33] maps, SR maps are designed to represent underlying data well rather than use an assigned representation. SR maps can be used as topological maps by representing the SRs as vertices in a graph with edges connecting neighboring SRs, and can also be used as qualitative maps by developing the SRs with relational information.

The proposed SaliEnt Region GEneration (SERGE) algorithm is unique in that it automatically constructs a salient region map, or a collection of SRs, given a probabilistic belief about the environment. The SERGE algorithm efficiently finds SR maps with high saliency through intelligent sampling. Two parameters allow a user to control the SR map resolution and saliency. A multi-resolution version of the SERGE algorithm creates multi-resolution SR maps, where the level of environmental discretization can change depending on the needs of a planner. The resulting SR maps can be used for further environmental mapping or coarse navigation.

3.2 Problem Definition

3.2.1 Map Saliency

Consider the following mapping problem. Assume a probability distribution of the environment has been generated in the form of K beliefs,

$$B_k = p(X_k|Z_k) \quad k \in [1, \dots, K], \quad (3.1)$$

where $X_k \in \mathbb{R}^M$ is the spatial domain and Z_k is raw sensor data. The belief, B_k , is a fine grained spatial probability representation of the environment, such as the output of SLAM [17], beliefs over relative positions of landmark triplets [39], or extended objects, or beliefs learned from human input [49].

Given that each belief is a probability distribution over a spatial region, a formal definition of saliency for a region R_i is $S_i = \int_{R_i} B_k dX_k$. A highly salient region is then defined as a region, R_i , where $S_i \rightarrow 1$. Intuitively, saliency is the total probability that the object with location represented by B_k is located in R_i . For navigation or search applications, high saliency regions indicate that a desired object has a high probability of residing in a given region.

The mapping problem is then defined as: given K beliefs, B_k , find the set of μ regions (the map), $\mathcal{R} = \{R_1, \dots, R_\mu\}$, that maximizes map saliency, or $S = \prod_{i=1:\mu} S_i$.

A key novelty of the proposed algorithm is to allow the user to define μ . When $\mu = K$, the resulting map is straightforward, with one region, R_i , per belief, B_k . More generally, $\mu \leq K$ and each belief must be assigned to a region. Defining this assignment to be \mathcal{L}_k^B , the mapping problem can be formally defined as

$$S_{\max} = \max_{\mathcal{R}, \mathcal{L}^B} \left(\prod_{k=1:K} \sum_{i=1:\mu} \delta(i - \mathcal{L}_k^B) \int_{R_i} B_k dB_k \right), \quad (3.2)$$

where \mathcal{R} is the set of μ unknown spatial regions and \mathcal{L}^B are the unknown belief to region assignments.

3.2.2 Representations

Directly inferring the μ regions is intractable since the set of all collectively exhaustive, mutually exclusive spatial regions is infinite due to the continuous

nature of the environment. Therefore, a set of N collectively exhaustive, mutually exclusive cells, C , is introduced to reduce the solution space. Possible cells include but are not limited to hexagonal tessellations of space, Voronoi regions generated using the mean of each belief, or regions generated by classifiers. In the proposed problem, each cell, $C_n \in C$, is assigned to one of the R_i , denoted as \mathcal{L}_n^C . Then, \mathcal{L}^C is a vector of N assignments for all cells, \mathcal{L}_n^C . Regions, R_i , are defined as the union of all cells with assignment $\mathcal{L}_n^C = i$, or

$$R_i = \bigcup_{\mathcal{L}_n^C=i} C_n. \quad (3.3)$$

While computing an SR map with exactly μ regions may be desired, in some cases desired resolution may be sacrificed for higher SR map saliency. Two parameters are defined to allow the user to trade between saliency and resolution. The desired number of SRs in the map, μ , is a natural number between 1 and $\min(K, N)$ set by the user. A tolerance factor, $\sigma \in \mathbb{R}$, allows the user to trade between an SR map that has exactly μ SRs and an SR map that has higher saliency. As $\sigma \rightarrow 0$, \mathcal{R} tends to have μ SRs in the map. A higher value of σ increases the importance of saliency and creates an SR map that may have higher saliency but fewer SRs.

3.2.3 The General Problem of Finding Salient Maps

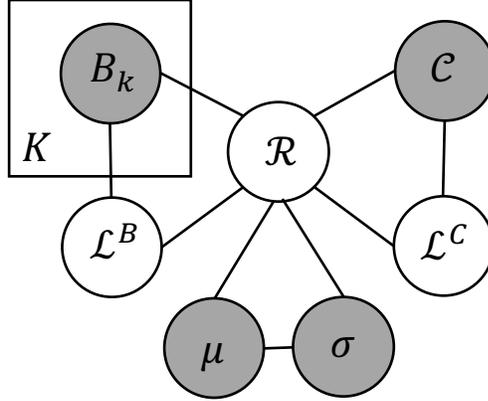


Figure 3.1: A graphical model showing the relationship between each of the variables in the salient region mapping problem. The shaded nodes represent observed variables, while the remaining nodes represent uncertain variables. This model shows three cliques that are used to factor the potential function in equation (3.4).

The key variables in solving for SR maps are given in Figure 3.1. The user parameters μ and σ guide the number of regions in \mathcal{R} . The variables \mathcal{L}^B and \mathcal{L}^C assign beliefs and cells to each SR, $R_i \in \mathcal{R}$, where $\mathcal{L}_i^B, \mathcal{L}_i^C \in [1, \dots, |\mathcal{R}|]$ and $|\mathcal{R}|$ is the number of SRs in the map. The SR map with the highest saliency and the desired number of regions is computed by maximizing a potential function,

$$\phi_{\max} = \max_{\mathcal{L}^B, \mathcal{L}^C, \mathcal{R}} \psi(B, \mu, \sigma, C, \mathcal{R}, \mathcal{L}^B, \mathcal{L}^C). \quad (3.4)$$

The potential function, $\psi(\cdot) = \psi_1 \cdot \psi_2 \cdot \psi_3$, has three factors given the relationships shown in Figure 3.1: $\psi_1(\mu, \sigma, \mathcal{R})$, $\psi_2(\mathcal{L}^C, C, \mathcal{R})$, $\psi_3(B, \mathcal{L}^B, \mathcal{R})$. The term $\psi_1(\mu, \sigma, \mathcal{R})$ is a potential over the number of SRs in the map, or

$$\psi_1(\mu, \sigma, \mathcal{R}) = \frac{1}{\beta \sqrt{2\pi\sigma^2}} e^{-\frac{(|\mathcal{R}| - \mu)^2}{2\sigma^2}}. \quad (3.5)$$

This potential is modeled after a normal distribution using the user defined parameters, where μ is the mean of the distribution, σ is the standard deviation, and β is an optional normalization factor, $\beta = \sum_{k=1}^K \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(k-\mu)^2}{2\sigma^2}}$. The potential ψ_1 is not a true normal distribution since $|\mathcal{R}|$ and μ are natural numbers; however the model has several nice properties. First, the maximum occurs when $|\mathcal{R}| = \mu$ and the desired number of regions is achieved. Second, as $(|\mathcal{R}| - \mu)^2$ becomes larger, the value of the potential decreases exponentially. The variable σ controls the rate of decrease as $(|\mathcal{R}| - \mu)^2$ becomes larger, and allows the user to control the importance of achieving exactly μ regions. If achieving exactly μ regions is critical, then the user can set $\sigma = 0$ and the only solution is $|\mathcal{R}| = \mu$.

The term $\psi_2(\mathcal{L}^C, C, \mathcal{R})$ is a potential that enforces SR contiguity, and is only non-zero when all $R_i \in \mathcal{R}$ have only one connected component. A connected component is one continuous portion of a spatial region. An SR has more than one connected component if it is comprised of multiple disjoint spatial regions. Formally, the contiguity potential is written as

$$\psi_2(\mathcal{L}^C, C, \mathcal{R}) = \begin{cases} 1 & \text{if } \prod_{i=1:|\mathcal{R}|} \Omega(R_i, C, \mathcal{L}^C) = 1, \\ 0 & \text{otherwise.} \end{cases}, \quad (3.6)$$

where Ω is a function that computes the number of connected components of R_i given C and \mathcal{L}^C . Further information about connected components can be found in [47].

Finally, the potential $\psi_3(B, \mathcal{L}^B, \mathcal{R})$ is the SR map saliency term, as defined in equation (3.2). Because each belief is considered independent, $\psi_3(B, \mathcal{L}^B, \mathcal{R})$ is

factored into K terms, and written as

$$\psi_3(B, \mathcal{L}^B, \mathcal{R}) = \prod_{k=1:K} \sum_{i=1:|\mathcal{R}|} \delta(i - \mathcal{L}_k^B) \int_{R_i} B_k dB_k. \quad (3.7)$$

3.3 The SERGE Algorithm

Solving equation (3.4) is combinatorial in K and N . The search can be efficiently performed by exploring portions of the solution space for which $\phi = \psi(B, \mu, \sigma, C, \mathcal{R}, \mathcal{L}^B, \mathcal{L}^C) > 0$. The SERGE algorithm (summarized in Algorithm 2) intelligently merges cells to efficiently generate a high saliency SR map. The SERGE algorithm is initialized with parameters C_{init} , $\mathcal{L}_{\text{init}}^C$, and $\mathcal{L}_{\text{init}}^B$, where $\mathcal{L}_{\text{init}}^B$ is a vector with K unique assignments, C_{init} is initialized to K contiguous regions with high saliency (each region has high saliency for one belief), and $\mathcal{L}_{\text{init}}^C$ is a vector with K unique assignments. The initial cells, C_{init} , are a best guess at the coarsest SR map, and can be generated in several ways; including ‘one vs one’ or ‘one vs all’ classifiers or Voronoi regions.

The SERGE algorithm ensures that potential ψ_2 is always positive, and thus does not search in regions of the solution space for which $\phi = 0$ when solving equation (3.4). Each iteration of SERGE simulates the result of merging a cell, C_n , with each of its neighbors, C_m , by calculating the value of the potential function, ϕ_m . The merge that results in the largest value of ϕ is selected. When two cells are merged, $C_m = C_m \cup C_n$, and a new region set \mathcal{R}_t is computed as in equation (3.3). At each iteration the number of cells is reduced by setting $C = \mathcal{R}_t$. This step simultaneously reduces the solution space and guarantees that $\psi_2 > 0$. Because the order in which C_n are chosen can influence the solution, the SERGE algorithm re-initialized and run over T trials, selecting C_n at random from C at

Algorithm 2: The SERGE algorithm

- 1: **Input** $C_{\text{init}}, \mathcal{L}_{\text{init}}^C, B, \mu, \sigma$ and $\mathcal{L}_{\text{init}}^B$
 - 2: set Best SR map Likelihood, $\phi_{\text{best}} = 0$.
 - 3: set Best SR map, $\mathcal{R}_{\text{best}} = C_{\text{init}}$
 - 4: **for** T trials **do**
 - 5: Set SR map $\mathcal{R}_t = C_{\text{init}}$
 - 6: Set $\mathcal{C} = C_{\text{init}}$
 - 7: Set $\mathcal{L}^B = \mathcal{L}_{\text{init}}^B$
 - 8: Set $\mathcal{L}^C = \mathcal{L}_{\text{init}}^C$
 - 9: **while** the \mathcal{R}_t is not converged **do**
 - 10: Randomly choose one $C_n \in \mathcal{C}$
 - 11: Set $\hat{\mathcal{L}}^B = \mathcal{L}^B$ and $\hat{\mathcal{L}}^C = \mathcal{L}^C$
 - 12: Compute $\phi_t = \psi(B, \mu, \sigma, C, \mathcal{R}_t, \mathcal{L}^B, \mathcal{L}^C)$
 - 13: **for** Each neighbor, C_m of C_n **do**
 - 14: **for** Each belief with assignment $\mathcal{L}_k^B = \mathcal{L}_n^C$ **do**
 - 15: Set $\hat{\mathcal{L}}_k^B = \mathcal{L}_m^C$
 - 16: **end for**
 - 17: Set $\hat{\mathcal{L}}_n^C = \mathcal{L}_m^C$
 - 18: Compute $\phi_m = \psi(B, \mu, \sigma, C, \mathcal{R}_t, \hat{\mathcal{L}}^B, \hat{\mathcal{L}}^C)$
 - 19: **end for**
-

```

20:     if  $\max(\phi_m) > \phi_t$  then
21:          $\phi_t = \max(\phi_m)$ 
22:          $\mathcal{L}^C = \hat{\mathcal{L}}^C$ 
23:          $\mathcal{L}^B = \hat{\mathcal{L}}^B$ 
24:         Compute  $\mathcal{R}_t$  from  $C$  and  $\mathcal{L}^C$  (equation (3.3))
25:          $C = \mathcal{R}_t$ 
26:     end if
27: end while
28: Compute  $\phi_t = \psi(B, \mu, \sigma, C, \mathcal{R}_t, \mathcal{L}^B, \mathcal{L}^C)$ 
29: if  $\phi_t > \phi_{\text{best}}$  then
30:      $\mathcal{R}_{\text{best}} = \mathcal{R}_t$ 
31:      $\phi_{\text{best}} = \phi_t$ 
32: end if
33: end for

```

each iteration. The best solution after T trials is picked as being close enough to the optimal SR map. The number of trials, T , must be large enough to allow the SERGE algorithm to explore the solution space well. A value of $T > 100 \cdot K$ is suggested.

3.3.1 Resolution Scaling

Since μ and σ guide the map resolution and SRs are formed via combinations of finer SRs, a multi-resolution SR map can be developed by iteratively applying the SERGE algorithm. A tree like structure (herein referred to as the resolution tree) is created to represent the multi-resolution SR map. The root of the resolution tree (level 0) is always a map with a single SR. Subsequent branches split the parent SR map based saliency and SRs generated in the parent. The final layer of the resolution tree is represented by K SRs, one assigned to each belief in the dataset (i.e. the initialization for the SERGE algorithm).

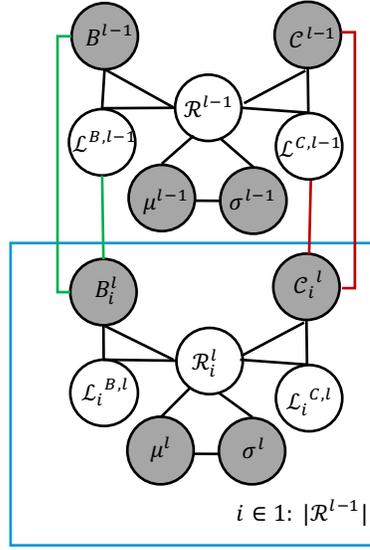


Figure 3.2: A graphical model showing the relationship between distributions in two levels of a resolution tree, $l, l - 1$. Two cliques (indicated by the connections shown in red and green) relate the two levels: $\{B^{l-1}, B_i^l, \mathcal{L}^{B,l-1}\}$ and $\{C^{l-1}, C_i^l, \mathcal{L}^{C,l-1}\}$.

A graphical model showing the relationship between variables at two levels of the resolution tree is shown in Figure 3.2. Superscripts refer to resolution tree level, $0 \leq l \leq L$, and subscripts refer to the parent SR R_i^{l-1} (the i^{th} SR in the previous tree level). The likelihood of the multi resolution solution, ϕ_{MR} , is the product of all the potentials in equation (3.4) and factors that relate variables between tree levels,

$$\begin{aligned} \phi_{MR} = & \prod_{l=0:L} \psi(B^l, \mu^l, \sigma^l, C^l, \mathcal{R}^l, \mathcal{L}^{B,l}, \mathcal{L}^{C,l}) \\ & \cdot \prod_{l=1:L} \prod_{i=1:|\mathcal{R}^{l-1}|} \psi_B(B^{l-1}, B_i^l, \mathcal{L}^{B,l-1}) \cdot \psi_C(C^{l-1}, C_i^l, \mathcal{L}^{C,l-1}). \end{aligned} \quad (3.8)$$

Two clique potentials are added in the factorization for $l > 0$. These are $\psi_B(B^{l-1}, B_i^l, \mathcal{L}^{B,l-1})$ and $\psi_C(C^{l-1}, C_i^l, \mathcal{L}^{C,l-1})$. These two potentials select beliefs and cells with the same assignment in the parent node, and are written as

$$\psi_B(B^{l-1}, B_i^l, \mathcal{L}^{B,l-1}) = \begin{cases} 1 & \text{if } B_i^l = \{B^{l-1} | \mathcal{L}^{B,l-1} = i\} \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

and

$$\psi_C(C^{l-1}, C_i^l, \mathcal{L}^{C,l-1}) = \begin{cases} 1 & \text{if } C_i^l = \{C^{l-1} | \mathcal{L}^{C,l-1} = i\} \\ 0 & \text{otherwise} \end{cases}. \quad (3.10)$$

where $C_i^l = \{C^{l-1} | \mathcal{L}^{C,l-1} = i\}$ indicates the set of cells in the parent node with assignment i and $B_i^l = \{B^{l-1} | \mathcal{L}^{B,l-1} = i\}$ indicates the set of beliefs with assignment i in the parent. This selection ensures that only beliefs and cells that are relevant to region R_i^{l-1} are considered when running SERGE to split R_i^{l-1} into child SRs.

Consider the following three layer resolution sample, shown in Figure 3.3. At the top (coarsest resolution), there is only one SR. Subsequent layers of the tree are created by splitting parent layers in a depth-first order. After the first layer, the SR map is split into two SRs by applying the SERGE algorithm with $\mu = 2$ to all cells and beliefs. Then, going down the left branch of the tree (blue), the SR on the left is further split into two SRs using $\mu = 2$. This process repeats until the desired resolution is achieved. Note that it is not always necessary to split using $\mu = 2$; one could choose a finer resolution for each level using $\mu > 2$. For subsequent use of the SR map by the robot, the SR map can be reconstructed at any level of the tree. Different resolutions in different branches of the tree can also be constructed (e.g. finer resolution green and blue on the left half of the SR map and coarser resolution magenta for the right side of the SR map).

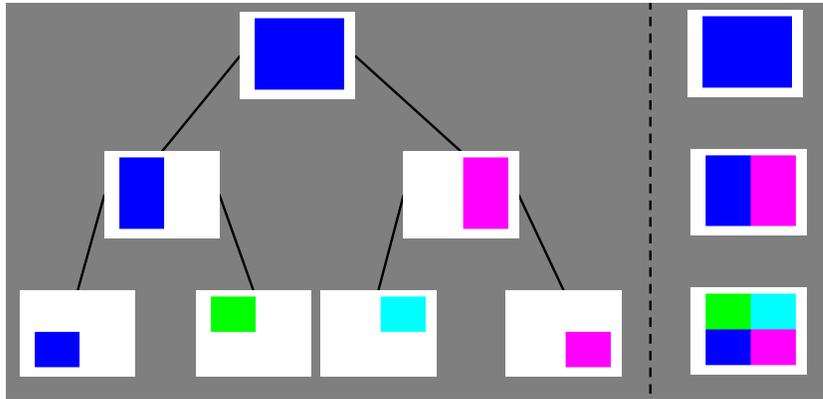


Figure 3.3: An example a multi resolution SR map. SRs are shown by the different colors. As shown on the left, at the top level the SR map has one SR; the second layer splits the SR. The third layer of the SR map is formed by splitting each parent SR into two additional SRs. The result is an SR map (at the finest resolution) with four SRs. The resulting full map at each layer of the tree is shown on the right.

3.3.2 Using SR maps for Qualitative Reasoning

Salient regions in SR maps can be used for reasoning with qualitative spatial relationships (QSR). In any QSR, objects in an environment are located in discrete regions, known as qualitative states, relative to a local coordinate frame [39]. SR maps may be used as QSR by learning an SR map on data that is transformed into relative, local coordinate frames, defining the salient regions as qualitative states, and developing a set of symbolic relationships that allow for reasoning within a QSR framework. The ternary QSR are considered here, where three landmarks (D , E , and F) are used to describe space; however the methods developed are easily extended to binary QSR.

In a ternary QSR, two landmarks establish a local coordinate frame, and the

third is localized in a qualitative state. The data are transformed into local coordinate frames by computing the transform such that the location of landmark D is $(0, 0)$ and the location of landmark E is $(0, 1)$. This transform is applied to all other landmarks that are observed simultaneously with D and E . For notational convenience, a coordinate frame in which landmark D is located at the origin and landmark E is located at $(0, 1)$ is referred to as ‘coordinate frame DE ’. In QSR, landmark F is located in a qualitative state (salient region) in coordinate frame DE . The qualitative state of F in coordinate frame DE is denoted as $\{DE : F\}$. SR maps for QSR are learned running the SERGE algorithm on beliefs over locations of landmark F in coordinate frame DE for all triplets of landmarks that are simultaneously observed. Beliefs for the k^{th} landmark triplet, \hat{B}_k^F , over locations of landmark F in coordinate frame DE are computed by applying the transform

$$\hat{B}_k^F(\hat{x}_F, \hat{y}_F) = \int_{x_D, y_D, x_E, y_E, x_F, y_F} B_k^D(x_D, y_D) B_k^E(x_E, y_E) B_k^F(x_F, y_F) \Gamma_k \frac{1}{\sqrt{(x_D - x_E)^2 + (y_D - y_E)^2}} [x_F - x_D, y_F - y_D]^T, \quad (3.11)$$

where $\Gamma_k = [\sin \theta_k \quad \cos \theta_k; \cos \theta_k \quad -\sin \theta_k]$ is the rotation matrix, $\theta_k = \text{atan} \left(\frac{y_D - y_E}{x_D - x_E} \right)$, and $B_k^D(x_D, y_D)$ represents the value of the belief of landmark D in triplet k at location (x_D, y_D) (the same notation is applied for landmarks E and F). The transformed beliefs, \hat{B}_k^F are used in the SERGE algorithm to generate a QSR with high saliency for qualitative mapping.

One of the most useful capabilities of QSRs is the ability to do efficient reasoning via a look up table of compose relationships. The compose operator uses two known relationships ($\{DE : F\}$ and $\{EF : G\}$) to reason about a third qualitative relationship ($\{DE : G\}$). An algorithm for computing a look up table for the compose operator with QSRs generated by the SERGE algorithm is presented

Algorithm 3: Compose Table Calculation

Input $\{DE : F\}$, $\{EF : G\}$ and a SR Map
for Each map cell, $c_i \in \{DE : F\}$ **do**
 Compute inverse transform, $\mathcal{T} c_i \rightarrow DE$
 for Each map cell, $d_i \in \{EF : G\}$ **do**
 Apply \mathcal{T} to d_i , resulting in \hat{d}_i
 Find cells in map that \hat{d}_i “lands in”
 Add label of these cells to $\{DE : G\}$
 end for
end for

in Algorithm 3. The algorithm inputs are an SR map and relationships $\{DE : F\}$ and $\{EF : G\}$. Given the relationship $\{DE : F\}$, landmark F can be anywhere in one salient region. All possible transformations between coordinate frames DE and EF given that F is located somewhere in $\{DE : F\}$ are computed. For each possible coordinate frame transformation, the set of possible locations of landmark G in $\{EF : G\}$ are mapped into coordinate frame DE . The compose look up table entry for all possible coordinate frame transformations and locations of $\{DE : G\}$ is the set of all salient regions in which G can reside in coordinate frame DE . The worst possible case run time for this algorithm is $O(N^2)$, where N is the number of grid cells in the SR map.

3.4 Experiments and Results

Two simulated datasets are constructed and an in vivo dataset is collected to examine the performance of the SERGE algorithm operating on beliefs generated from different types of data. SR maps are automatically generated for the given datasets. The number of SRs in the map, the saliency score (data likelihood), and the error rate are reported. An example of how the resolution tree can be used for robot navigation is presented.

3.4.1 The Data

The first dataset, shown in Figure 3.4, provides a baseline. Measurements, Z_k , from 49 unique landmarks are shown in different colors. This dataset is well separated, such that high likelihood regions of the beliefs, $B_k = p(X_k|Z_k)$, do not overlap. Beliefs are estimated from the data and assumed to be distributed normally as $B_k \sim \mathcal{N}(\alpha_k, \Sigma_k)$, where α_k is the mean and Σ_k is the covariance of data points, Z_k . This is a baseline dataset because the map saliency is guaranteed to be 1 due to the low noise relative to the separation of the landmarks.

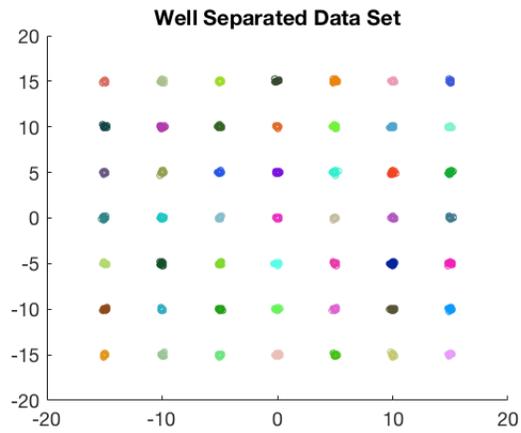
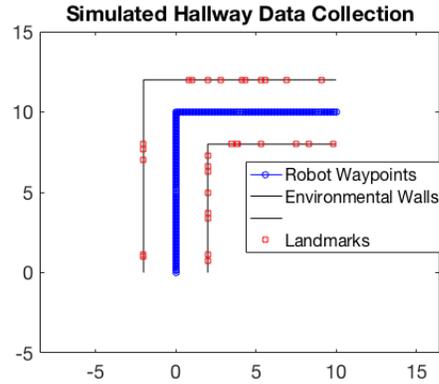
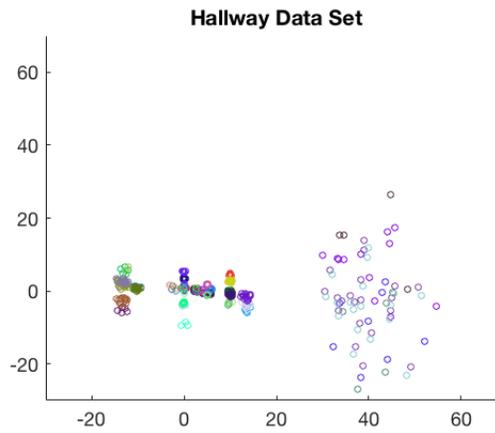


Figure 3.4: A well separated dataset where measurements from the different landmarks are shown in unique colors. The beliefs representing map landmarks have small variance relative to their spacing.



(a) The hallway environment with landmarks shown in red. The blue points show the robot trajectory.

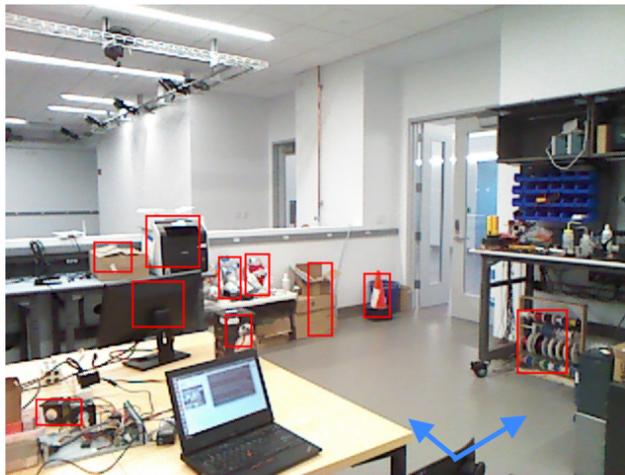


(b) The hallway dataset shown as triplets in the normalized coordinate (0,1) frame. The different triplet measurements are each assigned a random color.

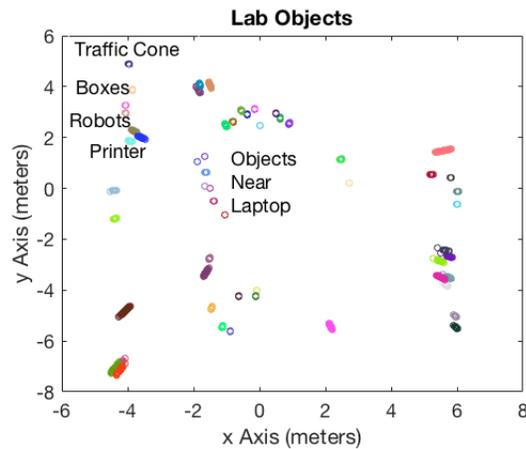
Figure 3.5: Data from a hallway simulation.

The second data set is used to demonstrate the automatic generation of SRs for qualitative mapping; a ternary calculus, similar to those in [39] is used. The second dataset is collected by simulating a robot driving down a hallway and observing point landmarks on the walls, as shown in Figure 3.5(a). A normal distribution with variance of 0.1m^2 is simulated for the measurements. Since this dataset is intended to show the utility of SR maps in qualitative mapping with

ternary relationships, measurements are observations of landmark triplets with known data associations, denoted as Z_k , where k is the triplet index. An observation, Z_k , is composed of three landmarks locations in 2-D (labeled D_k , E_k , and F_k), where $D_k = (x_D, y_D)_k$, $E_k = (x_E, y_E)_k$, and $F_k = (x_F, y_F)_k$, relative to the robot. Each observation is collected and processed into a normalized coordinate frame such that $\hat{D}_k = (0, 0)$, $\hat{E}_k = (0, 1)$, and $\hat{F}_k = \Gamma_k \frac{1}{\sqrt{(x_D - x_E)^2 + (y_D - y_E)^2}} [x_F - x_D, y_F - y_D]^T$, where $\Gamma_k = [\sin \theta_k \ \cos \theta_k; \cos \theta_k \ -\sin \theta_k]$ is the rotation matrix and $\theta_k = \text{atan} \left(\frac{y_D - y_E}{x_D - x_E} \right)$. The triplets converted into the normalized coordinate frame are shown in Figure 3.5(b), where the different colors represent measurements from different triplets. Note that \hat{D}_k and \hat{E}_k overlay $\hat{D} = (0, 0)$ and $\hat{E} = (0, 1)$; thus the scatter of points denote the locations of \hat{F}_k . The variance of the measurements, \hat{F}_k , is proportional to distance from the origin because the normalization amplifies noise. Each belief is a probability distribution over the location of \hat{F}_k in the normalized coordinate frame, or $B_k = p(\hat{X}_k | \hat{F}_k)$. The beliefs, B_k , are generated by sampling 10,000 points from a normal distribution with a variance of 0.1m^2 centered at each of (D_k^t, E_k^t, F_k^t) , where t denotes the t^{th} measurement of the k^{th} triplet. The samples are transformed into the normalized coordinate frame to yield 10,000 samples of \hat{F}_k . The belief, B_k , is represented as all of the samples \hat{F}_k .



(a) Examples of objects used as landmarks in the experiment. The red bounding boxes show the regions used to calculate the positions of the objects. The coordinate system is shown by the blue arrows on the bottom right of the figure.



(b) Objects from the lab plotted from an overhead view. A total of 55 objects are segmented from the data. Each object is plotted as a different color. Some of the data points are labeled to reference the picture.

Figure 3.6: The lab data acquired with a Kinect sensor. An example image (a) and a plot of the data (b) from an overhead view are shown.

The goal of the in vivo experiment is to examine the performance of the SERGE algorithm under realistic conditions for a robot. A Kinect sensor ob-

serves a lab environment for the in vivo study. Landmarks are extracted from the RGBD images by creating bounding boxes around objects and tracking those objects for several seconds. Examples of objects are shown in Figure 3.6(a). The 2-D data, Z_k , consist of the measured bounding box centroids for each object and are plotted from an overhead view as shown in Figure 3.6(b). For this plot, the coordinate frame is centered in the middle of the lab and pointing towards the doors (shown by the blue arrows in Figure 3.6(a)). The beliefs are assumed to be normally distributed and are estimated from the data, $B_k \sim \mathcal{N}(\alpha_k, \Sigma_k)$, where α_k is the mean and Σ_k is the covariance of the measurements, Z_k , from the k^{th} object.

For each dataset, the SERGE algorithm is evaluated in a 5-fold cross validation study where the data are separated into five disjoint sets and four of the sets are used as a training set to generate SR maps and one set is used as a testing set to evaluate the SR map in each validation trial. The metrics used to determine the quality of the SR maps produced are 1) the resulting data log likelihood as a measure of saliency and 2) error rate as a measure of accuracy on a test set. The data log likelihood is computed as $\log(\psi_3(B_{\text{test}}, \mathcal{L}^B, \mathcal{R}))$, where B_{test} is the set of beliefs generated from data points in a test set. This is used as a performance metric since the main purposes of creating SR maps is to maximize SR map saliency. The error rate is compute as $\frac{|Z_{\text{test}} \notin R_{\text{train}}^B|}{|Z_{\text{test}}|}$, where $|Z_{\text{test}}|$ is the total number of data points in the test set and $|Z_{\text{test}} \notin R_{\text{train}}^B|$ is the number of data points in the test set for belief B_k that do not fall within the same SR as the belief assignment learned from the training set.

The set of cells, C_{init} , for all tests is generated as follows. First, a fine grid of cells is generated, and the saliency (equation (3.2)) of each grid cell is computed for each belief. Grid cells are then assigned the belief that has the highest

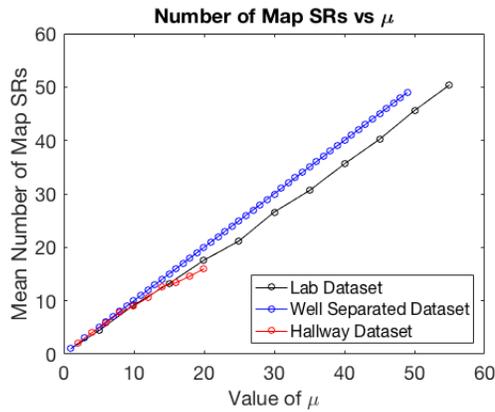
saliency. The cells in C_{init} are the union of all grid cells with the same assignment.

3.4.2 Evaluation of SR maps

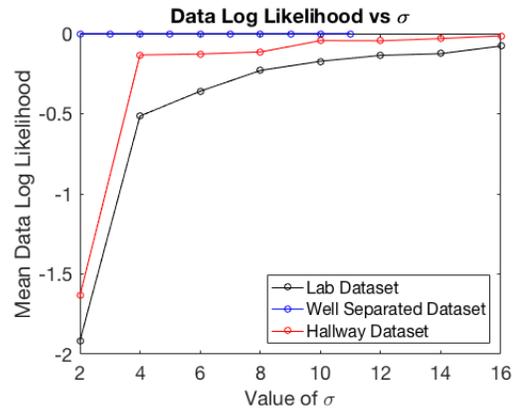
Two tests that study the effects of the user defined parameters, μ and σ , on the output of the SERGE algorithm are conducted for all three datasets: varying only μ and varying only σ . In one test, μ is varied from 1 to K for each dataset with $\sigma = 2$ fixed. In the other test, σ is varied from 2 to 16, and $\mu = 20$ is fixed. The results of both studies can be seen in Figure 3.7, which shows the total number of SRs in the final maps, the data log likelihood, and the error rate.

Figures 3.7(a,b) show the number of map SRs as a function of varying μ and σ . With increasing μ , the total number of SRs increases. However, whereas the well separated data set always shows $|\mathcal{R}| = \mu$, the lab and hallway datasets realize $|\mathcal{R}| < \mu$. This occurs because the $\sigma = 2$ parameter selection allows for fewer SRs in the case that the map saliency improves, and the hallway and lab datasets both have beliefs with high likelihood in similar locations. The well separated dataset achieves exactly the desired number of SRs because the map saliency does not increase if SRs are merged, due to spacing of the data.

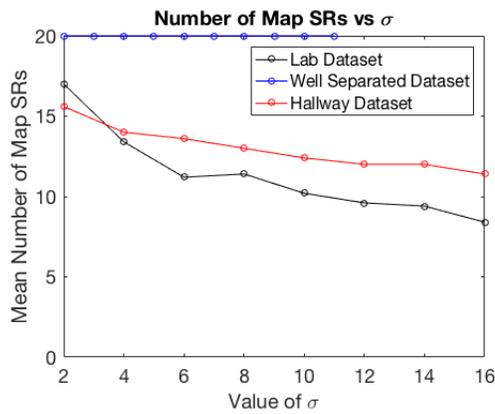
Figure 3.7(b) shows the result of varying σ with $\mu = 20$. The well separated data set SR maps have 20 SRs regardless of the value of σ because merging SRs does not increase the map saliency. This is supported by examining Figure 3.7(c,d), where the data log likelihood is 0 for all scenarios of the well separated dataset. The number of SRs in the lab and hallway SR maps decrease for increasing σ because larger σ allows for the map saliency to be a ‘more important’ factor in equation (3.4). Notice that the data log likelihood is higher when



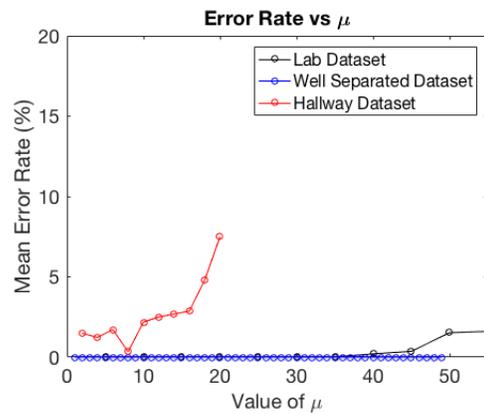
(a) The number of map SRs presented as a function of changing μ .



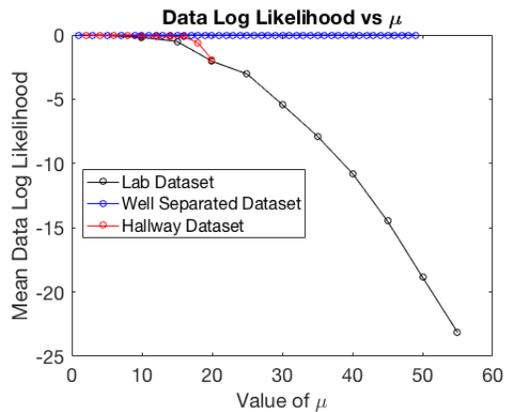
(d) The data log likelihood presented as a function of changing σ .



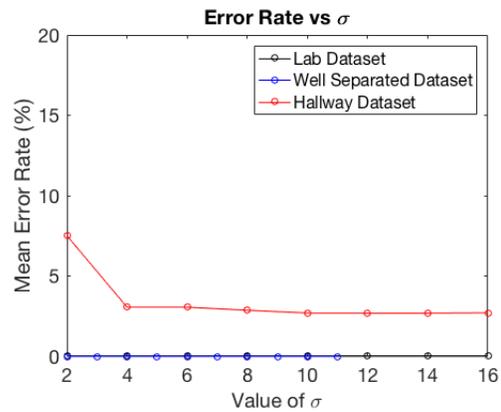
(b) The number of map SRs presented as a function of changing σ .



(e) The error rate presented as a function of changing μ .



(c) The data log likelihood presented as a function of changing μ .



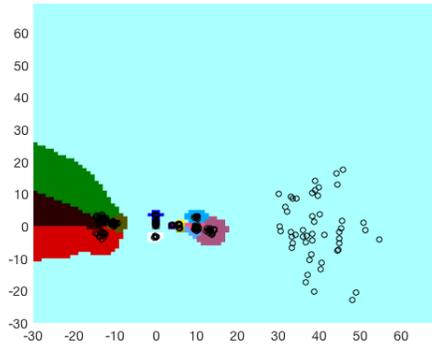
(f) The error rate presented as a function of changing σ .

Figure 3.7: Results from the 5-fold cross validation study on all datasets.

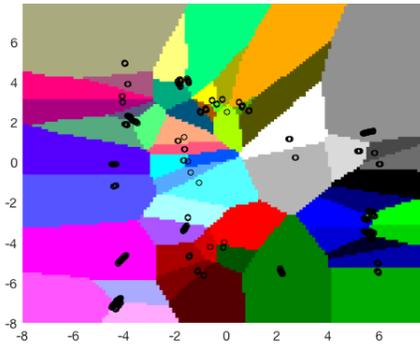
the map has fewer SRs.

Finally, Figures 3.7(e,f) plot the average error rates for the parameters tested. The error rate increases for the hallway and lab datasets as μ increases because a small σ prevents SERGE from merging cells to improve SR map saliency. The error rates reported for the hallway dataset are higher than other reported error rates due to the overlap of high likelihood belief locations that the algorithm is forced to categorize into different SRs. There are no errors in any test for the well separated dataset maps. Errors for the lab data set are less than 3% for all tests. As σ increases, the error rate decreases for the hallway dataset since regions merge to increase saliency.

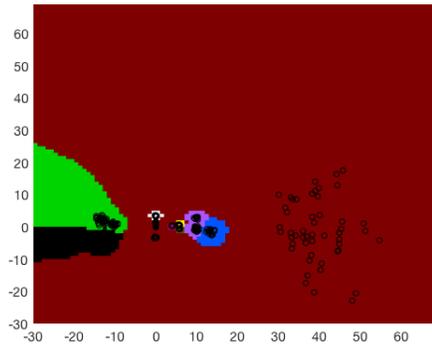
Examples of the generated SR maps for the hallway and lab datasets are shown in Figure 3.8; note that the well separated dataset yields SR maps that intuitively have the desired number of SRs, and are not shown. Figures 3.8(a,b) show two examples of the maps generated using the hallway dataset. A fine resolution SR map with 16 SRs (22 maximum possible SRs) is shown in Figure 3.8(a), and a coarser resolution SR map with 9 SRs is shown in Figure 3.8(b). Notice that several of the smaller SRs are merged when the coarser resolution SR map is generated. Examining the area near $(-30, 0)$ in Figure 3.8(a), 4 SRs describe the left region, but these are merged into 2 SRs with a lower μ as shown in Figure 3.8(b) since the data points are close. In both SR maps, the datasets that overlap highly (far right) are always included in one very broad SR because the data have a large variance, and are highly likely to reside anywhere in \mathbb{R}^2 . This can be further understood by examining the impact of sensor noise. Because the measurements for the landmarks on the far right have a higher variance, the likelihood of any one of these landmarks occupying any location in \mathbb{R}^2 is non-



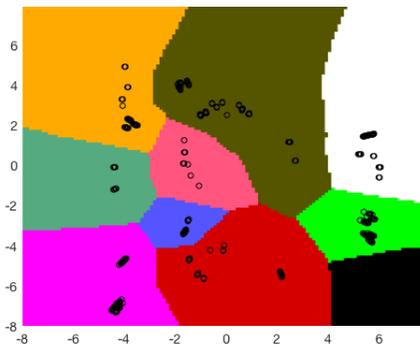
(a) A SR map generated with the hallway data and parameters $\mu = 18, \sigma = 2$. The resulting map has 16 SRs. Notice that the fine map details, near location $(0, 0)$, are well preserved.



(c) The finest map possible for the lab dataset. The map has 55 SRs, and is generated with $\mu = 55$ and $\sigma = 0.01$.



(b) SRs generated with the hallway data and parameters $\mu = 10, \sigma = 2$. Finer details near location $(0, 0)$ are merged, but details that distinguish between upper left and lower left are preserved near $(-30, 0)$.



(d) A coarser SR map for the lab environment comprised of 10 SRs, is generated using $\mu = 10$ and $\sigma = 2$.

Figure 3.8: Example SR maps for each dataset at two different parameter values. The unique colors represent the different qualitative regions. The black points are the data points (the entire dataset).

zero, whereas the landmarks in the center and on the left have distributions with small variance, and zero probability of occupying locations far away from the data. Therefore, the algorithm generates large SRs to encapsulate beliefs with high variance.

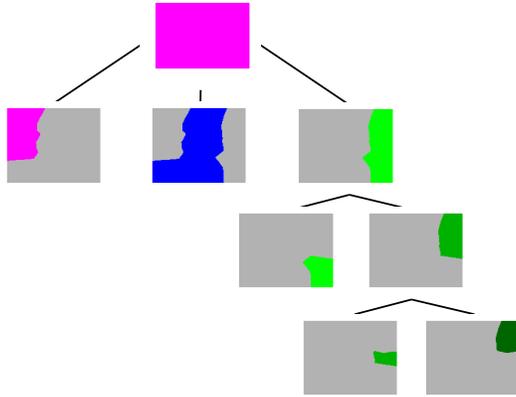
Figures 3.8(c,d) show examples of SR maps generated from the lab dataset. The 55 SRs shown in Figure 3.8(c) represent the finest SR map achievable with the lab dataset. Data points are close to the boundaries of some of these regions. Figure 3.8(d) shows a much coarser example of a SR map with only 10 SRs. The regions from Figure 3.8(c) where the data points are close to the boundaries are merged into the same SR. Notice that while beliefs that have close high likelihood regions are grouped into the same SR at finer resolutions, the SERGE algorithm combines beliefs that are not well clustered into the same SR to account for the smaller number of desired regions at coarse resolutions. However, the results are intuitive and create compact regions, as can be seen by examining the bottom left magenta SR in Figure 3.8(d). Four beliefs are assigned to one region, even though one is separated from the other three.

In practice, the value of μ should be chosen to generate an appropriate number of SRs for a given robotic task. If a robot only needs to reason about an environment at a coarse level, μ should be small. However, if a robot needs to reason about fine details or even individual beliefs, μ should be closer to K . The variable σ is chosen to prioritize saliency over fine detail. For example, the SR map shown in Figure 3.8(c) prioritizes resolution over saliency, and thus σ is set to 0.01. When σ is increased to 2 on this dataset (with $\mu = 55$) the number of regions decreases from 55 to 50, but the saliency (measured as a data log likelihood) increases from -28.1 to -22.6 .

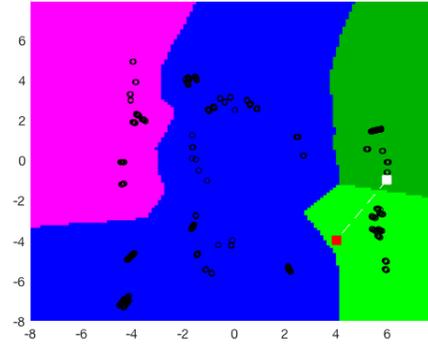
3.4.3 A Use Case for the Multi-Resolution SR Maps

A resolution tree and associated SR maps are constructed using the lab dataset; the use of these maps for robot planning is described. The goal of the robot is to navigate from a start point to a goal point located in one of the fine resolution regions. The planner efficiently generates a plan by only reasoning at the finest necessary resolution. First, it plans over the coarsest resolution map Figure 3.9(b), then over subsequent finer resolution maps, Figures 3.9(c,d). A partial resolution tree for the lab dataset is shown in Figure 3.9(a). The tree shows the finest resolution regions only for the branches on the right. The second level of the resolution tree is divided into three SRs by applying the SERGE algorithm with $\mu = 3$ and $\sigma = 1$ to all available beliefs and cells; the resulting SR map is shown in Figure 3.9(b).

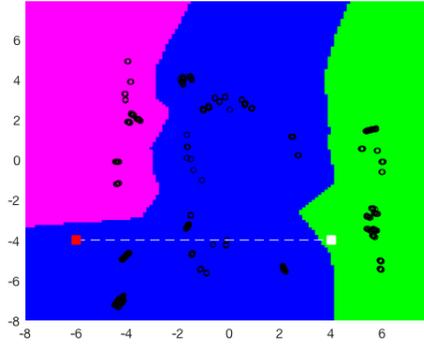
The specifics of the navigation scenario are as follows: a robot starts in the blue SR (red square) in Figure 3.9(b) and wishes to travel to a region corresponding to the ‘top right’ of the lab environment (green SR) in order to locate an object. The robot then navigates from the blue SR to the green SR (white square) shown in Figure 3.9(b). Using the coarse SR map in Figure 3.9(b), the robot needs only to reason over 3 SRs rather than up to 55 at the finest level of the tree. Once the robot reaches the white square shown in Figure 3.9(b), a finer resolution map of the green SR in Figure 3.9(b) is generated. The SERGE algorithm is applied to the SR with $\mu = 2$, $\sigma = 1$, which splits the region in two, as shown in Figure 3.9(c). The robot again navigates from the light green SR (red marker) to the dark green SR (white marker). This reduces the total number of map SRs that must be reasoned over from 10 to 4 at the third level of the tree. This process repeats for a third time, and the robot creates a fine representation of the



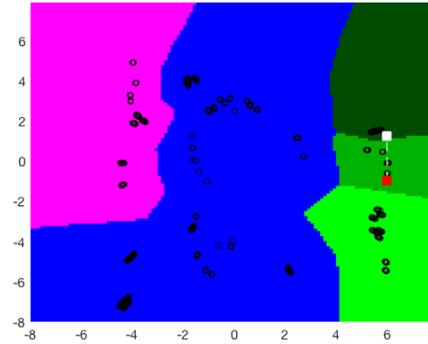
(a) A partial resolution tree generated for the lab dataset. The right side of the tree is developed, successively splitting green SRs with the SERGE algorithm.



(c) An example of the robot using fine information from only one portion of the resolution tree to navigate to an 'upper right' region.



(b) An example of a robot using the coarsest level of a resolution tree to navigate to the region labeled 'right'.



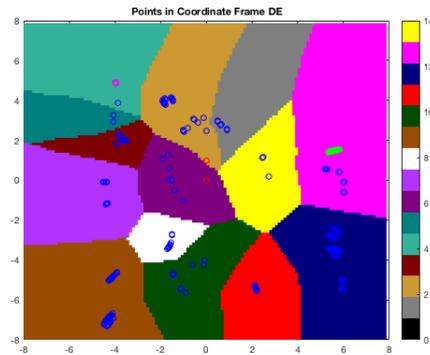
(d) An example of a robot further splitting the SRs until the 'topmost right' region in the workspace is available for navigation.

Figure 3.9: An example of a navigation scenario using a SR map and a resolution tree.

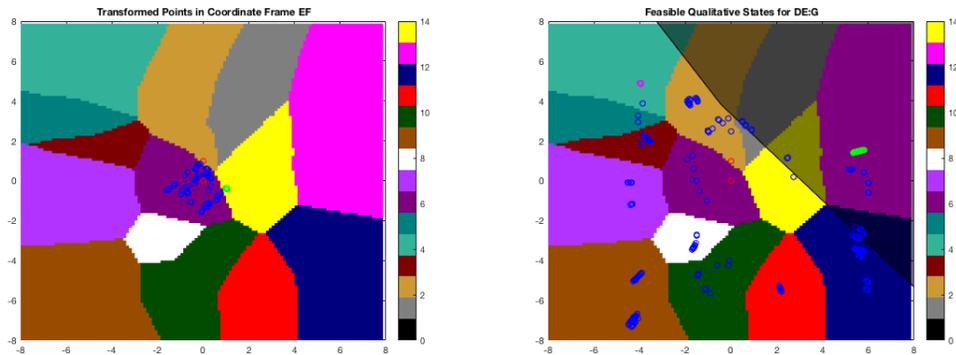
medium green SR, generating the representation shown in Figure 3.9(d). The robot generates the finest local representation in the green region (upper right where the object is located) of the map, and navigates to the darkest green SR. This strategy allows the robot to reason over a minimum number of necessary regions for planning, thus reducing computation time.

A Example of the Compose Operator

The compose operator for QSR generated by SERGE is shown for one of the maps created with the lab data. The map is shown in Figure 3.10(a) with all data in coordinate frame DE over-layed as circles. The coordinates of landmark D and landmark E are shown by the red points. The points highlighted in magenta are measurements of landmark F (the traffic cone seen in Figure 3.6(a)). Landmark G is shown by the green points. Consider the following scenario: only the relationships $\{DE : F\}$ and $\{EF : G\}$ are known, but $\{DE : G\}$ is not directly observed. The compose operator must be used to infer the set of possible locations of landmark G relative to D and E . The set of possible relationships is found by first finding the relationship $\{EF : G\}$, as shown in Figure 3.10(b). Then, considering all possible locations of G relative to E and F (all points in region 14), and all possible locations of F relative to D and E (all points in region 4), a set of constraints is computed to find the feasible qualitative states of $\{DE : G\}$. This is shown in Figure 3.10(c), where the shaded area represents the areas in which landmark G could be given the known relationships of $\{DE : F\}$ and $\{EF : G\}$. Notice that this means that states 1, 2, 4, 12, 13, and 14 are all feasible even though the constraints only pass through portions of some of these states.



(a) The lab data over-layed on an intermediate resolution map with 14 states. The coordinate frame shown in the normalized coordinate frame in which the measurements are acquired. The data points shown in magenta represent the landmark F and the data points shown in bright green represent the landmark G . It is assumed that the relationships $\{DE : F\}$ and $\{EF : G\}$ are known, but $\{DE : G\}$ must be inferred via the compose operator.



(b) The lab data shown in the coordinate frame of EF . The lime green points representing G are transformed, and are in qualitative state 14 relative to landmarks E and F .

(c) The possible positions of landmark D shadowed in on the map relative to D and E . Note that when finding this region, all possible points such that $\{DE : F\} = 4$ and $\{EF : G\} = 14$ are considered. Thus a large area is created, and the possible relationships for $\{DE : G\}$ include $\{1, 2, 4, 12, 13, 14\}$.

Figure 3.10: An example of the compose operator on the lab dataset. The possible relationships for $\{DE : G\}$ are shown given that $\{DE : F\} = 4$ and $\{EF : G\} = 14$

The volume of possible space in which landmark G can be relative to D and E depends both on the position of the qualitative states and on the map resolution. For instance, in the above example landmark G is can be anywhere in the upper right corner of the space given the known relationships. However, if instead $\{EF : G\} = 6$, then G can be almost anywhere in the entire space. If the map resolution is increased, the compose relationships may represent tighter constraints on possible locations of G . Consider a scaling tree where the maps shown in Figure 3.10 are an intermediate level. If any state is split into children states then the total compose relationships of the parent state is the union of the feasible regions of the children. Thus, a finer partition of space may result in tighter compose constraints.

3.5 Conclusion

The SaliEnt Region GEneration (SERGE) algorithm enables the automatic creation of SR maps from beliefs with two parameters that control the number of SRs and the map saliency. The SERGE algorithm can be used to define SRs for several types of data: distributions over landmark locations from SLAM algorithms, spatial distributions garnered from human input, relational measurements between sets of landmarks, or any other type of data from which a spatial distribution can be derived. The SERGE algorithm intelligently merges cells to create SR maps with high saliency, defined as the probability that a region contains objects from the environment. Experiments demonstrate a variety scenarios in which the SERGE algorithm can be used to generate SR maps. The SERGE algorithm generates intuitive, compact regions for each of these datasets at a resolution guided by the user. Qualitative spatial relationships that fit low-

level data well can be generated using the SERGE algorithm with data transformed into local coordinate frames. Results show that the SRs generated by the SERGE algorithm have high saliency and low error rates for classification (less than 7% worst case). A multi-resolution version of the SERGE algorithm is used to create multi-resolution SR maps that are useful for subsequent robotic applications such as planning. A presented example shows how the multi-resolution maps are used for robotic navigation.

CHAPTER 4

Q-LINK: A GENERAL PLANNING ARCHITECTURE FOR NAVIGATION WITH QUALITATIVE RELATIONAL INFORMATION

4.1 Introduction

Navigation using a map is an essential capability for any robot. There are many types of navigation and planning methods developed to enable robots to navigate over different map types, ranging from metrical to topological maps [48]. Methods such as rapidly exploring random trees [26], probabilistic roadmaps [18], the belief road map [40], and potential function methods [5] allow robots to efficiently plan over and navigate with metrical information derived from mapping algorithms, including the simultaneous localization and mapping (SLAM) algorithm [17]. Markov decision processes and partially observable Markov decision processes [48, 21] allow robots to generate plans over finite state representations of space such as occupancy grids [48]. Shortest path approaches such as Dijkstra’s algorithm and A* allow robots to efficiently generate plans for navigating over topological maps [5]. While the solution space of navigation and planning methods for metrical and topological maps is well developed, navigation methods for qualitative relational maps [7, 28, 32, 33, 44, 29, 53, 39] are not well explored and either rely on an additional topological map for planning [7, 28, 44] or are tied to one specific type of qualitative information [33]. This paper introduces a general approach to navigating in qualitative relational maps that is safe, generalizable to any type of qualitative information, and seamlessly incorporates map uncertainty into the planning architecture.

Qualitative relational maps (QRM) are a useful mapping paradigm in which

space is represented as a series of qualitative spatial relationships (QSR) in independent, local coordinate frames [7, 28, 32, 33, 44, 29, 53, 39]. QSRs are relationships between generalized landmarks, typically represented as a set of global visual orderings [7, 28], discrete states in local coordinate frames [32, 33, 44, 29, 53, 39], or topological relationships between pairs of landmarks [44, 41]. QRMs are highly adaptable and can use information about point-like or extended landmarks [7, 28, 32, 33, 44, 39]. Probabilistic sensor readings can be incorporated to generate a probabilistic QRM (PQRM) [39]. While qualitative maps are increasingly used to represent environments for robotic applications, a general method for navigating with QSR information does not yet exist. To date, navigation methods are either proposed for a specific QSR representation [33] or a supplemental topological map is constructed for navigation [7, 28, 44]. These approaches are limited because they only incorporate information from one type of QSR representation, may be robot-specific, and can not incorporate probabilistic information [7, 28, 33, 44].

A general planning architecture for navigating with qualitative information is developed; the approach allows a robot to reason directly about a QRM without a supplemental topological map. The qualitative linking planning architecture (Q-Link) uses three levels to interpret general QRM information while also generating low-level controls. The architecture levels are adaptable to QRMs with different QSR representations at the high level and different robot controls at the low level, allowing Q-Link to be tailored to many types of map. Q-Link can also be used with QRMs that contain multiple QSR representations. Since Q-Link does not rely on topological map information for navigation it can be used for environmental exploration. Q-Link is also unique because it incorporates uncertainty from controls, perception, and the map into the architecture; to

date, very few planning methods incorporate map uncertainty [34, 23, 11]. Path completion guarantees and trajectory length bounds are provided for environments with point landmarks. For more general settings path completion and path length of Q-Link are evaluated in simulation and in vivo experiments. The simulations study Q-Link performance in conditions when path completion is not guaranteed, and the in vivo experiments demonstrate Q-Link performance in a real-world environment with three distinct environmental conditions and a hybrid QRM that uses two QSR and both point and extended landmarks.

4.2 Qualitative Spatial Relationships and Qualitative Relational Maps

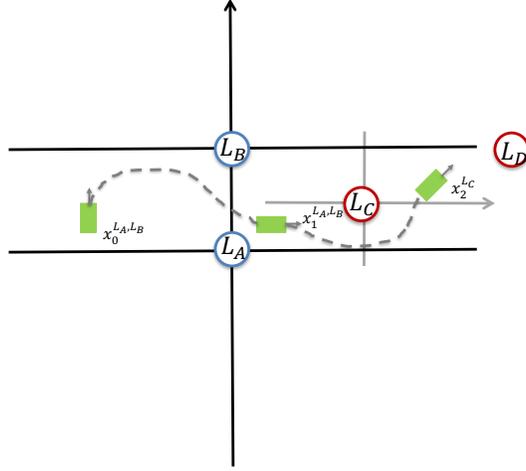


Figure 4.1: Two different QSR representations, the double cross [53] shown by the black solid lines, and a star QSR [27] shown by the gray solid lines. Qualitative states, q , are the regions of space separated by the lines. The blue landmarks are non-oriented point landmarks, and the red landmarks are oriented point landmarks. The positions of qualitative states are determined by \mathcal{L}^{ref} . For the double cross $\mathcal{L}^{\text{ref}} = \{L_A, L_B\}$ and for the star calculus $\mathcal{L}^{\text{ref}} = \{L_C\}$ in this example. A robot is shown by the green rectangle at three time instances with poses $x_0^{L_A, L_B}$, $x_1^{L_A, L_B}$, and $x_2^{L_C}$. The trajectory traveled is shown by the gray dashed line.

Qualitative spatial relationships (QSR) are a broad class of methods used to describe the relative positions of generalized landmarks (points or extended features) in an environment. QSRs partition space into discrete regions called qualitative states, which are either used to localize one landmark relative to other landmarks in the environment [32, 33, 44, 29, 53, 39] or to partition space into regions from which a specific clockwise or counter-clockwise visual ordering of landmarks is observed [7, 28]. This paper is primarily concerned with QSR rep-

representations that localize landmarks relative to other landmarks; however, the methods developed can easily be extended to QSR representations that partition space based on visual orderings.

Formally, QSRs are defined over landmark ‘sets’, denoted as \mathcal{L} . Each landmark set, \mathcal{L} , contains n landmarks, where n varies depending on the type of QSR representation used; for example, in a ternary QSR representation $n = 3$ [32, 33, 53], while in a binary QSR representation $n = 2$ [44, 29]. The specific landmarks in each set can be point-like features, extended features, point landmarks with an orientation feature. For all QSR representations, a landmark L_C is localized relative to a set of ordered reference landmarks, denoted as \mathcal{L}^{ref} and $|\mathcal{L}^{\text{ref}}| = n - 1$, which are used to define a local coordinate frame. Landmark L_C is localized relative to \mathcal{L}^{ref} in a qualitative state. The qualitative states, q , are regions of space that are either defined by constraint equations with parameters determined by \mathcal{L}^{ref} or topological relationships with respect to \mathcal{L}^{ref} . The qualitative state, q , of L_C relative to \mathcal{L}^{ref} is denoted as $\{\mathcal{L}^{\text{ref}} : L_C\} = q \in Q$, where Q is the set of all qualitative states in a given QSR representation. Generally, relationships between extended objects are represented by topological QSR representations which contain information such as ‘a landmark L_C is a tangential proper part of landmark L_B' ’ [44, 41], while relationships between point-like objects are represented by constraint-based QSR representations which contain information such as ‘a landmark L_C is to the left of oriented point landmark L_B' ’ [44, 27] or ‘a landmark L_C is to the right of the vector drawn between non-oriented point landmarks L_A and L_B' ’ [32, 33, 53].

Two example constraint-based QSR representations created with point landmarks are shown in Figure 4.1. The lines are constraints that divide space into

qualitative states, q . The blue point landmarks (L_A and L_B) are non-oriented and the red point landmarks (L_C and L_D) are oriented. The ternary QSR representation shown by the black lines with L_A and L_B is a double cross [53] and the binary QSR representation shown by the gray lines with landmark L_C is a star QSR [27]. For the double cross shown in Figure 4.1 $|Q| = 6$ and $|\mathcal{L}^{\text{ref}}| = 2$, while for the star QSR shown in Figure 4.1 $|Q| = 4$ and $|\mathcal{L}^{\text{ref}}| = 1$. Two landmarks are needed to establish a local coordinate system and define the positions of qualitative states for QSR representations with non-oriented point landmarks (L_A and L_B) while only one is necessary for oriented point landmarks (L_C and L_D). In the double cross example, $\mathcal{L}^{\text{ref}} = \{L_A, L_B\}$ and the relationship observed is $\{L_A, L_B : L_C\} = \text{'middle-right'}$. However, if the order is reversed and $\mathcal{L}^{\text{ref}} = \{L_B, L_A\}$, then the vector drawn between L_A and L_B flips and $\{L_B, L_A : L_C\} = \text{'middle-left'}$. The ordering of reference landmarks in the star example is not important because only one reference landmark is needed. For the remainder of this paper, the ordering of the reference landmarks in a ternary QSR representation is not considered to have an impact on planning because localization information for any ordering of \mathcal{L}^{ref} is either available or can be inferred via a unitary operator [33, 53, 27]. Therefore, for simplicity the reference landmarks are referred to as \mathcal{L}^{ref} without a specified ordering.

Qualitative relational maps (QRM) represent an environment as a set of recorded QSRs. Types of QRMs created in the literature include those that record constraint-based QSR relationships in local coordinate frames [33, 44, 39] and hybrid maps that record both topological and constraint-based QSR relationships [44]. Each QRM map entry, $M_i = \{\mathcal{L}_i, Q_i\}$, is composed of a landmark set, \mathcal{L}_i , and Q_i , the qualitative states of the landmarks for all possible ordered sets $\mathcal{L}_i^{\text{ref}} \subset \mathcal{L}_i$. Q_i is indexed so as to allow multiple types of QSRs in the map;

for example in a binary QSR $|\mathcal{Q}_i| = 2$ and in a ternary QSR $|\mathcal{Q}_i| = 6$. In a typical QRM, one entry in \mathcal{Q}_i is $\{\mathcal{L}_i^{\text{ref}} : (\mathcal{L}_i \setminus \mathcal{L}_i^{\text{ref}})\} = q_i$; in a PQRM [39] one entry in \mathcal{Q}_i is a probability distribution over qualitative states, q .

Localization in a QRM is performed locally. Let $x_t^{\mathcal{L}_i^{\text{ref}}} \in \mathbb{R}^N$, where N is the dimension of the workspace and t is the time index, be the pose of a robot in the coordinate frame established by landmarks in $\mathcal{L}_i^{\text{ref}}$. The robot is localized in a qualitative state, $\{\mathcal{L}_i^{\text{ref}} : x_t^{\mathcal{L}_i^{\text{ref}}}\}$, with respect to some $\mathcal{L}_i^{\text{ref}} \subseteq \mathcal{L}_i$ for a map element M_i . Let $\mathcal{L}^{\text{vis}_t}$ be the set of landmarks visible to the robot at time t . When $\mathcal{L}_i^{\text{ref}} \subseteq \mathcal{L}^{\text{vis}_t}$, the robot can estimate its qualitative state from measurements of $\mathcal{L}_i^{\text{ref}}$, and when $\mathcal{L}_i^{\text{ref}} \not\subseteq \mathcal{L}^{\text{vis}_t}$ the robot may use an estimator to track its qualitative state. A transition to a new local coordinate frame requires transitioning between localization $\{\mathcal{L}_i^{\text{ref}} : x_t^{\mathcal{L}_i^{\text{ref}}}\}$ and $\{\mathcal{L}_j^{\text{ref}} : x_t^{\mathcal{L}_j^{\text{ref}}}\}$ for the robot. While the new coordinate frame can be estimated even if $\mathcal{L}_j^{\text{ref}} \not\subseteq \mathcal{L}^{\text{vis}_t}$ for some t , here it is required that $\mathcal{L}_i^{\text{ref}} \subseteq \mathcal{L}^{\text{vis}_t}$ when a robot switches the coordinate frame in which it is localized.

4.3 The Q-Link Planning Architecture

Navigation with a QRM poses a unique challenge because all information about the environment is preserved as coarse relationships relative to local reference landmarks. The planning problem is defined over elements of a QRM, such that the robot starts at a map element M_{start} and ends at a map element M_{goal} . Navigating between M_{start} and M_{goal} requires the ability to transition between intermediate map elements (transitions between navigating with information in different coordinate frames). Transitions are enabled by qualitative links, and

a high level plan is generated from M_{start} to M_{goal} over the links. Given a high level plan from M_{start} to M_{goal} , a finite state controller and a low level controller allow the robot to travel using qualitative links and reach M_{goal} from M_{start} .

4.3.1 Qualitative Links

A transition between map elements M_i and M_j occurs when the robot has sufficient information to reason about qualitative states in Q_j . This requires that a set of reference landmarks from M_j is sensed by the robot, or $\mathcal{L}_j^{\text{ref}} \subseteq \mathcal{L}^{\text{vis}_t} \cap \mathcal{L}_j$ for one of the possible reference landmark sets $\mathcal{L}_j^{\text{ref}} \subset \mathcal{L}_j$. Link landmarks that lie within both landmark sets \mathcal{L}_i and \mathcal{L}_j , $L_{i,j}^\alpha \in \mathcal{L}_i \cap \mathcal{L}_j$, allow a robot to execute a transition between M_i and M_j . Links are landmarks that allow a robot to transition between coordinate frames defined by $\mathcal{L}_i^{\text{ref}}$ and $\mathcal{L}_j^{\text{ref}}$. The number of links necessary to complete a transition is typically one for a binary QSR representation and two for a ternary QSR representation; however, in some cases, the ternary QSR representation only requires one link if the robot can safely search for other landmarks in the environment.

Links are used as follows during navigation. If the link landmarks are not sensed by the robot at time t , $L_{i,j}^\alpha \notin \mathcal{L}^{\text{vis}_t}$, then the links, $L_{i,j}^\alpha$, are found by searching in qualitative state $\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\}$. When the link landmarks are visible, $L_{i,j}^\alpha \in \mathcal{L}^{\text{vis}_t}$, then either $\{L_{i,j}^\alpha\} = \mathcal{L}_j^{\text{ref}} \subset \mathcal{L}^{\text{vis}_t}$ and the robot has transitioned to M_j or the robot searches for an additional landmark in \mathcal{L}_j to find $\mathcal{L}_j^{\text{ref}}$ and transition to M_j .

To understand conceptually how the robot navigates using links, consider the example shown in Figure 4.1, where there are two map elements, M_1 and

M_2 , with corresponding landmark sets $\mathcal{L}_1 = \{L_A, L_B, L_C\}$ and $\mathcal{L}_2 = \{L_C, L_D\}$. The goal is to reach landmark L_D , starting from the position denoted by $x_0^{L_A, L_B}$ to the left of landmarks L_A and L_B . At the start, $\mathcal{L}^{\text{vis}_t} = \{L_A, L_B\}$, but L_C and L_D are not visible. The robot must transition from M_1 to M_2 to reach its goal. First, the robot computes $L_{1,2}^\alpha = L_C \in \mathcal{L}_1 \cap \mathcal{L}_2$. Then the robot determines that $\mathcal{L}_1^{\text{ref}} = \{L_A, L_B\}$ and locates L_C in $\{L_A, L_B : L_C\} = \text{'middle-right'}$ (the states are shown by the black lines). The robot drives along the trajectory shown by the gray dashed line in Figure 4.1 to position $x_1^{L_A, L_B}$ in qualitative state $\{L_A, L_B : L_C\} = \text{'middle-right'}$, and here $\mathcal{L}^{\text{vis}_t} = \{L_A, L_B, L_C\}$. When the robot is at position $x_1^{L_A, L_B}$, $L_C \in \mathcal{L}^{\text{vis}_t}$ the robot can transition to M_2 and use the information in M_2 to reach the goal. The robot defines $\mathcal{L}_2^{\text{ref}} = \{L_C\}$, and locates $\{L_C : L_D\} = \text{'top-left'}$ (qualitative states shown in gray). The robot then drives towards this qualitative state along the trajectory shown by the gray dashed line, ends at position $x_2^{L_C}$, and finds L_D .

4.3.2 The Q-Link Planning Architecture

Q-Link is a three level planning architecture that takes qualitative map information and generates a navigation strategy to enable a robot to navigate from a start to a goal. A diagram of the architecture is seen in Figure 4.2. The navigation strategy is planned at a high level over QRM elements and executed by lower level controllers that enable transitions between map elements. The three levels are the Link Graph level, the QSR level, and the Trajectory level. The Link Graph level contains a graph that is used to plan over links and pass qualitative state information, $\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\}$, to the QSR level. The QSR level contains a finite state controller that generates plans over qualitative states. The trajectory level takes a command, U , from the QSR level and interprets the command into con-

trols, u , the robot can execute to travel along a desired trajectory. Each of these are described in detail next.

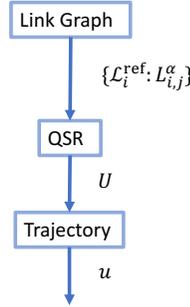


Figure 4.2: The Q-Link architecture, with three levels: the Link Graph level generates plans over a QRM, the QSR level reasons over qualitative states, and a Trajectory level generates low level controls, u , given commands, U , from the QSR level.

Link Graph level

The Link Graph level contains a Link Graph that shows connectivity between map elements. Formally, the Link Graph, \mathcal{G} , is a graph $\mathcal{G} = \{M, \alpha, C\}$ where M , the qualitative map elements, are the nodes, α are the edges, and C are the edge costs. An edge $\alpha_{i,j}$ exists between M_i and M_j in the Link Graph if $|\mathcal{L}_i \cap \mathcal{L}_j| \geq \beta$, where β is a threshold on the minimum number of links required to successfully transition between M_i and M_j . For any binary QSR representation $\beta = 1$, and for higher order QSR $\beta \geq 1$, depending on the requirements of the lower level controllers and restrictions of the robotic system.

The lowest cost path through the Link Graph is found by searching for the path, P^* , that starts at map element M_{start} , ends at map element M_{goal} , and has

minimum cost over the path,

$$P^* = \underset{P}{\operatorname{argmin}} \sum_{\alpha_{i,j} \in P} C_{i,j}. \quad (4.1)$$

This is a well studied problem, and can be solved efficiently using Dijkstra’s algorithm or A* [12]. Edge costs are assigned using one of several metrics: unit cost, expected path length to complete a transition, or the negative log probability of success for the transition. The unit cost prioritizes minimum map element transitions, while the expected path length prioritizes a minimum number of qualitative state traversals. The negative log probability of success prioritizes path completion regardless of path length. The choice of edge cost is best informed by the type of QRM and lower-level controllers used. For example, the negative log probability of success for transitions is a good choice of edge cost if the finite state controller used is a markov decision process or a partially observable decision process while unit cost is a good choice for edge cost if the system is deterministic.

The Link Graph level is used to plan over map elements M_1 and M_2 in the example given by Figure 4.1. At the start of the example shown in Figure 4.1, the robot is at location $x_0^{L_A, L_B}$ and only landmarks L_A and L_B are in its field of view. The robot defines its start as M_1 since it has a set of reference landmarks from M_1 in view. The robot uses the Link Graph, which has an edge between M_1 and M_2 since both map elements share L_C , to plan a path to any map element containing landmark L_D . The path generated is a two-hop path with map elements M_1 and M_2 .

QSR level

The QSR level contains a finite state controller that enables transitions from M_i to M_j by guiding the robot to qualitative state $\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\}$. Formally, a finite state controller is defined as a tuple for each map element, M_i , as $\Omega_i = (U_i, O_i, \psi_i, Q_i, \Gamma_i)$, where Q_i is the set of qualitative states located relative to $\mathcal{L}_i^{\text{ref}}$ in the environment, U_i is the available set of actions to the robot, $\psi_i \in U_i \times Q_i$ is the transition function between states, O_i is the set of observations the robot can make about the environment, and $\Gamma_i \in U_i \times Q_i$ is a reward function. The finite state controller parameters are flexible, and are defined based on the needs of the robotic system. For example, the states $q \in Q_i$ are the qualitative states that the robot can be located in relative to $\mathcal{L}_i^{\text{ref}}$. Let $x_t^{\mathcal{L}_i^{\text{ref}}} \in \mathbb{R}^N$ be the position of a robot at time t relative to $\mathcal{L}_i^{\text{ref}}$. The initial qualitative state of the robot is, $q_{\text{start}} = \{L_i^{\text{ref}} : x_0^{\mathcal{L}_i^{\text{ref}}}\}$. The observations, O_i , include a boolean variable indicating whether $L_{i,j}^\alpha \in \mathcal{L}^{\text{vis}_t}$ and an observation about the current robot qualitative state, $\{L_i^{\text{ref}} : x_t^{\mathcal{L}_i^{\text{ref}}}\}$. The reward function, Γ_i , has a small cost, such as -1 for all actions, U_i , that result in $q_t \neq \{L_i^{\text{ref}} : L_{i,j}^\alpha\}$ at time t , and high reward, such as 100 , for $q_t = \{L_i^{\text{ref}} : L_{i,j}^\alpha\}$. The transition function, ψ_i , and controls, U_i , are determined by the QSR representation used for every map element, M_i . The controls, U_i , direct a robot into any qualitative state in the QSR representation used in a map element, and the transition function, ψ_i , chooses which controls are feasible given that the state of a robot at time t is q_t . Controls are feasible only if the qualitative state that the robot is directed towards is adjacent to the current qualitative state. For example, in the double cross representation shown in Figure 4.1, a transition from ‘top left’ to ‘middle-left’ is feasible, but a direct traversal from ‘top left’ to ‘bottom left’ is not because the robot pass through at least one additional qualitative state. Any finite state controller may be used; however, if

there is uncertainty in the controls, perception, or the map then a markov decision process or a partially observable markov decision process is recommended [21].

The QSR level is used to provide the robot with commands for transitions to different qualitative states. At the first hop in the path shown by the example in Figure 4.1, the robot reasons over qualitative state information contained in map element M_1 . The QSR level directs the robot towards the ‘middle-right’ qualitative state relative to L_A and L_B using finite state controller Ω_1 . Once the robot is located in $\{L_A, L_B : x_1^{L_A, L_B}\}$ = ‘middle-right’ at time $t = 1$ and views landmark L_C the finite state controller changes to one that reasons over map element M_2 , Ω_2 . The states representing robot poses change to the qualitative states represented in M_2 , Q_2 and the robot receives new commands for the different qualitative representation. The robot is directed to a position $\{L_C : x_2^{L_C}\}$ = ‘top left’ and views landmark L_D once $\{L_C : x_2^{L_C}\}$ = ‘top left’ is reached.

Trajectory level

The trajectory level contains a low-level controller that takes commands, U_i , from the QSR level and moves a robot along a trajectory, $x_{t-1:t}^{\text{ref}}$, to achieve a pose, $x_t^{\text{ref}} \in \{q_t | q_{t-1}, U_i\}$, where $q_t \in Q_i$ denotes the qualitative state of the robot given the prior qualitative state $q_{t-1} \in Q_i$ and received command U_i . The interpretation of commands from the finite state controller influences the outcome of Q-Link navigation plans and depends on the environment. In an environment with good visibility with a robot that has a large field of view, a command may be interpreted as sweeping the sensor over the qualitative state from one position, $x_t^{\text{ref}} \in q_t$, in the environment. Another interpretation is that the robot moves

the closest point that is in qualitative state q_i . Finally, in sparse environments or environments in which the sensor has a limited field of view, the robot may do a ‘safe search’ to cover as much of the qualitative state as possible using methods such as [13, 4].

The trajectory level enables the robot to execute commands from the QSR level such as ‘drive to the middle-right qualitative state’ as shown by the example in Figure 4.1. Starting at initial position $x_0^{L_A, L_B}$, the robot receives a command from the QSR level to drive to a position such that $\{L_A, L_B : x_1^{L_A, L_B}\} = \text{‘middle-right’}$. The robot executes the trajectory shown by the gray dashed line and ends at position $x_1^{L_A, L_B}$ in the ‘middle-right’ qualitative state. Once the robot views L_C the QSR level controller switches to a controller for the star calculus, and the trajectory level switches coordinate frames to localize the robot relative to oriented landmark L_C . Once the robot receives command to make $\{L_C : x_2^{L_C}\} = \text{‘top left’}$, the robot executes the trajectory shown by the gray dashed line in Figure 4.1 and ends at point $x_2^{L_C}$.

4.3.3 Incorporation of Probabilistic Information

Uncertain information from controls, perception, and the QRM are seamlessly incorporated into the Q-Link planning architecture. The finite state controller in the QSR level uses probabilistic information to generate controls with high probability of reaching a local goal. The link level incorporates probabilistic information into the edge costs of the Link Graph so that the total cost of a path reflects the probability of successful path completion.

The finite state controller in the QSR level naturally incorporates uncer-

tainty from controls and perception if a partially observable markov decision process (POMDP) [48, 21] is used. The POMDP reasons over probabilities of the system state to yield controls that result in a maximum expected reward. Formally, a POMDP for every map element, M_i , is defined as a tuple $\Omega_i = (U_i, O_i, \psi_i, X_i, \Gamma_i, \gamma_i, \Pi_i, \tau_i)$, where O_i are the set of observations the robot can make about its environment, U_i are the set of controls available, $\psi_i \in X_i \times U_i$ is a probabilistic transition function between states, Γ_i is a reward function, X_i are the discrete environmental states, Π_i is a set of conditional observation probabilities, $\gamma_i \in [0, 1]$ is an optional discount factor that trades off between immediate rewards and a greater sum of reward in the future, and τ_i is an optional time horizon.

If there is map uncertainty and the qualitative state $\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\}$ is unknown, the POMDP is formulated as a search problem where the goal is to localize $L_{i,j}^\alpha$ and have $\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\} = \{\mathcal{L}_i^{\text{ref}} : x_i^{\mathcal{L}_i^{\text{ref}}}\}$. The states are defined as $X_i \in Q_i \times Q_i$, where Q_i is the set of qualitative states in the QSR representation used in map element M_i . The reward function, Γ_i has high reward for all states $\{\mathcal{L}_i^{\text{ref}} : x_i^{\mathcal{L}_i^{\text{ref}}}\} = \{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\}$ and a small cost otherwise. When a transition into map element M_i is made the POMDP is initialized as follows. First, the robot estimates the probability of being in a qualitative state, $p(\{\mathcal{L}_i^{\text{ref}} : x_i^{\mathcal{L}_i^{\text{ref}}}\} = q)$. This can be done using the methods developed in [39]. Then, the robot retrieves the probability distribution over qualitative states of $L_{i,j}^\alpha$, $p(\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\} = q')$, from the PQRM. The initial probability distribution over robot and local goal landmark states X_i is the joint distribution over the robot state and the state of $L_{i,j}^\alpha$, $p(\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\} = q', \{\mathcal{L}_i^{\text{ref}} : x_i^{\mathcal{L}_i^{\text{ref}}}\} = q)$. Given this initial distribution, the robot computes the optimal control, U_i^* , given the POMDP policy and sends U_i^* to the low-level controller in the Trajectory level. While POMDP control policies are often computationally expensive to

calculate, approximate methods exist to compute policies for large state spaces [24]. However, the state space of a QSR is typically small, and an exact solution for a POMDP can be computed in a reasonable amount of time [48, 24].

The Link Graph further supports the incorporation of probabilistic QRM information in the edge costs of the graph. Given a POMDP policy and $p(\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\} = q', \{\mathcal{L}_i^{\text{ref}} : x_0^{\mathcal{L}_i^{\text{ref}}}\} = q)$, the expected probability of success can be computed for each edge transition. Let $S \in X_i$ be the set of states for which $\{\mathcal{L}_i^{\text{ref}} : x_t^{\mathcal{L}_i^{\text{ref}}}\} = \{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\}$ and $p(S_i|U_{i,t}, X_{i,t-1})$ be the probability at time t , of the robot reaching some state in S_i given control $U_{i,t}$ and previous state $X_{i,t-1}$. Also, let $p(O_i|X_{i,t})$ be the probability of observing $L_{i,j}^\alpha$ in state $X_{i,t}$. The probability of the robot both reaching a goal state and observing $L_{i,j}^\alpha$ at time t is

$$p_{i,j}^{\alpha,t} = \sum_{X_{i,t-1}^k \in X_i} \sum_{s \in S} p(s|U_{i,t}, X_{i,t-1}^k) p(O_{i,t}|s) p(X_{i,t-1}). \quad (4.2)$$

The probability of a robot finding $L_{i,j}^\alpha$ between times t and τ_i with $0 \leq t < \tau_i$ is computed recursively as

$$p_{i,j}^{\alpha,t:\tau_i} = p_{i,j}^{\alpha,t} + (1 - p_{i,j}^{\alpha,t}) p_{i,j}^{\alpha,t+1:\tau_i}. \quad (4.3)$$

The probability $p_{i,j}^{\alpha,0:\tau_i}$ is the probability that the robot successfully transitions between M_i and M_j within a time horizon of τ_i . This probability is incorporated into the Link Graph as $C_{i,j} = -\log(p_{i,j}^{\alpha,0:\tau_i})$. The total success probability of any path through the environment is

$$p_{\text{success}} = \exp\left(-\sum_{\alpha_{i,j} \in P} -\log(p_{i,j}^{\alpha,0:\tau_i})\right), \quad (4.4)$$

where $\alpha_{i,j} \in P$ are edges in the Link Graph path, P . This edge cost prioritizes path completion over distance traveled. If distance traveled during a transition between two map elements is more important than probability of path comple-

tion, then the expected number of qualitative state transitions can be computed using information from the POMDP and used as an edge cost in the Link Graph.

4.3.4 Path Completion Guarantees

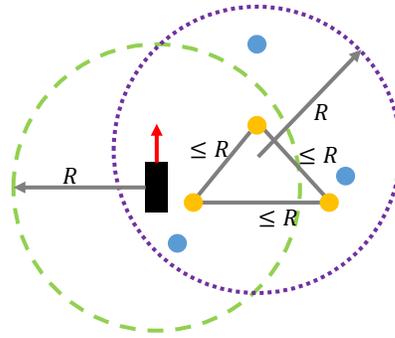


Figure 4.3: An example of the two scenarios for which path completion is guaranteed. The robot is represented by the black rectangle, and its field of view is represented by the dashed green line. The landmarks for two scenarios are presented (yellow and blue) and are all contained in a circle with radius R (purple dotted line). Landmarks for the general scenario are shown in blue, and landmarks for the more restricted scenario are shown in yellow.

Path completion guarantees and distance bounds are provided for environments that have point-like landmarks and no occlusions. More specifically, the path completion guarantees include guarantees that a robot can transition from any M_i to any M_j without having $\mathcal{L}^{\text{vis}_t} = \emptyset$ at any time t . The following additional assumptions must be made. A robot must have perfect landmark recognition and sensor range $R' \geq R$, where R is the sensor range of the robot that mapped the environment and R' is the sensor range of the robot navigating in the environment. This ensures that the robot never leaves the QRM mapped space. Two

cases are considered: 1) the general case that results from a robot recording a QRM entry for all landmark sets it observes, and 2) a more restricted case that results from applying a distance threshold between landmarks in a set. Both scenarios are shown in Figure 4.3.

The most general scenario (blue landmarks in Figure 4.3) occurs naturally while mapping an environment with a robot that has an omnidirectional sensor with radius R . In this scenario, the following assumptions are met.

Assumptions 1 (i) $\forall \mathcal{L}_j$ in the QRM, $\exists x \in \mathbb{R}^N$ s.t. $d(\vec{L}_n, x) \leq R \forall L_n \in \mathcal{L}_j$.
(ii) Let $\chi = \{x \in \mathbb{R}^N \mid d(x, \vec{L}_n) = R\}$ then, $\exists x_{0:T}$ s.t. $\chi \subseteq x_{0:T}$ and $\forall x \in x_{0:T} d(\vec{L}_n, x) \leq R$

where $d(\cdot)$ is the euclidean distance function, \vec{L}_n is the point location of landmark L_n , and $x_{0:T}$ is a robot trajectory between times 0 and T . All landmarks in any landmark set \mathcal{L}_j must be contained in a ball of radius R at some point in space. Additionally, there must exist some trajectory where all points on a ball of radius R centered at some landmark $L_n \in \mathcal{L}_j$ are visited and at no point on the trajectory L_n is not sensed by the robot.

Proposition 1 If Assumption 1 holds, then for some $x_t \in x_{0:T}$ and $L_k \in \{\mathcal{L}_j \setminus L_n\}$, $d(x_t, \vec{L}_n) = R \wedge d(x_t, \vec{L}_k) \leq R \rightarrow \{L_n, L_k\} \subseteq \mathcal{L}^{\text{vis}_t}$.

A robot can locate any landmark $L_k \in \mathcal{L}_j$ from any other $L_n \in \mathcal{L}_j$ by driving a trajectory that includes all points distance R from L_n . If L_n is a link, $L_{i,j}^\alpha$, then $\{L_n\} = \mathcal{L}_j^{\text{ref}}$ for a binary QSR representation and $\{L_n, L_k\} = \mathcal{L}_j^{\text{ref}}$ for a ternary QSR representation. Any map transition can be made, then, with one link landmark. This is guarantee of path completion for any QSR with any $\beta \geq 1$ required for

an edge in the Link Graph. Additionally, if $\mathcal{L}_i^{\text{ref}} \subseteq \mathcal{L}^{\text{vis}_i}$, the robot can locate $L_{i,j}^\alpha$ by driving a trajectory that includes an arc of radius R in $\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\}$. The path can be restricted to a circle of radius R if the following assumption holds.

Assumptions 2 Let $\chi = \{x \in \mathbb{R}^N \text{ s.t. } d(x, \vec{L}_n) = R\}$, $\exists x_{0:T} \text{ s.t. } \chi = x_{0:T} \wedge D(x_{0:T}) = 2\pi R$

where $D(\cdot)$ is a function that returns the length of a trajectory. This assumption restricts the trajectory from Assumption 1 to the trajectory that only includes points on a circle of radius R , centered at L_n .

Proposition 2 Provided that Assumptions 1 and 2 hold, the worst case trajectory length for one map element transition is $(2\pi + 1)R$, and the worst case trajectory length for $|P|$ transitions is $(2\pi + 1)R|P|$.

If the trajectory in Assumption 2 can be executed by the robot, then the worst case distance it will travel for one transition on a path is $(2\pi + 1)R$. In the worst case the robot must drive a distance R to reach an initial point on the circle of radius R then travel the entire circle.

While this set of guarantees represents the most general scenario a robot may encounter while mapping, the trajectories generated may be circuitous, and reasoning over qualitative states may not always be feasible. Therefore, guarantees are also provided in a more restrictive case (yellow landmarks shown in Figure 4.3) that allows a robot to bypass the finite state controller and navigate by driving directly between landmarks in the environment. In this case, the following assumption must hold.

Assumptions 3 (i) $\forall \mathcal{L}_j$ in the QRM, $d(L_k, L_n) \leq R \forall L_k, L_n \in \mathcal{L}_j$.

(ii) $\exists x_{0:T}$ s.t. $\vec{L}_k \in x_{0:T} \wedge \vec{L}_n \in x_{0:T}$ and $\forall x_t \in x_{0:T} d(\vec{L}_n, x_t) \leq R \wedge d(\vec{L}_k, x_t) \leq R \forall L_k, L_n \in \mathcal{L}_j$.

This assumption restricts the pairwise distance between any two landmarks in a landmark set to R or less. Enforcing this assumption requires some additional processing when building a QRM. Additionally, the assumption that the robot can drive a trajectory from landmark L_n to landmark L_k without L_k leaving the robot's field of view is made.

Proposition 3 If Assumption 3 holds, then $x_t = \vec{L}_n \rightarrow \mathcal{L}_j \subseteq \mathcal{L}^{\text{vis}_t} \forall L_n \in \mathcal{L}_j$. Furthermore, $\forall x_t \in x_{0:T} \{L_n, L_k\} \subseteq \mathcal{L}^{\text{vis}_t}$.

The robot can view all $L_n \in \mathcal{L}_j$ from any $L_k \in \mathcal{L}_j$. Furthermore, at any point along the trajectory $x_{0:T}$ both L_n and L_k are visible. Supposing the robot can drive a straight line trajectory, a distance bound is provided.

Assumptions 4 Let $\chi = \{x_{0:T} \in \mathbb{R}^N \text{ s.t. } x_0 = \vec{L}_n \wedge x_T = \vec{L}_k\}$, $\exists x_{0:T} \text{ s.t. } \chi = x_{0:T} \wedge D(x_{0:T}) = R$

Proposition 4 Let $|P|$ be the Link Graph path length for a plan over a QRM and assumptions 3 and 4 hold. Over a series of $|P|$ transitions, the robot will then travel at worst a length of $|P|R$.

If a robot travels in straight lines between landmarks (via Assumptions 4), the worst case distance for one transition is R . For robots that are not able to drive perfectly straight lines, other bounds may be provided given the dynamics of a

robot and guarantee that a trajectory from which two landmarks L_n and L_k are always visible can be executed.

The guarantees and bounds are provided for two specific environmental scenarios. In general, the visibility assumptions are violated because objects in the physical world provide occlusions and the sensors may not always detect landmarks that are in view. Therefore, having a finite state controller and a low level controller that increase the robustness of the system is essential. Contingency plans that re-plan at the Link Graph level may also increase the robustness of Q-Link.

4.4 Simulation and Experimental Result and Analysis

Q-Link is evaluated in simulation and in an in vivo robot experiment to study the performance under realistic conditions when path completion can not be guaranteed. Paths are evaluated for trajectory length and path failure rate. The failure rate is used as a metric since the goal of any navigation routine is to ensure success. Trajectory length is reported as a function of Link Graph path length to determine whether length bounds hold in the absence of completion guarantees.

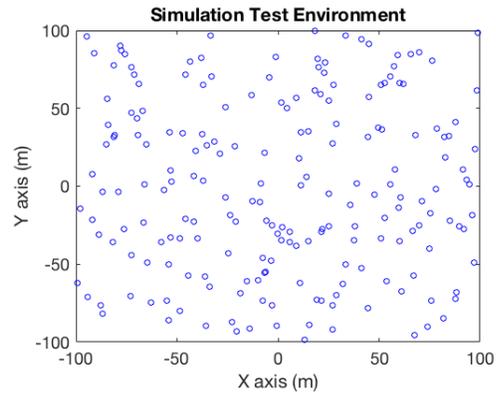


Figure 4.4: A $200\text{m} \times 200\text{m}$ test environment for Q-Link. Landmarks are indicated by the blue circles.

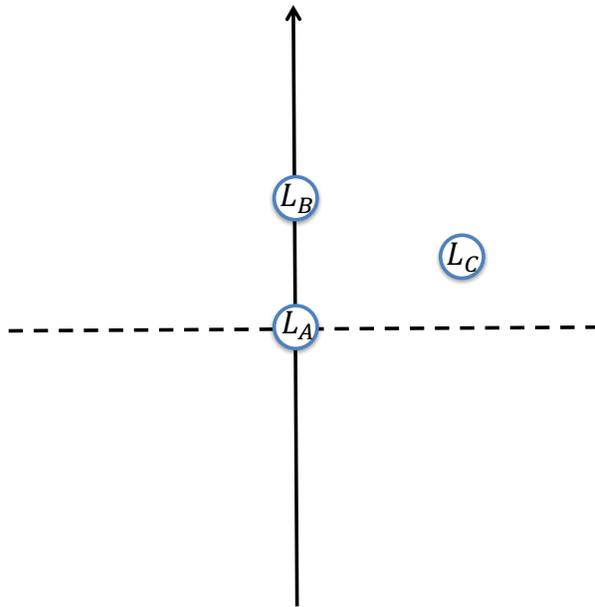


Figure 4.5: Two QSR representations defined with non-oriented point landmarks that are used for testing. The first, represented by both the solid and dashed lines, partitions space into quadrants. The second, represented by the solid line only, partitions space into left and right halves.

4.4.1 Simulation Studies

A test environment with 200 point landmarks (shown in Figure 4.4) is generated to test the Q-Link planning architecture in scenarios when path completion is not guaranteed. A PQRM is developed by simulating a robot with a sensor range of $R = 20\text{m}$ driving the environment with measurement noise modeled as $\mathcal{N}(0, 0.3)$. First, the path failure rate is analyzed as a function of insufficient sensor range by testing a decreasing robot sensor range from R to $0.6R$. In a second study, the path failure rate is analyzed as a function of landmark occlusion by simulating an environment where the landmarks have size ranging from 0.1m to 0.7m . In the occlusion studies, a landmark, L_A , is only recognized if no portions of another landmark, L_B , occlude or partially occlude L_A . A total of 40,000 trials were run for each study, and every pairwise combination of start landmark and goal landmark was evaluated.

The specifics of the studies were as follows: the map used is a PQRM with the quadrant QSR representation, shown in Figure 4.5 (includes the dashed line); the Link Graph has edges between map elements for which $|\mathcal{L}_i \cap \mathcal{L}_j| = 1$ and the edge costs are unit cost to ensure a minimum Link Graph path length, $|P|$. The finite state controller is a POMDP that produces controls, U_i , that direct the robot into qualitative states, $\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\}$. The low level controller drives an arc of radius R in the qualitative state specified by U_i and searches for reference landmarks in a circle of radius R when necessary. A trajectory is considered failed if the robot cannot successfully transition between any map elements M_i and M_j between M_{start} and M_{goal} on the path. A transition is successfully made if the reference landmarks in map element M_j are visible at some time, or $\mathcal{L}_j^{\text{ref}} \subseteq \mathcal{L}^{\text{vis}_t}$ at some time t . The paths are evaluated for completion and trajectory

length upon completion.

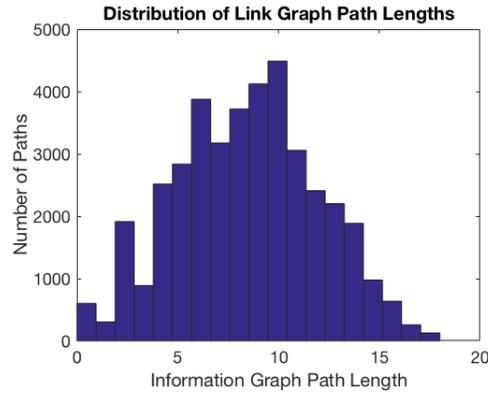


Figure 4.6: A histogram showing the distribution of Link Graph path lengths for the Monte Carlo simulations.

A histogram of Link Graph path lengths for the baseline trial when $R' = R$, there are no occlusions, and path completion is guaranteed is shown in Figure 4.6. Each histogram bin represents the total number of samples for Link Graph path length in the Monte Carlo trials. The majority of the paths have $|P| \leq 10$, and the path lengths are distributed approximately Gaussian. This is primarily a function of the environment size and the sensor range used to map the environment. The paths for which $|P| = 1$ are paths where the goal is visible from the start of the path.

Trajectory length as a function of Link Graph path length for each set of trials is shown in Figure 4.7 along with the distance bound provided by Proposition 2. Trajectory length increases with increasing Link Graph path length in the sensor range trial and the occlusion trial for all sets of parameters. The distance bound provided (shown by the dashed line) is very conservative, especially for paths with $|P| = 18$, where the bound on trajectory length is 1000m longer than the executed trajectories. The bound is conservative with a worst case transition

bound of $2\pi R + R$. For the quadrant QSR representation, a robot typically drives an arc of length $\frac{\pi}{2}R$ when searching in a qualitative state, and at worst another R to transition to the next map element. It is uncommon that all qualitative states must be searched. In the sensor range trials, the trajectory lengths increase with decreasing sensor range. This occurs because the robot must search farther to view landmarks before transitioning between map elements successfully. When the sensor range is larger, the robot views more landmarks and can make transitions more easily. Trajectories in all occlusion trials are longer than trajectories in the sensor range trials because the robot must search for a location where $L_{i,j}^\alpha$ is not occluded when transitioning between map elements, and this search may lead the robot to visit more than one qualitative state.

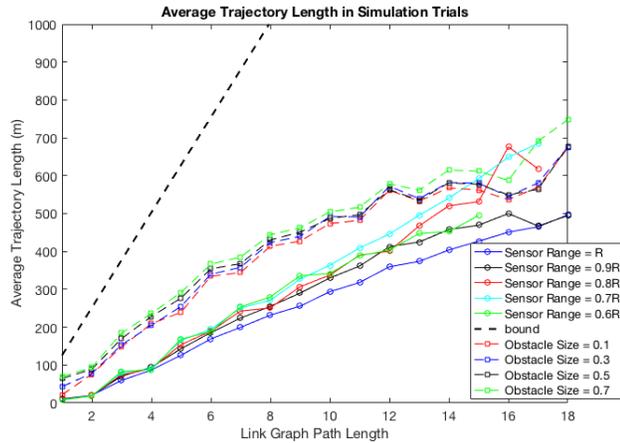


Figure 4.7: Average trajectory length (m) reported as a function of Link Graph path length for the occlusion and sensor range trials.

Failure rates for the sensor range and occlusion trials are reported in Table 4.1. As the obstacle size increases, the path failure rate increases, and as the sensor range decreases the failure rate increases. A large increase in the path failure rate occurs when $R' = 0.8R$, due to the spacing of landmarks in the environment. The spacing of landmarks in one landmark set, \mathcal{L}_i (at most $40\text{m} = 2R$)

Failure Rates for Simulation Studies		
Trial	Value	Path Failure Rate
Sensor Range Trial	R	0%
	$0.9R$	2.0%
	$0.8R$	31.9%
	$0.7R$	55.5%
	$0.6R$	74.7%
Occlusion Trial	0.1m	2.0%
	0.3m	2.8%
	0.5m	12.0%
	0.7m	16.3%

Table 4.1: Failure rates for the different simulation scenarios. The parameters for the sensor range trial is the robot sensor range, and the parameter for the occlusion trial is the obstacle size.

is too large for robots with a sensor range of $0.8R = 16\text{m}$. Finally, in the occlusion trials the worst failure rate occurs when the landmarks are largest. A sharp increase is seen in the failure rate when the landmarks reach a size of 0.5m. These results show the sensitivity of Q-Link to insufficient landmark features, while also showing that Q-Link can achieve path completion sometimes even when the set of assumptions in Assumptions 1 are violated. The failure rates in both cases may be improved by incorporating low level controllers that perform safe search beyond an arc of radius R and localization in qualitative states relative to $\mathcal{L}_i^{\text{ref}}$. Safe search controllers such as the one presented in [13] may be applied to this problem.

The simulation results indicate that having a robot with sufficient sensor range and the ability to find landmarks that are not occluded is critical to path success in the Q-Link architecture. However, when a robot can successfully find landmarks, having a smaller sensor range or environments with landmarks that occlude each other does not have a large effect on the trajectory length of the robot. Navigation can be made more robust by relaxing the requirement that

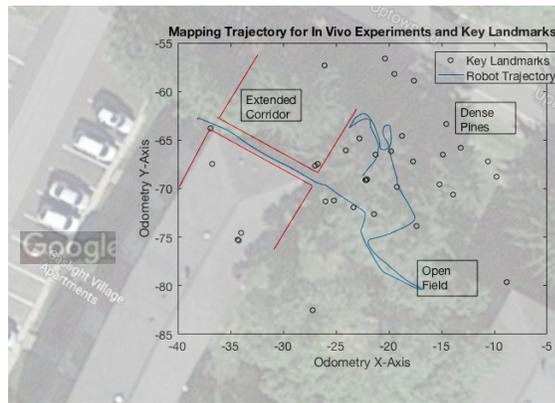
landmarks are visible at all times, or $\mathcal{L}^{\text{vis}_t} \neq \emptyset$ for all t , and having a controller at the trajectory level that can ‘safe search’ qualitative states to increase the chances of transitioning between map elements. This addition will increase the chance of success, but at the expense of trajectory length. One example of a controller that can be used is [13]. Additionally, the failure rate for the occlusion trials can be improved if the robot can recognize landmarks that are only partially occluded, which is a current area of research in the computer vision community.

4.4.2 In Vivo Experiments

An in vivo experiment in an outdoor environment with synthetic and natural objects (Figure 4.8) was conducted to show the adaptability and robustness of navigation with Q-Link. A Clearpath Jackal robot (Figure 4.8(a)), with a Hokuyo UTM-30LX Lidar sensor is used to map and navigate through the environment. The environment has three distinct environmental conditions: dense pines, open fields, and a corridor, all shown in Figure 4.8(b). An overhead view of the environment along with the path the robot traverses when developing the qualitative map of the environment is shown in Figure 4.8(c). The robot was teleoperated by a human to collect the mapping data.



(a) The Clearpath Jackal robot used for the in vivo experiment. The Hokuyo UTM-30LX lidar sensor is mounted above the ground. (b) Montage of key areas in the test environment. Left shows the extended objects as two walls, one the side of an apartment building, the other artificial walls. The middle shows a dense tree-filled environment with 1" thick layer of pine needles covering the ground. Right shows an open environment with soft pitched terrain.



(c) An overhead view of the environment for the in vivo study. Key point landmarks are shown in black, and the extended objects are shown by the red lines. The trajectory the robot traverses while mapping the environment is shown in blue.

Figure 4.8: Setup for the in vivo navigation trials.

Two different types of ternary QSR representation are used to map the environment: one that represents relationships between point landmark triplets and another that represents the relationship between two oriented, extended landmarks and one point landmark. The second ternary QSR representation used for point landmark triplets is the flip-flop calculus [30] that divides space into left and right (solid line in Figure 4.5). The QSR representation that rep-

resents the relationships between two oriented, extended landmarks and one point landmark contains the relationships ‘in front’, ‘between’, and ‘behind’. In this QSR representation, only the extended objects shown in 4.8(c) are reference landmarks. The ‘front’ represents the portion of the building near the cars, ‘between’ is the portion of space between the buildings, and ‘behind’ is the portion of space where there are dense pines and the open field.

Lidar data is used to sense features for the map of the environment. Candidate landmarks are extracted using a two-pass clustering routine to segment distinct objects. Clusters are classified as landmarks if they have between 10 and 50 lidar points or the cluster has a corner feature. Extended object candidates are detected via corner features, which are extracted according to the method in [38]. Landmark recall is not possible from the lidar data alone, since the clusters do not have distinct appearance. Therefore, the landmarks are recognized as members of distinct triangles. As triplets of landmarks are found, side lengths of triangles are recorded and used as a feature to recognize landmarks [15].

The robot is driven along the trajectory shown by the blue line in Figure 4.8(c) to map the environment. As the robot maps the environment, a probability distribution over the qualitative states of landmark sets is created, as in the PQRM [39]. Two different QSR representations are used. The measurement noise is modeled as $\mathcal{N}(0, 0.7)$ to account for noise in the lidar sensor measurements and sensor processing. The robot does not need to revisit the start of the map since loop closures are not necessary in QRM. The map has 25 point landmark features (black circles) and 2 extended landmark features (red lines), as shown in Figure 4.8(c). Notice that some of the point features are inside of the building - this occurs because windows in the building are on the ground

level, and the robot can detect objects through the windows. The map has 44 distinct QRM map elements, and the distribution over qualitative states for all map elements and landmark sets has low-entropy.

The Q-Link levels have elements chosen to increase the probability of the robot successfully completing a path. The Link Graph contains edges between map elements if $|\mathcal{L}_i \cap \mathcal{L}_j| = 2$. To ensure robustness, edge costs in the Link Graph are the expected probability of success as in equation (4.4). The finite state controller is a POMDP with a time horizon of 4, allowing the robot to search qualitative states more than once if $L_{i,j}^\alpha$ is not found. Direct transitions are allowed between the states, and controls, U_i , direct the robot to a desired $\{\mathcal{L}_i^{\text{ref}} : L_{i,j}^\alpha\} = q$. The low level controls leverage safety along with the robot sensor range. To ensure that the robot does not ‘get lost’ the low level controls direct the robot to the center of the line segment between landmarks in L_i^{ref} . Once the robot reaches this position it performs a sensor sweep in the qualitative state designated by U for a transition from M_i to M_j . This action is chosen because of the high likelihood that the robot will observe all of the landmarks required for a transition without driving into the qualitative state because the sensor range of the robot is large (60m). If a transition between M_i and M_j fails a contingency plan is implemented where the robot attempts to plan a different path at the Link Graph level and execute the new path. When the goal map element, M_{goal} is reached, the robot drives to the centroid of $\mathcal{L}_{\text{goal}}$.

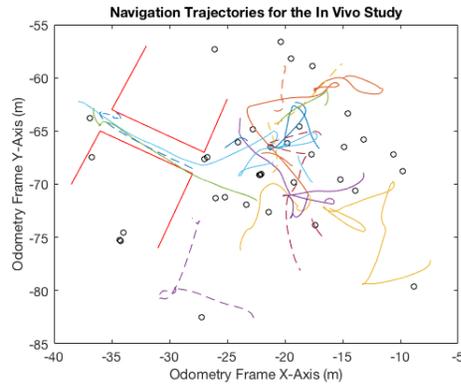


Figure 4.9: Navigation trajectories in the in vivo study. The solid lines represent successful paths, and the dashed lines represent paths that did not reach their goal.

A total of 18 trials were conducted where the robot navigates using the qualitative information in the PQRM using different start and goal locations in each trial. Paths are considered successfully completed if they reach the goal triplet, even if a contingency plan is executed. Trajectories are considered failed if they either reach an incorrect goal triplet (perceptual aliasing) or they cannot re-plan after a first failure due to insufficient environmental features in view. All of the trajectories are shown in different colors in Figure 4.9. The successfully completed paths are shown in solid lines, while the paths that failed to reach the goal are shown in dashed lines. Of the 18 paths trials, 13 were successful (72% success rate).

In five of the successful trajectories, the robot does not view the next map element on its path and re-plans at the link level. One example is shown by the trajectory in light blue, passing through the corridor, where the robot completes loop. The robot initially plans a path with $|P| = 3$ to reach its goal triplet. The first map element contains the QSR that represents relationships between two extended landmarks and one point landmark. The Trajectory level receives and

executes the command 'drive between the landmarks towards the back' successfully. However, after executing this command and reaching the qualitative state behind the extended landmarks, the robot does not view the next map element expected from the Link Graph path. It executes a re-plan at the Link Graph level and finds a path through map elements with the QSR representation for triplets of point landmarks. The robot is able to successfully transition between all map elements in the second path, and reaches the goal landmark then drives to the centroid.

Failure occurs for two reasons: perceptual aliasing and insufficient landmark features. The most frequent cause of failure is lack of sufficient features (all five failed paths) and in one case a combination of perceptual aliasing and insufficient features. The most frequent causes of insufficient features are ground plane interference due to the varying terrain pitch and the conservative nature of the lidar landmark processing. Often the robot encounters pitched terrain and the line scan lidar only views the ground plane. In one case (shown by the dashed blue line between the extended objects), the robot exits the area between the buildings and drives into a natural basin where the lidar only views ground plane. In one of the failure cases, there is an issue with perceptual aliasing and insufficient features (shown by the orange dashed line by $(-20, -60)$). The robot erroneously detects a part of the roadway near the testing environment as a landmark, and matches the roadway along with two other detected landmarks to a triplet in the map. This results in the robot driving at an incorrect heading and ending in an area where there are insufficient features to replan.

Trajectory length as a function of Link Graph path length is reported in Table 4.2. The average trajectory length increases monotonically with information

Path Lengths in the In Vivo Study		
$ P $	Number of Paths	Average Trajectory Length (m)
1	2	2.5
2	3	6.5
3	2	12.7
4	3	14.3
6	2	32.9
8	1	38.4

Table 4.2: Average trajectory lengths in the in vivo study reported as a function of Link Graph path length, $|P|$.

path length, as in the simulation experiments (Figure 4.7). The path length dramatically increases for $|P| > 5$. These paths (blue, orange, and yellow solid lines in Figure 4.9) all have long segments through areas that are not well populated with features. None of the paths exceed either distance bound provided by Propositions 2 or 4.

While Q-Link has a high rate of completion in the tested varied environment, several changes can be made to further improve performance. First, landmark detection can be made more robust to decrease the probability that the robot fails to find sufficient landmark features. The landmark detection used for line scan lidar data is conservative, and rejects many landmark candidates. The detection can be made less conservative, but at the price a more false landmarks being detected. Use of a different sensor may improve landmark detection by providing richer data. Specifically a 3-D lidar scanner or a vision sensor will not be as prone to ground plane occlusion as the single line scan lidar. Vision may also be used to provide visually distinct landmarks that can be reliably detected. Second, better exploration of qualitative states at the trajectory level can be enabled by a low-level controller that performs a ‘safe search’ by exploring unmapped areas of space while maintaining localization. This expands

the amount of space searched in a qualitative state before the robot attempts to explore another qualitative state. Third, the predicted probability of success in the Link Graph level can be made more accurate if the terrain profile of an environment is known a priori or developed on-line in real time. In this experiment, the conditional observation probabilities are assumed to be the same for all QRM elements. However, the interaction of terrain features with the single line scan lidar makes observations less likely in some areas than others. If these probabilities are accurately modeled, the likelihood that the robot will avoid map entries in areas with low visibility will increase.

4.5 Conclusion

A general three level planning architecture for navigation in qualitative relational maps (QRM) is presented. The Q-Link architecture is a general planner that can be used with any type of QRM, including QRM developed with ternary and binary QSR and extended or point-like landmarks. Importantly, Q-Link generates plans that can be executed in environments with extended and point-like landmarks. Q-Link also seamlessly incorporates uncertainty from the map, controls, and perception. Safety guarantees and path length bounds are provided in environments with point landmarks and no occlusions. Studies in simulation show the importance of landmark visibility to achieving goals when using Q-Link. An in vivo study demonstrates the performance of Q-Link in an environment with three distinct environmental conditions: a corridor, dense pines, and an open field. The Q-Link algorithm successfully completes paths 72% of the time in the in vivo study. Paths primarily failed due to lack of sufficient landmark features from which the robot can localize itself.

CHAPTER 5

CONCLUSION AND CONTRIBUTIONS

A general method for incorporating uncertain sensor readings into qualitative relational maps is presented in Chapter 2. The probabilistic qualitative relational mapping (PQRM) algorithm is designed for use on any robotic system for sensor readings that can provide unique landmark identifications with a probability distribution over the point locations of landmarks in independent coordinate frames. The probabilities of qualitative states are computed to obtain categorical distributions over the qualitative state positions of landmarks. Results are presented for maps created with several ternary qualitative spatial relationships. Studies show that the map entropy is low for qualitative relational maps even in the presence of high sensor noise. Additionally a study of qualitative relational mapping on the New College dataset shows the utility of probabilistic qualitative relational mapping in a real world scenario.

List of contributions:

- Development of a method to incorporate uncertain sensor measurements into qualitative spatial relationships that is general and extends to any type of qualitative spatial relationship.
- The probabilistic qualitative relational mapping algorithm, a general algorithm for creating probabilistic qualitative relational maps for any type of qualitative spatial relationship.
- Simulations studies that test the performance and convergence properties of probabilistic qualitative relational maps under several cases of sensor noise. Additional testing on the convergence of probabilistic qualitative

relational maps for qualitative spatial relationships of several different resolutions.

- A study on probabilistic qualitative relational mapping with visual landmarks in an outdoor environment on the New College dataset. Six different maps are created, each with a different qualitative spatial relationship and three different processing methods are tested, showing the performance for different types of camera set ups.

A general method for creating highly salient discrete environment representations from continuous data is presented in Chapter 3. The salient regions are used for mapping in a global context and are extended to be qualitative spatial relationships. The salient regions are created by maximizing data likelihood in discrete regions of space. Two user defined parameters adjust the number of discrete regions and allow a user to trade between achieving exactly the number of desired regions and higher saliency. The parameters can be iteratively adjusted to create a multi-resolution salient region map. The SERGE algorithm is general and can be applied to any data that has a probability distribution over locations of landmarks in an environment. Simulation studies show the performance in saliency and mean error rate in a 5-fold cross validation study, and also show how the SERGE algorithm can be applied in qualitative relational mapping. A study on data from a kinect sensor shows the performance on data that may be acquired from a common robotic sensor.

List of contributions:

- Development of the SERGE algorithm, which allows users to create high saliency discrete representations of space.

- Two tuning parameters that control map saliency and resolution.
- Development of an algorithm to compute multi-resolution salient region maps.
- An example navigation scenario in multi-resolution maps
- Application of discrete salient regions in qualitative relational mapping.
- An algorithm to compute compose relationships of the salient region maps for use with qualitative spatial reasoning.
- Five fold cross validation studies that detail the performance of SERGE on different data types with different tuning parameter settings.

A general planning and navigation architecture for qualitative relational maps (Q-Link) is presented in Chapter 4. Q-Link allows a robot to reason over disjoint qualitative spatial relationships in a qualitative relational map by linking landmarks that are necessary to transition between coordinate frames. Q-Link is flexible, and can be tailored for any scenario. Low-level trajectory planners are chosen for specific robots, and high level controls are chosen for the type of qualitative relational map in use. Qualitative relational maps using any type of qualitative spatial relationship, or multiple qualitative spatial relationships are used with Q-Link. Q-Link can incorporate uncertain information from probabilistic qualitative relational maps in its framework. Path completion guarantees and distance bounds are provided in environments with point landmarks and under some assumptions about sensor range. Studies in simulation and in vivo show the flexibility and robustness of navigation with qualitative information using Q-Link.

List of contributions:

- Identification of the minimum amount of information needed to robustly navigate with qualitative information.
- A general, three level architecture that allows robots to reason about qualitative relational maps and navigate to a goal.
- The introduction of link landmarks as the key piece of information that allows robot to transition between different map elements.
- Path completion guarantees for environments with point landmarks and one assumption about robotic sensor range
- Distance bounds for two types of control behavior
- A methodology for the seamless incorporation of probabilistic information into the planner.
- Simulation studies that study Q-link performance as a function of limited sensor range and landmark occlusion
- An in vivo study that studies the performance of Q-link with a hybrid probabilistic qualitative relational map in a real world environment on a robot with a single scan lidar sensor.

All methods presented for qualitative relational mapping and navigation are general, flexible, and applicable to several scenarios. Importantly, none are tied to any robotic platform, one mode of perception, or one specific qualitative spatial representation. Each of the chapters provides algorithms and methodologies that gear qualitative relational mapping towards further, practical use in modern robotics. All of the ideas are complimentary and may be combined into one system or chosen for different applications.

BIBLIOGRAPHY

- [1] Nisar R Ahmed, Eric M Sample, and Mark Campbell. Bayesian multi-categorical soft data fusion for human–robot collaboration. *Robotics, IEEE Transactions on*, 29(1):189–206, 2013.
- [2] Christopher M Bishop. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.
- [3] Jose-Luis Blanco, Juan-Antonio Fernandez-Madrigal, and Javier González. A novel measure of uncertainty for mobile robot slam with rao-blackwellized particle filters. *The International Journal of Robotics Research*, 27(1):73–89, 2008.
- [4] Luca Carlone and Daniel Lyons. Uncertainty-constrained robot exploration: A mixed-integer linear programming approach. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1140–1147. IEEE, 2014.
- [5] Howie M Choset. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.
- [6] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
- [7] Paolo Fogliaroni, Jan Oliver Wallgrün, Eliseo Clementini, Francesco Tarquini, and Diedrich Wolter. A qualitative approach to localization and navigation based on visibility information. In *Spatial Information Theory*, pages 312–329. Springer, 2009.
- [8] Christian Freksa et al. Conceptual neighborhood and its role in temporal and spatial reasoning. 1991.
- [9] Simone Frintrop. *VOCUS: A visual attention system for object detection and goal-directed search*, volume 3899. Springer, 2006.
- [10] Simone Frintrop and Patric Jensfelt. Attentional landmarks and active gaze control for visual slam. *Robotics, IEEE Transactions on*, 24(5):1054–1065, 2008.

- [11] Leonidas J Guibas, David Hsu, Hanna Kurniawati, and Ehsan Rehman. Bounded uncertainty roadmaps for path planning. In *Algorithmic Foundation of Robotics VIII*, pages 199–215. Springer, 2009.
- [12] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [13] Alexander Ivanov and Mark Campbell. Uncertainty constrained robotic exploration: An integrated exploration planner. *arXiv preprint arXiv:1612.01392*, 2016.
- [14] Margaret E Jefferies and Wai K Yeap. The utility of global representations in a cognitive map. In *International Conference on Spatial Information Theory*, pages 233–246. Springer, 2001.
- [15] Brandon M Jones, Mark Campbell, and Lang Tong. Maximum likelihood fusion of stochastic maps. *IEEE Transactions on Signal Processing*, 62(8):2090–2099, 2014.
- [16] Simon J Julier and Jeffrey K Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 4238–4243. IEEE, 2001.
- [17] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, page 0278364911430419, 2011.
- [18] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [19] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. Constructing symbolic representations for high-level planning. 2014.
- [20] Kirill Kouzoubov and David Austin. Hybrid topological/metric approach to slam. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 872–877. IEEE, 2004.

- [21] Vikram Krishnamurthy. Partially observed markov decision processes, 2016.
- [22] Lars Kunze, Keerthi Kumar Doreswamy, and Nick Hawes. Using qualitative spatial relations for indirect object search. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [23] Hanna Kurniawati, Tirthankar Bandyopadhyay, and Nicholas M Patrikalakis. Global motion planning under uncertain motion, sensing, and environment map. *Autonomous Robots*, 33(3):255–272, 2012.
- [24] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008. Zurich, Switzerland., 2008.
- [25] Labbé, M. <http://introlab.github.io/find-object>.
- [26] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.
- [27] Jae Hee Lee, Jochen Renz, Diedrich Wolter, et al. Starvars-effective reasoning about relative directions. In *IJCAI*, 2013.
- [28] Tod S Levitt and Daryl T Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44(3):305–360, 1990.
- [29] G É Ligozat. Reasoning about cardinal directions. *Journal of Visual Languages & Computing*, 9(1):23–44, 1998.
- [30] Gérard F Ligozat. Qualitative triangulation for spatial reasoning. In *Spatial Information Theory A Theoretical Basis for GIS*, pages 54–68. Springer, 1993.
- [31] Mark McClelland, Mark Campbell, and Tara Estlin. Qualitative relational mapping for robotic navigation. *AIAA Infotech@ Aerospace*, 2012.
- [32] Mark McClelland, Mark Campbell, and Tara Estlin. Qualitative relational mapping for mobile robots with minimal sensing. *Journal of Aerospace Information Systems*, 11(8):497–511, 2014.
- [33] Mark McClelland, Mark Campbell, and Tara Estlin. Qualitative relational

mapping and navigation for planetary rovers. *Robotics and Autonomous Systems*, 83:73–86, 2016.

- [34] Patrycja E Missiuro and Nicholas Roy. Adapting probabilistic roadmaps to handle uncertain maps. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1261–1267. IEEE, 2006.
- [35] Reinhard Moratz. Representing relative direction as a binary relation of oriented points. In *ECAI*, volume 6, pages 407–411, 2006.
- [36] Reinhard Moratz and Jan Oliver Wallgrün. Spatial reasoning with augmented points: Extending cardinal directions with local distances. *Journal of Spatial Information Science*, 2012(5):1–30, 2012.
- [37] Jonathan Mugan and Benjamin Kuipers. Autonomous learning of high-level states and actions in continuous environments. *IEEE Transactions on Autonomous Mental Development*, 4(1):70–86, 2012.
- [38] Pedro Nunez, Ricardo Vazquez-Martin, Jose C del Toro, Antonio Bandera, and Francisco Sandoval. Feature extraction from laser scan data based on curvature estimation for mobile robotics. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1167–1172. IEEE, 2006.
- [39] Jennifer Padgett and Mark Campbell. Probabilistic qualitative mapping for robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5388–5394. IEEE, 2016.
- [40] Samuel Prentice and Nicholas Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, 28(11-12):1448–1465, 2009.
- [41] David A Randell, Zhan Cui, and Anthony G Cohn. A spatial logic based on regions and connection. *KR*, 92:165–176, 1992.
- [42] Ananth Ranganathan and Frank Dellaert. Online probabilistic topological mapping. *The International Journal of Robotics Research*, page 0278364910393287, 2011.
- [43] Emilio Remolina and Benjamin Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, 152(1):47–104, 2004.

- [44] Pascal Rost, Lothar Hotz, and Stephanie Von Riegen. Supporting mobile robot's tasks through qualitative spatial reasoning. In *ICINCO (2)*, pages 394–399. Citeseer, 2012.
- [45] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *The International Journal of Robotics Research*, 28(5):595–599, May 2009.
- [46] Niko Sunderhauf and Peter Protzel. Switchable constraints for robust pose graph slam. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1879–1884. IEEE, 2012.
- [47] Kenji Suzuki, Isao Horiba, and Noboru Sugie. Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, 89(1):1–23, 2003.
- [48] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.
- [49] Rina Tse and Mark Campbell. Human-robot information sharing with structured language generation from probabilistic beliefs. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 1242–1248. IEEE, 2015.
- [50] Jan Oliver Wallgrün. Hierarchical voronoi-based route graph representations for planning, spatial reasoning, and communication. In *Proceedings of the 4th International Cognitive Robotics Workshop (CogRob-2004)*, pages 64–69, 2004.
- [51] Jan Oliver Wallgrün. Qualitative spatial reasoning for topological map learning. *Spatial Cognition & Computation*, 10(4):207–246, 2010.
- [52] Jay Young and Nick Hawes. Learning by observation using qualitative spatial relations. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 745–751. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
- [53] Kai Zimmermann and Christian Freksa. Qualitative spatial reasoning using orientation, distance, and path knowledge. *Applied Intelligence*, 6(1):49–58, 1996.