

A STUDY OF THE BENEFITS OF CELL CORRELATIONS
IN OCCUPANCY GRID MAPPING

by

Rehman Shamsh Merali

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Aerospace Science and Engineering
University of Toronto

© Copyright 2020 by Rehman Shamsh Merali

Abstract

A Study of the Benefits of Cell Correlations
in Occupancy Grid Mapping

Rehman Shamsh Merali

Doctor of Philosophy

Graduate Department of Aerospace Science and Engineering

University of Toronto

2020

A mobile robot equipped with a range sensor can create a map of its environment given the range measurements and corresponding robot pose(s). These maps, although not perfect, are useful for path planning, exploration, search, and often the objective is to create an accurate map of the environment. Occupancy grid mapping (OGM) is a popular mapping technique that discretizes the environment into cells (or voxels) and seeks to estimate the occupancy probability of each cell. Computing the probability of all possible maps is computationally intractable and thus it is typically estimated using Bayesian inference to compute the probability of each cell and assuming that the cells are independent.

This thesis revisits the assumptions made in traditional OGM and reintroduces cell correlations in the prior, the measurements, and the posterior. While the full Bayesian posterior is intractable, we develop improved approximations and show how two popular techniques, variational inference (to optimize parameters of a chosen model type) and Markov Chain Monte Carlo (MCMC), can be used to relax traditional assumptions. We show that mutual information between cells in the full Bayesian posterior is concentrated locally and approaches zero for cells that are distant. Therefore, we have developed OGM algorithms that maintain local cell correlations to a varying degree and this degree can be user-defined.

We also introduce a data-driven prior that correlates cells locally, which is shown to be particularly effective when occupancy measurements are sparse. The result is an OGM

algorithm that is able to interpolate between measurement rays, reduce the effect of sensor noise, better estimate the occupancy of each cell and also the joint marginal of a group of cells. The algorithms are shown to work on simulated and hardware datasets, where computing the full Bayesian posterior would be computationally intractable. Depending on the mapping application, the methods offer some different points on the tradeoff curve between speed of operation and map quality in their ability to capture the full Bayesian posterior.

Acknowledgements

This thesis was only made possible with the encouragement and assistance of many people. I am fortunate to have strong support in both my personal and academic life.

First and foremost, I am forever grateful to my family. To my parents, Shamsh and Salma, who have given me every opportunity, encouraged me at every step and have always put their children above all else. To my brother, Hasan, his wife, Zuleikha, and my niece, Arya: thank you for your endless motivation and for setting an incredible example towards which I will continue to strive. To my sister, Noor, who has always been there to support me both personally and professionally, and whom I can always count on to bring a smile when times are tough. To Heather, who has taught me so much and helped me pursue my goals. Finally, to my loving wife, Safina, without whom I cannot (or don't want to) imagine life without, you've made me stronger, smarter and happier than I ever thought possible. I love each of you.

In my academic life, this thesis would not have been possible without the patience, guidance, and direction of my supervisor, Professor Tim Barfoot. Tim, thank you for setting a high standard of excellence and supporting me as I strive towards it; your work has been a constant motivation for me. I have been at the Autonomous Space Robotics Lab (ASRL) for some time and would like to specifically thank Chi Hay Tong, Jonathan Gammell and David Yoon for their academic collaborations. I would also like to thank many others for their friendship and moral support over the years (in the order we met): Braden, Pat C, Paul, Keith, Hien, Goran, Billy, Colin, Andrew, Hang, Sean, Chris, Peter, Pat M, Mona, Michael, Keenan, Jeremy and Mollie.

I would also like to thank Professor Chris Damaren and Professor Gabriele D'Eleuterio for guiding my research at the University of Toronto Institute for Aerospace Studies (UTIAS). I thank Professor Radford Neal from the University of Toronto and Dr. Joseph Bakambu from MDA Space Missions for their valuable contributions to my research. For funding my research, I would like to thank the Canadian Space Agency (CSA), MDA Space Missions, and the National Science and Engineering Research Council (NSERC) of Canada. Thank you to my examiners, Professor Florian Shkurti from the University of Toronto and Professor Michael Jenkin from York University, for their consideration and feedback on this dissertation. Finally, I thank Dr. Alberto Elfes (1953–2020) who introduced occupancy grid mapping in his doctoral dissertation and made many contributions to the field of mobile robotics, upon which this research is based.

Contents

1	Introduction	1
1.1	Literature Review and Motivation	4
1.1.1	Grid-Based Mapping	7
1.1.2	Mapping without a Grid Structure	11
1.1.3	Mapping Applications and Metrics	13
1.2	Thesis Background and Overview	16
1.3	Thesis Contributions	20
2	Occupancy Grid Mapping (OGM) Fundamentals	22
2.1	OGM Formulation	23
2.2	Maximum a Posteriori (MAP)	25
2.3	Full Bayesian Solution	27
2.3.1	Pose-Constrained Full Solution	29
2.4	Traditional Occupancy Grid Mapping (OGM)	31
2.5	Optimizing Traditional OGM	33
2.6	Mutual Information from Measurements	42
2.7	Chapter Summary	44
3	Patch Map	46
3.1	Online Patch Map	47
3.2	Offline Patch Map	52
3.3	Chapter Summary	56
4	Markov Chain Monte Carlo (MCMC)	58
4.1	Sampling Cells	60
4.1.1	Cell Sampling Formulation	61

4.1.2	Cell Sampling Application	63
4.2	Sampling Patches	67
4.2.1	Patch Sampling Formulation	67
4.2.2	Patch Sampling Application	68
4.2.3	Estimating the First-Occupied-Cell	71
4.3	Chapter Summary	72
5	Cell Correlations in the Prior	73
5.1	Full Bayesian Solution with a Data-Driven Prior	74
5.2	MCMC with a Nonuniform Prior	78
5.2.1	Higher-Order Markov Random Field	79
5.2.2	Data-Driven Patch Prior	82
5.2.3	Data-Driven Patch Prior with Border	88
5.3	Chapter Summary	95
6	Experimental Results	96
6.1	Simulation Results	98
6.1.1	Patch Map Results	99
6.1.2	Optimizing Traditional OGM	101
6.1.3	MCMC with a Uniform Prior	105
6.1.4	MCMC with a Data-Driven Patch Prior	106
6.2	Hardware Results	114
6.2.1	Intel Lab Dataset	115
6.2.2	MIT Stata Dataset	127
6.3	Chapter Summary	134
7	Conclusion	136
7.1	Review of Novel Contributions	138
7.2	Future Work	139
	Bibliography	142

List of Figures

1.1	Mobile robots performing repetitive tasks	1
1.2	Mobile robots performing dangerous tasks	2
1.3	Range Sensors on Mobile Robots	3
1.4	Narrow Sensing Frustum of LiDAR	4
1.5	Semantic Mapping	5
1.6	Various 2D map representations	6
1.7	Various 3D map representations	7
1.8	Incremental GPOM	12
1.9	Hilbert Maps compared to OGM	14
1.10	Overview of OGM Results in 2D	18
1.11	Mutual Information in OGM	19
2.1	1D Occupancy Grid	24
2.2	Inverse sensor model for traditional OGM	33
2.3	Optimized update terms for traditional OGM algorithms	34
2.4	1D OG from various inverse sensor models	37
2.5	Optimized update term for increasing parameters	38
2.6	Cross entropy for Δ optimization	39
2.7	Average D_{KL} for traditional OGM algorithms	39
2.8	Δ optimization results in 2D	41
2.9	Mutual information in the full Bayesian posterior	44
3.1	2D Patch Connectivity Neighbourhoods	47
3.2	1D OG: Online Patch Map	50
3.3	1D OG: Online Patch Map with Offset	51
3.4	1D OG: Offline Patch Map	54
3.5	Patch Map OGM in 2D	56

4.1	MCMC OG in 2D with Uniform Prior	63
4.2	MCMC OG in 2D with Data-Driven Prior	64
4.3	MCMC Convergence	65
4.4	1D OG: MCMC with Increasing Samples	66
4.5	1D OG: MCMC with Uniform Prior	69
4.6	MCMC OG in 2D with Increasing Samples	70
4.7	Probability of First-Occupied Cell in 1D	71
5.1	3-cell Map Example for Data-Driven Prior	75
5.2	Mutual Information for Data-Driven Prior	75
5.3	1D OG: Data-Driven Prior	76
5.4	First-Occupied Cell with Data Driven Prior	77
5.5	First-Occupied Cell with Data-Driven Prior after 50 Experiments	77
5.6	MRF Cell Connectivity	80
5.7	1D OG: Higher-Order MRF	81
5.8	2D OG: Higher-Order MRF	83
5.9	2D Maps for Simulation	85
5.10	Sampling patches from data	86
5.11	Example 3×3 patches	87
5.12	2D OG with a Data-Driven Prior	88
5.13	Example 3×3 patches given their border	89
5.14	Sub-sampling 5×5 patches	90
5.15	Patch Probability given Border	91
5.16	1D OG: Data-Driven Prior with Borders	92
5.17	2D OG with Border-Based Data-Driven Prior	94
6.1	Simulation Screenshot	99
6.2	2D Results from Offline Patch Map	100
6.3	2D Results for Optimized Δ Function	104
6.4	2D MCMC Convergence	105
6.5	2D Results for MCMC with a Uniform Prior	107
6.6	2D OG with Simulated Data	109
6.7	2D OG thresholded at different values	111
6.8	P-R Curve for Simulated Results	113
6.9	F_1 Score for Simulated Results	113

6.10	Benchmark OG Maps for Hardware Datasets	116
6.11	2D OG of Intel Lab Dataset	117
6.12	OG of Intel Lab Dataset from MCMC with $L=1$	119
6.13	Hardware Results – Experiment 1	121
6.14	Hardware Results – Experiment 2	122
6.15	Hardware Results – Experiment 3	123
6.16	Masked Benchmark Map of Intel Lab Dataset	124
6.17	Intel Lab Dataset – F_1 Scores	125
6.18	Hardware Results – Experiment 4	129
6.19	Hardware Results – Experiment 5	130
6.20	Hardware Results – Experiment 6	131
6.21	Masked Benchmark Map of Intel Lab Dataset	132
6.22	Intel Lab Dataset – F_1 Scores	133

Acronyms

ANN	: Artificial Neural Network
COM	: Continuous Occupancy Mapping
CSA	: Canadian Space Agency
EG	: Evidence Grid
EM	: Expectation Maximization
FN	: False negative
FOV	: Field of View
FP	: False positive
GMM	: Gaussian Mixture Model
GP	: Gaussian Process
GPIS	: Gaussian Process Implicit Surfaces
GPOM	: Gaussian Process Occupancy Mapping
GPS	: Global Positioning System
GT	: Ground truth
HMC	: Hamiltonian Monte Carlo
LiDAR	: Light Detection And Ranging
LRF	: Laser Rangefinder
MAP	: Maximum a posteriori
MCMC	: Markov Chain Monte Carlo
MDA	: MacDonald, Dettwiler and Associates
MET	: Mars Emulation Terrain
MI	: Mutual Information
MIT	: Massachusetts Institute of Technology
MRF	: Markov Random Field
NBV	: Next-Best-View

NDT : Normal Distributions Transform
NDT-OM : Normal Distributions Transform Occupancy Map
NSERC : Natural Sciences and Engineering Research Council of Canada
OG : Occupancy Grid
OGM : Occupancy Grid Mapping
P-R : Precision-Recall
PRF : Polygon Random Fields
PTU : Pan-and-Tilt Unit
RADAR : Radio Detection and Ranging
RBPF : Rao-Blackwellised Particle Filter
RMS : Root-Mean-Square
SLAM : Simultaneous Localization and Mapping
SONAR : Sound Navigation Ranging
UTIAS : University of Toronto Institute for Aerospace Studies
VI : Variational Inference

Notation

- K : Number of cells in the map.
 R : 2^K possible maps.
 N : Number of measurements and robot poses.
 L : Number of cells in a patch.
 W : Number of patch configurations.
 F : Maximum number of cells in a measurement ray.
- m : Set of all cells in the map.
 m^r : Configuration r of the map.
 $m^{(i)}$: Map sample i .
 \tilde{m} : Estimate of map m .
 m_k : Cell k of the map.
 m_{-k} : Set of all cells other than of m_k .
- a_k : Patch of cells, centered at cell m_k .
 $a_{k,w}$: Patch of cells, centered at cell m_k with configuration w .
 $a_k^{(i)}$: Patch sample i of cells centered at m_k .
- z : Set of all measurements.
 $z_{1:n-1}$: Set of measurements from 1 through $n-1$.
 x : Set of all robot poses.
 x_n : Robot pose corresponding to measurement z_n .
 κ : Cell along measurement ray z_n , relative to x_n .
 f : First occupied cell.
 $f_{r,n}$: First occupied cell on map m^r from pose x_n .

- λ : Constant prior occupancy probability of a cell, $p(m_k) = \lambda$.
- η : Normalizing constant.
- ϵ : Random variable representing the edge weight in a MRF.
- ψ : Constant value of ϵ for neighbouring cells.

$l(\cdot)$: Log-odds.

$\mathcal{O}(\cdot)$: Big O notation for complexity analysis.

$\mathcal{N}(f, \sigma^2)$: Normal distribution with mean f and standard deviation σ .

$D_{\text{KL}}(p||q)$: Kullback–Leibler divergence between a benchmark, p , and an estimated, q , probability distribution.

Chapter 1

Introduction

Many vehicles that have traditionally been piloted by human operators on board (e.g., planes, trains, and automobiles) are being equipped with sensors and controls to be piloted remotely (i.e., teleoperated) and computers on board the vehicle are even capable of semiautonomous or fully autonomous control of the vehicle. Furthermore, vehicles are increasingly being designed to be operated without a human pilot on board and we broadly describe these vehicles as mobile robots. The utility of mobile robots continues to expand as they are often utilized for tasks that are seen as dull, dirty, or dangerous for human operators. For example, these robots are deployed in repetitive applications such as floor cleaning, security and warehouse robotics, as seen in Figure 1.1. They are also used for tasks that could be dangerous to a human pilot, such as mining, driving, surveillance, and extraterrestrial exploration, as seen in Figure 1.2. As sensors, computers



(a) Autonomous floor cleaning robot. Photo credit: Avidbots (2020).



(b) Autonomous warehouse robot. Photo credit: OTTO Motors (2020).

Figure 1.1: Mobile robots are often used in industry to perform repetitive tasks like cleaning floors and moving pallets in a warehouse.



(a) Robocar set the record for world's fastest autonomous car in 2019 with a speed of 282.42 km/h. Photo credit: Roborace (2020).



(b) NASA's Perseverance rover, which launched on July 30, 2020 and is scheduled to land on Mars in February 2021. Photo credit: NASA (2020).

Figure 1.2: Mobile robots can protect human pilots when performing dangerous tasks, such as driving at high speed and exploring extraterrestrial environments.

and algorithms continue to evolve, mobile robots will become more ubiquitous in our society.

Mobile robots are equipped with an array of sensors to measure changes in the robot and to detect the environment. These sensors provide inputs to a variety of algorithms that enable the mobile robot to complete an assigned task or mission. The enabling algorithms can be mission specific, but some tasks that are common to most mobile robots are localization, mapping and path planning. Localization algorithms deal with determining the pose (position and orientation) of the robot in the environment. Mapping algorithms deal with creating a map of the environment that the robot senses. Path planning algorithms generally use the output of localization and mapping algorithms to determine where the robot can move. This thesis will focus on the mapping problem, which is an enabling technology for path planning and other algorithms and is often the goal of the mission in and of itself. The mapping problem is often coupled with the localization problem, but this thesis separates the two and assumes that the localization problem is solved separately. This decoupling has been shown to be effective in the literature (Thrun et al., 2001; Tong et al., 2012) and allows the work in this thesis to focus on mapping independent of localization error.

Mobile robots can use a variety of sensors to map the environment, some of which are highlighted in Figure 1.3. Tactile sensors can be inexpensive, but require the robot to touch the environment, which means that each measurement can take considerable time. Furthermore, these sensors may change the environment by interacting with it. Therefore,

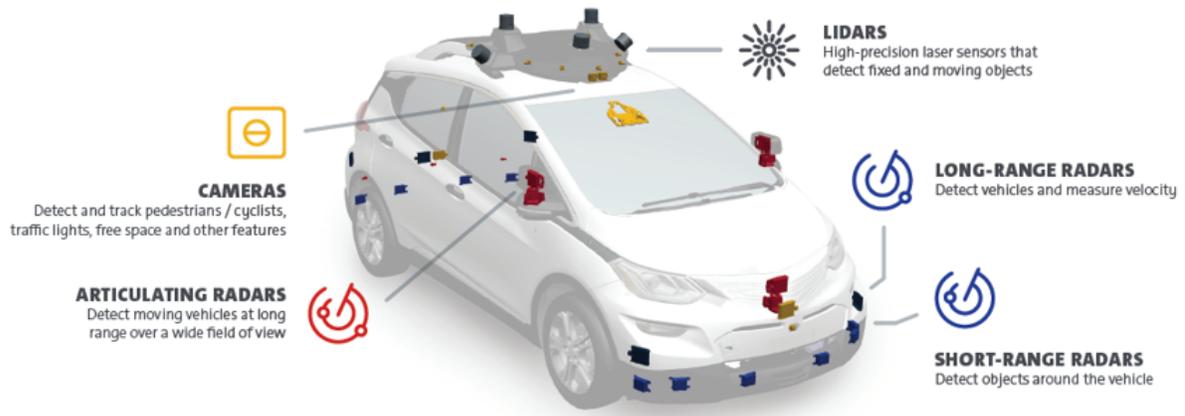


Figure 1.3: A mobile robot can be equipped with a variety of range sensors, which are often used for mapping the environment. Image credit: Cardinal (2020).

many robots will use light, sound, or radio frequencies to detect the environment. Some sensors will only detect these frequencies, however, when the sensor both emits and detects the frequency, these sensors are known as LiDAR, sonar and radar, respectively. Radar is often used on aerial and ground vehicles as radio waves can travel great distances. Sonar sensors are less expensive and often used on ground vehicles and submarines. Sonar is commonly used on underwater vehicles because sound waves travel faster in water than air and also travel farther than radio waves in sea water. Although radar and sonar sensors are more robust to weather conditions, LiDAR is commonly used on ground vehicles because it can provide more accurate range measurements. Infrared sensors are also less accurate than LiDAR because they compute depth based on the angle of received light, whereas LiDAR computes depth based on the time-of-flight of each laser ray. Furthermore, LiDAR has a narrow sensing frustum (or beam divergence) as compared to other range sensors, which enables more accurate measurements. Figure 1.4 illustrates the narrow beam divergence of LiDAR as compared to radar. This thesis will assume the use of narrow-beam range sensor, such as LiDAR, but the theory may be applied more broadly.

Cameras are also a popular mapping sensor in mobile robotics, but they do not inherently measure depth. These sensors do not generally emit light, but detect the reflection of light in the environment. Multiple cameras, such as stereo cameras, can be used to infer depth from camera images. Alternatively, cameras can be coupled with other range sensors to provide depth information. In addition to the visible spectrum, infrared cameras are commonly used for thermal imaging and night vision. For a mapping

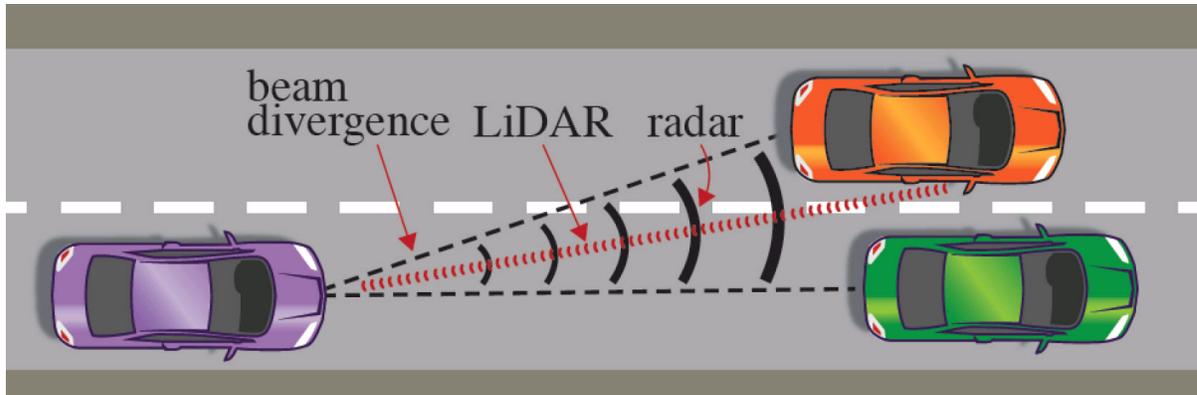


Figure 1.4: LiDAR has a narrow beam divergence (or sensing frustum) compared to other range sensors like radar. This enables LiDAR to more accurately detect the object that reflected the beam. This thesis will focus on narrow-beam range sensors, such as LiDAR. Image credit: Schiavullo (2018).

algorithm, cameras and other sensors can be modeled as a series of narrow-beam range measurements; some sensors even estimate the depth of each pixel in the camera image. Therefore the algorithms presented in this thesis may be applied to these sensors as well, or the algorithms may be adapted for different sensor models.

Given a series of range measurements from on board a mobile robot, some mapping algorithms use this data to create a map of the environment relative to the robot’s pose. However, given a series of robot poses and range measurements from each, many mapping algorithms seek to create a globally consistent map of the environment. The mapping technique used may depend on the computation available, the sensor, the mission, or other factors. Section 1.1 will explore various mapping techniques presented in the literature and highlight the benefit of occupancy grid mapping (OGM) relative to other methods. This thesis will then focus on OGM and Section 1.2 will present an overview of the thesis and Section 1.3 will highlight the novel contributions made.

1.1 Literature Review and Motivation

Mobile robots have been using range sensors to map their environment for decades (Martin and Moravec, 1996; Moravec and Elfes, 1985). Given a series of range measurements and the respective robot pose (position and orientation) from where each measurement was taken, various methods exist to represent this information. Of the mapping techniques developed in the literature, some of the most popular can be broadly categorized as semantic maps, feature maps, geometric maps and grid maps (Tipaldi, 2009).

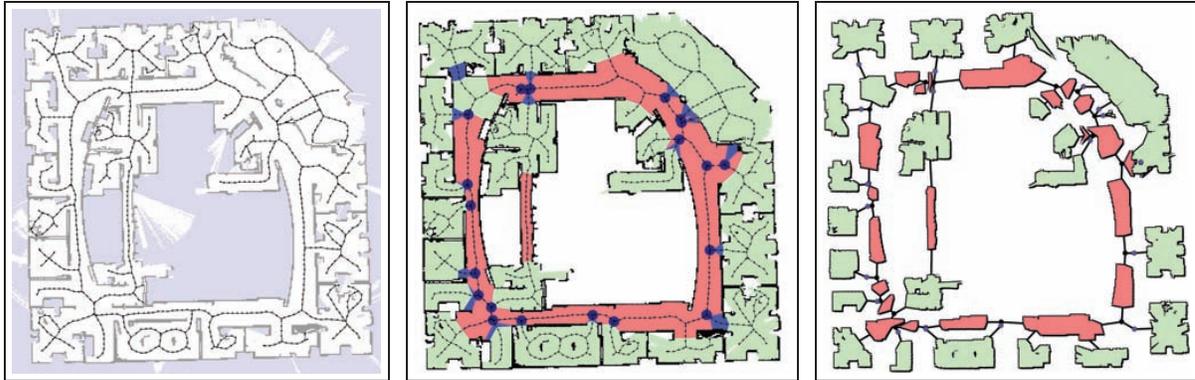


Figure 1.5: 2D maps produced using LiDAR and the Intel Lab dataset. The image on the left shows the occupancy grid map with the robot path. The image in the middle shows a semantic map where hallways are shown in red, rooms in green, doorways in blue and junctions as black circles. The image on the right shows a topological-metric map highlighting the connectivity between rooms and hallways at junctions and doorways. Image credit: Friedman et al. (2007).

Semantic maps provide an abstraction of the space and a means for human-robot communication. For example, the space may be categorized according to characteristics such as scalability, inference model, temporal coherence, and topological map usage (Kostavelis and Gasteratos, 2015). One such semantic map is shown in Figure 1.5, where the map is segmented into rooms and hallways, connected by junctions and doorways. The semantic map shown in the third image may discard some of the information in the first and second image after processing it to develop the semantic map.

A feature map detects and stores the position of specific features, such as lines, corners, or scale invariant feature transform (SIFT) features (Lowe, 2004). In addition to the position of each feature, these mapping techniques typically store a measure of uncertainty. Figure 1.6d illustrates a feature map representation of the true map shown in Figure 1.6a. The feature map extracts S-shapes, Z-shapes, and doors from the true map (or measurements of the true map). Note that much of the environment is not represented as it is not recognized as one of these features based on the measurements.

Geometric maps use range data to detect shapes in the environment, such as circles or polygons. These mapping algorithms are typically storage efficient as they extract and store high-level geometries. Figure 1.6b shows the true map in Figure 1.6a represented as lines. This representation works well for the man-made indoor environment in Figure 1.6. However, this representation would struggle to represent unstructured data such as that shown in Figure 1.7.

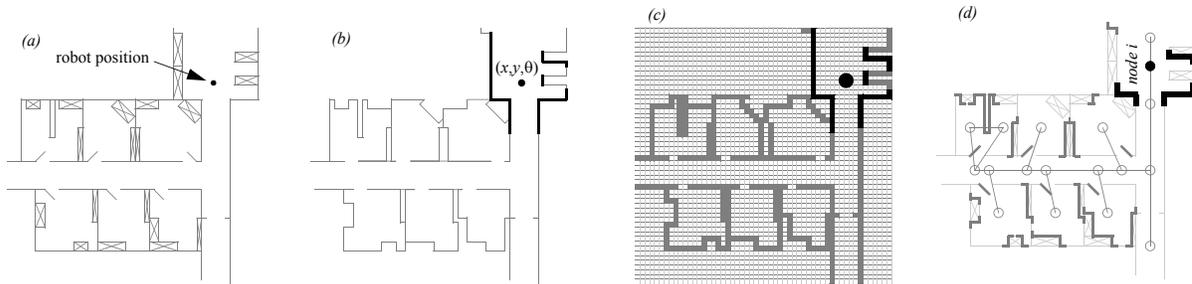


Figure 1.6: Mapping a 2D environment using various mapping techniques. The robot’s currently perceived map is shown in bold. (a) The real map with walls, doors and furniture. (b) A geometric map representation using lines. (c) A grid-based map representation, where dark cells are occupied. (d) A topological map using line features (Z/S shapes) and doors. Image credit: Siegwart et al. (2014).

Figure 1.5 shows the development of a semantic from a grid-based map. Grid-based mapping will be the focus of this thesis and several examples are discussed in Section 1.1.1. However, many maps are generated without a predefined grid and these are discussed in Section 1.1.2. For example, a simple representation of range measurements is a point-cloud where only the location of each reflected point is stored. This method is shown in Figure 1.7a and is able to quickly represent the data and is often sufficient for a robot to interact with its environment. In many cases, objects and features can be extracted from the point-cloud which further aid in more advanced interaction (e.g., localization or grasping). Alternatively, a mesh can be applied to the point-cloud data to create a terrain map. A mesh map representation is popular in video game graphics as surfaces appear opaque.

Both point-cloud and mesh representations summarize the range measurement using the single point where it was reflected and thus neglect the non-reflective or *unoccupied* space that was measured between the source (robot) and the reflective surface. There are several scenarios where the robot would benefit from accurately representing the unoccupied space and differentiating it from *unknown* (i.e., not measured) space. For example, exploration algorithms often seek to explore the boundary between unoccupied and unknown areas (Yamauchi, 1997) or seek to reduce uncertainty in the map (Hou et al., 2019; Stachniss, 2009). Furthermore, some applications will have the robot plan its interaction in the environment (i.e., navigation, grasping, etc.) with a previously generated map. For example, if the robot plans to move through the environment, it would be necessary to plan that movement through unoccupied space as opposed to

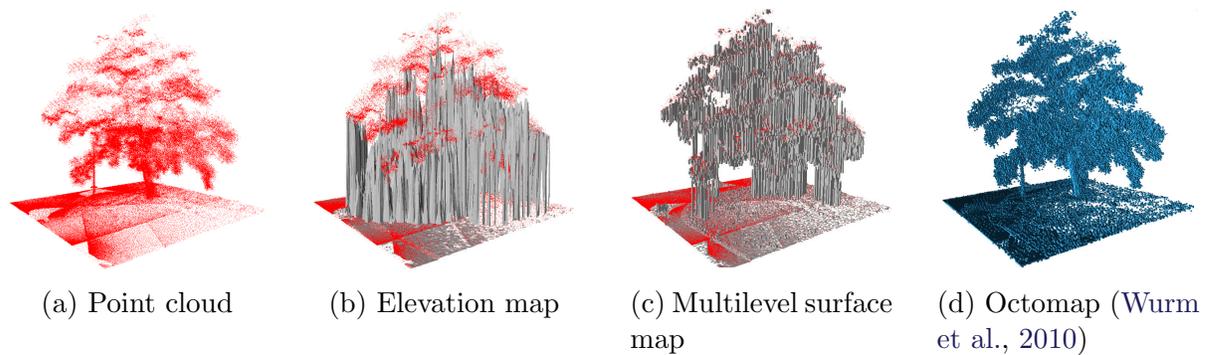


Figure 1.7: 3D representation of a tree using various mapping techniques. (a) Point clouds store a single point per range measurement. (b) Elevation maps discretize the map to a 2D Cartesian grid and assume a single height value per discretization. (c) Multilevel surface maps enable elevation maps to estimate multiple height values per discretization. (d) Octomap is based on OGM and discretizes the 3D environment and estimates the occupancy probability of each discretization. Image credit: Wurm et al. (2010).

unknown areas that have not been mapped. Further applications of maps and metrics to compare them are discussed in Section 1.1.3.

1.1.1 Grid-Based Mapping

Grid-based mapping algorithms begin by discretizing the environment, then proceed to estimate some information about each discretization. The elevation map shown in Figure 1.7b has discretized the environment in a 2D grid, then seeks to estimate the height of each grid cell. This type of representation works well for plains, hills, and other terrain where each grid cell has a single height, but as Figure 1.7b illustrates, it fails to represent environments with *overhang*. A multilevel surface map seeks to overcome this limitation of elevation maps, without the storage burden of discretizing the environment in a 3D grid. As shown in Figure 1.7c, the multilevel surface map is able to improve upon the elevation map by estimating multiple heights per grid cell, but is still not able to capture the numerous heights per grid cell in the tree.

A popular paradigm in robotics to represent the environment is to discretize the environment and subsequently estimate the occupancy of each discrete unit (referred to as a *cell* in 2D and a *voxel* in 3D). Although there are some variations to this approach (such as Reflection Probability Mapping (Tipaldi, 2009) and different discretizations (Wurm et al., 2011)), generally the environment is discretized as a Cartesian grid and the robot

seeks to estimate the occupancy of each discrete unit, or cell. This paradigm is commonly known as Occupancy Grid Mapping (OGM) (Elfes, 1989a; Moravec, 1988; Moravec and Elfes, 1985) and is shown in Figure 1.6c. Occupancy grid mapping has been extensively used in robotics for over 30 years and although it was originally developed with sonar sensors, it has since been used for a variety of range sensors including LiDAR (Limketkai et al., 2003; Yguel et al., 2006), radar (Bauer et al., 2019; Himm et al., 2010; Li et al., 2019; Werber et al., 2015; Weston et al., 2019), and cameras (Li and Ruichek, 2013; Muffert, 2018; Perrollaz et al., 2012; Shankar and Michael, 2020). OGM can be used in 1D, 2D, and 3D environments. However, the storage cost increases exponentially with each dimension, which has prompted some researchers to pursue compression algorithms for this type of mapping algorithm (Andriamahefa, 2017; Cain and Leonessa, 2016; Hornung et al., 2013; Joubert et al., 2015; Li and Ruichek, 2013; Peasley et al., 2012; Wurm et al., 2011, 2010; Zhou et al., 2018). Figure 1.7d illustrates one such compression based on an octree representation that estimates occupied voxels with greater resolution than unoccupied voxels.

For computational and storage efficiency, the occupancies of cells in the map are generally considered independent and are typically only correlated by measurements. However, even when OGM was first developed, Elfes (1989a) noted in the conclusion of his PhD thesis that OGM would benefit if we “maintain and update inter-cell links that provide connectivity information about the set of cells that compose a single object”. Elfes wrote that this would aid in the recognition and handling of objects. In this thesis, we examine the cell independence assumption and explore the benefits of including cell correlations in the posterior and the prior, as well as in the measurements. We show that accounting for these cell correlations yields more accurate maps.

Occupancy grids provide a means of mapping the occupied and unoccupied areas of an environment, given, for example, range measurements gathered by a mobile robot. Moravec and Elfes (1985) first introduced occupancy grids, and Moravec (1988) subsequently derived a more efficient method of storing and updating them. The computed occupancy grid is generally, although not always, thresholded as occupied or unoccupied (on a cell-by-cell basis), then used for path planning, navigation (Elfes, 1987), localization (Fox et al., 1998; Pfaff et al., 2007; Thrun et al., 2001) and other tasks (Elfes, 1989b; Martin and Moravec, 1996). However, knowing the occupancy probability of each map or of each cell lends itself well to entropy and information gain calculations. These calculations are often used in exploration algorithms when evaluating the expected information

gain of scanning a particular location on the map (Hou et al., 2019; Kaufman et al., 2018; Makarenko et al., 2002; Merali et al., 2012; Stachniss and Burgard, 2003). Before using occupancy grids to compute the information in the map, we must ensure that the map contains the correct information, given the noisy sensor measurements and occupancy prior used to construct it.

Occupancy Grid mapping was originally developed as a binary Markov Random Field (MRF), where each cell is considered independent. In the conclusion of his PhD thesis, Elfes (1989a) recognized that “the application of higher-order Markov Random Field estimation methods to Occupancy Grids needs to be investigated.” Recently, Muffert (2018) and Shankar and Michael (2020) have explored a higher-order MRF in OGM to model dependencies between neighbouring grid cells using cameras. Muffert (2018) showed that by using a MRF to bias neighbouring cells to have the same occupancy probability the map was better able to estimate unoccupied space and did slightly better in occupied areas. Shankar and Michael (2020) use a higher-order MRF and loopy belief propagation to correlate cells along measurement rays. In this thesis, we explore higher-order MRFs for OGM by introducing cell correlations in the prior, as well as correlating cells from narrow-beam range measurements. We highlight some drawbacks of this approach and propose a novel data-driven prior for neighbouring cells.

Researchers commonly assume a uniform cell prior (Dhiman, 2019) for OGM, but this thesis explores both a data-driven prior and correlating neighbouring cells in the prior. Researchers have previously tried to represent the prior structure in the environment. For example, Anguelov et al. (2004) use an expectation maximization algorithm to detect doors in an indoor hallway environment. Schaefer et al. (2018a) is able to exploit the structure of a man-made structure to generate straight walls in the environment. Shrestha et al. (2019) use a deep neural network to predict unseen regions of building floor plans using floor plan training data. The data-driven prior highlighted in this thesis offers the ability to introduce cell correlations in the prior beyond straight walls and orthogonal structures.

In an occupancy grid, each cell in the map is occupied or unoccupied for some discretization. However, if a larger discretization is selected, then a cell may be partially occupied. Therefore, there has been some research that discusses the idea of a certain portion of the cell being occupied (Agha-Mohammadi et al., 2019), but this ambiguity is often resolved by further discretizing the environment until an adequate resolution is reached where each cell can be classified as occupied or unoccupied (Hornung et al., 2013;

Wurm et al., 2010). In addition, O’Meadhra et al. (2019) present a method for computing variable resolution OG maps using Gaussian mixture models (GMM). For larger grid resolutions, Biber and Strasser (2003) introduced the normal distribution transform (NDT) for scan matching of 2D laser scans. This method is similar to OGM in that it discretizes the map into a grid, but it then assigns a normal distribution to each cell that locally models the probability of measuring a point in that cell. This approach has since been extended to the concept of a NDT occupancy map (NDT-OM) (Saarinen et al., 2013a; Saarinen et al., 2013) and applied to 3D maps and has proven effective for large maps and large cell (or voxel) sizes (Stoyanov et al., 2013). Furthermore, Saarinen et al. (2013b) highlights the effectiveness of NDT-OM for Monte Carlo localization over traditional OGM. The NDT-OM uses a minimum of 11 variables per cell to estimate the various parameters (mean, upper diagonal of covariance, number of points and occupancy probability). However, Saarinen et al. claim that the NDT-OM is still more storage efficient than traditional OGM because they find a NDT-OM with 80cm cell resolution to be as accurate as an occupancy grid with 10cm resolution; thus implying a storage efficiency of storing 11 variables as opposed to $(8 \times 8 =) 64$. This thesis will introduce a different concept, the data-driven patch prior, which can be limited to any number of patch configurations to consider per cell. Moreover, the patch-based methods presented in this thesis are able to estimate cells with few or even no measurements based on the occupancy of neighbouring cells and do not presume the measurements to be normally distributed in each cell. Saarinen et al. present their results on maps with cells of resolution 40cm to 1.2m, whereas all of the results presented in this thesis are on finer-resolution grids (e.g., 2cm).

NDT algorithms are able to model a 2D as well as a 3D environment. Stoyanov et al. (2013) compare NDT algorithms to traditional OGM and a triangle mesh method of mapping the terrain and find that NDT algorithms are more likely to correctly predict unseen measurements in the sensor’s field of view (FOV). Triangle meshes are a popular method to represent a 3D surface (Thrun et al., 2003), especially in computer graphics. In 2D, the mapped surface is represented by line segments (Chatila and Laumond, 1985). But like point clouds, these methods do not represent the unoccupied or unknown space or differentiate between them. Another popular method to represent a surface is a Gaussian process (GP). Given a 2D grid, a GP can represent the height of the terrain using a Gaussian distribution in function space. The sensor measurements are therefore used to learn the hyperparameters of the GP, which can then be used to estimate the height for

any 2D position. A key criterion for a GP is selecting the type of covariance function associated with the function-space Gaussian PDF, known as the kernel. Lang et al. (2007) and Plagemann et al. (2008) used a nonstationary kernel that was able to handle different surface types. These methods are shown to provide conservative height predictions in occluded areas, but do not differentiate between unknown and unoccupied space.

1.1.2 Mapping without a Grid Structure

Grid-based mapping algorithms, including OGM, require that the map be discretized according to a particular coordinate system. This could introduce some limitation in their ability to model the world. Paskin and Thrun (2005) present a unique occupancy mapping technique using polygon random fields (PRF), which are able to estimate occupancy in the map without a grid structure. PRFs were developed by Arak (1982) and later extended by Arak and Surgailis (1989) to allow the polygons to be in multiple states and further extended by Arak et al. (1993) for more complicated graph structures. Paskin and Thrun were the first to bring the idea from statistical literature and apply it to real data in robotics. Kluszczyński et al. (2007) later applied the technique to images in computer graphics.

Unlike traditional OGM, the method developed by Paskin and Thrun is able to estimate areas of the map that are not mapped directly as the algorithm seeks to fit polygons to the data that interpolate between measurements. The authors use a Metropolis-Hastings MCMC algorithm to sample PRF maps from the posterior and are therefore able to estimate the occupancy of any area in the map by averaging over these samples. This thesis also presents a MCMC algorithm, but it is applied to grid-based occupancy mapping. The grid-based algorithm is easily initialized by traditional OGM methods, whereas the PRF method can take hours to converge to a stationary distribution. In addition, this thesis is able to exploit the grid structure to introduce a data-driven prior that results in maps that better estimate the underlying map. The PRF methods are not confined to a grid resolution, but do have a scale parameter that dictates the smoothing applied to the structures in the map.

A Gaussian Processes (GP) (Rasmussen and Williams, 2006) is a stochastic process that enables high-dimensional regression by using measurements to learn covariance and mean functions. GPs enable Continuous Occupancy Mapping (COM) that estimates a map as a continuous smooth surfaces (O’Callaghan et al., 2009; O’Callaghan and Ramos,

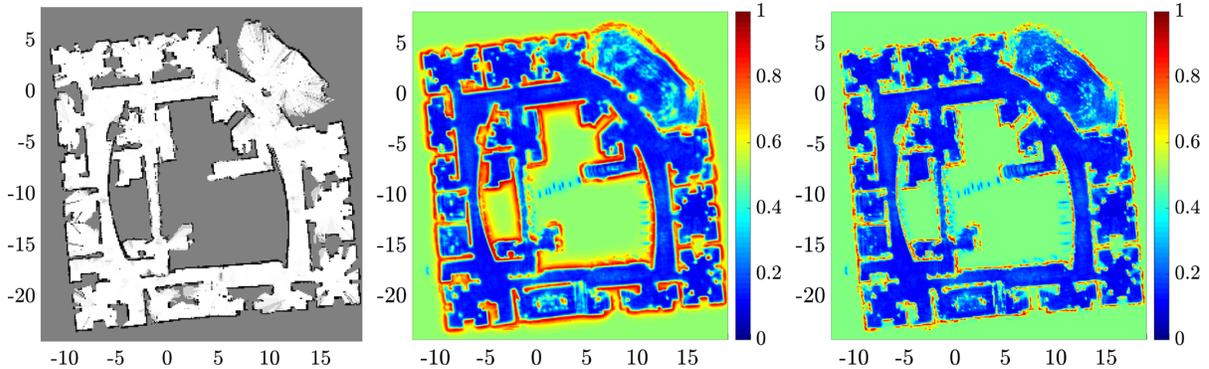


Figure 1.8: A figure from Jadidi et al. (2018) comparing traditional OGM to incremental GPOM on the Intel Lab dataset. The image on the left is traditional OGM and offers more crisp boundaries between occupied and unoccupied regions. The two images on the right are two variants of incremental GPOM and are able to estimate unmapped areas of the environment, but have a smoothing effect on the occupancy maps.

2012). O’Callaghan and Ramos (2012) introduced Gaussian process occupancy mapping (GPOM) that is also able to estimate the occupancy of areas in the map without a grid structure. Unlike traditional OGM, this method is also able to estimate the occupancy of unmapped areas. Furthermore, the GPOM algorithm is *anytime* and thus able to provide accurate representations of large environments at arbitrary resolutions. The MCMC OGM algorithms presented in this thesis are also *anytime* and able to estimate unmapped areas of the map. However, the algorithms presented here provide a finite resolution, but do not smooth sharp features in the environment.

A major drawback of GPOM is its cubic time complexity. Lee et al. (2019) present a Gaussian process implicit surfaces (GPIS) map representation that can be used online by clustering the data using quadtrees (Finkel and Bentley, 1974) and octrees (Meagher, 1982) and training each cluster locally. Jadidi et al. (2018) developed an incremental GPOM which is able to run online and can be used for frontier-based exploration of the map. Figure 1.8 compares traditional OGM to incremental GPOM on the Intel Lab dataset (which is used in this thesis). The GPOM algorithms are able to estimate unmapped areas, but have a smoothing effect on the map and require batch offline optimization of hyper-parameters. OGM algorithms are able to produce maps with a sharp transition from occupied to unoccupied and the algorithms in this thesis address their ability to estimate unmapped areas. GPs have also been used for other map representations such as implicit shape potentials that enable a grasping robot to grasp an object from occluded views (Dragiev et al., 2011). Hilbert maps (Ramos and Ott, 2016) are continuous occupancy maps built by projecting the LiDAR measurements in a Hilbert

space, learning a logistic regression classifier, and then classifying each point in space as free or occupied. However, this approximation for continuous occupancy mapping yields less accuracy than GPOMs as seen in Figure 1.9 that shows Hilbert maps on the same Intel Lab dataset as used in Figure 1.8. Figure 1.9 also illustrates the smoothing effect of Hilbert maps relative to OGM. This smoothing can lead to narrow unoccupied areas being classified as occupied and therefore hindering a navigation algorithm that uses the map for path planning. Senanayake and Ramos (2017) extend Hilbert maps using Bayesian techniques to run faster, but with similar accuracy and Guizilini and Ramos (2018) use Hilbert maps to estimate occluded areas of a map, although it sacrifices some detail in the map in doing so. In addition, Guizilini et al. (2019) have recently used Hilbert maps to predict occupancy in dynamic environments, and Zhi et al. (2019) extend Bayesian Hilbert maps to a method that reduces their cubic complexity to below quadratic.

Schaefer et al. (2018b) have recently presented a mapping technique that stores the map parameters of a decay rate sensor model in the discrete frequency domain. This method is memory efficient but unable to distinguish between unoccupied and unknown areas of the map.

1.1.3 Mapping Applications and Metrics

This thesis shows that traditional OGM techniques can be overconfident in their estimates of the map and this overconfidence can lead to inaccurate information gain calculations. Before developing the mapping techniques presented in this thesis, we used traditional OGM techniques to evaluate next-best-view candidates by computing the expected information gain at the candidate locations (Merali et al., 2012). However, the information gain calculation led to similar results as less computationally expensive methods (Yamauchi, 1997) using traditional OGM. Recently, Hou et al. (2019) have been able to use one of the mapping methods developed in this thesis to better perform information-based next-best-view (NBV) planning. Furthermore, Kaufman et al. (2018) have recently proposed an entropy-based exploration algorithm using occupancy grids and Kaufman et al. (2019) highlight that robots often have limited onboard computing, but may have the ability to send data to an off-board computer for further computation. This thesis develops some *offline* and some *anytime* algorithms that are able to produce more accurate OG maps than traditional OGM.

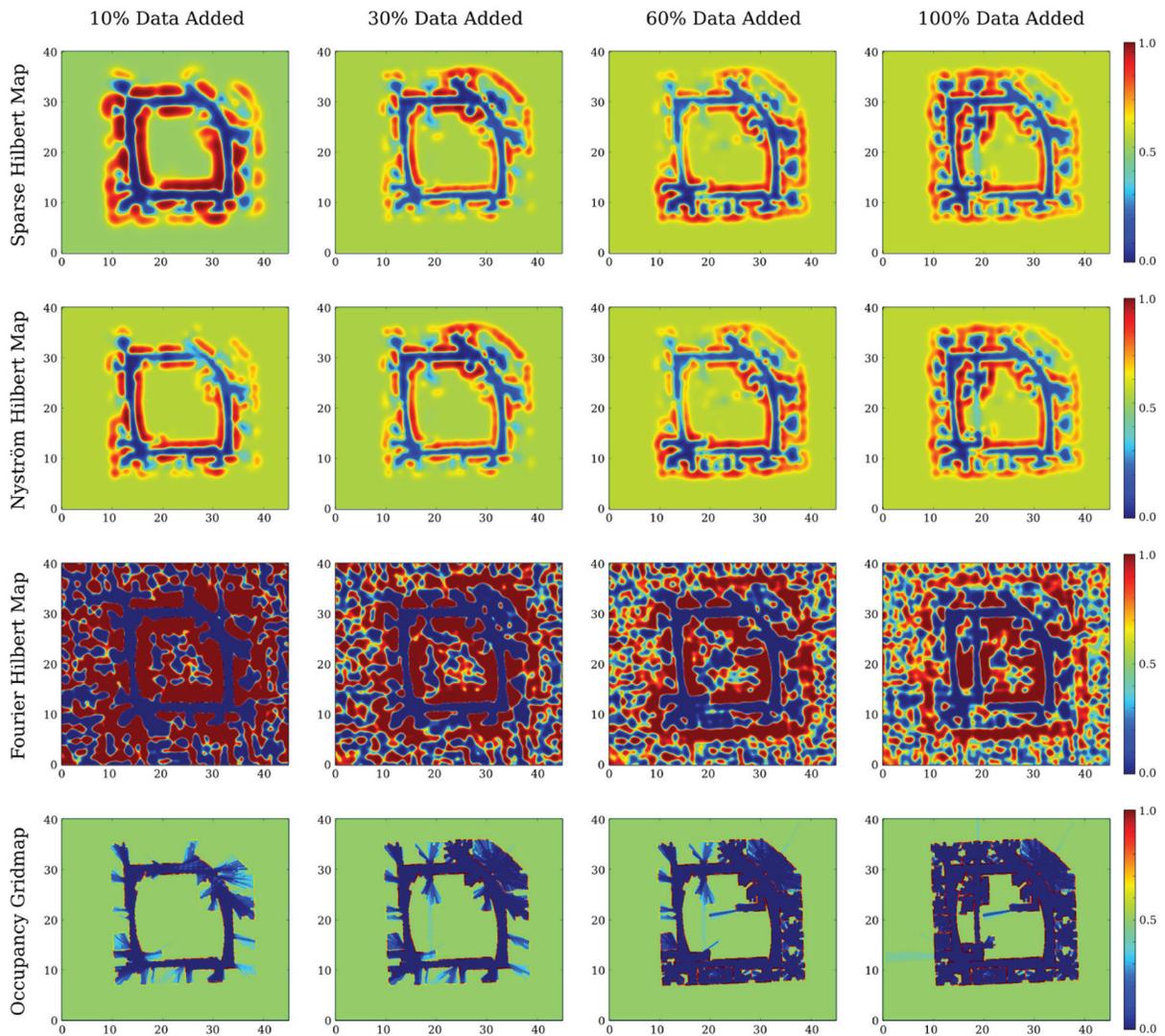


Figure 1.9: A figure from Ramos and Ott (2016) comparing various Hilbert Map algorithms to traditional OGM using the Intel Lab dataset. Hilbert maps are able to estimate unobserved cells whereas traditional OGM techniques are not. The algorithms presented in this thesis extend OGM algorithms to estimate unobserved cells while maintaining or improving the sharpness depicted here for OGM.

Occupancy grid mapping (OGM) is extensively used in mobile robotics because of its wide range of applications and robustness to noise and error in the range measurements (Elfes, 1989a; Thrun, 2002). To quantify how well an occupancy grid mapping algorithm has done, a benchmark occupancy grid is required. Moravec and Cho (1989), Moravec and Blackwell (1992), and Thrun (1998) compared the computed occupancy grid to the ground truth occupancy grid, whose cells are either occupied or unoccupied (available in simulation). However, it is difficult for a mapping algorithm to produce the ground

truth map from a finite number of sparse measurements. Thus this thesis uses the full Bayesian solution (and approximations of it) as a benchmark for OGM. To compare the OG map to a GT map, each cell is typically thresholded at a single value to produce a binary map. Thus, we also examine OGM as a binary classification problem (Georgiou et al., 2017) and compare the metrics of *precision* and *recall* (Davis and Goadrich, 2006) to compare the maps on a cell-by-cell basis for various cell threshold values.

Thrun (2003) proposes a batch hill-climbing algorithm to determine the maximum a posteriori (MAP) estimate of the occupancy grid, given the sensor measurements and respective robot poses. The MAP estimate does not represent the uncertainty in the map as all cells are either occupied or unoccupied. Jang et al. (2006) apply this method to small groups of cells in the map, as opposed to the entire map, to improve the computational efficiency. Thrun (2003) goes a step further and seeks to quantify the uncertainty in the map by computing the residual occupancy uncertainty of each cell in the map.

Although this thesis assumes a static world, several researchers have used OGM for dynamic environments (Hoermann et al., 2018; Li and Ruichek, 2014; Meyer-Delius et al., 2012; Oh and Kang, 2016; Tipaldi et al., 2011; Ziebart et al., 2009) and the research presented here could be extended to a dynamic environment. Whether the environment is dynamic or the robot is moving while mapping, the sensor measurements can be affected by motion distortion. Luo et al. (2019) projects 3D LiDAR data onto a 2D OG to account for motion distortion. This thesis uses a stop-and-scan approach to avoid these effects. OGM lends itself well to cooperative robots sharing the same map as each map is developed in global coordinates as opposed to coordinates relative to the robot. Li et al. (2014) highlight a method to perform map merging with cooperative robots and Mazuran et al. (2014) proposes a measure to determine global consistency in the map. Furthermore, Sodhi et al. (2019) highlights a method to correct the robot pose based on OG map consistency in a SLAM algorithm (Durrant-Whyte and Bailey, 2006). Although this thesis does not address robot localization error or map merging, cooperative robots in communication would also lend themselves to computing higher-fidelity maps using the algorithms developed in this thesis as each robot could compute a portion of the map.

Given the various mapping techniques in the mobile robotics literature, OGM stands out in its ability to map known, unknown, and unoccupied areas of the map. This technique is widely used in mobile robotics and has been improved by numerous methods since its creation. However, the majority of these methods still make an independence

assumption between cells and therefore assume no mutual information between them. This thesis highlights the benefits of keeping these cell correlations and shows novel methods to estimate the full Bayesian solution, which is otherwise intractable for real maps.

1.2 Thesis Background and Overview

The 3D site mapping project began in 2008 as a collaboration between the University of Toronto (UTIAS), industry (MDA Space Missions) and the Canadian Space Agency (CSA), with funding from the Canadian government (NSERC). The project came to a successful conclusion in 2011 with a site-mapping mission on the CSA’s Mars Emulation Terrain (MET), which measures $60\text{m} \times 120\text{m}$ and includes inclines, rocks, cliffs, and a 5.5m-diameter crater (Merali et al., 2012). The research conducted was broadly divided into mapping and planning. The mapping research was led by Chi Hay Tong, and the planning was led by the author. The next-best-view (NBV) planning research was presented at several conferences (SVAR-2009, SVAR-2010, SVAR-2011, i-SAIRAS-2012, SVAR-2012, SVAR-2013) and served to motivate our interest in capturing the residual uncertainty in occupancy grid mapping (OGM).

Therefore, this thesis focuses on occupancy grid mapping (OGM) and provides a study of cell correlations in this popular mapping paradigm. This thesis assumes that the robot moves in a static world and that each robot pose is known. These assumptions enable a study of the mapping technique without the dynamics of a changing environment or pose uncertainty for range measurements. Therefore, the experiments in this thesis are conducted using a stop-and-scan approach to minimize or eliminate pose uncertainty and dynamic effects. This thesis highlights that OGM is a complex Bayesian inference problem and is intractable for even simple 2D maps like that shown in Figure 1.10. This thesis applies both variational inference (VI) and Markov Chain Monte Carlo (MCMC) methods to estimate the true posterior in OGM.

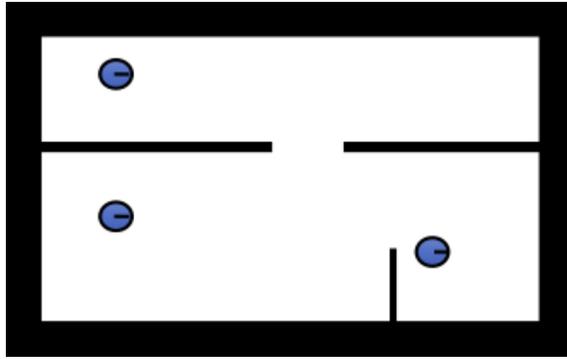
Figure 1.10 highlights some of the results of this thesis on a simple 2D example. Figure 1.10a shows the ground truth map and the three robot scan locations. A series of narrow-beam range measurements are taken from each of these locations. Given the range measurements and corresponding robot pose, a mapping algorithm seeks to estimate the underlying map. Figure 1.10b illustrates the result of applying traditional OGM and the other four are results from algorithms presented in this thesis using the same

measurements. Figure 1.10g shows that each cell is displayed on a scale from white, to yellow, to red, to black, to represent the occupancy probability of each cell (darker cells are more likely to be occupied); this scale is used to visualize OG maps throughout this thesis.

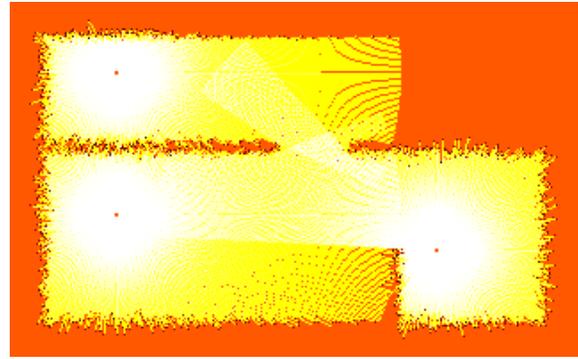
In Chapter 2, this thesis will introduce the mathematical formulation of occupancy grid mapping. Since OGM estimates the map probabilistically, Section 2.2 will discuss a method of computing the most likely occupancy grid from measurement data. Section 2.3 will derive the full Bayesian solution for OGM and estimate the probability of each map (as opposed to cell) without any assumptions. We first published this work in Merali and Barfoot (2012) and it highlights the computation and storage burden of the paradigm without simplifying assumptions. Section 2.4 will then introduce the commonly made assumption in OGM that allows it to run efficiently online. Section 2.5 will apply variational inference techniques to traditional OGM by optimizing the update term to better capture the uncertainty in each cell. This improvement is observable in comparing Figure 1.10c to Figure 1.10b and we published this work in (Merali and Barfoot, 2014). Although traditional OGM maintains an estimate of the occupancy of each cell, it is important to note that in a real map, each cell will be either occupied or unoccupied for some discretization.

Chapter 2 concludes with a discussion of cell correlations in the sensor model in Section 2.6 that highlights that cell correlations are local. Figure 1.11 illustrates the mutual information between proximate cells and shows that the mutual information approaches zero as cells are farther apart. Traditional OGM assumes that cells are independent and therefore assumes zero mutual information in the posterior, but this thesis explores various methods of retaining these cell correlations to improve the map estimate.

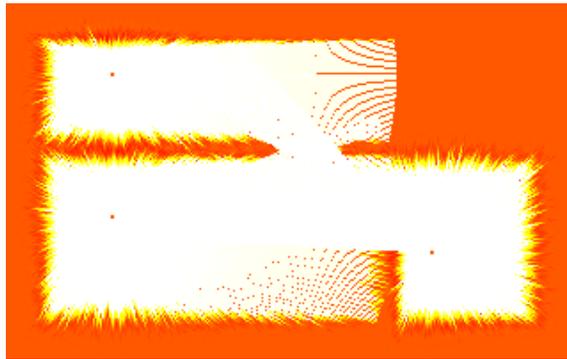
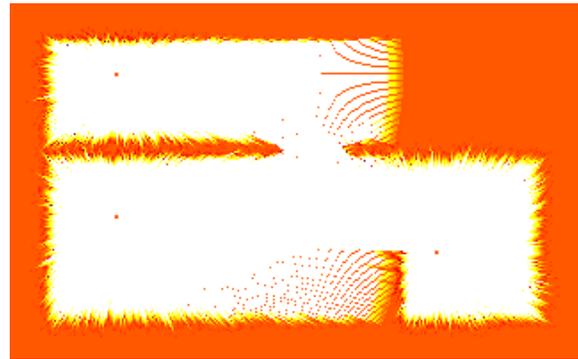
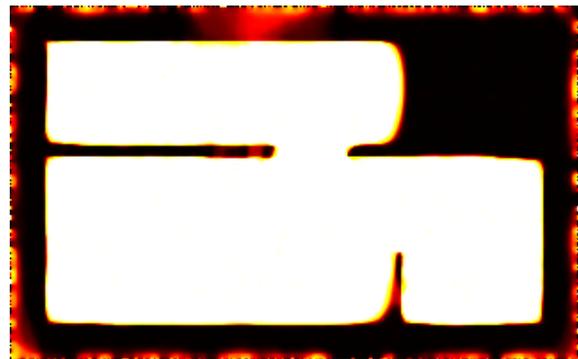
Given that cell correlations are local, Section 3.1 extends traditional OGM to compute the probability of a *patch* of cells as opposed to individual cells. This method assumes that cells in the same patch are fully correlated, but that patches are independent of one another. This extension is novel, but unpublished. Section 3.2 extends the online patch map algorithm to an offline, batch solution that is more computationally expensive, but is better able to capture the residual uncertainty in the map. We first published the offline patch map solution in Merali and Barfoot (2012) and its ability to better capture the residual uncertainty in the map is seen in Figure 1.10d.



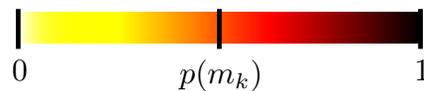
(a) Ground Truth



(b) Traditional OGM

(c) Section 2.5: Traditional OGM with optimized Δ (d) Chapter 3: Offline patch map with K patches(e) Chapter 4: MCMC with a cell prior of $p(m_k)=0.234$ 

(f) Chapter 5: MCMC with border-based, data-driven, patch prior



(g) Legend for occupancy grids presented throughout this thesis

Figure 1.10: Various OGM algorithms presented in this thesis from the same 2D data. Each algorithm is trying to estimate the ground truth map based on range measurements. Algorithms (c), (d), (e), and (f) are novel to this thesis.

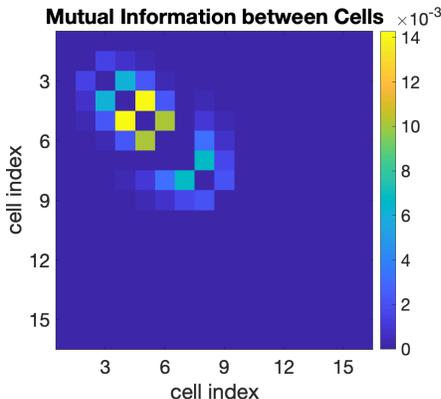


Figure 1.11: Mutual information between cells in the posterior using the full Bayesian solution to OGM. Note that values are higher for proximate cells and zero for distant cells. The cell independence assumption in traditional OGM assumes no mutual information between cells.

Chapter 4 discusses a method called Markov Chain Monte Carlo (MCMC) and its ability to sample from the full posterior. Using the same input data as traditional OGM, this method is better able to capture the cell correlations in the posterior and does not suffer from the negative effects of patch-boundaries as the patch map algorithms do. Furthermore, the MCMC method continues to improve its estimate as more samples are drawn from the posterior. We first published this work in Merali and Barfoot (2013), but generalize the approach in Section 4.2 to sample patches of cells. The result of the MCMC algorithm is illustrated in Figure 1.10e, but the cell prior has been decreased from 0.5 to a more realistic 0.234 and thus unmapped cells appear lighter in the figure (i.e., have a lower occupancy probability).

Chapter 5 introduces cell correlations in the prior. By sampling patches of cells from realistic maps, we obtain a data-driven patch prior that better represents prior cell correlations in OGM. Using the MCMC algorithm developed in Chapter 4, we apply a data-driven prior to each patch sampled by the algorithm. The result is a map that better estimates the original map as seen in Figure 1.10f. This work is novel and being prepared for publication. It is worth noting that both the MCMC algorithm and the data-driven prior estimate will only benefit from more samples and thus these algorithms lend themselves well to the growing trend of faster computing on board the robot and computing these results on more powerful computers that are not physically on the robot (Kaufman et al., 2019).

Throughout this thesis, each of the algorithms are explained with 1D toy examples and simple 2D examples like that depicted in Figure 1.10. Finally, Chapter 6 highlights the results of the algorithms presented in this thesis on large scale 2D data – both simulated and hardware data. The simulated data provides perfect robot localization, whereas the hardware data has a localization accuracy of 2-3cm. The simulated data also provides a range of maps to showcase that the algorithms work in various environments. We highlight that introducing a data-driven, patch prior is particularly beneficial when data is sparse and this result is also observed in Figure 1.10f in that the algorithm is able to estimate the occupancy of cells “between” measurements.

1.3 Thesis Contributions

The novel contributions presented in this thesis are:

- Section 2.3.1 presents a method of computing the full Bayesian solution for OGM by constraining the robot’s pose. This can be computed for a 1D map of any size and is useful in evaluating other OGM algorithms.
- Section 2.5 optimizes traditional OGM to capture the residual uncertainty, resulting in more accurate maps without sacrificing the speed of traditional methods. This work was presented and published at ICRA-2014 (Merali and Barfoot, 2014).
- Section 3.1 introduces *Online Patch Map*, which generalizes traditional OGM to estimate patches of cells. This formulation produces more accurate maps by enabling the posterior to keep cell correlations for cells in the same patch (novel but not published).
- Section 3.2 introduces *Offline Patch Map*, which is a batch solution to better estimate the residual uncertainty in the map. This method is a useful benchmark for estimating the full Bayesian solution in 2D. This research was presented and published at IROS-2012 (Merali and Barfoot, 2012).
- Chapter 4 uses MCMC Gibbs sampling to estimate the full Bayesian solution in OGM, which was first presented and published at ICRA-2013 (Merali and Barfoot, 2013). Given enough samples, this *anytime* algorithm can produce more accurate maps than patch map methods.

- Chapter 5 introduces a data-driven patch prior and a method to correlate neighbouring patches in MCMC OGM. This extension to MCMC OGM is faster and more accurate than the algorithm without this extension (being prepared for publication).

Chapter 2

Occupancy Grid Mapping (OGM) Fundamentals

Chapter 1 highlighted the importance of mapping in robotics. This thesis is focused specifically on occupancy grid mapping, which is a technique that discretizes the environment and seeks to estimate the true map based on the occupancy of each discrete unit. This technique was introduced in 1985 and is still heavily used in robotics today. This method lends itself to a variety of range sensors such as sonar (Nagla et al., 2012; Thrun, 2003), radar, cameras (Pathak et al., 2007; Perrollaz et al., 2012), infrared, but this thesis will focus on LiDAR as this technique has a narrow sensing frustum, can measure long distances and is becoming more ubiquitous in robotics as its cost decreases and capabilities increase.

This chapter begins by describing the problem mathematically in Section 2.1. Discretizing an analog environment into binary cells (or voxels) results in a finite number of cell combinations to represent the environment. OGM refers to each combination as a potential map of the environment and seeks to estimate the likelihood of each map. However, the number of possible maps is large and it is therefore difficult to estimate the likelihood of each of them. Many applications are simply concerned with the most likely map and therefore Section 2.2 describes how it is possible to focus on the single most likely map, which can be computed relatively quickly. In Section 2.3 we develop the mathematical formulation to estimate the probability of every possible map, but highlight that it becomes computationally intractable for reasonably sized maps. Therefore, Section 2.3.1 examines a special case of this formulation and shows that the probability of every map can be computed if the robot is stationary. Although this formulation is

not useful for real experiments, it provides a useful benchmark to compare other OGM techniques and is used throughout this thesis. For example, since OGM was first developed, it has focused on estimating the occupancy probability of each cell (as opposed to each map) by assuming that cells in the map are independent and that measurements are conditionally independent. These assumptions and this formulation is presented in Section 2.4 and compared to the benchmark developed for a stationary robot. Section 2.5 goes a step further and uses this benchmark to develop a novel update technique for traditional OGM. The new update function can be used in existing OGM algorithms to better estimate the uncertainty in each cell with no increase in computational cost and it better estimates the uncertainty in each cell. Finally, Section 2.6 highlights that the sensor introduces cell correlations in the map, but traditional OGM methods disregard these correlations by assuming cells are independent. However, the cells are only locally correlated, which prompts the algorithms presented in subsequent chapters to improve OGM by retaining local cell correlations.

2.1 OGM Formulation

In this section, we develop the OGM formulation used throughout this thesis. Given a set of range measurements, z , and corresponding robot poses, x , mapping algorithms seek to estimate the map, m . OGM is a commonly used mapping framework that represents the unoccupied space as well as the occupied areas measured by range sensors. Specifically, these algorithms discretize an analog environment, into a regular grid of K cells (or voxels), $m_{1:K}$. Each cell in the map is represented by a binary random variable, m_k , where $k = 1 \dots K$, that indicates whether the cell is *occupied*, $m_k = 1$, or *unoccupied*, $m_k = 0$. The true map, known as the ground truth map, correctly indicates the value of each m_k . An occupancy grid mapping algorithm seeks to determine the map, m^r , that best represents the world, given a set of N scalar range measurements, $z_{1:N}$, and corresponding robot pose vectors, $x_{1:N}$. To simplify the notation, we will not include the subscript $1:N$ when referring to the set of all measurements or poses, just as all K cells in the map are referred to as m . Each robot pose, x_n , represents the position and orientation of the robot and is therefore a vector representing each degree of freedom of the robot – in a 2D environment this is typically the x- and y-position and the yaw angle for orientation. As most narrow-beam range sensors measure the distance to the first occupied cell, f , (and not beyond that) the occupancy of cells inside walls or obstacles cannot be measured. The ground

truth map indicates the true state of these unmeasurable cells, but a mapping algorithm cannot use range measurements to estimate the state of these cells. It is therefore typical in the literature to refer to these cells as *unknown* and estimate them at the cell prior (Collins et al., 2007) as opposed to occupied or unoccupied.

Therefore, OGM algorithms are concerned with using Bayesian methods to compute the *probability* of each map, $p(m^r)$, given the range measurements and corresponding robot poses,

$$p(m^r|z, x). \quad (2.1)$$

Since the map is discretized into K cells and a binary random variable is used to represent the occupancy of each cell, there are therefore $R=2^K$ possible occupancy grids. We use a superscript to represent these maps: m^r , where $r = 1 \dots R$. Of the R possible maps, there is one true map, known as the ground truth (GT) map. The full Bayesian solution to OGM therefore seeks to estimate the probability of all possible maps given the range measurements and their corresponding pose, which can become computationally intractable when the number of cells is large. Therefore, OGM methods seek to estimate Equation (2.1) for all values of r , but it is computationally intractable for realistic maps. This thesis employs both variational inference and MCMC techniques to estimate this quantity better than traditional OGM methods, as outlined in the following chapters. In addition, Chapter 5 will introduce the use of prior information that can be used to improve the map estimates.

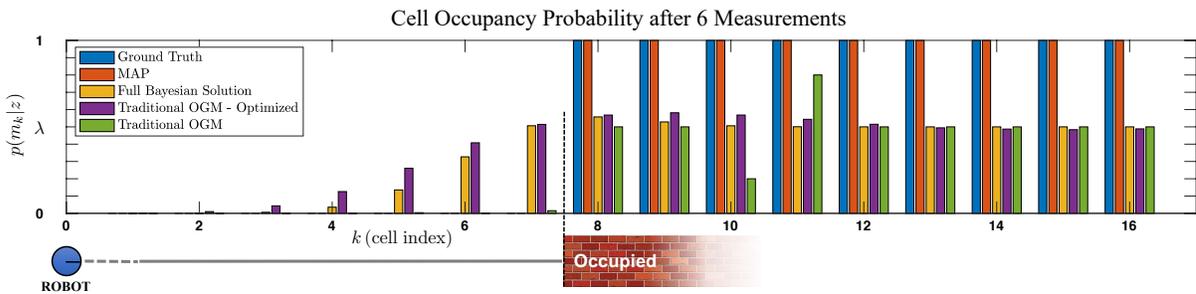


Figure 2.1: Occupancy grid after six measurements drawn from $p(z|f) = \mathcal{N}(f, d^2)$. The robot is at $k = 0$, the true first occupied cell is at $k = 8$ (therefore $f = 8$). The full Bayesian solution estimates all $R = 2^K$ possible maps, but the marginal cell occupancy is illustrated here. The ground truth map is perfectly confident about the occupancy of cells that can be mapped (irrespective of the measurements). The MAP estimate also does not represent uncertainty, but is obtained from the measurements. The traditional OGM algorithm is shown for comparison with both an update term from the literature and optimized values.

For even a small number of cells in the map, K , the true Bayesian posterior is difficult to illustrate for all possible maps, $R=2^K$. Therefore, throughout this thesis we illustrate the marginal cell occupancy, $p(m_k|z, x)$, as we showed in Figure 1.10. Because many of the algorithms in this thesis aim to accurately capture the residual uncertainty in each cell, a 1D occupancy grid is often depicted to illustrate the marginal cell probabilities on a bar plot, such as the one seen in Figure 2.1. This figure illustrates a 1D map, where the robot is stationary at cell m_0 and the wall, or first-occupied-cell, is at cell m_8 . The sensor model will be discussed in more detail in Section 2.6, but unless stated otherwise, this thesis will model the sensor as having Gaussian noise about the true range, f_n , and standard deviation of σ cells, $p(z_n|f_n) = \mathcal{N}(f_n, \sigma^2)$. Thus, in the example in Figure 2.1, the robot has taken six range measurements and seeks to estimate $p(m|z_{1:6})$ and the figure depicts the marginal occupancy probability of each cell, $p(m_k|z_{1:6})$ using various algorithms. This example will be repeated several times in this thesis to compare and contrast the various OGM algorithms.

2.2 Maximum a Posteriori (MAP)

Most OGM algorithms provide a method to estimate the probability of each of the possible maps, but [Thrun \(2003\)](#) introduces a method of computing the single most likely map, known as the maximum a posteriori (MAP) estimate, \tilde{m} , by maximizing $p(z|x, \tilde{m})$. It is an iterative Expectation-Maximization (EM) algorithm that begins with a random map, $\tilde{m}^{(i=0)}$, and updates a single cell, $\tilde{m}_k^{(i)}$, given the occupancy of all other cells in the map at the previous iteration, $\tilde{m}_{-k}^{(i-1)}$. The notation $\tilde{m}_{-k}^{(i-1)}$ indicates an estimate of all cells in the map except cell m_k at iteration $(i-1)$. The algorithm iterates through all K cells in the map several times and performs hill-climbing until each cell reaches a steady-state value of 0 or 1. [Thrun](#) computes the MAP estimate for sonar sensors, which have a wide sensing frustum and may measure the distance to one of multiple obstacles, or measure the absence of obstacles. Because any one of several obstacles could have caused the sensor measurement, [Thrun](#) accounts for multiple possibilities. In this thesis, however, we only consider narrow-beam range sensors (e.g., laser rangefinders) that measure the distance to only the first occupied cell, f , along the measurement ray. Thus, computing

the MAP estimate is simplified. To compute the MAP estimate, we recognize that

$$p(z|x, \tilde{m}) = \frac{p(\tilde{m}_k|z, x, \tilde{m}_{-k})p(z|x, \tilde{m}_{-k})}{p(\tilde{m}_k|x, \tilde{m}_{-k})}. \quad (2.2)$$

From Equation (2.2) we note that $p(z|x, \tilde{m}_{-k})$ does not depend on the occupancy of cell \tilde{m}_k and the denominator is simply the prior occupancy probability of each cell. Therefore, we rearrange Equation (2.2), and seek to maximize

$$\begin{aligned} p(\tilde{m}_k|z, x, \tilde{m}_{-k}) &= \frac{p(\tilde{m}_k|x, \tilde{m}_{-k})p(z|x, \tilde{m})}{p(z|x, \tilde{m}_{-k})} \\ &= \frac{p(\tilde{m}_k|\tilde{m}_{-k}) \prod_{n=1}^N p(z_n|x_n, \tilde{m})}{p(z|x, \tilde{m}_{-k})}, \end{aligned} \quad (2.3)$$

where we recognize that the occupancy of cell, \tilde{m}_k , is independent of the set of robot poses, x . Note that we have assumed that each measurement, z_n , is independent given the occupancy of all cells in the map, \tilde{m} . This assumption is realistic for a static world, unlike the measurement independence assumption made in traditional occupancy grid mapping that will be discussed in Section 2.4. Computing the denominator of Equation (2.3) is avoided by introducing the *log-odds* notation. That is, for a random variable, y , the log-odds of y is

$$l(y) = \log \left(\frac{p(y)}{1 - p(y)} \right).$$

Note that the probability of y can still be recovered from the log-odds notation:

$$p(y) = \frac{\exp(l(y))}{1 + \exp(l(y))}.$$

Therefore, in the log-odds domain, we use Equation (2.3) to compute

$$\begin{aligned} l(\tilde{m}_k|z, x, \tilde{m}_{-k}) &= \log \left(\frac{p(\tilde{m}_k=1|z, x, \tilde{m}_{-k})}{p(\tilde{m}_k=0|z, x, \tilde{m}_{-k})} \right) \\ &= l(\tilde{m}_k|\tilde{m}_{-k}) + \sum_{n=1}^N \log \left(\frac{p(z_n|x_n, \tilde{m}_{-k}, \tilde{m}_k=1)}{p(z_n|x_n, \tilde{m}_{-k}, \tilde{m}_k=0)} \right). \end{aligned} \quad (2.4)$$

Computing $p(z_n|x_n, \tilde{m}_{-k}, \tilde{m}_k)$ in Equation (2.4), amounts to evaluating the sensor model because together \tilde{m}_{-k} and \tilde{m}_k give the occupancy of every cell in the map, \tilde{m} . Given the occupancy of all cells, and the robot pose, x_n , we can extract the first occupied cell, f_n . The sensor model will be discussed in more detail in Section 2.6, but unless stated otherwise, this thesis will model the sensor as having Gaussian noise about the true range, f_n , and standard deviation of σ cells, $p(z_n|f_n) = \mathcal{N}(f_n, \sigma^2)$.

The term $l(\tilde{m}_k|\tilde{m}_{-k})$ in Equation (2.4) is equal to zero if we assume a uniform prior and cell independence in the prior. Therefore, the term is set equal to zero for our experiments, but the uniform prior assumption is revisited in Chapter 5. To compute the MAP estimate of the occupancy grid, we employ an Expectation-Maximization (EM) algorithm (Dempster et al., 1977). The E-Step, or expectation, is calculated using Equation (2.4). After computing $l(\tilde{m}_k|z, x, \tilde{m}_{-k})$, the M-Step, or maximization, sets the value of cell $\tilde{m}_k^{(i)}$ according to

$$\tilde{m}_k^{(i)} = \begin{cases} 1 & \text{if } l\left(\tilde{m}_k|z, x, \tilde{m}_{-k}^{(i-1)}\right) > l(\lambda) \\ 0 & \text{if } l\left(\tilde{m}_k|z, x, \tilde{m}_{-k}^{(i-1)}\right) < l(\lambda) \\ \tilde{m}_k^{(i-1)} & \text{otherwise} \end{cases} ,$$

where $l(\lambda)$ is a constant value and commonly set to 0. The new value of $\tilde{m}_k^{(i)}$, is then used to compute the next cell in the grid, $\tilde{m}_{k+1}^{(i)}$. After numerous iterations of all cells in the map, the solution converges to a local maximum likelihood estimate. The algorithm works well when the initial map is randomly generated and it converged to a global maximum in each case we tested. However, we observed that convergence is faster when the initial map has all cells unoccupied (as did Thrun (2003)). We also observed that changing the prior has the largest effect on the unmapped cells. Our offline patch map algorithm, described in Chapter 3, requires that the unknown cells be marked as occupied to ensure that they will reflect the range sensor. Therefore, we sacrifice the convergence speed of the MAP estimate and initialize all cells as occupied, $\tilde{m}_k^{(i=0)} = 1, \forall k = 1 \dots K$. In our 1D toy example, the MAP solution is shown to exactly estimate the ground truth map in Figure 2.1.

2.3 Full Bayesian Solution

Occupancy grid mapping algorithms are concerned with using Bayesian methods to compute the *probability* of each map given a set of range measurements and corresponding

robot poses, $p(m^r|z, x)$. Section 2.2 highlighted a method of computing which of these maps is the most probable, but did not compute the probability of each map. Besag (1986) illustrates that the MAP solution often inaccurately represents the marginal likelihood of a specific random variable (the occupancy of a cell in this case). Thus, it would be more effective, although more computationally expensive, to estimate the probability of every possible map, than to compute marginal probabilities. The probability of each map, m^r , given the range measurements and corresponding poses, is computed as,

$$\begin{aligned} p(m^r|z, x) &= \frac{p(z_n|m^r, z_{1:n-1}, x)p(m^r|z_{1:n-1}, x)}{p(z_n|z_{1:n-1}, x)} \\ &= \frac{p(z_n|m^r, x_n)p(m^r|z_{1:n-1}, x_{1:n-1})}{\sum_{j=1}^R p(z_n, m^j|z_{1:n-1}, x)} \\ &= \frac{p(z_n|m^r, x_n)p(m^r|z_{1:n-1}, x_{1:n-1})}{\sum_{j=1}^R p(z_n|m^j, x_n)p(m^j|z_{1:n-1}, x_{1:n-1})}. \end{aligned}$$

This equation is written recursively to incorporate each new sensor measurement, z_n , incrementally. In this chapter, we assign a uniform prior probability for each map, $p(m^r) = 1/R$, but we revisit this assumption in Chapter 5.

Similar to Section 2.2, we introduce the hidden state of the first occupied cell to write $p(z_n|m^r, x_n)$ as the sensor model, $p(z_n|f_{r,n})$, where $f_{r,n}$ is dependent on the map, m^r , and the robot pose, x_n . Hence, the occupancy grid mapping algorithm reduces to

$$p(m^r|z, x) = \frac{p(z_n|f_{r,n})p(m^r|z_{1:n-1}, x_{1:n-1})}{\sum_{j=1}^R p(z_n|f_{r,n})p(m^j|z_{1:n-1}, x_{1:n-1})}. \quad (2.5)$$

Note that the denominator is a normalizing constant that does not increase the computational complexity since the numerator is computed for all values of r . Computing Equation (2.5) of each of the R maps, will yield the probability of every possible map. As we illustrated in Figure 1.10 and Figure 2.1, occupancy grid maps are often visualized and compared on a cell-by-cell basis. Therefore, the marginal probability of a cell being occupied is calculated by summing the map probabilities for those maps in which the cell

of interest is occupied:

$$p(m_k|z, x) = \sum_{r=1}^R p(m_k|m^r)p(m^r|z, x), \quad (2.6)$$

where $p(m_k|m^r) \in \{0, 1\}$. This can be used as a benchmark for occupancy grid mapping algorithms, introduced in later sections, that only seek to compute the probability of each cell being occupied, as opposed to the probability of a map. However, the computation time for this formulation is exponential in the number of cells in the map, K , because $R = 2^K$. As a result, this formulation is limited to computing the full solution for a map with a small number of cells (e.g., $K=20$ cells requires several hours). Hence, this solution is only suitable for small 1D maps, and not for any reasonably sized 2D or 3D maps.

2.3.1 Pose-Constrained Full Solution

In the preceding section, we presented the full Bayesian solution to OG mapping but noted that it is computationally expensive to compute for R maps. In this section, we show that by constraining the robot’s pose in 1D, we only need to compute the full solution for K unique maps. By introducing the hidden state of the first occupied cell, $f=1 \dots F$, we are able to compute the probability of each map given range measurements and corresponding poses, $p(m^r|z, x)$. Throughout this thesis, we assume that the range sensor is a narrow-beam sensor such as a laser rangefinder. Thus, the sensor will measure the distance to the first occupied cell along the measurement ray, as opposed to a sensor with a wide sensing frustum (e.g., ultrasonic range sensor), which Thrun (2003) handles by taking the expectation over possible obstacles in the sensing frustum.

Therefore, given the probability of each map from Equation (2.5), computing the probability of R maps is computationally expensive. Given that it is common for a range sensor measurement to trace more than 50 cells per measurement, it would be computationally intractable to compute the full solution of even a 1D map of this size. However, by assuming the robot’s pose is fixed, we note that Equation (2.5) only yields K unique solutions. Specifically, $p(m^r|z, x)$ is different for each value of f . Therefore the R possible maps can be grouped by their respective value of f , $m^r \in f$. To compute

$p(m^r|z)$, we compute the K possible solutions and then multiply by the frequency of that value of f in the denominator,

$$p(m^r|z) = \frac{p(z_n|f_r)p(m^r|z_{1:n-1})}{\sum_{j=1}^K 2^{(K-j)}p(z_n|f=j)p(m^j|z_{1:n-1})},$$

where the notation m^j refers to any $m^r \in f = j$. This equation is written recursively to incorporate each new sensor measurement, z_n , incrementally. Here, we use a uniform prior probability for each map, $p(m^r) = 1/R$, but revisit this assumption in Chapter 5.

Finally, the probability of a cell being occupied is calculated by summing the map probabilities for those maps in which the cell of interest is occupied:

$$p(m_k|z) = \sum_{r=1}^R p(m_k|m^r)p(m^r|z), \quad (2.7)$$

where $p(m_k|m^r) \in \{0, 1\}$. This equation can be used to compare the full solution to traditional OG mapping algorithms that only calculate the probability of a cell being occupied, as opposed to the probability of a map. This result is highlighted in Figure 2.1; although the full solution approaches the ground truth map as the number of measurements increases, it has not yet converged after six measurements. We should not be more confident in the estimate of a cell's occupancy than the full Bayesian solution allows. Hence, having a Bayesian mapping algorithm converge to the ground truth given a finite set of measurements, would imply that the algorithm is overconfident in its estimate. This conclusion is further evidenced in cells with no measurements; the ground truth map will be perfectly confident about the cell's occupancy, whereas the full solution will remain at the prior. Note that computing the full solution is typically exponential in the number of cells in the map, K , because $R = 2^K$. However, the pose-constrained formulation presented here allows us to compute the probability of only K maps. Therefore, we can use this formulation to compare any size map in 1D against the full Bayesian solution, as we will demonstrate throughout this thesis. While this does not admit a practical algorithm for real-world use, it is a convenient benchmarking tool in one dimension.

2.4 Traditional Occupancy Grid Mapping (OGM)

The full solution derived in Section 2.3 is too computationally expensive to be used in real time and on realistic 2D and 3D maps. Therefore, traditional occupancy grid mapping (Elfes, 1989a; Moravec, 1988; Moravec and Elfes, 1985) makes simplifying assumptions to reduce the computation time to be linear in the number of cells that the measurement, z_n , traverses.

The first assumption is that the occupancy of a cell is independent of all other cells:

$$p(m) = \prod_{k=1}^K p(m_k).$$

This assumption results in losing any mutual information between cells, but dramatically reduces the computational and storage burden because the problem is reduced to computing $p(m_k|z, x)$ for K cells, as opposed to computing $p(m^r|z, x)$ for $R=2^K$ maps:

$$p(m_k|z, x) = \frac{p(z_n|m_k, z_{1:n-1}, x)p(m_k|z_{1:n-1}, x)}{p(z_n|z_{1:n-1}, x)}.$$

This equation is written iteratively to incorporate each new measurement, z_n , using the previously computed cell occupancy probability, $p(m_k|z_{1:n-1}, x)$.

The second assumption is that measurements are conditionally independent, given the occupancy of a single cell of interest:

$$p(z|m_k) = \prod_{n=1}^N p(z_n|m_k). \quad (2.8)$$

However, this assumption made in traditional occupancy grid mapping is not strictly true, even in a static world (Kaufman et al., 2016). Note that this assumption differs from the static-world assumption, made in Section 2.3, that measurements are independent given the occupancy of all cells in the map,

$$p(z|m) = \prod_{n=1}^N p(z_n|m).$$

But traditional OGM uses this second assumption because it reduces the cell update

equation to

$$\begin{aligned} p(m_k|z, x) &= \frac{p(z_n|m_k, x_n)p(m_k|z_{1:n-1}, x_{1:n-1})}{p(z_n|z_{1:n-1}, x)} \\ &= \frac{p(m_k|z_n, x_n)p(z_n|x_n)p(m_k|z_{1:n-1}, x_{1:n-1})}{p(m_k|x_n)p(z_n|z_{1:n-1}, x)}. \end{aligned} \quad (2.9)$$

Computation of the terms that do not depend on m_k in this expression are avoided by using the log-odds domain, and therefore this formulation is often referred to as the log-odds formulation of OGM in the literature:

$$l(m_k|z, x) = \underbrace{l(m_k|z_{1:n-1}, x_{1:n-1})}_{\text{stored value}} + \underbrace{l(m_k|z_n, x_n)}_{\text{update term}} - \underbrace{l(m_k)}_{\text{cell prior}}. \quad (2.10)$$

The three terms in this equation are easily computed. The first term, $l(m_k|z_{1:n-1}, x_{1:n-1})$, is computed at the previous iteration and stored in the evidence grid (an *evidence grid* is simply an occupancy grid in the log-odds domain (Martin and Moravec, 1996)). The second term, $l(m_k|z_n, x_n)$, is the *update term* that must be computed at each iteration. Finally, the third term, $l(m_k)$, is a constant based on the prior for each cell. Setting the prior to $p(m_k) = \lambda = 0.5$ (maximum entropy prior) ensures the third term is equal to zero. This assumption is not necessarily correct and is explored in further detail in Chapter 5, but it is commonly used in the literature (Dhiman, 2019; Xu et al., 2019).

For the update term, the robot pose, x_n , is only used to convert the cell, m_k , from global coordinates to the robot's local coordinates so that it may be compared to z_n (which is in local coordinates). Therefore, we introduce the variable κ , where $\kappa = 0 \dots F$, to enumerate the cells, m_κ , that the measurement z_n could possibly measure. Hence, $m_{\kappa=0}$ is the cell that the robot occupies at pose x_n (the starting pose of the measurement ray z_n), and $m_{\kappa=F}$ is the cell at the maximum range of the sensor given the position and orientation of the sensor, x_n . The update term can now be written $l(m_\kappa|z_n)$. This update term is known as the log-odds of the inverse sensor model, $p(m_\kappa|z_n)$. A commonly used form in the literature (Borenstein and Koren, 1991; Hähnel, 2004; Murray and Little, 2000; Stachniss, 2006; Vasto, 2011; Xu et al., 2019) is

$$l(m_\kappa|z_n) = \begin{cases} l_{\text{free}} & \text{if } \kappa < z_n \\ l_{\text{occ}} & \text{if } \kappa = z_n \\ 0 & \text{otherwise} \end{cases},$$

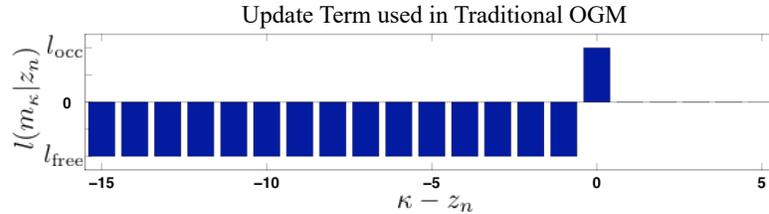


Figure 2.2: A commonly used inverse sensor model for traditional OGM in the log-odds domain. The parameters l_{occ} and l_{free} are normally selected such that $l_{\text{occ}} > 0 \geq l_{\text{free}}$.

where l_{free} and l_{occ} are manually selected, and generally $l_{\text{occ}} \geq -l_{\text{free}}$, and $l_{\text{occ}} > 0$. Various values for l_{free} and l_{occ} are presented in the OG literature. Section 2.5 will explore the inverse sensor model in detail. However, unless stated otherwise, this thesis will use the values reported by Hähnel (2004) of $l_{\text{free}} = -1.3863$ and $l_{\text{occ}} = 1.3863$ ¹ and refer to this as *traditional OGM*. Figure 2.2 illustrates this commonly used inverse sensor model. Again, the two assumptions are beneficial because $l(m_\kappa | z_n)$ is precomputed making the update linear in the number of cells through which z_n passes. Hence, traditional OGM is an online algorithm that is often used for large maps (e.g., $K \gg 20$).

Traditional OGM is depicted in Figure 2.1 and highlights that it is overconfident in its occupancy estimate of some cells and underconfident in others. Again, the full solution represents the true occupancy probability of each cell. The next section will improve this estimate using traditional OGM methods without sacrificing the speed and ease of implementation that have contributed to the wide-spread use of traditional OGM.

2.5 Optimizing Traditional OGM

Section 2.4 highlighted Equation (2.10) as the formula used in traditional occupancy grid mapping. In this equation, $l(m_k)$ is a prior and $l(m_k | z_{1:n-1}, x_{1:n-1})$ is the value stored in the *evidence grid* before incorporating the measurement z_n . Traditional occupancy grid mapping uses the so-called *inverse sensor model*, $p(m_k | z_n, x_n)$, to update the map. Therefore, it is important to compute the inverse sensor model in the log-odds domain, which we call the update term, $\Delta = l(m_k | z_n, x_n) = l(m_\kappa | z_n)$. As explained in Section 2.4, the expression m_κ is the cell m_k relative to the pose x_n and Δ is part of the log-odds Bayesian update rule. This update term is known as the log-odds of the inverse sensor model, $p(m_\kappa | z_n)$, and can be computed a priori.

¹Hähnel (2004) reports these values in the probability domain as $p_{\text{free}} = 0.2$ and $p_{\text{occ}} = 0.8$ and we use the natural logarithm of these values.

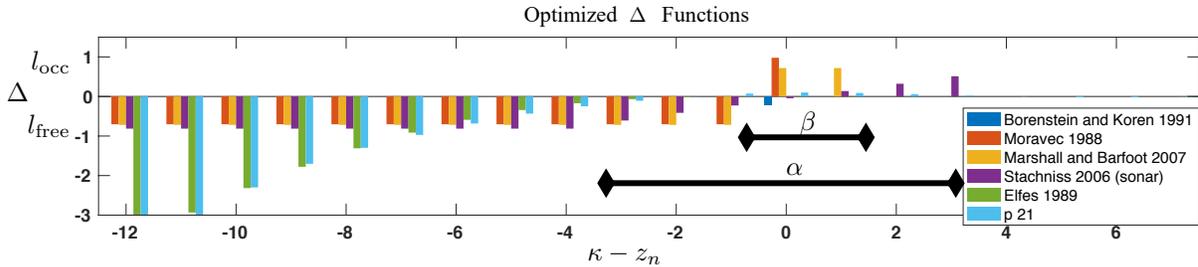


Figure 2.3: Optimized update terms from five parameterizations found in the literature and our parameterization with 21 parameters (p21). The tuning parameters for each update term are indicated (l_{occ} , l_{free} , β , α) with the exception of the Elfes 1989 parameters, which could not be displayed on this plot. κ is the number of cells from the robot to the cell being updated and z_n is the range measurement in units of cells.

The update term governs the amount of information that is added to the map with each new measurement. However, the OG mapping literature contains several different update functions. Therefore, in this section we use variational inference (VI) techniques to better estimate the various update functions found in the literature. In Merali and Barfoot (2014) we compared and optimized the update terms to best capture the residual uncertainty in the OG map. Note that we only reviewed update terms for narrow-beam range sensors; other researchers have provided update terms for different types of sensors. For example, Thrun (1998) used an artificial neural network (ANN) to optimize the update term for multiple sonar sensors on the robot. In addition, Kaufman et al. (2016) proposed update functions for both a *ray* (narrow-beam range sensor) and a *scan* without the measurement-independence assumption used in traditional OGM. Without this assumption, however, the OG update is slower and therefore this section focuses on traditional OGM. Figure 2.3 illustrates five update functions from the literature and Table 2.1 specifies the tuning parameters of each.

The tuning parameters for each update term listed in Table 2.1 are shown in Figure 2.3. Borenstein and Koren (1991) developed a reactive obstacle avoidance algorithm. To update the map quickly, they elected to use a simple update term that only adds information where the range sensor detected an obstacle, $\kappa = z_n$. In contrast, all of the other update terms add information (positive or negative) along the length of the measurement ray and some even beyond that. The update term described in Section 2.4 and first presented by Moravec (1988) is the most commonly found in the literature (Murray and Little, 2000; Stachniss, 2006). This update term will add l_{occ} to the evidence grid where the range sensor detected an obstacle, $\kappa = z_n$, and l_{free} to all cells between the

Table 2.1: Tuning parameters for various update functions, Δ , from the literature

	Tuning Parameters	Optimized Values
Borenstein and Koren (1991)	l_{occ}	-0.2197
Moravec (1988)	$l_{\text{occ}}, l_{\text{free}}$	0.9787, -0.7021
Marshall and Barfoot (2007)	$l_{\text{occ}} = -l_{\text{free}}, \beta$	0.7183, 2
Stachniss (2006) (sonar)	$l_{\text{occ}}, l_{\text{free}}, \alpha$	0.5115, -0.8135, 7
Elfes (1989a)	σ, l_{min}	2.9627, -12.9047

sensor and the obstacle, $\kappa < z_n$. Arbuckle et al. (2002) and Hähnel (2004) used a similar update term, but enforced the constraint that $l_{\text{occ}} = -l_{\text{free}}$. Although not discussed in the paper, Marshall and Barfoot (2007) also enforced this constraint but found that it resulted in the OG map overestimating the distance to the obstacle. Therefore, they allowed the update term to add l_{occ} to the evidence grid for β cells, $\kappa = z_n$ to $\kappa = z_n + \beta - 1$, which addressed their issue. Stachniss (2006) used two update terms in his PhD thesis. For a laser rangefinder, he used the common update term first presented by Moravec (1988). However, for a noisier sonar sensor, he linearly interpolated between l_{free} and l_{occ} with the width of this interpolation, denoted α , centered on $\kappa = z_n$. Finally, Elfes (1989a) presented an update term in his PhD thesis that is based on a range sensor with Gaussian noise, $p(z_n|f) = \mathcal{N}(f, \sigma^2)$. Specifically, Elfes defines the update term as

$$\Delta = \log \left(\frac{\sum_{f=1}^F p(z_n|f)p(f|m_{\kappa} = 1)}{\sum_{f=1}^F p(z_n|f)p(f|m_{\kappa} = 0)} \right).$$

In our research, we found that thresholding the minimum value of Δ yields better results. Therefore, for this parameterization, we also include a variable, l_{min} , at which we threshold the minimum value of Δ . Neither of the two parameters for (Elfes, 1989a) are shown in Figure 2.3 because the l_{min} value is beyond the range of the y -axis displayed and σ is used to compute Δ .

Section 2.3 explained how to compute the full Bayesian solution for OG mapping. This solution makes no assumptions, unlike the traditional OG mapping algorithm discussed in Section 2.4. Therefore, we use variational inference techniques to optimize the Δ parameters in 1D where the full solution can be computed quickly by assuming the robot’s pose is constrained, as shown in Section 2.3.1. In addition to the parameters in the literature, we treat each value of the update term as variable and optimize the update

term for an increasing number of parameters centered around $\kappa = z_n$. The optimization seeks to minimize the divergence between the OG produced using the update term and traditional OGM methods as compared to the full Bayesian solution. These optimized parameters can then be used in 2D or 3D to better capture the residual uncertainty.

The Kullback-Leibler divergence (Kullback and Leibler, 1951), $D_{\text{KL}}(p||q)$, quantifies the difference between two probability distributions, p and q , where p is the benchmark probability, and q is the estimated probability. Variational inference techniques seek to minimize the Kullback-Leibler divergence between the two distributions. Given the cell independence assumption in traditional OGM, we are able to compare two occupancy grids by computing $D_{\text{KL}}(p_k||q_k)$ for each cell, m_k , in the map between the full solution and estimated map. Once

$$D_{\text{KL}}(p_k||q_k) = \sum_{m_k=0}^1 p(m_k) \log \left(\frac{p(m_k)}{q(m_k)} \right)$$

is computed for each cell, m_k , the sum over all cells is the D_{KL} between the two maps:

$$D_{\text{KL}}(p||q) = \sum_{k=1}^K D_{\text{KL}}(p_k||q_k).$$

However, to optimize the tuning parameters, we need only optimize the cross entropy between the two maps,

$$H(p_k, q_k) = - \sum_{m_k=0}^1 p(m_k) \log(q(m_k)),$$

$$H(p, q) = \sum_{k=1}^K H(p_k, q_k),$$

because the cross entropy and Kullback-Leibler divergence are equal to within an additive constant if the benchmark probability, p , is constant. This is known as the *principal of minimum discrimination information* (Kullback, 1959).

Each of the tuning parameters for the five update terms were optimized using the BFGS (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) Quasi-Newton method with a mixed quadratic and cubic line search procedure. Specifically, we used the same 1D dataset with 1000 OG maps as we did in Merali and Barfoot (2012) and Merali and Barfoot (2013). However, in Merali and Barfoot (2014), we divided the 1000

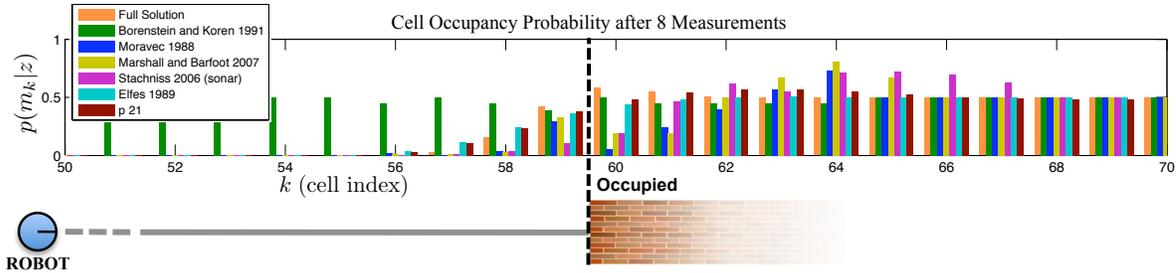


Figure 2.4: The occupancy grids computed by the various algorithms in the literature and our parameterization with 21 parameters (p21) for one of the 1D datasets. The full solution is also shown for reference. Note that our algorithm most closely resembles the full solution near the obstacle, $f=60$.

datasets into 10 equal sets and performed 10-fold cross validation. In other words, we optimized on 900 datasets, then tested on 100 datasets and repeated this procedure 10 times. In this optimization, the range sensor had a maximum range of $F=75$ cells, but the obstacle was $f=60$ cells from the (static) robot and the range sensor had Gaussian noise with a standard deviation of $\sigma=3$ cells, $p(z|f)=\mathcal{N}(60, 3^2)$. Each dataset produced an OG map using traditional OGM and the tuning parameters being optimized for the update term. This occupancy grid was compared to the full solution using cross entropy to quantify the divergence of the two maps. The best update terms from this optimization are shown in Figure 2.3 and the third column of Table 2.1. Figure 2.4 illustrates a portion of the OG created by each of the optimized algorithms for one of the 1000 datasets. Figure 2.4 also shows the full Bayesian solution for reference.

Figure 2.3 shows the optimized update terms for p21 and Figure 2.4 uses the optimized terms to illustrate an example OG created by p21. The p21 algorithm refers to an optimization that we conducted by optimizing an increasing number of $\Delta(\kappa - z_n)$ values centered around $\kappa = z_n$. We assumed that $\Delta=0$ for values beyond (away from the robot) the parameters being optimized and we assumed that Δ is equal to the first value being optimized (nearest to the robot) for the values of delta that were closer to the robot than those being optimized. For one parameter being optimized, p1, our method outperformed Borenstein and Koren (1991) because we were able to add information to all cells between the rover and the obstacle. However, our method is equivalent to Moravec (1988) for two parameters, p2. Thus, as we increase the number of parameters, this methodology should be able to capture and outperform all of the other Δ functions that we have identified in the literature. Figure 2.5 illustrates the optimized parameters for the increasing number of parameters. Interestingly, as the number of parameters increases, the update term

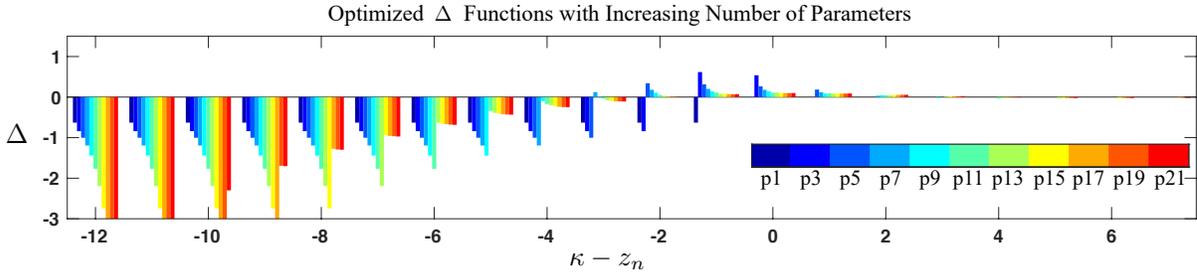


Figure 2.5: The optimized update term for an increasing number of parameters centered around $\kappa = z_n$. The optimization was completed for parameters 1 to 21, but only every other parameterization is illustrated here.

Table 2.2: Optimized update function, Δ , with 21 tuning parameters - referred to as p21.

$\kappa - z_n$																						
≤ -11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	≥ 10	
Δ	-5.4028	-2.2993	-1.7025	-1.2979	-0.9722	-0.6836	-0.4341	-0.2476	-0.1083	0.0051	0.0719	0.0995	0.0877	0.0564	0.0242	-0.0077	-0.0215	-0.0213	-0.0149	-0.0068	0.0034	0

becomes more negative for values of $\kappa < z_n$. Also, near $\kappa = z_n$ the update term adds less information per cell but is distributed over more cells as the number of parameters increases. Table 2.2 shows the optimized Δ function for p21. As expected, the highest value is at $\kappa = z_n$ and approaches zero for larger values of κ . Notably, the Δ function adds significant negative information to the evidence grid for cells near the robot, $\kappa - z_n \leq 11$, indicating that the inverse sensor model is confident that these cells are unoccupied.

Figure 2.6 shows how the cross entropy decreases as the number of parameters being optimized increases. Figure 2.6 also shows the best cross entropy values for the other algorithms in the literature for reference; the parameterization from Borenstein and Koren (1991) was excluded from Figure 2.6 because its cross entropy values were significantly higher than the rest. Interestingly, adding parameters near the robot, as opposed to beyond the obstacle, more dramatically decreases cross entropy. Intuitively this makes sense as the range sensor should not be changing the occupancy probability of cells beyond the obstacle. Furthermore, Figure 2.6 shows that the parameterization proposed by Elfes (1989a) did extremely well and our method had to optimize 19 or more parameters to do better. Figure 2.3 gives more insight into this result as we see where Elfes 1989 and p21 differ. Figure 2.7 further highlights this result by illustrating the average D_{KL} on a cell-by-cell basis for the cells near the obstacle. These figures show that our parameterization better models the full solution near the obstacle at the expense of

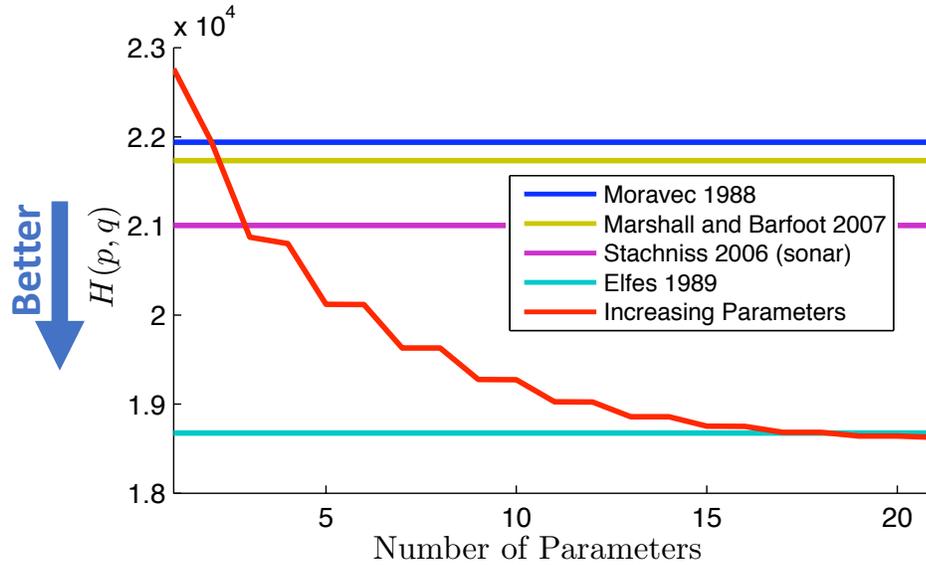


Figure 2.6: The best cross entropy values for optimized two-assumption OG mapping algorithms as compared to the full Bayesian solution.

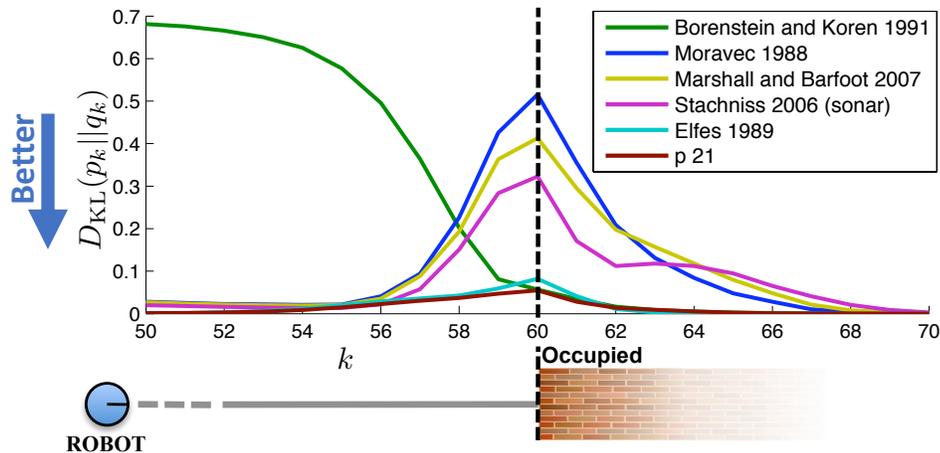


Figure 2.7: The average D_{KL} for each of the two-assumption OGM algorithms in the literature and our solution with 21 parameters, compared to the full Bayesian solution. The obstacle is at $k = 60$, so we see that the largest divergence is centered around the obstacle and that our parameterization has the lowest average divergence.

performing slightly worse beyond the obstacle. We clearly see that the algorithms (with the exception of Borenstein and Koren (1991)) diverge most from the full solution near the obstacle and that our parameterization of the update function diverges the least. Particularly, the parameterization proposed by Moravec (1988) diverges the most near the obstacle boundary. This is the most common OG mapping algorithm found in the literature and what we refer to as *traditional OGM* in this thesis.

The optimized p21 algorithm is applied to the 1D toy example and presented in Figure 2.1 alongside the other OGM algorithms discussed in this chapter. Furthermore, the optimized Δ functions are applied to simulated 2D results and presented in Section 6.1.2. However, throughout this thesis, we will show the results of various OGM algorithms on a simple 2D example to further highlight the difference between the algorithms. Some of these results were shown in Figure 1.10 and the legend shown in Figure 1.10g is commonly used in this thesis for the 2D OG maps. Figure 2.8 introduces this simple 2D example and illustrates some of the results from this chapter. Figure 2.8a illustrates the ground truth map and the three robot locations where narrow-beam range measurements were taken from. Specifically, the robot acquires 720 measurements at each location in 0.5° increments, for a total of 2160 measurements. Each of these measurements has Gaussian random noise about the true range measurement, $p(z_n|f) = \mathcal{N}(f, 3^2)$. Figure 2.8b shows these measurement rays, where blue cells indicate that the ray passed through the cell without being reflected and red cells indicate that the measurement ray was reflected at that cell. Note that some measurement rays do not terminate in a red cell, indicating that the measurement ray reached its maximum range without being reflected. Most significantly, no measurement rays reach the upper-right corner in this example. Figure 2.8c illustrates an occupancy grid created from Figure 2.8b by showing blue cells as white (unoccupied), red cells as black (occupied) and the remaining cells as orange (unknown). This map has no notion of uncertainty and cannot deal with conflicting range measurements in a cell and is therefore only shown for reference.

The remaining three maps shown in Figure 2.8 are computed using traditional OGM methods, but with different Δ functions. Figure 2.8d illustrates the occupancy grid map generated using a common Δ function from the literature (Hähnel, 2004). Figure 2.8e illustrates the occupancy grid map generated using the optimized values of a Δ function of the same form. These optimized values, shown in the second row of Table 2.1, are closer to zero, indicating that traditional OGM is overconfident and adds too much information to the map for each measurement. Finally, Figure 2.8f illustrates the result of our optimized Δ function with 21 parameters, explained in this section. This map is more confident that mapped cells near the robot are unoccupied, but it is also less confident that cells that reflected the range sensor are occupied. The result is an occupancy grid that better captures the uncertainty in the map.

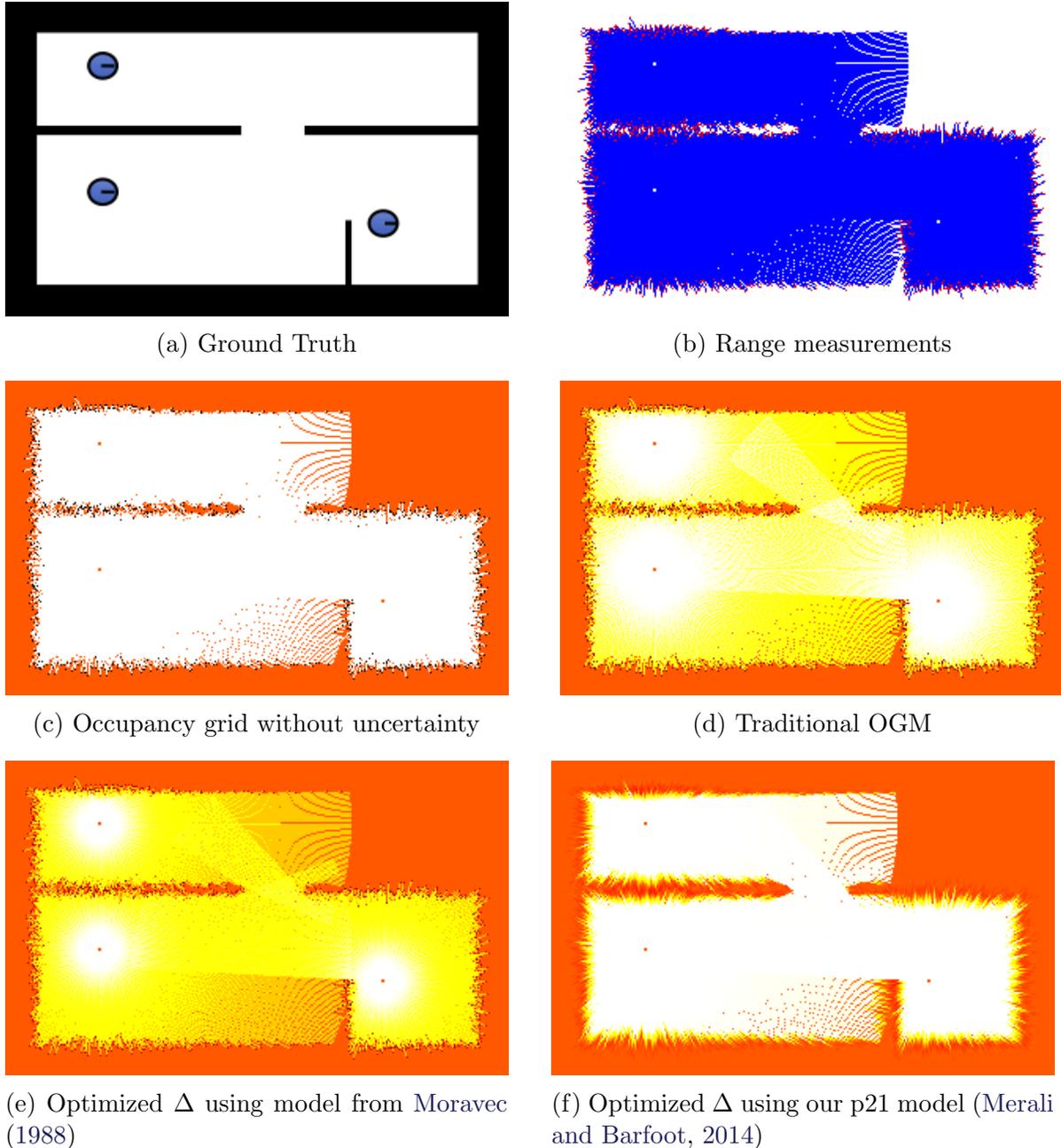


Figure 2.8: 2D example illustrating the ground truth map and robot positions in (a), the range measurements in (b), and four possible OG maps resulting from these range measurements in (c)-(f). The OG map in (c) does not model uncertainty in the occupancy map and is shown for reference. The remaining three OG maps use traditional OGM methods with a different Δ function for each. The result in (d) uses a common Δ function from the literature, the result in (e) uses an optimized version of this Δ parameterization. By allowing the Δ function to have 21 different parameters (f), we are able to add more or less information to the map along each measurement ray to better approximate the full Bayesian solution. The legend used, with the exception of (b), is shown in Figure 1.10g.

2.6 Mutual Information from Measurements

The preceding sections have shown the need for incorporating sensor measurements in OGM with an accurate inverse sensor model. The literature discusses various sensor models, but this thesis is concerned with narrow-beam range sensors. In robotics, laser range finders are a common example of narrow-beam range sensors, but several other sensors may be modeled as narrow-beam range sensors, such as cameras, ultrasonic sensors, and even radar (e.g., Shankar and Michael, 2020).

The nature of a narrow beam range sensor is to return the distance to the first occupied cell, f ; a signal will pass through unoccupied cells and reflect off the first occupied cell. Therefore, by introducing the hidden state of the first occupied cell, $p(z_n|m^r, x_n)$ can be written as the sensor model, $p(z_n|f_{r,n})$, where $f_{r,n}$ is dependent on the map, m^r , and the robot pose, x_n . Furthermore, this thesis is concerned with a static environment and therefore assumes that measurements are independent of one another given the map,

$$p(z|m^r) = \prod_{n=1}^N p(z_n|m^r).$$

This assumption is valid for a static environment unlike the assumption shown in Equation (2.8) for traditional OGM that assumes measurements are independent given the occupancy of a single cell.

As discussed in Section 2.4, traditional OGM assumes that the occupancy probability of cells are not correlated and that measurements are independent of one another given the occupancy of a single cell (as opposed to all cells in the map). Therefore, traditional OGM uses an *inverse sensor model*, but range sensors are more accurately modeled as $p(z_n|m^r, x_n) = p(z_n|f_{r,n})$, which is known as the *forward sensor model*. Several examples of the forward sensor model exist in the literature (Agha-Mohammadi et al., 2019; Dhiman, 2019; Hou et al., 2019; Pathak et al., 2007; Thrun, 2003), but two of the most common are Gaussian noise about the true measurement and pattern-matching. For a standard deviation of σ cells and the first occupied cell $f_{r,n}$, the Gaussian sensor model is,

$$\begin{aligned} p(z_n|f_{r,n}) &= \mathcal{N}(f_{r,n}, \sigma^2) \\ &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(z_n - f_{r,n})^2}{2\sigma^2}\right). \end{aligned}$$

The pattern-matching sensor model will increase the probability of cell configurations that *match* the measurement and decrease the probability of other cell configurations. To reduce computation time, the pattern-matching sensor model will typically increase the probability of one configuration and decrease the others equally. This version of the pattern-matching sensor model is equivalent to the Gaussian sensor model with a small standard deviation – such that the probability mass is concentrated at $p(z_n = f_{r,n})$. For example,

$$p(z_n|f_{r,n}) = \begin{cases} p_{\text{occ}} & \text{if } z_n = f_{r,n} \\ (1 - p_{\text{occ}})/F & \text{otherwise} \end{cases},$$

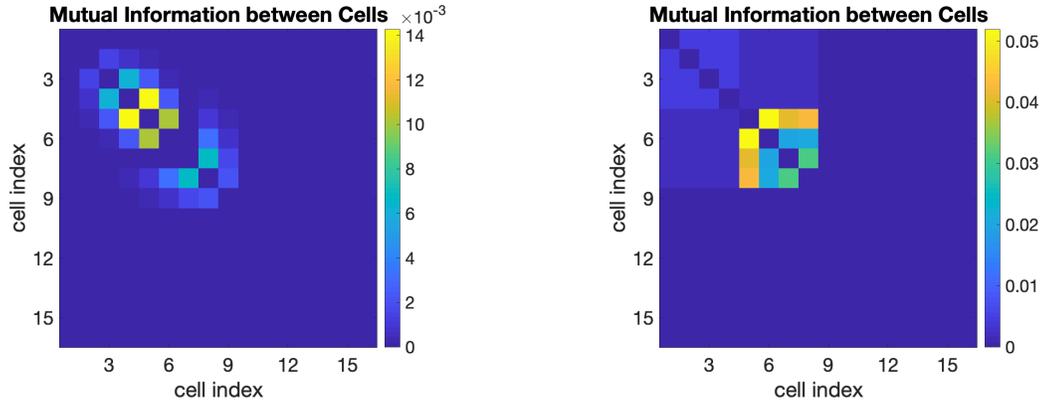
where $p_{\text{occ}} \in (0, 1]$ is a user-defined value and is typically greater than 0.5 and the remaining probability mass is equally distributed over the other F possible values of $f_{r,n}$. The pattern-matching sensor model is also referred to as the piecewise constant sensor model (PCSM) in the literature (Dhiman, 2019), and is predominately used because it can be applied faster than a Gaussian sensor model. Note that both models can be precomputed and therefore do not need to be computed online. However, the pattern-matching sensor model need only update one map, m^r , or a subset of maps that satisfy the cell configuration to allow, $f_{r,n} = z_n$. Further details about the sensor model can be found in Section 6.3 of Thrun et al. (2005).

In this thesis, we wish to highlight that the forward sensor model will introduce cell correlations from measurements, whereas the inverse sensor model will not. One measure of cell correlations is the mutual information (MI) between cells, which measures the dependence between the two cells. Specifically, MI quantifies the amount of information in one cell by observing another. For two cells, m_j and m_k , the mutual information between them is quantified as,

$$\text{MI}(m_j, m_k) = \sum_{m_j=0}^1 \sum_{m_k=0}^1 p(m_j, m_k) \log \left(\frac{p(m_j, m_k)}{p(m_j)p(m_k)} \right).$$

Therefore, if two cells are independent, $p(m_j, m_k) = p(m_j)p(m_k)$, then the MI between them is zero.

Given the ability to quantify mutual information, Figure 2.9 illustrates the MI between all pairs of cells in a 1D occupancy grid. In the two examples, the OG was computed using the full Bayesian solution with the same measurements and a forward



(a) Mutual information with a Gaussian sensor model of $p(z_n|f_{r,n}) = \mathcal{N}(f_{r,n}, 3^2)$.

(b) Mutual information with a pattern-matching sensor model with $p_{occ} = 0.9$.

Figure 2.9: Comparing the mutual information between cells in the posterior using the full Bayesian solution to OGM, where the robot is at cell $k=0$ and $z = \{11, 7, 8, 8, 5, 9\}$. Note that values are higher for proximate cells and zero for distant cells. In contrast, the cell independence assumption made by traditional OGM assumes no mutual information between cells.

sensor model. However, Figure 2.9a shows the result of using a Gaussian sensor model, whereas Figure 2.9b uses a pattern-matching sensor model. The resulting plots highlight that both forward sensor models add mutual information to the map, but in both cases the mutual information is highest between proximate cells and is zero or near-zero for cells that are farther apart. Critically, all of the mutual information introduced by the forward sensor model is assumed to be zero by the inverse sensor model and traditional OGM techniques.

The insight that mutual information is highest between proximate cells and near-zero between cells that are farther apart, serves to motivate much of the work in this thesis. By recognizing that mutual information is local and not global, we are able to develop algorithms that approximate the full Bayesian solution by accounting for these local cell correlations. In the next chapter, we develop OGM algorithms that seek to retain the mutual information between proximate cells. In Chapter 5, we are able to extend the notion of local cell correlation to the prior and not just measurements.

2.7 Chapter Summary

In this chapter we have formulated the full Bayesian solution to OGM and shown that it is computationally intractable for realistic maps. Section 2.3.1 demonstrates a novel method

to compute the full Bayesian solution for a 1D map when the robot's pose is constrained. This result is then used to optimize the update term in traditional OGM in Section 2.5. The resulting novel update terms can be used in traditional OGM to better capture the residual uncertainty in the map with no additional computation or storage cost. Finally, Section 2.6 highlights that the sensor introduces mutual information between proximate cells, but traditional OGM is unable to capture this information. Therefore, the following chapters will introduce methods to better estimate the full posterior in OGM by leveraging the fact that mutual information is well approximated by using nearby cells.

Chapter 3

Patch Map

Chapter 2 highlights that the full Bayesian solution is computationally intractable for real maps and that traditional OGM makes simplifying assumptions that enable it to run online. One of these simplifying assumptions is that cells are independent and therefore discard any mutual information between cells. However, Section 2.6 illustrates that the forward sensor model introduces local cell correlations in the map, but that traditional OGM cannot represent these correlations because it assumes no mutual information between cells by only storing the occupancy probability of each cell, $p(m_k|z, x)$. In addition, Section 2.6 also reveals that mutual information is greatest between proximate cells. Therefore, this chapter presents a novel *patch map* that retains mutual information between proximate cells yielding a better estimate of the true information in the map, given the measurements and the poses from which they were taken. Specifically, we introduce the notion of a *patch*, or grouping, of cells and discretize the map into a coarse set of *patches*. We then compute the full Bayesian solution for each patch, but make an independence assumption between patches for efficiency. The goal of the patch map is to capture local cell correlations as the full solution does, but to be computationally tractable to be used for realistic maps (2D or 3D).

Section 3.1 generalizes traditional OGM to estimate patches of cells as opposed to individual cells. This novel formulation simplifies to traditional OGM for a patch size of one cell and can therefore be viewed as a generalization. However, by increasing the patch size, the algorithm is able to keep cell correlations between cells in the patch and therefore better estimate the true information in the map. The independence assumption between patches allows the algorithm to run online. However, Section 3.2 relaxes this assumption by introducing an offline map to correlate patches. We published this novel offline patch

map algorithm in Merali and Barfoot (2012) as a benchmark for OGM algorithms in 2D, where the full solution is computationally intractable. Further experimental results of this algorithm are shown in Section 6.1.1.

3.1 Online Patch Map

Section 2.3 highlights that the full solution for OGM does not make the cell independence assumption. As a result, the occupancy of each cell is correlated with all other cells in the map through the measurements. However, given the mutual information study in Section 2.6, we feel that the full solution can be approximated by accounting for dependence between only proximate cells as opposed to all cells. Thus, we propose a solution where the full solution is computed for a particular *patch*, a_k , of the map, centered on cell, m_k . Then we compute the probability of all possible patch configurations, $w = 1 \dots W$, where there are $W = 2^L$ possible patch configurations for a patch size of L cells. If each patch has L cells and assuming that no patches overlap, then there are K/L patches in the map. Hence, for a map with K cells, the computation is $\mathcal{O}(K/L \times W)$ as opposed to $\mathcal{O}(2^K)$ for the full solution. In 1D, we define the patch as a cluster of adjacent cells and in 2D we suggest three possible patch connectivity neighbourhoods shown in Figure 3.1.

Equation (2.5) shows that the full solution uses the forward sensor model, $p(z_n | f_{r,n})$. Typically, the occupancy of all cells along the measurement ray must be known to determine $f_{r,n}$. The patch map algorithm iterates through all possible values of $a_{k,w}$, but for the *online patch map*, we assume that the first-occupied-cell, $f_{k,w}$, is independent of cells outside of $a_{k,w}$. Note that we will revisit this assumption in Section 3.2. Therefore, we

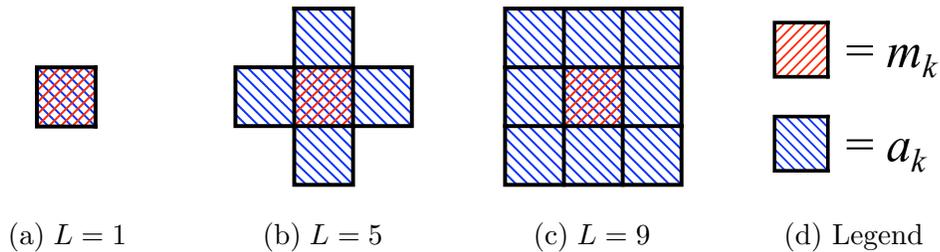


Figure 3.1: Three possible patch connectivity neighbourhoods for a 2D map. Each patch, a_k is centered on a specific cell, m_k . The patch method increases in complexity as the number of cells in the patch, L , increases, but it better captures the true information in the map, given the measurements, z .

are able to use a similar formulation to traditional OGM and compute the probability of each patch, $p(a_{k,w}|z, x)$, independently. By storing the value of each patch configuration, $p(a_{k,w}|z_{1:n-1}, x_{1:n-1})$, in the grid we are able to update the stored value incrementally with each new measurement, using the forward sensor model. Using Bayes rule, the probability of a particular patch configuration centered on cell m_k is

$$\begin{aligned}
p(a_{k,w}|z, x) &= \frac{p(z_n|a_{k,w}, z_{1:n-1}, x)p(a_{k,w}|z_{1:n-1}, x)}{p(z_n|z_{1:n-1}, x)} \\
&= \frac{p(z_n|a_{k,w}, x_n)p(a_{k,w}|z_{1:n-1}, x_{1:n-1})}{p(z_n|z_{1:n-1}, x)} \\
&= \frac{p(z_n|a_{k,w}, x_n)p(a_{k,w}|z_{1:n-1}, x_{1:n-1})}{\sum_{j=1}^W p(z_n|z_{1:n-1}, x, a_{k,j})p(a_{k,j}|z_{1:n-1}, x)} \\
&= \frac{p(z_n|a_{k,w}, x_n)p(a_{k,w}|z_{1:n-1}, x_{1:n-1})}{\sum_{j=1}^W p(z_n|x_n, a_{k,j})p(a_{k,j}|z_{1:n-1}, x_{1:n-1})}. \tag{3.1}
\end{aligned}$$

This computation is required for all patch configurations. As a consequence, the denominator adds little computational cost, because the numerator must be computed for each patch configuration, w . Thus, this formulation reduces to the formulation of traditional OGM in Equation 2.9 for a patch size of $L = 1$ and can therefore be seen as a generalization of traditional OGM. However, the formulation in Equation (3.1) uses the forward sensor model, $p(z_n|a_{k,w}, x_n)$, as opposed to the inverse sensor model used in traditional OGM. Similar to the cell inverse sensor model, $p(m_k|z_n, x_n)$, used in traditional OGM, we introduce a patch inverse sensor model, $p(a_{k,w}|z_n, x_n)$. The inverse sensor model can be used by applying Bayes' Rule to the forward sensor model,

$$\begin{aligned}
p(a_{k,w}|z, x) &= \frac{p(z_n|a_{k,w}, x_n)p(a_{k,w}|z_{1:n-1}, x_{1:n-1})}{p(z_n|z_{1:n-1}, x)} \\
&= \frac{p(a_{k,w}|z_n, x_n)p(z_n|x_n)p(a_{k,w}|z_{1:n-1}, x_{1:n-1})}{p(a_{k,w}|x_n)p(z_n|z_{1:n-1}, x)} \\
&= \eta \frac{\underbrace{p(a_{k,w}|z_n, x_n)}_{\text{patch inverse sensor model}} \underbrace{p(a_{k,w}|z_{1:n-1}, x_{1:n-1})}_{\text{stored value}}}{\underbrace{p(a_{k,w}|x_n)}_{\text{patch prior}}}, \tag{3.2}
\end{aligned}$$

where η is a normalizing constant. By introducing the patch inverse sensor model, $p(a_{k,w}|z_n, x_n)$, this formulation clearly shows that the online patch map algorithm is a generalization of traditional OGM, shown in Equation 2.9. However, as highlighted in Section 2.5, optimizing an inverse sensor model requires a great deal of tuning. Recently, Bauer et al. (2019) have used a neural network to learn a similar patch prior for a radar sensor with the patch centered around the robot. One possible patch inverse sensor model we propose is to increase the likelihood of patch configurations that match the sensor measurement and decrease the others. For example,

$$p(a_{k,w}|z_n) = \begin{cases} p_{\text{match}} & \text{if cells in } a_{k,w} \text{ correspond to } z_n \\ \eta & \text{otherwise} \end{cases},$$

where $p_{\text{match}} \in (0, 1]$ is a user-defined value and the sum of all matching probabilities is typically greater than 0.5 and the remaining probability mass is equally distributed over the other possible patch configurations. Note that the term $p(a_{k,w}|z_{1:n-1}, x_{1:n-1})$ is stored in the patch-based occupancy grid for each patch configuration. Regardless of whether the forward sensor model (Equation (3.1)) or the inverse sensor model (Equation (3.2)) is used to compute $p(a_{k,w}|z, x)$, the marginal probability of a cell being occupied may be calculated by summing the patch probabilities for those patches in which the cell of interest is occupied:

$$p(m_k|z, x) = \sum_{w=1}^W p(m_k|a_{k,w})p(a_{k,w}|z, x),$$

where $p(m_k|a_{k,w}) \in \{0, 1\}$.

Although the patch map has been developed for higher dimensions, the theory was implemented in 1D with the forward sensor model so that it could be compared to the full Bayesian solution. Figure 3.2 illustrates the resulting occupancy grid using different values of L . As expected, the OG approaches the full solution for increasing values of L . However, this figure also highlights that the marginal occupancy probability of a cell is more accurate if it is correlated to its neighbours on both sides as opposed to one side (i.e., the cell is near the center of the patch). All of the patch algorithms depicted draw the first patch boundary starting before cell $k = 1$, then every L cells thereafter. Therefore the algorithms with $L = 3$ and $L = 5$ better predict the occupancy probability (closer to the full solution) of the cell at $k = 9$, than the algorithms with $L = 4$ and $L = 8$

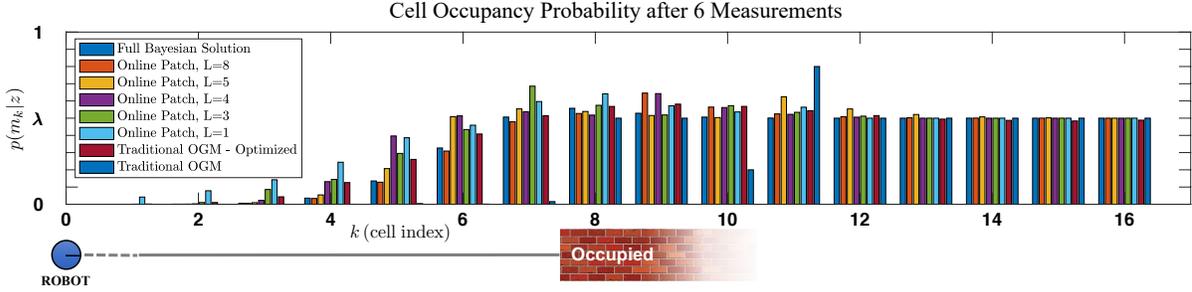


Figure 3.2: Occupancy grid after six measurements. The robot is at $k = 0$, the true first occupied cell is at $k = 8$ (therefore $f = 8$). The full Bayesian solution illustrates the true occupancy probability of each cell. The online patch map is computed for various values of L and approaches the full solution as L increases. All patch algorithms start the patch boundaries before $k = 1$. The cell occupancy is less accurate near patch boundaries where cells are independent of neighbouring cells in another patch. The traditional OGM algorithm is also shown for comparison with an update term from the literature and optimized values.

because the latter two algorithms have a patch boundary between cells 8 and 9. And the algorithm with $L = 5$ performs poorly for cell 11 because it has a patch boundary between cells 10 and 11. Thus, we see the negative impact of patch boundaries – a problem that we will seek to address in Chapter 4.

Patch boundaries are selected before incorporating measurements, but this choice can lead to different map estimates for the same patch size, L . Figure 3.3 highlights this problem by illustrating two patch maps resulting from a patch size of $L = 4$, but with different patch boundaries. The first has patch boundaries starting before $k = 1$ and the second is offset by two cells. The figure illustrates that the first algorithm poorly estimates cells 5 and 6, but then estimates cell 7 well. However, the opposite result is true for the second algorithm that has a patch boundary between cells 6 and 7. Chapter 4 will propose a solution to the problem of patch boundaries.

To quantitatively compare two occupancy grids, we compute the Kullback-Leibler divergence, $D_{\text{KL}}(p_k || q_k)$, for each cell, m_k , in the map between the benchmark, p , and estimated, q , maps. Once

$$D_{\text{KL}}(p_k || q_k) = \sum_{m_k=0}^1 p(m_k) \log \left(\frac{p(m_k)}{q(m_k)} \right)$$

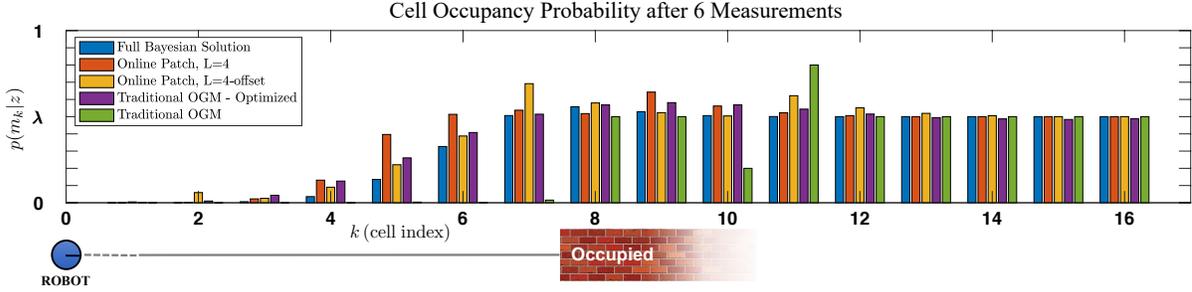


Figure 3.3: Occupancy grid after six measurements. The robot is at $k = 0$, the true first occupied cell is at $k = 8$ (therefore $f = 8$). The full Bayesian solution illustrates the true occupancy probability of each cell. The online patch map is illustrated for $L = 4$, where one has a boundary starting at $k = 1$ and the second is offset by two cells. The cell occupancy is less accurate near patch boundaries where cells are independent of neighbouring cells in another patch. The same algorithm can therefore yield different results depending on where the patch boundary lies – this is clearly shown in cells 6 and 7. The traditional OGM algorithm is also shown for comparison with an update term from the literature and optimized values.

Table 3.1: The average D_{KL} (from 1000 1D datasets) between maps produced by OG mapping algorithms and the full solution

	Traditional OGM	Traditional OGM – Optimized	Patch Map $L = 1$	Patch Map $L = 3$	Patch Map $L = 5$
Online	6.7870	0.2509	0.5297	0.3100	0.2290
Offline			0.2834	0.0749	0.0274
Offline w/overlap			n/a	0.0287	0.0038

is computed for each cell, m_k , the sum over all cells is the D_{KL} between the two maps:

$$D_{\text{KL}}(p||q) = \sum_{k=1}^K D_{\text{KL}}(p_k||q_k).$$

Table 3.1 shows the Kullback-Leibler divergence between various mapping algorithms and the full solution, averaged over 1000 datasets. Each dataset includes between one and eight measurements drawn from $p(z_n|f) = \mathcal{N}(f, \sigma^2)$ with $f = 20$ and $\sigma = 2$ cells. The values in Table 3.1 indicate that as the patch size, L , increases, the patch map more accurately models the full solution. The table also highlights that the optimized Δ function from Section 2.5 is a great improvement over traditional OGM techniques and that the online patch map algorithm with a forward sensor model (first row) only performs better with $L \geq 5$.

3.2 Offline Patch Map

Section 3.1 shows that the online patch map algorithm does well at estimating the map by keeping cell correlations between cells in the patch. However, as highlighted in Figure 3.2, the patch independence assumption results in inaccuracies near the patch boundaries. As a consequence, for the offline patch map algorithm presented here, we relax this independence assumption to achieve a more accurate map estimate, but at the cost of additional computation. We first introduced the *patch map* algorithm in Merali and Barfoot (2012) as a benchmark that could be used to evaluate other OGM techniques. The offline patch map algorithm incorporates knowledge of an occupancy grid such as the ground truth map or MAP estimate that does not model the uncertainty in the map. This additional map can be used to determine the first occupied cell, which is used by the forward sensor model. In contrast, the online patch map algorithm presented in Section 3.1 assumes that the range measurement will not be reflected by cells outside of the patch being updated, a_k . The formulation presented in this section is known as the *offline patch map* algorithm as it is a batch solution and is not iterative like the online version. Therefore, the offline patch map algorithm lies somewhere between the computationally intractable full Bayesian solution and the online patch map algorithm on the accuracy vs. efficiency tradeoff curve.

Equation (2.5) shows that the full solution requires the forward sensor model, $p(z_n|f_{r,n})$, which is conditioned on the first occupied cell, $f_{r,n}$. To compute $f_{r,n}$, the occupancy of all cells along the measurement ray must be known. The online patch map algorithm assumes that patches are independent and therefore the forward sensor model only depends on the patch being updated, $p(z_n|f_{r,n}) \approx p(z_n|a_{k,w}, x_n)$. To determine the first occupied cell, f , the offline patch method uses the ground truth (GT) or MAP occupancy of all cells outside the current patch; this subset of cells is denoted \tilde{m}_{-a} . The offline patch map algorithm uses $a_{k,w}$ and \tilde{m}_{-a} to obtain the occupancy of all cells in the map and estimate f for the forward sensor model, $p(z_n|f_{r,n}) \approx p(z_n|a_{k,w}, x_n, \tilde{m}_{-a})$. For the offline patch map algorithm, we assume that unknown cells in the GT map or MAP estimate are represented as occupied in \tilde{m}_{-a} and will reflect the range sensor. This consideration is necessary because some patch configurations will allow the range sensor to map these unknown cells. Therefore, by incorporating \tilde{m}_{-a} , the offline patch map algorithm estimates the probability of a particular patch configuration centered on cell m_k as

$$\begin{aligned}
p(a_{k,w}|z, x, \tilde{m}_{-a}) &= \frac{p(a_{k,w}|x, \tilde{m}_{-a})p(z|a_{k,w}, x, \tilde{m}_{-a})}{p(z|x, \tilde{m}_{-a})} \\
&\approx \frac{p(a_{k,w}) \prod_{n=1}^N p(z_n|a_{k,w}, x_n, \tilde{m}_{-a})}{\sum_{j=1}^W \left(p(a_{k,j}) \prod_{n=1}^N p(z_n|a_{k,j}, x_n, \tilde{m}_{-a}) \right)}. \tag{3.3}
\end{aligned}$$

This computation is required for all patch configurations. As a consequence, the denominator adds little computational cost, because the numerator must be computed for each w . In this formulation, we assume that the prior probability of a particular patch configuration, $p(a_{k,w})$, is independent of all other cells in the map and we also assume a uniform patch prior, $p(a_{k,w}) = 1/W$. Both of these assumptions will be explored further in Chapter 5. We have also made the static-world assumption that measurements are independent given the map. Therefore, the measurements may be incorporated incrementally using the forward sensor model, $p(z_n|a_{k,w}, x_n, \tilde{m}_{-a})$. This forward sensor model is more accurate than that used in the online patch map algorithm because $a_{k,w}$ and \tilde{m}_{-a} together yield the occupancy of every cell in the map, which must be m^r for some r . Note that $p(a_{k,w}|z, x, \tilde{m}_{-a})$ need only be computed for the subset of measurements that pass through the patch, a_k . Measurements that do not pass through a_k will yield the same probability for all W configurations of $a_{k,w}$, and the factors on the numerator of Equation (3.3) will cancel with common factors on the denominator. Similarly to the full solution, the probability of a cell being occupied may be calculated by summing the patch probabilities for those patches in which the cell of interest is occupied:

$$p(m_k|z, x, \tilde{m}_{-a}) = \sum_{w=1}^W p(m_k|a_{k,w})p(a_{k,w}|z, x, \tilde{m}_{-a}), \tag{3.4}$$

where $p(m_k|a_{k,w}) \in \{0, 1\}$.

The benefit of the offline patch map algorithm is highlighted in Figure 3.4. In contrast to the online algorithm presented in Figure 3.2 for the same measurements, the offline algorithm better estimates the full Bayesian solution. For example, comparing $L = 3$ in Figures 3.4 and 3.2, the estimate of $p(m_{k=7}|z)$ is quite poor for the online algorithm and quite good for the offline algorithm. However, the offline patch map algorithm is still

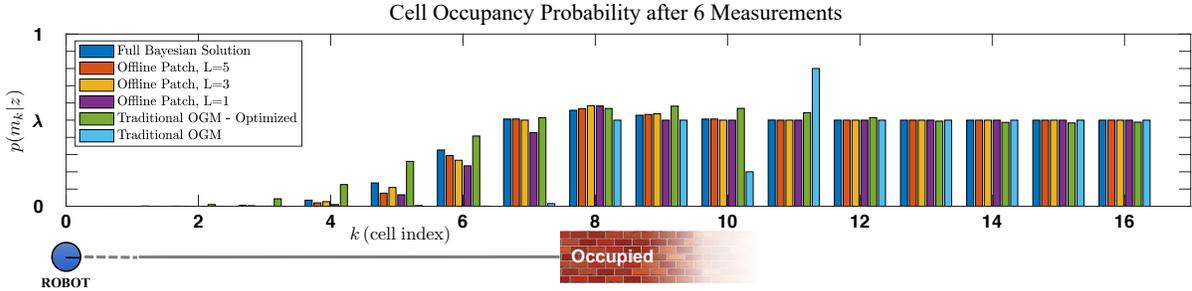


Figure 3.4: Occupancy grid after six measurements. The robot is at $k = 0$, the true first occupied cell is at $k = 8$ (therefore $f = 8$). The full Bayesian solution illustrates the true occupancy probability of each cell. The offline patch map is computed for various values of L and approaches the full solution as L increases. All patch algorithms start the patch boundaries before $k=1$. The cell occupancy is less accurate near patch boundaries where cells are independent of neighbouring cells in another patch. The traditional OGM algorithm is also shown for comparison with an update term from the literature and optimized values.

adversely effected by patch boundaries. Figure 3.4 highlights this for the $L=3$ algorithm as the occupancy estimate for cells in the patch $a_{k=5}$, $k = \{4, 5, 6\}$, become progressively worse until the next patch boundary. In the subsequent patch, $a_{k=8}$, the estimate is quite accurate again for cell $k=7$, but worse again for cells $k = \{8, 9\}$.

It is clear that the marginal cell occupancy probability is more accurate for cells in the middle of a patch as opposed to near the patch boundaries. As a consequence, we are better able to estimate the marginal probability $p(m_k|z)$ by applying Equation (3.4) for a patch, a_k , centered at each cell, m_k , as we showed in Merali and Barfoot (2012). Computing the probability of a patch centered at each cell will reduce the effects of patch boundaries and ensure that each cell is symmetrically correlated to its neighbours. This approach is further supported by the analysis in Section 2.6 that highlights that mutual information is greatest between proximate cells.

This variant of the patch map is well suited as an offline benchmark that can be used to compare other OGM techniques. In Section 2.5, we used the full Bayesian solution as a benchmark in 1D to optimize the Δ function, but the patch map can be used as a benchmark in 2D, where the full solution is computationally intractable. Just as we compared OGM algorithms using the Kullback-Leibler divergence with the full solution as the reference map in 1D, the patch map algorithm can be used in 2D for the same purpose.

To further highlight the accuracy of this approach, Table 3.1 shows the Kullback-Leibler divergence between various mapping algorithms and the full solution in 1D, averaged over 1000 datasets. Each dataset includes between one and eight measurements drawn from $p(z_n|f) = \mathcal{N}(f, d^2)$ with $f = 20$ and $d = 2$ cells. The values in Table 3.1 indicate that as the patch size, L , increases, the patch map more accurately models the full solution. In fact, a small increase in the patch size makes a significant difference in the D_{KL} values. This result supports the finding in the mutual information study in Section 2.6 that MI between cells quickly approaches zero for cells that are farther apart. Table 3.1 also highlights that the offline patch map provides a significant improvement over the online version. Furthermore, allowing the patches to overlap and only computing the marginal cell occupancy of the one cell at the center of each patch yields the lowest D_{KL} of all. Note that in Merali and Barfoot (2012) we showed that using the MAP estimate as the reference map yielded slightly better performance than using the ground truth map, so in this analysis we have only used the MAP estimate for \tilde{m}_{-a} . Therefore, for larger maps where the full solution is computationally intractable, a patch map using the MAP estimate and the largest possible patch size would best approximate the full Bayesian solution.

Note that the cost of computing $p(a_{k,w}|z, x, \tilde{m}_{-a})$ for a patch centered at each cell, increases the computational complexity of the algorithm from $\mathcal{O}(K/L \times 2^L)$ to $\mathcal{O}(K \times 2^L)$. In either case, selecting L is a trade-off between computation time and fidelity. The strength of the patch map is the ability to make this trade, as opposed to the ‘all-or-nothing’ offered by the full solution. Note that the patch map generalizes to the full solution for $L = K$, as there will only be a single patch. Conversely, the offline patch map algorithm with a patch size of $L = 1$ is how Thrun (2003) computed the residual uncertainty in the map after computing the MAP estimate.

Figure 3.5 illustrates the various patch map algorithms from this chapter using the same data used to generate Figure 2.8. Comparing the two figures illustrates, on a 2D example, the same findings that were shown analytically in 1D in Table 3.1, that each of the patch map algorithms perform near or above the level of the optimized traditional OGM algorithms presented in Section 2.5. Comparing the patch map algorithms, Figure 3.5b and Figure 3.5c appear pixelated as a result of the patch boundaries between the 3×3 patches. Figure 3.5d does not appear pixelated due to overlapping the patches and can be used as a benchmark for OGM in 2D. Furthermore, the online patch map algorithm shows greater uncertainty near the wall boundaries.

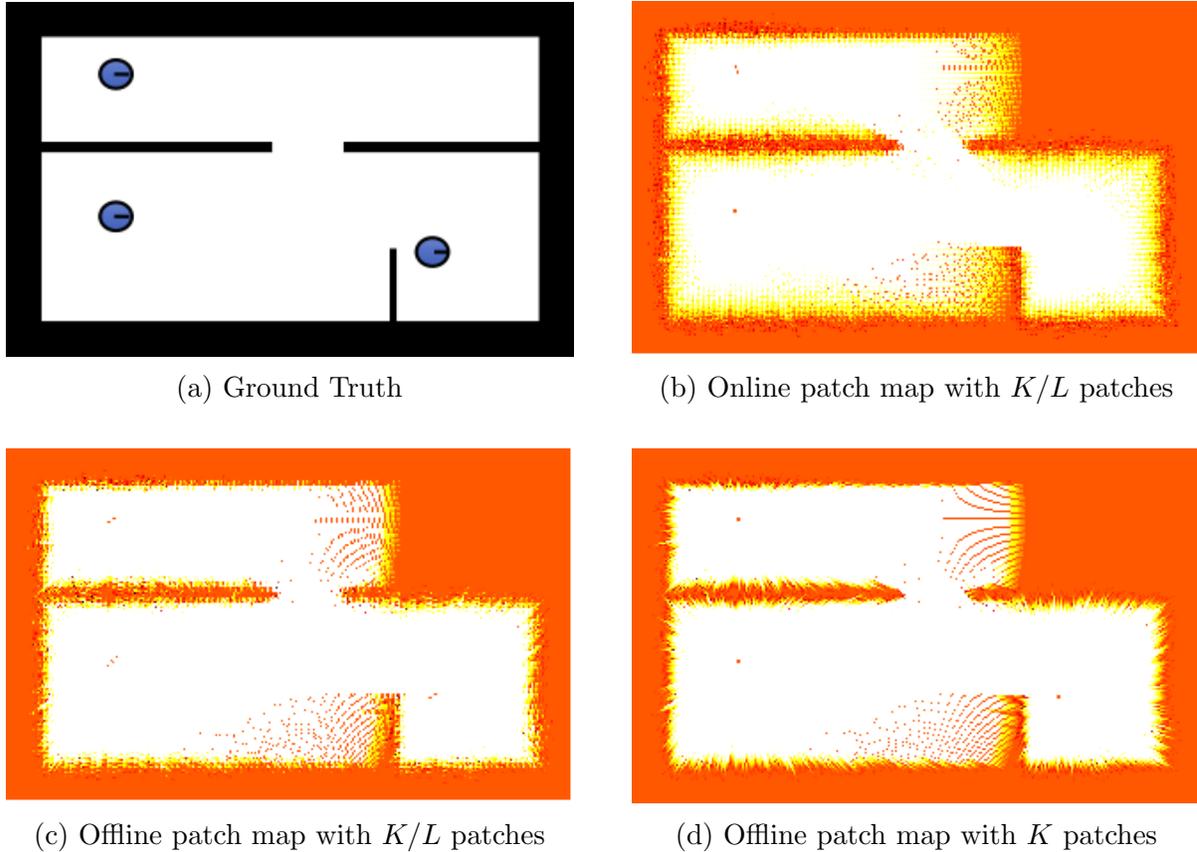


Figure 3.5: 2D example illustrating the resulting OG maps from the same measurements taken from three robot positions. The patch maps depicted use a patch size of $L = 9$. Each of these patch map algorithms performs better than traditional OGM. The offline algorithms incorporate information from the MAP solution. Computing the probability of a patch centered at each cell, yields the best marginal occupancy probability for each cell.

3.3 Chapter Summary

In conclusion, this chapter has presented a method to retain mutual information between neighbouring cells in occupancy grid mapping. The novel patch map algorithms are better able to capture the residual uncertainty in the map and their development was motivated by the observation in Chapter 2 that mutual information is greatest between proximate cells in OGM and that traditional OGM techniques do not retain this information. The online patch map is a generalization of traditional OGM using patches as opposed to individual cells. However, the offline patch map algorithm is a more computationally expensive batch solution that more accurately represents the marginal cell occupancy. The offline solution is a useful benchmark in comparing OGM techniques in 2D where the

full solution is computationally intractable. Section 6.1.1 will show experimental results of the patch map algorithm applied to 2D data. Also, Section 6.1.2 uses the offline patch map algorithm as a benchmark for large scale 2D experiments.

Chapter 4

Markov Chain Monte Carlo (MCMC)

Chapter 2 highlights that the full Bayesian solution to OGM is computationally intractable for real maps, but that mutual information is well approximated using approximate correlations. Therefore, Chapter 3 presents two novel patch-based algorithms to approximate the full solution. These algorithms are able to account for cell correlations between cells in the same patch, but the patch map algorithms have predefined patch boundaries. The patch map algorithms correlate cells within a patch, but cells near patch boundaries are therefore asymmetrically correlated with neighbouring cells. Section 3.2 proposes an extension to the offline patch map algorithm that computes a patch centered at each cell to ensure that each cell is symmetrically correlated to its neighbouring cells. However, this extension requires significantly more computation than even the offline patch map algorithm and is mainly useful for computing the marginal probability of each cell – a commonly used metric for evaluating an OG. In this chapter, we explore the usefulness of OGM beyond strictly the marginal cell probabilities. We present a method of approximating the full solution using an inference technique based on numerical sampling, also known as *Monte Carlo* techniques. These approximations work well in situations where the posterior distribution may not be of particular interest itself, but instead its expectation is the focus. Furthermore, these sampling algorithms provide *anytime* solutions as they improve their estimate with more computation, while also being able to provide a reasonable estimate quickly.

The offline patch map algorithm extends the work of Thrun (2003) to approximate the full solution. However, both of these methods require either the maximum a posteriori

(MAP) estimate of the occupancy grid or the ground truth map (available in simulation) before computing the residual uncertainty in each cell. In contrast, this chapter outlines a *Markov Chain Monte Carlo* (MCMC) method known as *Gibbs sampling* (Geman and Geman, 1984) that does not require these offline estimates of the map. MCMC methods are able to sample from a complex distribution of interest (in this case, the full Bayesian solution), thus enabling an accurate approximation of the distribution by averaging over many samples. MCMC approaches apply the Markov assumption by generating the next sample based solely on the current state or latest sample. This process generates a Markov chain, as the transition probabilities between samples are only a function of the single previous sample.

The *Metropolis algorithm* (Metropolis et al., 1953; Metropolis and Ulam, 1949) was the first MCMC method. It sampled from a proposal distribution and accepted the new sample based on an acceptance probability. The *Metropolis-Hastings algorithm* (Hastings, 1970) extended this algorithm by using an arbitrary transition probability function. Both of these algorithms typically sample from a multivariate distribution. In contrast, the Gibbs sampler (Geman and Geman, 1984) only considers a univariate conditional distribution, which is generally easier to compute. Moreover, the Gibbs sampler does not have parameters that require tuning like most other MCMC methods including Hamiltonian Monte Carlo (HMC) (Neal, 2011). In this chapter, we show that because we are able to compute the probability of a patch in the occupancy grid being occupied, given the occupancy of all other cells in the map, we can therefore use Gibbs sampling to sample from the full posterior with no approximations.

We first published this novel MCMC Gibbs sampling algorithm in Merali and Barfoot (2013). It has since been extended to a factor graph approach by Dhiman et al. (2014) and Dhiman (2019) and used for viewpoint planning by Hou et al. (2019). In addition, Liu and von Wichert (2014) used an MCMC algorithm to provide semantic information about the environment. Liu and von Wichert (2014) extend the work of Song-Chun Zhu et al. (2000) by applying an MCMC algorithm to occupancy grid mapping. Their algorithm does not seek to improve the occupancy estimate as ours does, but instead uses an OG map to estimate the semantic map from which it was derived. The improved OGM techniques discussed in this thesis could serve to aid algorithms seeking to estimate semantic map information.

Section 4.1 presents the novel MCMC Gibbs sampling algorithm that we published in Merali and Barfoot (2013). This algorithm samples one cell at a time to generate map

samples from the full posterior. We show that with sufficient samples, this algorithm can accurately estimate the full Bayesian solution. Further experimental results of this algorithm will be shown in Section 6.1.3. Section 4.2 generalizes the cell sampling approach to sample patches of cells. The two algorithms produce similar estimates of the posterior, but the patch samples are more computationally expensive. However, the benefit of this extension will be realized in Chapter 5 where we examine the patch prior. Both of the MCMC sampling algorithms presented in this chapter are able to estimate the full posterior without the patch-boundary effect observed in Chapter 3 and both provide an *anytime* solution, in contrast to the patch map algorithms.

4.1 Sampling Cells

One powerful family of sampling techniques are known as *Markov Chain Monte Carlo* (MCMC) techniques. These techniques have their origins in physics (Metropolis and Ulam, 1949), but only in the late 1980s did they make a significant impact on the field of statistics (Bishop, 2006). The Markov assumption states that the next state is independent of all previous states, given the current state. Therefore, for a sampling algorithm, this means that the next sample depends on the current state and no other states.

The key to a MCMC algorithm is the acceptance probability. Once a new sample is proposed, it is only accepted with the probability given by the *acceptance probability*. The basic *Metropolis* algorithm (Metropolis et al., 1953) assumes that the proposal distribution is symmetric. This algorithm is generalized by the *Metropolis-Hastings* algorithm (Hastings, 1970) that removes the assumption that the proposal distribution is a symmetric function of its arguments. All of these MCMC methods require that the state transition probabilities leave the posterior distribution invariant and that the Markov chain be ergodic (Barber, 2012).

Gibbs sampling (Geman and Geman, 1984) can be viewed as a special case of the Metropolis-Hastings algorithm. In fact, it is seen as an efficient case since the acceptance probability is always one (Bishop, 2006). The idea behind Gibbs sampling is to update each component of the state by sampling from its conditional distribution given all other components of the state. In this section, we achieved this by updating one cell in the occupancy grid, m_k , given all of the measurements, z , their respective poses, x , and the occupancy of all other cells in the map, m_{-k} . The new value computed for each cell is used to condition the next distribution. The algorithm states that we do this for each

cell in the map to complete one iteration. By taking the average of several iterations, the resulting map will asymptotically converge to the true posterior.

To show that this method samples from the correct distribution, we must show that $p(m|z, x)$ is invariant of each Gibbs sampling step and therefore the whole Markov chain. To show this, we note that $p(m|z, x) = p(m_k|z, x, m_{-k})p(m_{-k}|z, x)$. $p(m_{-k}|z, x)$ is clearly invariant, as m_{-k} is left unchanged at each step. Furthermore, each step samples from the correct conditional distribution, $p(m_k|z, x, m_{-k})$, by definition. Therefore, the joint distribution is invariant.

Furthermore, we must show that it is ergodic. A sufficient condition for ergodicity is that none of the conditional distributions be zero for any value of m_k . As long as we can ensure this, the Gibbs sampling technique will sample from the correct posterior distribution.

4.1.1 Cell Sampling Formulation

To implement Gibbs sampling for OGM we compute the marginal occupancy probability of a specific cell, m_k , given all of the measurements, z , their respective poses, x , and the occupancy of all other cells in the map, m_{-k} . We compute this probability as

$$\begin{aligned} p(m_k = 1|z, x, m_{-k}) &= \frac{p(m_k = 1|x, m_{-k})p(z|x, m_{-k}, m_k = 1)}{p(z|x, m_{-k})} \\ &= \frac{p(m_k = 1|m_{-k})p(z|x, m_{-k}, m_k = 1)}{p(z|x, m_{-k})}. \end{aligned} \quad (4.1)$$

Similarly, we can compute the probability that the cell, m_k is unoccupied as

$$p(m_k = 0|z, x, m_{-k}) = \frac{p(m_k = 0|m_{-k})p(z|x, m_{-k}, m_k = 0)}{p(z|x, m_{-k})}.$$

Comparing the two equations above, we note that the denominators are the same. Therefore, by computing the *odds* of $p(m_k = 1|z, x, m_{-k})$, we avoid having to compute the denominator:

$$\begin{aligned} o(m_k|z, x, m_{-k}) &= \frac{p(m_k = 1|z, x, m_{-k})}{p(m_k = 0|z, x, m_{-k})} \\ &= \frac{p(m_k = 1|m_{-k})p(z|x, m_{-k}, m_k = 1)}{p(m_k = 0|m_{-k})p(z|x, m_{-k}, m_k = 0)}. \end{aligned}$$

The computational cost is further reduced by computing the logarithm of this odds:

$$\begin{aligned}
 l(m_k|z, x, m_{-k}) &= \log \left(\frac{p(m_k = 1|z, x, m_{-k})}{p(m_k = 0|z, x, m_{-k})} \right) \\
 &= l(m_k|m_{-k}) + l(z|x, m_{-k}, m_k) \\
 &= \underbrace{l(m_k|m_{-k})}_{\text{cell prior}} + \sum_{n=1}^N \underbrace{l(z_n|x_n, m_{-k}, m_k)}_{\text{sensor model}}, \tag{4.2}
 \end{aligned}$$

where the notation $l(\cdot)$ is used to represent the *log odds*. Note that we have made the assumption that measurements are independent of one another given the occupancy of all cells in the map. This assumption is true for a static world, unlike traditional occupancy grid mapping, which assumes that measurements are independent of one another given the occupancy of a single cell. Also, note that the probability $p(m_k = 1|z, x, m_{-k})$ can be recovered from Equation (4.2),

$$p(m_k = 1|z, x, m_{-k}) = \frac{\exp(l(m_k|z, x, m_{-k}))}{1 + \exp(l(m_k|z, x, m_{-k}))}.$$

Computing $l(z_n|x_n, m_{-k}, m_k)$ in Equation (4.2), is simply the log odds of the forward sensor model. As we did in previous chapters, we model the sensor as having Gaussian noise about the true range, f , and standard deviation of d cells, $\mathcal{N}(f, d^2)$. The MCMC algorithm samples from $p(m|z, x)$ using Gibbs sampling via the conditional distribution in Equation (4.2). The log odds domain is convenient to prevent numerical instability. Therefore a cell is occupied if,

$$\log \left(\frac{\text{rand}}{1 - \text{rand}} \right) < l(m_k|z, x, m_{-k}),$$

and unoccupied otherwise; rand is a random variable drawn from the uniform distribution $(0, 1)$.

The term $l(m_k|m_{-k})$ in Equation (4.2) is the prior for the cell occupancy. This value will be explored further in Chapter 5, but in this chapter, we assume that the cell is independent of all other cells, i.e. $l(m_k|m_{-k}) = l(m_k)$. Furthermore, if we assume a uniform cell prior, i.e. $l(m_k) = 0$, then Equation (4.2) is simplified to,

$$l(m_k|z, x, m_{-k}) \approx \sum_{n=1}^N l(z_n|x_n, m_{-k}, m_k). \tag{4.3}$$

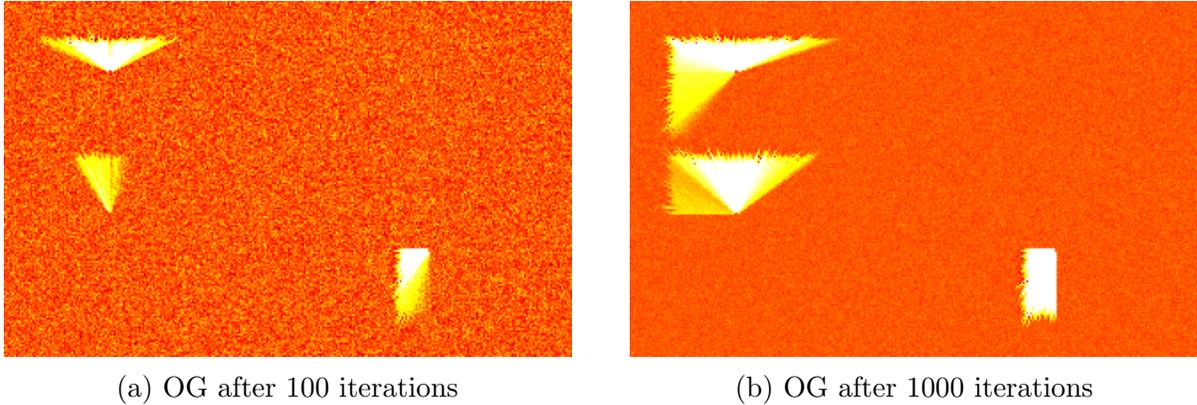


Figure 4.1: 2D example of MCMC OGM with a uniform prior. The algorithm has not reached a stationary distribution after 1000 iterations. The legend used is shown in Figure 1.10g.

4.1.2 Cell Sampling Application

In Merali and Barfoot (2013) we showed that using a uniform cell prior, $p(m_k) = 0.5$, required more iterations (> 200) to reach a stationary distribution, than using a more realistic prior of $p(m_k = 1) = 0.15$ that was able to reach a stationary distribution in less than 20 iterations. However, these results were from experiments that used an exploration algorithm to ensure the cells were well mapped. Repeating this experiment on the sparsely mapped 2D example used thus far in this thesis, we found that the algorithm had difficulty converging to a stationary distribution using a uniform prior. Figure 4.1 illustrates the resulting OG map after 100 and 1000 iterations using a uniform prior and it can be seen that the map has still not reached a stationary distribution. Furthermore, we see that given the Gaussian sensor model, shorter range measurements converge before longer ones. The Gaussian sensor model depends on the first occupied cell, f , which requires that all cells between the robot and the first occupied cell be unoccupied. Therefore, longer range measurements (i.e., higher values of f) are statistically less likely than shorter ones if cells are independent in the prior. However, each range measurement introduces cell correlations between the cells it traces in the OG map. Therefore, the cells are correlated in the posterior and the MCMC algorithm provides a way of representing those correlations, unlike traditional OGM.

By reducing the cell prior to a more realistic $p(m_k = 1) = 0.234$, Figure 4.2a shows that the MCMC algorithm is better able to converge to a stationary distribution than Figure 4.1b, because each cell in the map was initialized by drawing from the cell prior.

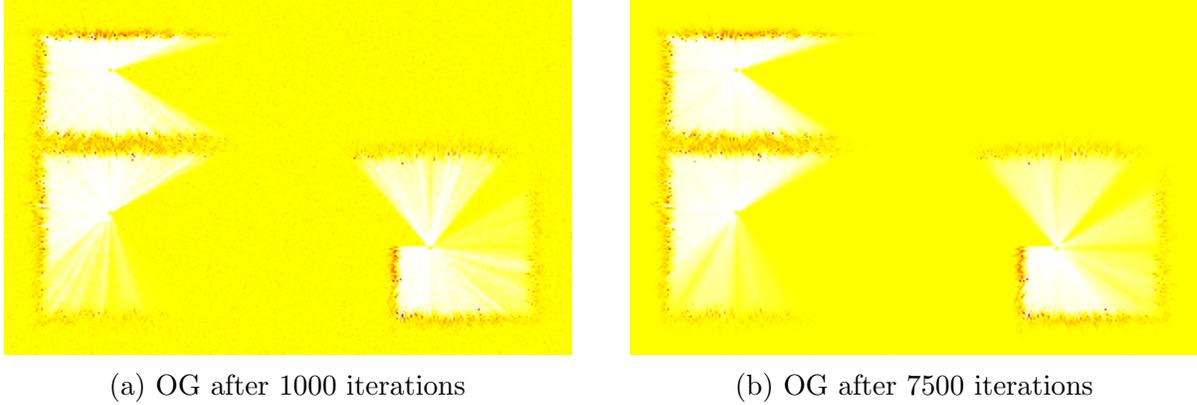


Figure 4.2: 2D example of MCMC OGM with a prior of $p(m_k = 1) = 0.234$. The algorithm has not reached a stationary distribution after 7500 iterations. The legend used is shown in Figure 1.10g.

The prior will be examined in depth in Chapter 5, where we show that $p(m_k) = 0.234$ is more realistic. However, Figure 4.2b illustrates that even after many more iterations, the algorithm has not reached a stationary distribution. We therefore find that starting the MCMC algorithm with a realistic map is especially important for sparsely mapped environments. Hence, we start the MCMC algorithm with a map obtained by thresholding the result of traditional OGM. Beginning with a stationary distribution, the MCMC algorithm will continue to draw samples from the posterior and these samples can be used to estimate various statistics about the map. For example, by averaging the occupancy of a specific cell over many MCMC samples, we are able to estimate the marginal cell occupancy of each cell as depicted in Figure 4.6.

Similar to the observation by Paskin and Thrun (2005), we find that the MCMC algorithm can be sped up by preprocessing and storing all of the measurements that affect a given cell (or patch). Because only measurements that map m_k will affect its conditional distribution, the algorithm only considers those relevant measurements to draw each sample. Furthermore, when iterating through the relevant measurements, some will traverse an occupied cell before m_k . These measurements can also be omitted according to the Gaussian sensor model since the value of m_k will not influence the conditional distribution. Finally, the *burn-in* period can be virtually eliminated by initializing the MCMC algorithm with a thresholded OG generated by traditional OGM. Thresholding each cell in an OG map is a common technique to generate a binary map from an OG and discussed in further detail in Chapter 6. To initialize the MCMC algorithm we suggest thresholding each cell in the OG map generated using traditional OGM at the cell prior

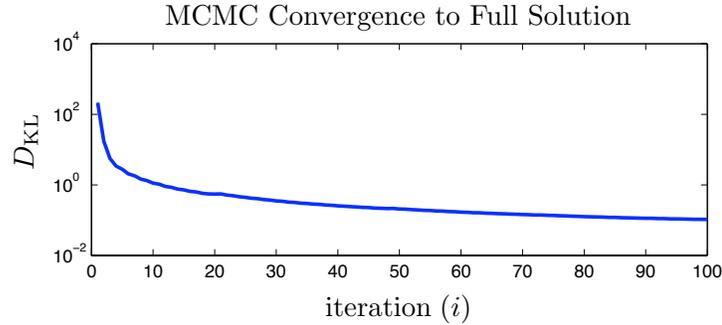


Figure 4.3: The MCMC Gibbs sampling method converges to the full Bayesian solution for occupancy grid mapping with more samples. The Kullback-Leibler divergence (D_{KL}) quantifies the difference between the two probability distributions.

($p(m_k) = 0.234$ in this case). The MCMC algorithm is therefore able to immediately begin drawing samples from the full posterior.

The MCMC algorithm draws map samples (binary maps in the case of OGM) from the posterior, $p(m|z, x)$. These samples can then be used to compute various statistics such as the marginal occupancy probability of each cell. The occupancy grid mapping literature most often uses this metric to compare occupancy grids because traditional OGM makes the cell independence assumption, so other statistics are computed from these marginals. In this regard, the MCMC algorithm performs quite well and we highlight 2D experimental results with this technique in Section 6.1.3. When compared to the full Bayesian solution on a 1D toy example, the MCMC algorithm is visually indistinguishable from the full solution after only 100 iterations. This convergence to the full Bayesian solution is illustrated in Figure 4.3 by computing the Kullback-Leibler divergence, on a cell-by-cell basis, from the full solution at each iteration. In particular, the convergence plot highlights that the MCMC method moves quickly from a random sample to sampling from the desired posterior. Thus by discarding the first few samples (approximately 10 in this case), we are left with a Gibbs sequence that samples from the correct posterior distribution. To further highlight this fact, we compute the MCMC algorithm on the same data used to generate Table 3.1 with 10^5 MCMC iterations and the resulting average D_{KL} was 3.72×10^{-5} ; more than two orders of magnitude lower than any of the algorithms presented in Section 2.5 or Chapter 3.

This highlights another key benefit of the iterative MCMC algorithm over the batch solution presented for the offline patch map algorithm, which is the fact that the MCMC algorithm provides an *anytime* solution and will continue to improve its estimate given

more computation time. This is beneficial for both a quick estimate when computation time is limited and a more accurate estimate when computation time is available. This is particularly true when a thresholded OG generated by traditional OGM is used as the first iteration of the MCMC algorithm; thus eliminating the *burn-in* period.

Figure 4.4 illustrates that the MCMC algorithm is able to produce marginal cell probabilities that approach the full Bayesian solution and that its accuracy increases with more samples. This result was also computed by initializing the algorithm with a random map and the first 10 samples are thus used as a *burn-in* period. Compared to the full solution, the sampling algorithm is more computationally efficient. The full solution is exponential in the number of cells in the map, $\mathcal{O}(2^K)$, whereas the MCMC Gibbs sampling method is polynomial, $\mathcal{O}(\text{maxSamples} \times K \times N)$. Thus, each sample approaches the complexity of traditional OG mapping, $\mathcal{O}(N \times F)$, but without the assumptions made by traditional OGM that were discussed in Section 2.4. However, traditional OGM allows the map to be updated incrementally with each new measurement, where each update is linear in the number of cells that the measurement affects, $\mathcal{O}(F)$. In contrast, the MCMC sampling algorithm presented here is a batch solution that estimates the univariate probability given all measurements that affect it, at each iteration. Because the MCMC algorithm samples from the full posterior, it is able to estimate various marginal probabilities. For example, the algorithm is able to compute the mutual information between cells to produce estimates that are indistinguishable from the full solution for mutual information, like that shown in Figure 2.9. Therefore, the MCMC algorithm is a robust method to approximate the full solution with increasing fidelity based on the computation available.

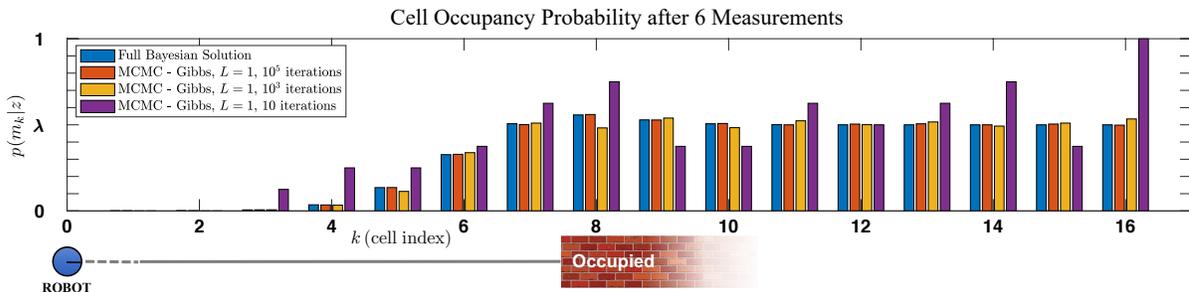


Figure 4.4: Occupancy grid after six measurements. The robot is at $k = 0$, the true first occupied cell is at $k = 8$ (therefore $f = 8$). The full Bayesian solution illustrates the true occupancy probability of each cell. The MCMC algorithm estimates the full solution with increasing fidelity as it draws more samples from the posterior.

4.2 Sampling Patches

Section 4.1 presented a novel MCMC Gibbs sampling algorithm for OGM and Chapter 3 developed algorithms that are able to compute the full solution for a patch of cells. In this section, we generalize the the MCMC Gibbs sampling algorithm presented in Section 4.1 by incorporating the insights of the patch map algorithms. The patch map algorithms were developed to capture the cell correlations between proximate cells, but Section 4.1 explained that the MCMC cell sampling algorithm is able to capture the mutual information between cells. However, Chapter 5 will highlight the importance of cell correlations in the prior and how patch sampling can incorporate these correlations. Therefore, this section extends the MCMC algorithm to sample a patch of cells as opposed to a single cell.

4.2.1 Patch Sampling Formulation

Equation (4.1) defines the MCMC update term for a single cell, which is equivalent to a patch size of $L=1$. Therefore, we can generalize this to the conditional distribution for a patch as

$$\begin{aligned}
 p(a_k = w | z, x, m_{-a_k}) &= \frac{p(a_k = w | x, m_{-a_k}) p(z | x, m_{-a_k}, a_k = w)}{p(z | x, m_{-a_k})} \\
 &= \frac{p(a_k = w | m_{-a_k}) \prod_{n=1}^N p(z_n | x_n, m_{-a_k}, a_k = w)}{p(z | x, m_{-a_k})}, \quad (4.4)
 \end{aligned}$$

where the term m_{-a_k} represents the occupancy of all cells in the map, m , except those in the patch a_k .

Equation (4.4) has made the static-world assumption that measurements are independent of one another given the occupancy of all cells in the map. This assumption is quite different than the assumption made in traditional OGM that assumes measurements are independent given the occupancy of one cell. The patch prior will be explored further in Chapter 5, but in this section we assume that the prior occupancy probability of a patch is independent of all other cells in the map, $p(a_k = w | m_{-a_k}) = p(a_k = w)$, and we assume

a uniform patch prior, $p(a_k = w) = 1/W$. Therefore Equation (4.4) is simplified to

$$p(a_k = w | z, x, m_{-a_k}) \approx \eta \prod_{n=1}^N p(z_n | x_n, m_{-a_k}, a_k = w), \quad (4.5)$$

where η is a normalizing constant.

4.2.2 Patch Sampling Application

Algorithm 1 describes the MCMC Gibbs sampling algorithm. Note that a patch size of $L = 1$ is equivalent to the MCMC method presented in Section 4.1 and in Meralli and Barfoot (2013). The algorithm will return the desired number of samples, `maxSamples-burnIn`, from the full posterior. If the algorithm begins with a random map (typically drawn from the patch prior), it is common to discard the early samples as the Gibbs sequence has not converged (the discarded samples as known as *burn-in*). The remaining samples in the Gibbs sequence can be used to compute various statistics about the full solution. For example, the occupancy probability of a particular cell or patch can be computed by averaging the occupancy of that cell or patch over many samples. This result is highlighted in Figure 4.5 for various patch sizes, L . Note that the value chosen for the patch size has little effect on the accuracy of the marginal probability, but a larger patch size increases the computational complexity as each update requires computing the probability of $W = 2^L$ patch configurations. The need to compute all W patch configurations will be revisited in Chapter 5. Figure 4.5 shows that the MCMC algorithm is not overconfident or underconfident near the obstacle boundaries like traditional OGM.

To showcase the effectiveness of the MCMC algorithms, Figure 4.6 illustrates the 2D OG map computed using the MCMC algorithm and a patch size of $L = 1$. As explained in Section 4.1, the *burn-in* period was avoided by having the MCMC algorithm start with an OG created by thresholding a map developed by traditional OGM. Figure 4.6b shows the MCMC algorithm is able to accurately estimate the cell marginals with only 100 iterations. Figure 4.6c illustrates that the result is more accurate with more iterations – this is particularly visible in the *unknown* cells approaching a homogeneous prior probability. However, Figure 4.6d illustrates the map after 10 000 iterations and some of the cells mapped by long range measurements appear to be approaching the prior. In fact, this only appears in cells mapped by few, long range measurement. This anomaly is a result the MCMC algorithm sampling one of the cells along the measurement ray as

Algorithm 1: MCMC Gibbs sampling patches for OGM

Given $z, x, L, \text{maxSamples}, \text{burnIn}$
 Generate a starting map, $m^{(0)}$, from the prior, $p(a_k|m_{-a_k})$, or by thresholding an OG
 Define m^r as $m^{(0)}$

for $i = 1$ to maxSamples **do**
 for $t = 1$ to K/L **do**
 Select random patch a_k from m^r
 Sample $a_k^{(i)}$ from $p(a_{k,w}|z, x, m_{-a_k}^r)$
 Define a_k^r , in m^r , as $a_k^{(i)}$
 end for
 Set $m^{(i)} = m^r$
end for

return set of samples, $m^{\text{burnIn}:\text{maxSamples}}$, from the full posterior

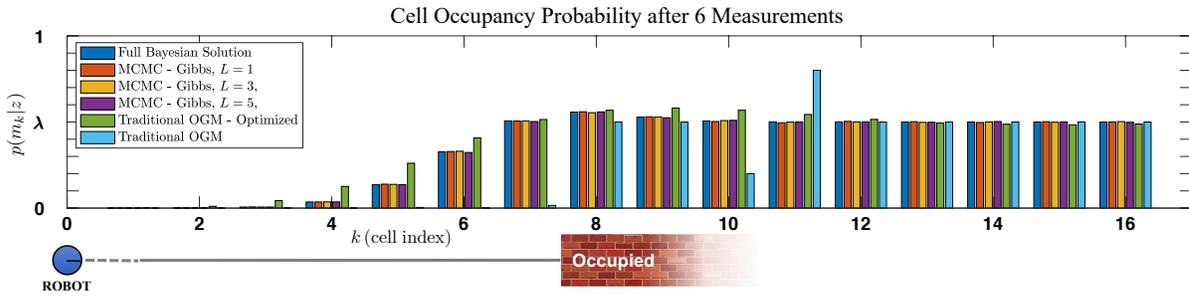
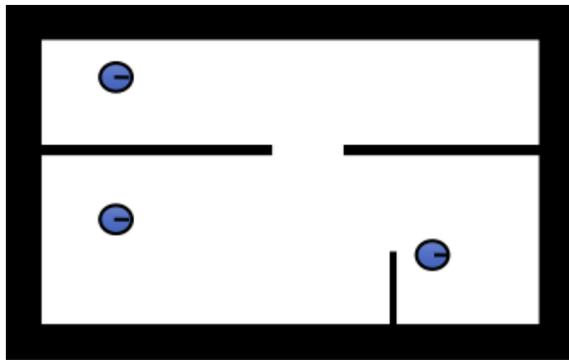


Figure 4.5: Occupancy grid after six measurements. The robot is at $k = 0$, the true first occupied cell is at $k = 8$ (therefore $f = 8$). The full Bayesian solution illustrates the true occupancy probability of each cell. The MCMC algorithms are computed for various values of L with 10^5 iterations each. Each MCMC algorithm approaches the full solution with more samples. The traditional OGM algorithm is also shown for comparison with an update term from the literature and optimized values.

occupied and thus changing the value of f . The cells beyond this occupied cell therefore revert to the prior, $p(m_k) = 0.5$, and are more likely to be occupied without the influence of the range measurement that indicates that they are unoccupied. As highlighted in Section 4.1, decreasing the prior to a lower, more realistic, value helps achieve convergence from a random prior map. In Figure 4.6e and Figure 4.6f, we see that lowering the cell occupancy prior to a more realistic value of $p(m_k) = 0.234$, yields a more accurate map without the anomaly observed with a uniform prior. Again, a larger patch size would increase computational cost, but would not provide a more accurate result than a patch size of $L = 1$ with a uniform prior.



(a) Ground Truth.

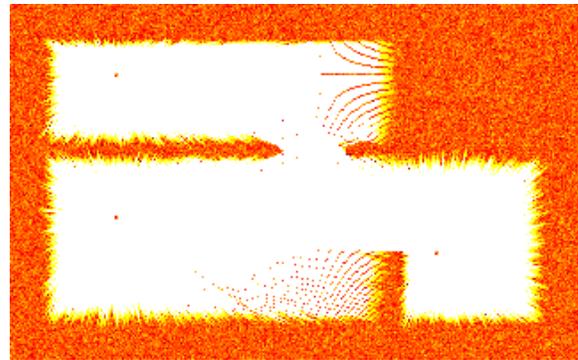
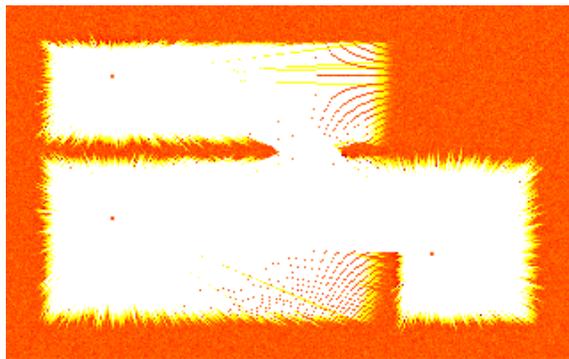
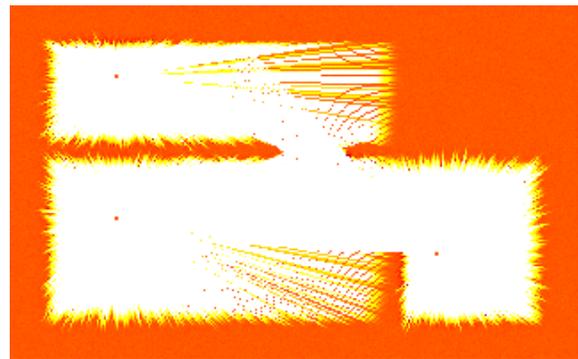
(b) MCMC with $L=1$ and 100 iterations.(c) MCMC with $L=1$ and 1000 iterations.(d) MCMC with $L=1$ and 10 000 iterations.(e) MCMC with $L = 1$, $p(m_k) = 0.234$ and 1000 iterations.(f) MCMC with $L = 1$, $p(m_k) = 0.234$ and 10 000 iterations.

Figure 4.6: 2D example illustrating the resulting OG maps from the same measurements taken from three robot positions. The MCMC algorithm is able to better approximate the full Bayesian solution with more iterations and thus perform better than traditional OGM and the patch map algorithms. A more accurate cell prior yields a better result. The legend used is shown in Figure 1.10g.

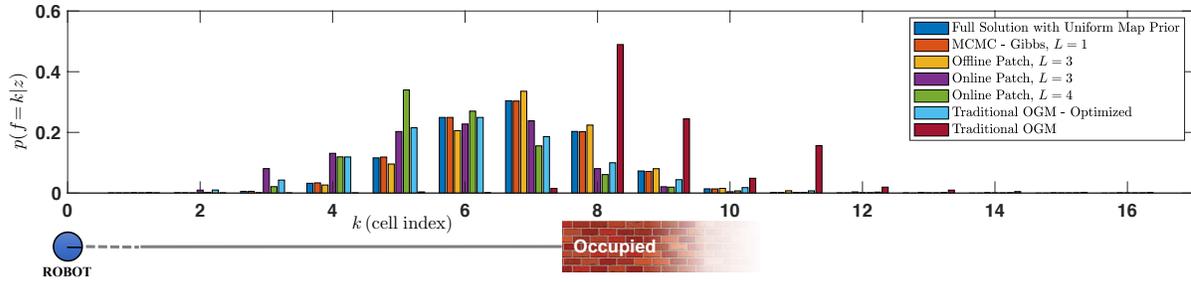


Figure 4.7: Probability that cell k is the first occupied cell, f , after six measurements. The robot is at $k = 0$, the true first occupied cell is at $k = 8$. With enough iterations, the MCMC algorithm is able to accurately approximate the full Bayesian solution. The patch map algorithms do not approximate it as well due to the patch boundaries. Increasing the patch size can even make the estimate worse by moving the patch boundary. The traditional OGM algorithm is also shown for comparison with an update term from the literature and optimized values.

4.2.3 Estimating the First-Occupied-Cell

To further illustrate the benefit of the MCMC algorithm over the patch map algorithms, we introduce a new metric by which to compare OGM techniques that uses the mutual information between cells. In the 1D toy example, we use the resulting OG to compute the first occupied cell, f . Note that we define $p(f = k|z, x)$ as the probability that the cell m_k is occupied and that all cells between the robot (at cell $k = 0$) and this cell are unoccupied,

$$p(f = k|z, x) = p(m_{1:k-1} = 0, m_k = 1|z, x). \quad (4.6)$$

Figure 4.7 illustrates that the MCMC algorithm is able to estimate $p(f|z, x)$ nearly as well as the full Bayesian solution by counting the number of samples that have the pattern f and dividing that sum by the total number of samples. Given that each map is equally likely in the prior, the full solution estimates that f is most likely 7 as opposed to 8 because higher values of f are less likely; we address this concern in Chapter 5. In contrast to the full solution, both the online and offline patch map algorithms multiply the probability of each patch configuration that matches the pattern f , thus assuming independence between patches. Figure 4.7 highlights that a larger patch size does not necessarily ensure a better estimate as it moves the patch boundary, which can result in a worse estimate. Similarly, the traditional OGM algorithms compute this quantity on

a cell-by-cell basis,

$$p(f = k|z, x) \approx p(m_k = 1|z, x) \prod_{i=1}^{k-1} p(m_i = 0|z, x). \quad (4.7)$$

The MCMC algorithm is shown to perform better than the other algorithms because it does not make an independence assumption and samples from the full posterior. Note that the extension to the offline patch map algorithm presented in Section 3.2, where the patches overlap, was developed to produce the best marginal cell occupancy probability, but it is not clear how to compute statistics like $p(f = k|z, x)$ from this method, except to have nonoverlapping patches.

4.3 Chapter Summary

In conclusion, the MCMC Gibbs sampling algorithm for OGM presented here is a novel method to draw samples from the full posterior and is therefore a powerful method to estimate the full Bayesian solution and is computationally tractable in higher dimensions. The MCMC method is an *anytime* algorithm and will continue to improve its estimate as more iterations are drawn and these iterations can be used to compute statistics like marginal cell occupancy, or the occupancy of a cluster of cells like the the first occupied cell, f . Furthermore, this method does not have the asymmetric cell correlations that were observed in the patch map. We also extended the MCMC algorithm to sample patches of any size (as opposed to cells), and the value of this extension will be presented in Chapter 5.

Chapter 5

Cell Correlations in the Prior

Thus far, this thesis has assumed that the map (or cell) prior is uniform, with the exception of the offline patch map algorithm in Section 3.2 that has the ability to use the ground truth map or MAP solution. Recently, Georgiou et al. (2015); Georgiou et al. (2017) have presented a method to extract ground truth information from an architectural drawing or a floor plan, then initialize the probability of each cell in traditional OGM based on the ground truth map. Specifically, Georgiou et al. (2017) use a FastSLAM algorithm (Montemerlo et al., 2003) because of its ability to separate the localization and mapping tasks using Rao-Blackwellization. Where such prior information is available, most of the OGM algorithms could benefit from incorporating them as an initial estimate. In Chapter 4 we showed that initializing the MCMC algorithm to a better estimate of the map reduced or eliminated the *burn-in* period typically seen with such sampling algorithms. If a floor plan map is available, an MCMC mapping algorithm would benefit from starting with that map as an initial estimate.

Some researchers have assumed that the general structure of the map is known a priori, as opposed to an estimate of the map. For example, Milstein (2008) uses prior information about the shape of the building to construct a more accurate OG in the skeletal FastSLAM algorithm. Specifically, Milstein recognizes that a robot traveling indoors is typically traveling in a straight corridor and can use this information to improve the robot’s localization estimate and therefore the map in the SLAM algorithm. Similarly, Peasley et al. (2012) are able to produce more accurate OG maps by focusing on the reconstruction of man-made environments. Peasley et al. do this by removing drift and rotation from the robot pose estimate in a SLAM algorithm, thus yielding an improved map estimate. Furthermore, Schaefer et al. (2018a) do not incorporate a priori

information in a SLAM algorithm, but instead seek to extract polylines (e.g., straight walls in a man-made environment) from the data.

This thesis has thus far focused on the full Bayesian solution for OGM presented in Section 2.3 as the standard by which all other OGM algorithms should be measured. However, when formulating the full Bayesian solution in Section 2.3, we began by assuming that we have no prior information and therefore every possible map is equally likely before measurements are incorporated. In this chapter, we question this assumption and explore how to incorporate a more realistic prior into occupancy grid mapping. Section 5.1 will illustrate a method of incorporating prior information into the full solution, so it can be used as a new benchmark for mapping in 1D.

Section 5.2 moves beyond the 1D toy example and introduces a novel approach to incorporating prior information into the MCMC algorithm presented in Chapter 4. This approach is more general than other approaches that assume a specific structure for the map being estimated. We show that accounting for local cell correlations in the prior enables the mapping algorithm to better estimate areas between range measurements and to reconcile range measurement noise to maps that more closely represent the true underlying map.

5.1 Full Bayesian Solution with a Data-Driven Prior

In Section 2.3 we presented the full Bayesian solution for OGM, where we assumed that each map is equally likely, $m^r = 1/R$. In this chapter, we seek to apply a more realistic data-driven prior to each map. Specifically, for the 1D toy example, we assume that the robot is stationary at cell $k=0$ and that there is one obstacle that is equally likely to start at any cell in the map, m_k , or to not exist. As a consequence, there are $K+1$ possible values of f and the prior probability of each first-occupied cell is $p(f) = 1/(K+1)$. In addition to assuming that all cells to the left of the obstacle are unoccupied (one obstacle per map), we also assume that all cells to the right are occupied (e.g., a thick wall). By making this simplifying assumption, only $K+1$ of the $R=2^K$ maps have a nonzero probability, yielding a large computational efficiency. Therefore, we need only compute $p(m^r|z, x)$ for $K+1$ maps as the others have a probability of zero. Figure 5.1 illustrates this prior for a map with $K=3$ cells and therefore $2^K=8$ possible maps. In particular, Figure 5.1a shows the $K+1$ maps with a nonzero (and equal) prior probability, while Figure 5.1b shows the maps that have zero probability given this data-driven prior.

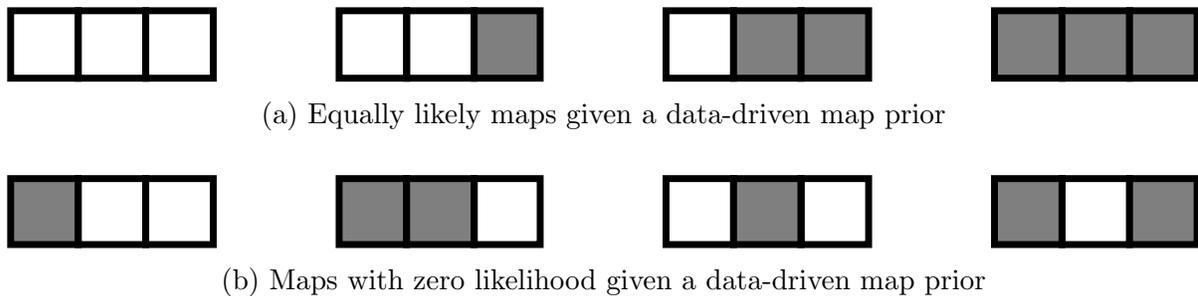


Figure 5.1: A 1D Map with $K = 3$ cells has these $2^K = 8$ possible maps. A uniform prior assumes all 8 maps are equally likely, whereas a data-driven prior assumes that the $K + 1 = 4$ maps in (a) are equally likely and the remaining maps in (b) have zero probability. The robot is to the left of the map and measures to the right in this example.

After computing the posterior probability of the $K + 1$ maps, statistics about the map including marginal cell occupancy, the first occupied cell, and mutual information between cells can be computed using the same methods presented in previous chapters. Figure 5.2 shows the mutual information between every pair of cells in the 1D map from the map probabilities computed using the data-driven prior. The figure highlights that most of the mutual information is at the wall boundary between cells $k = 7$ and $k = 8$, and this is generally the most critical part of an OG map as most algorithms seek to estimate the transition from unoccupied to occupied areas. Notably, traditional OGM discards this mutual information by assuming cells are independent. Similar to Figure 2.9, Figure 5.2 shows that there is higher mutual information between neighbouring cells and that the mutual information quickly approaches zero for cells that are farther

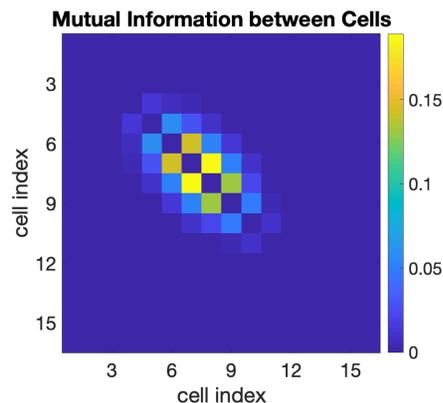


Figure 5.2: Mutual information between cells in a 1D map from the full solution using a data-driven prior. Note that the wall, or first-occupied-cell, is at $f = 8$ and the robot is at cell $k = 0$. A data-driven prior results in more mutual information between neighbouring cells than a uniform prior, but it still approaches zero as cells are farther apart.

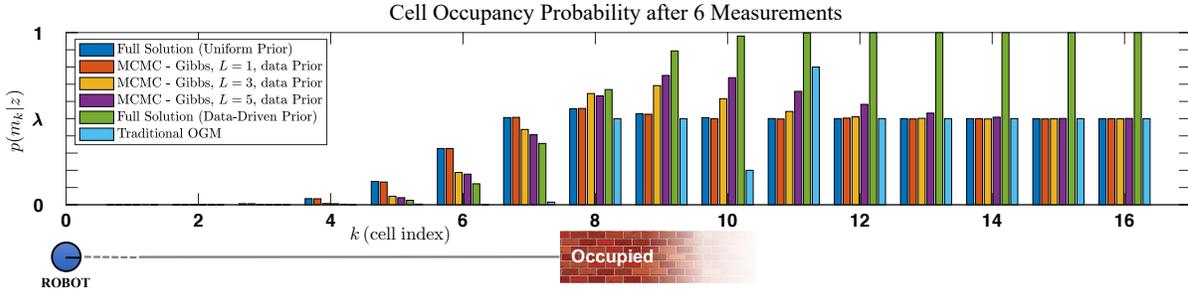


Figure 5.3: Occupancy grid after six measurements. The robot is at $k = 0$, the true first occupied cell is at $k = 8$ (therefore $f = 8$). The MCMC algorithms compute cell marginals from 10^5 map samples, with varying patch sizes.

apart in the map (approximately two cells in this example). However, comparing the magnitude of the mutual information in the two figures reveals that a data-driven prior adds additional mutual information to the map from the prior (the uniform prior added no mutual information). Thus, the case for keeping this mutual information is stronger with a data-driven prior and the argument for focusing on proximate cell correlations remains valid as the mutual information approaches zero for non-proximate cells.

Figure 5.3 illustrates the marginal cell occupancy probability as derived from the full solution with a uniform prior and with a data-driven prior, after six range measurements drawn from $p(z_n | f_n = 8) = \mathcal{N}(f_n, d^2)$. The MCMC algorithms plotted in the figure will be discussed in Section 5.2. The data-driven prior results in an occupancy grid that is more confident that cells before the wall are unoccupied, and that cells after the wall are occupied. Furthermore, the data-driven prior results in cells past the wall being likely to be occupied, as opposed to being estimated at the cell prior as is the case with the uniform map prior. Intuitively, this result makes sense as there are no measurements of these cells and we have defined the data-driven prior to have all cells past the wall to be occupied. Figure 5.3 also shows the OG created using traditional OGM for comparison.

Often, OG mapping algorithms are evaluated by their accuracy in computing the marginal cell probability of each cell in the map, but this marginalization does not use the mutual information between cells. Alternatively, Section 4.2.3 shows that in the 1D toy example, the maps can be evaluated on their ability to estimate the first-occupied cell, f . Specifically, Equation (4.6) defines $p(f = k | z, x)$ as the probability that cell m_k is occupied and all cells between the robot (at cell $k = 0$) and this cell are unoccupied. The algorithms plotted in Figure 5.4 show the estimate of $p(f | z, x)$ for various OGM techniques. In particular, the figure shows the probability of the first-occupied cell using

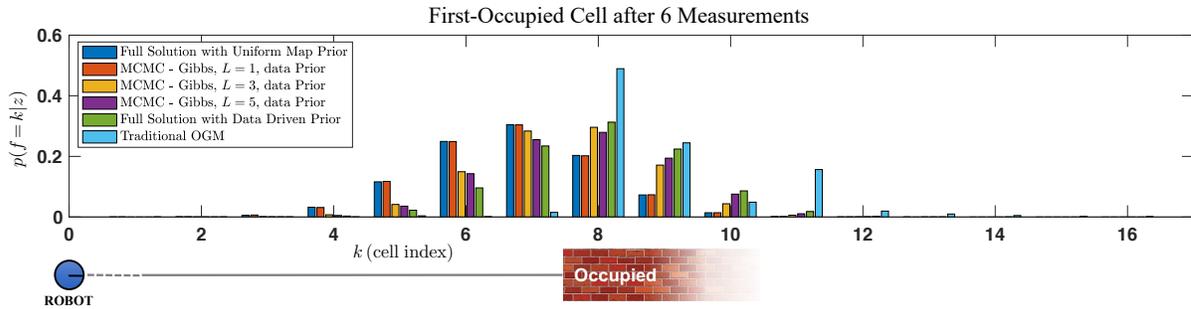


Figure 5.4: Using various occupancy grid mapping algorithms to estimate $p(f = k|z, x)$. The robot is at $k = 0$ and the true first-occupied cell is at $k = 8$. The full Bayesian solution is computed with both a uniform prior over maps and a data-driven prior. The MCMC algorithm approximates the full solution with increasing fidelity as the patch size, L , increases. The traditional OG mapping algorithm does not capture mutual information between cells and can therefore be overconfident in estimating the first-occupied cell.

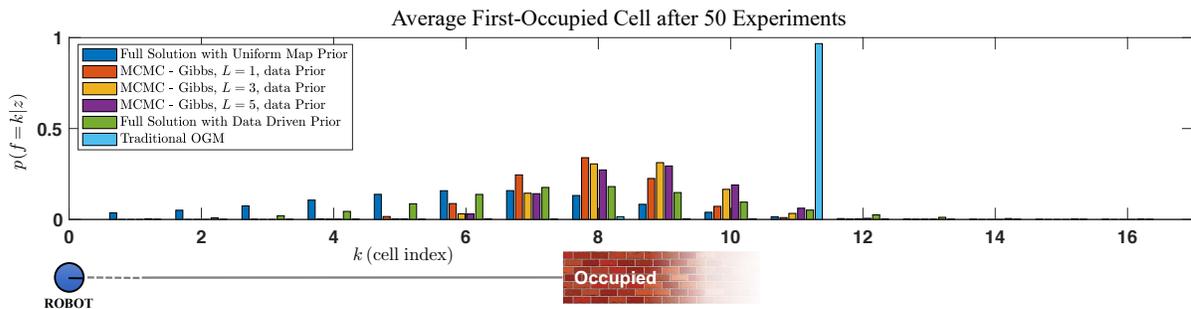


Figure 5.5: The average first-occupied cell, $p(f|z, x)$, for various OGM algorithms after 50 trials, where each trial had 2 to 8 measurements drawn from $p(z_n|f_{r,n}) = \mathcal{N}(8, 3^2)$. The robot is at $k = 0$ and the true first-occupied cell is at $k = 8$. The full Bayesian solution is computed with both a uniform prior over maps and a data-driven prior. The MCMC algorithm approximates the full solution with increasing fidelity as the patch size, L , increases. The traditional OG mapping algorithm does not capture mutual information between cells and can therefore be overconfident and inaccurate in estimating the first-occupied cell.

the same algorithms and measurements used in Figure 5.3. The MCMC algorithms plotted in Figure 5.4 will be discussed in Section 5.2. Figure 5.4 highlights that the full Bayesian solution with a uniform prior over all maps estimates the most likely first-occupied cell to be at $k = 7$, whereas with a data-driven prior it is correctly estimated at cell $k = 8$. The estimate using traditional OGM has the correct expectation using this set of measurements, but the estimate is overconfident.

To highlight this overconfidence, Figure 5.5 shows the average probability for $p(f|z, x)$, for each of the algorithms after 50 trials, where each trial had 2 to 8 measurements drawn from $p(z_n|f_{r,n}) = \mathcal{N}(8, 3^2)$. Similar to Figure 5.4, the expected value of the first-occupied

cell using the full solution with a uniform prior over all maps is at $k=7$ and with a data-driven prior it is at $k=8$. However, averaging over 50 trials illustrates that traditional OGM can not only be overconfident in its estimate, but also have an inaccurate expected value. In practice, this inaccuracy may lead to a poor localization estimate of the robot or obstacle if the OG map was used for this purpose. Furthermore, this plot highlights the issue observed by Marshall and Barfoot (2007), and discussed in Section 2.5, that traditional OGM can overestimate the distance to an obstacle.

5.2 MCMC with a Nonuniform Prior

The full solution presented in Section 2.3 assumes that every map (a combination of occupied and unoccupied cells) is equally likely. However, Section 5.1 illustrates a new benchmark for comparison – the full Bayesian solution with a more realistic or data-driven prior. This prior is based on the observation that occupied and unoccupied cells tend to be clustered together in real environments, and is thus simplified to a cluster of unoccupied cells, followed by a cluster of occupied cells for the 1D toy example. Furthermore, Figure 5.2 shows that a data-driven prior increases the mutual information between neighbouring cells but that the MI still approaches zero for cells that are farther apart. Therefore, capturing mutual information between proximate cells in the prior may accurately estimate the full Bayesian solution with a data-driven prior, presented in Section 5.1.

Although, the full solution remains computationally intractable for higher dimensions, we have presented several methods to approximate it in the preceding chapters. Moreover, some of these approximations utilize a prior over a patch of neighbouring cells. For example, Equation (3.3) in Section 3.2 shows that the offline patch map algorithm incorporates a patch prior, $p(a_{k,w})$. Even the online variant of the patch map, which is formulated iteratively in Equation (3.1) in Section 3.1, begins with an estimate of $p(a_{k,w})$ before incorporating any measurements. Finally, Equation (4.4) in Section 4.2, incorporates a patch prior, $p(a_k = w | m_{-a_k})$, for each MCMC sample drawn in Algorithm 1. Therefore, although a prior over all possible maps may be computationally intractable for higher dimensions, a prior over all possible patch configurations may not be. As highlighted in Chapter 4, the MCMC algorithm has several advantages over the patch map algorithm, namely the lack of patch boundaries and its ability to better estimate the full solution given more computation. Therefore, we will illustrate the benefit of a

nonuniform patch prior using the MCMC Gibbs sampling algorithm, but note that it would also benefit the patch map algorithms. Furthermore, since online patch map is a generalization of traditional OGM, this algorithm would also benefit from the data-driven prior presented in this chapter.

In Chapter 4, we used MCMC Gibbs sampling as a method to sample from the full solution in occupancy grid mapping. Specifically, Equation (4.4) extends the formulation to sample a patch of size L . In Section 4.2, we assumed a uniform prior over all patch configurations, but were able to incorporate any patch prior in the MCMC algorithm by sampling using the update equation,

$$p(a_{k,w}|z, x, m_{-a_k}) = \eta \underbrace{p(a_{k,w}|m_{-a_k})}_{\text{patch prior}} \prod_{n=1}^N \underbrace{p(z_n|x_n, m_{-a_k}, a_{k,w})}_{\text{sensor model}}, \quad (5.1)$$

where η is a normalizing constant. By using Equation (5.1) in Algorithm 1 we are able to sample the full posterior using a nonuniform patch prior. The following three sections will analyze various methods for computing the patch prior, $p(a_{k,w}|m_{-a_k})$, and highlight the benefits of each using the MCMC Gibbs sampling algorithm.

5.2.1 Higher-Order Markov Random Field

When occupancy grid mapping was first developed, Elfes (1989b) described the occupancy grid map representation as a Markov Random Field (MRF) of order 0, meaning that all cells in the map are independent of one another. However we can also consider a higher-order MRF to introduce correlations between cells, as seen in Figure 5.6. In a 2D grid, an order 1 MRF assumes that each cell is connected to its four nearest neighbours (known as 4-connected), and an order 2 MRF assumes that each cell is connected to its eight nearest neighbours (known as 8-connected). Muffert (2018) presented the order 1 MRF for OGM in 2D to introduce correlations between cells. In our 1D toy example, an order 1 MRF assumes that each cell is connected to its two nearest neighbours (known as 2-connected), and an order 2 MRF assumes that each cell is connected to its four nearest neighbours (known as 4-connected), etc. Therefore, a cell, m_k , is independent of all other cells in the map, given its neighbours, $p(m_k|m_{-k}) \approx p(m_k|\text{neighbours}_{m_k})$; this is referred to as a Markov blanket.

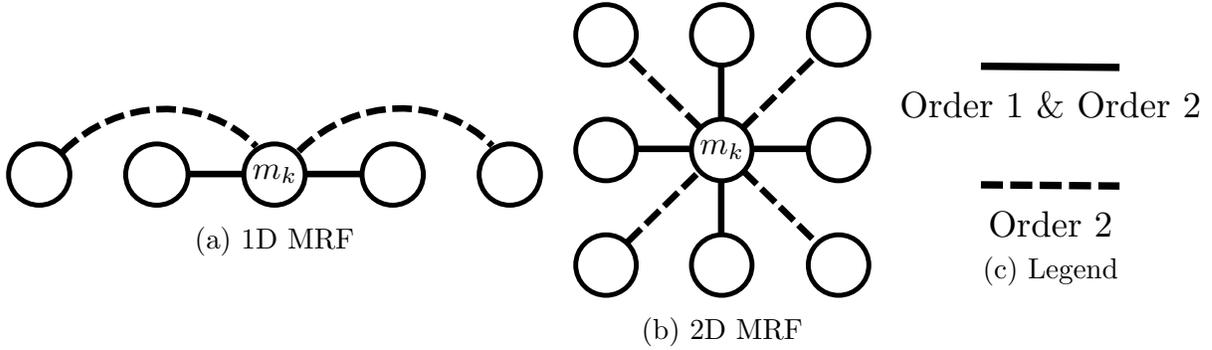


Figure 5.6: Cell connectivity for higher-order MRFs. An Order 0 MRF assumes cells are independent. An Order 1 MRF has cells connected to its two nearest neighbours in 1D and four nearest neighbours in 2D. An Order 2 MRF has cells connected to its four nearest neighbours in 1D and eight nearest neighbours in 2D.

In this section, we compute the prior for a single cell (patch size of $L = 1$) and we assume that the MRF is symmetric, meaning that the edge weights, ϵ , are the same for any two neighbouring cells independent of their location in the MRF (Koller and Friedman, 2009). Specifically, we state that the edge weight between two neighbouring cells, m_j and m_k , is

$$\epsilon_{j,k}(m_j, m_k) = \begin{cases} 1 & \text{if } m_j = m_k \\ \psi & \text{if } m_j \neq m_k \end{cases},$$

where $0 \leq \psi \leq 1$. Therefore the marginal cell occupancy probability can be computed by counting the number of neighbouring cells that are occupied, n_1 , or unoccupied, n_0 , then computing

$$p(m_k = 1 | \text{neighbours}_{m_k}) = \frac{\psi^{n_0}}{\psi^{n_0} + \psi^{n_1}}.$$

Or in the log-odds domain

$$\begin{aligned} l(m_k | \text{neighbours}_{m_k}) &= \log \left(\frac{\psi^{n_0}}{\psi^{n_1}} \right) \\ &= (n_0 - n_1) \log(\psi). \end{aligned}$$

Therefore the minimum sufficient statistic to compute $p(m_k | \text{neighbours}_{m_k})$ is $n_0 - n_1$. By using a higher-order MRF each cell in the OG is correlated to its neighbouring cells. The goal of this approach is to correlate neighbouring cells because realistic maps generally have occupied and unoccupied areas clustered together.

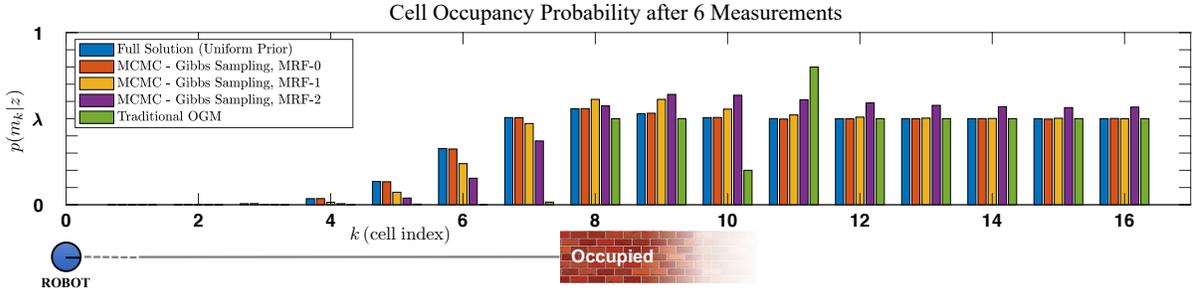


Figure 5.7: Occupancy grid after six measurements. The robot is at $k=0$, the true first occupied cell is at $k=8$ (therefore $f=8$). The full Bayesian solution is illustrated with a uniform prior. The MCMC algorithms are computed with $L=1$ and 10^5 map samples, with varying degrees of cell correlations in the prior. The higher order MRF algorithms have a smoothing effect, which may be useful in homogeneous regions but not where the map transitions from unoccupied to occupied.

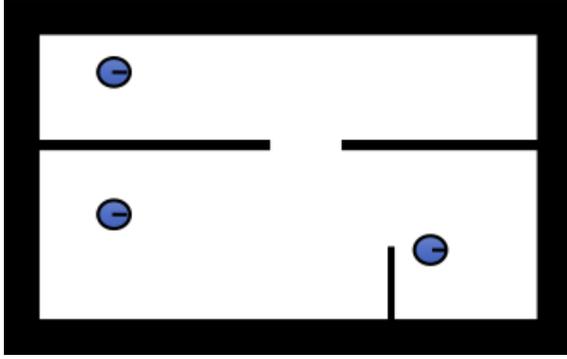
We can now use this updated prior in Equation (5.1). More specifically, in Equation (4.2), where we have formulated the MCMC update term for $L=1$ in the log odds domain. The resulting occupancy grid is seen in Figure 5.7 for increasing MRF orders. Note that an MRF of order 0 is equivalent to an independent patch prior and therefore models the full solution with a uniform map prior. However, increasing the MRF order, results in higher correlation between neighbouring cells and has a smoothing effect on the marginal cell occupancies. Thus, correlating the cell occupancy to its neighbours works well for the majority of a map where large homogeneous sections are unoccupied (e.g., free space) or occupied (e.g., a wall or other body). However, much of the OGM research is focused on the transition from unoccupied to occupied (e.g., the location of the wall or obstacle). Therefore the MRF approach is effective in “filling in the gaps” for homogeneous regions of the occupancy grid that may have few or even no measurements. However, the higher-order MRF approach performs worse than the data-driven approach from Section 5.1 at the transitions from unoccupied to occupied, where there is a discontinuity. Therefore, Section 5.2.2 will introduce an algorithm that incorporates a data-driven prior and can be used in dimensions higher than 1D.

The high-order MRF adds little computation cost to the MCMC algorithm presented in Section 4.1 and is therefore able to be applied to higher dimensions. Figure 5.8 illustrates the marginal cell probability on a 2D map for various MRF orders. Figure 5.8b illustrates the order 0 MRF and is therefore the same as Figure 4.6c. Figure 5.8c shows the resulting OG for a first-order MRF, where the smoothing effect of the MRF can be seen near the walls. This effect is more pronounced on the second-order MRF in Fig-

ure 5.8d. Notably, the cell correlations in the prior have a significant effect on unmapped or unknown cells. For example, the unknown cells “behind the wall” tend to be biased toward the occupied value of the walls and Figure 5.8c shows the clustering of occupied and unoccupied cells, whereas Figure 5.8d has stronger cell correlations resulting in all of these cells being occupied. Comparing the three MCMC results in Figure 5.8, a major benefit of neighbouring cell correlations in the prior is seen and that is the ability to more accurately estimate the occupancy probability of unknown cells between range measurements. The nature of narrow-beam range sensors (like laser range finders) is to map cells along a narrow ray, then the next measurement ray is rotated slightly, but from the same source location as the previous measurement. Coupling this effect with the Cartesian discretization of OG mapping yields the Moiré pattern observed by other researchers (e.g., Yguel et al., 2006) and seen in the unoccupied regions of Figure 5.8b. But introducing cell correlations in the prior enables the algorithm to minimize or eliminate this effect as seen in Figures 5.8c or 5.8d. However, Figure 5.8d also highlights the negative impact of a strong, and symmetric, cell correlation in the prior, and that is that a narrow wall surrounded by unoccupied cells can be “eaten away”, i.e., occupied cells are estimated as unoccupied.

5.2.2 Data-Driven Patch Prior

The 1D toy example from Section 5.1, showed the benefit of introducing a more realistic prior, as opposed to a uniform prior over all maps. For the 1D toy example, the prior probability of a patch configuration, $p(a_{k,w})$, can be computed by taking the sum of every instance of each patch configuration in the $K+1$ possible maps and dividing by the total number of patches in those maps. Since we have assumed that there is at most one obstacle per map and that all cells to the right of the obstacle are occupied, we have eliminated half of the possible patch configurations and thus the additional computation of computing $W/2$ patch configurations. In fact, we find that the two most likely patch configurations (fully occupied or fully unoccupied), account for the majority of the probability mass of the prior. Specifically, each has a probability of $(K-L+2)/(2K+2)$ and the remaining probability mass is distributed equally over the remaining, $2^{(L-1)}-2$, possible patch configurations. Therefore, as the number of cells in the map, K , increases the prior probabilities of the fully occupied and fully unoccupied patches approach 50% in our 1D toy example. For example, for a patch size of $L=1$, there



(a) Ground Truth.

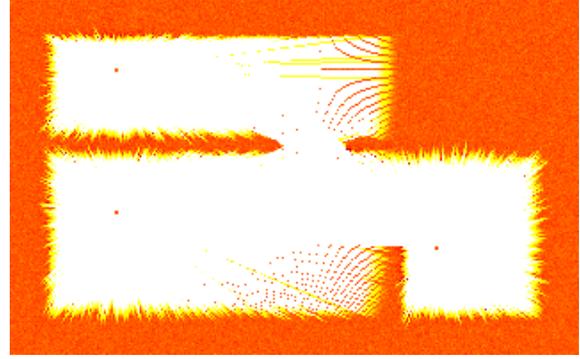
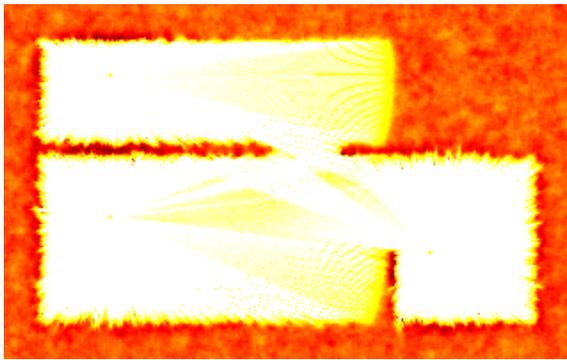
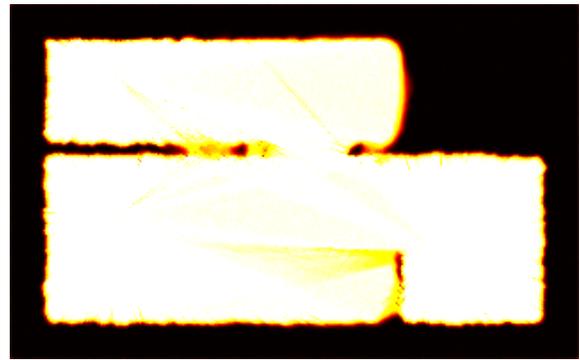
(b) MCMC with MRF-0; $L = 1$; 1000 iterations.(c) MCMC with MRF-1; $L = 1$; 1000 iterations.(d) MCMC with MRF-2; $L = 1$; 1000 iterations.

Figure 5.8: 2D example illustrating the resulting OG maps from the same measurements taken from three robot positions. The MCMC algorithm uses higher order MRF to correlate prior cell occupancy. These correlations in the prior allow the OG to effectively update the cell occupancy of unmapped cells that are proximate to mapped cells. The legend used is shown in Figure 1.10g.

are only two possible configurations and each has equal prior probability, $p(m_k) = 0.5$, for any sized map. In another example, for a map with $K = 3$ cells and a patch size of $L = 3$, there are $W = 2^L = 8$ possible patch configurations, of which half have zero likelihood. The example of $K = 3$ and $L = 3$ generates the eight patch configurations shown in Figure 5.1, and the four configurations in Figure 5.1a have equal probability, whereas the four configurations shown in Figure 5.1b have zero probability. However, for a slightly larger map with $K = 4$ cells and the same patch size of $L = 3$, the fully occupied and fully unoccupied patch configurations are more likely. Specifically, each has a probability of $p(a_{k,w}) = 0.3$ and the remaining two patch configurations shown in Figure 5.1a have a probability of $p(a_{k,w}) = 0.2$. The four patch configurations depicted in Figure 5.1b continue to have zero prior probability.

Algorithm 2: MCMC Gibbs sampling patches with a data-driven patch prior

Given $z, x, L, p_{\text{randPatch}}, m^{(0)}, \text{maxSamples}, \text{burnIn}$
Define m^r as $m^{(0)}$

for $i = 1$ to maxSamples **do**
 for $t = 1$ to K/L **do**
 Select random patch a_k from m^r
 if $\text{rand}(0,1) < p_{\text{randPatch}}$ **then**
 Set $a_k^{(i)}$ to a random patch configuration
 else
 Sample $a_k^{(i)}$ from $p(a_{k,w}|z, x, m_{-a_k}^r)$
 end if
 Define a_k^r , in m^r , as $a_k^{(i)}$
 end for
 Set $m^{(i)} = m^r$
end for

return set of samples, $m^{\text{burnIn}:\text{maxSamples}}$, from the full posterior

Therefore, we have formulated a data-driven patch prior for the 1D toy example, $p(a_{k,w})$, and see that half of these are zero. Because of this, we must modify Algorithm 1 to ensure that the sampling algorithm is ergodic. We could achieve this result by applying a nonzero probability to each of the prior patch configurations, but this would not yield the computational efficiency of not having to compute the probability of each of these patch configurations. Instead, we elect to meet the ergodicity requirement by having the sampling algorithm select a random patch configuration with probability $p_{\text{randPatch}}$. In this thesis, we set $p_{\text{randPatch}} = 0.001$, which we found strikes a balance between the time it takes to reach a stationary distribution while still exploring the solution space. Algorithm 2 presents an updated Gibbs sampling algorithm that is suitable for a data-driven patch prior where some patch priors may be zero.

Given Algorithm 2, we can now use the data-driven values of $p(a_{k,w})$ in Equation (5.1) to draw samples from the full posterior for varying patch sizes, L . Figure 5.3 illustrates the OG after six measurements, using this data-driven patch prior. The result is a better approximation of the full Bayesian solution with a data-driven prior, as compared to the higher order MRF estimate in Figure 5.7. Specifically, the data-driven prior is able to better estimate the marginal cell occupancy of the occupied cells. However, note that unknown cells that are far from the wall, return to the cell prior, $p(m_k) = 0.5$, because unlike the higher order MRF, patch priors are independent of neighbouring cells.

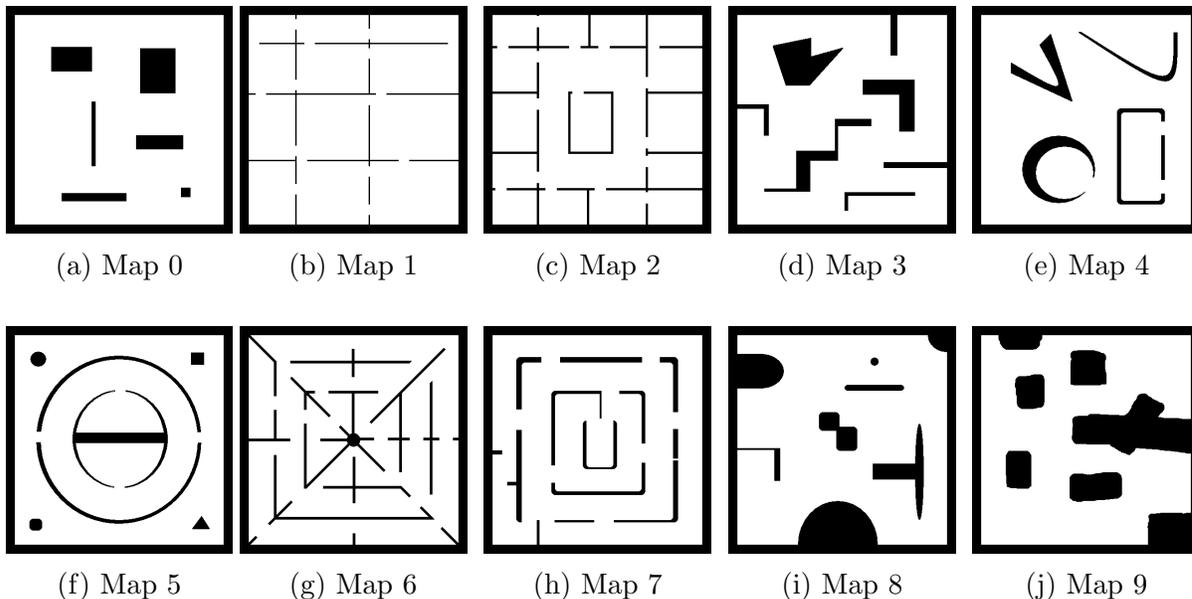


Figure 5.9: Ten 2D ground truth maps. Each map is designed to represent a different type of environment and each measures 500×500 cells with a 20 cell occupied border.

The data-driven patch prior, $p(a_{k,w})$, can be computed analytically for the 1D toy example because we know every possible map and its likelihood. However, the precise structure of realistic maps is not known. Therefore, to extend the concept to 2D, we conducted a data-driven analysis of local cell correlations. We did this by sampling 5×5 patches of cells centered at randomly selected cells from ten simulated maps with $K = 250\,000$ (500×500) cells. These are the same ten maps used in our previous research (Merali and Barfoot, 2012, 2013, 2014) and are illustrated in Figure 5.9. The maps have varying shapes and thicknesses of walls and distances between them to provide a diverse set of patch priors. Each patch sample was flipped and rotated to capture all eight permutations to remove orientation bias. From the 10^{10} patch samples drawn, we are able to subsample other cell configurations, such as 3×3 patches or single cells. From the samples, we determined that the probability of a cell being occupied is $p(m_k) = 0.234$, which is effectively the proportion of occupied cells in the ten sample maps. This value for the cell prior was used in the MCMC algorithm in Section 4.2 and yielded a more accurate result.

Figure 5.10 illustrates the frequency of the 3×3 and 5×5 patch configurations on a logarithmic scale to highlight how few patch configurations were ever sampled in the data. Specifically, of the $2^9 = 512$ possible patch configurations with a patch size of $L = 9$, only 124 patch configurations were ever sampled and 99.9% of those samples were of the

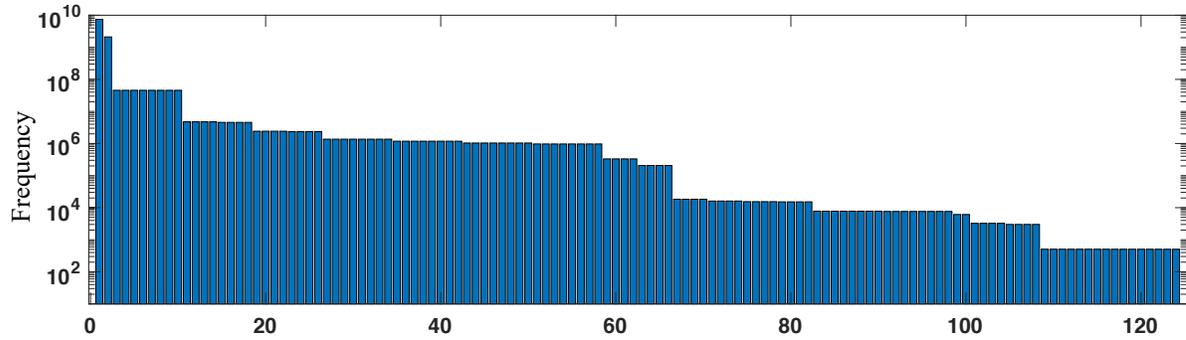
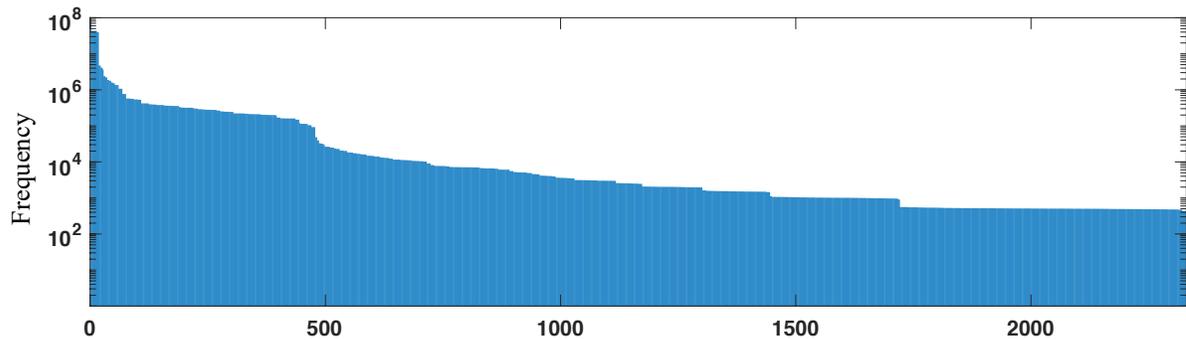
(a) Histogram of 3×3 patches sampled.(b) Histogram of 5×5 patches sampled.

Figure 5.10: Histogram of patch configurations from 10^{10} patches sampled from data for (a) 3×3 and (b) 5×5 patches. Many patch configurations do not appear in the data and therefore need not be computed when using a data-driven patch prior.

50 most likely patch configurations. Furthermore, 74.3% of 3×3 patches sampled were fully unoccupied and 21.1% were fully occupied and therefore together made up $> 95\%$ of samples. Similarly, for a patch size of $L = 25$, only 2329 possible patch configurations (of 2^{25}) were ever sampled, which is 0.0069% of the patch configurations. Furthermore, 99.9% of samples drawn are of the most likely 436 patch configurations. Although not visible in Figure 5.10b, the two most popular patch configurations were sampled more than the y-axis depicts. Namely, 71.98% of the sampled 5×5 patches are fully unoccupied and 18.92% are fully occupied; therefore these two patch configurations together are $> 90\%$ of samples. These samples are mainly in homogeneous areas that are fully occupied or fully unoccupied. Therefore these homogeneous areas could use a more coarse resolution, but the fine resolution is helpful to capture map details near the boundary between occupied and unoccupied cells. Much of the OG mapping research is concerned with accurately mapping this boundary as it effects the maps usefulness to perform tasks such as object recognition, localization and path planning.

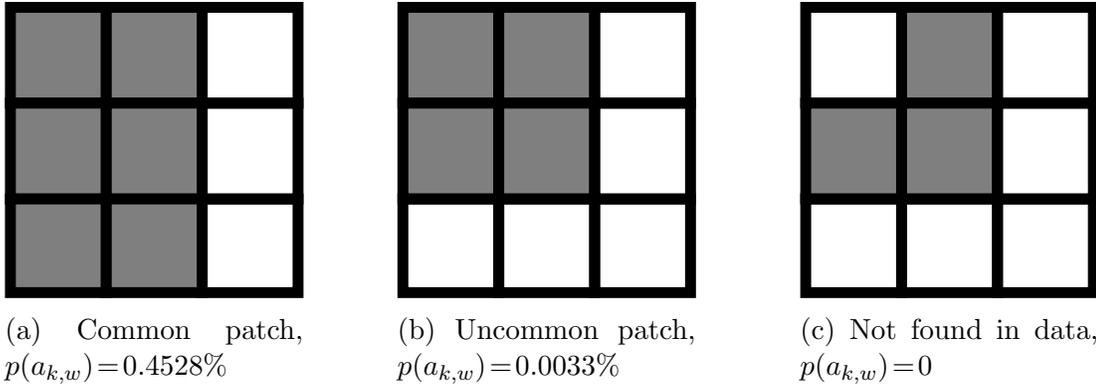


Figure 5.11: Three examples of 3×3 patches. It is computationally more efficient to only compute configurations common in the data, like (a), or less common configurations, like (b). Configurations like (c) never appear in the data and need not be computed.

In addition to the most common fully occupied and fully unoccupied patch configurations, Figure 5.11 shows other examples of patch configurations that are likely and unlikely. The insight that very few patch configurations occur in the data yields significant computational savings in addition to resulting in maps that more closely model real maps. In our patch map research (Merali and Barfoot, 2012), we found that computing patches larger than $L=9$ became computationally intractable. However, this new insight shows that computation of a patch size of $L=25$ would only require computing 2329 patch configurations as opposed to 2^{25} . Moreover, most of the benefit could be realized by computing only the most likely 436 patch configurations, which is less than the 2^9 configurations computed for $L=9$. This efficiency will be more pronounced as the patch size increases and if this work is extended from 2D to 3D.

Figure 5.12 illustrates the result of applying the MCMC Gibbs sampling algorithm with a 2D data-driven patch prior for various patch sizes. Note that Figure 5.12b was previously presented in Figure 4.6e to highlight that a data-driven prior overcomes some limitations of a uniform prior. Also, Figures 5.12c and 5.12d were computed by only considering the subset of patches that account for 99.9% of the samples for computational efficiency – all other patch configurations are assumed to have zero prior probability. Comparing the three MCMC results in Figure 5.12 highlights that the data-driven patch prior has the ability to better estimate the unknown cells between range measurements, as we saw with the higher-order MRF. We also observe that the cell occupancy probability returns to the cell prior for unknown cells that are far from the wall, as observed in 1D in Figure 5.3. In addition, the data-driven patch prior does not suffer from the “eating into

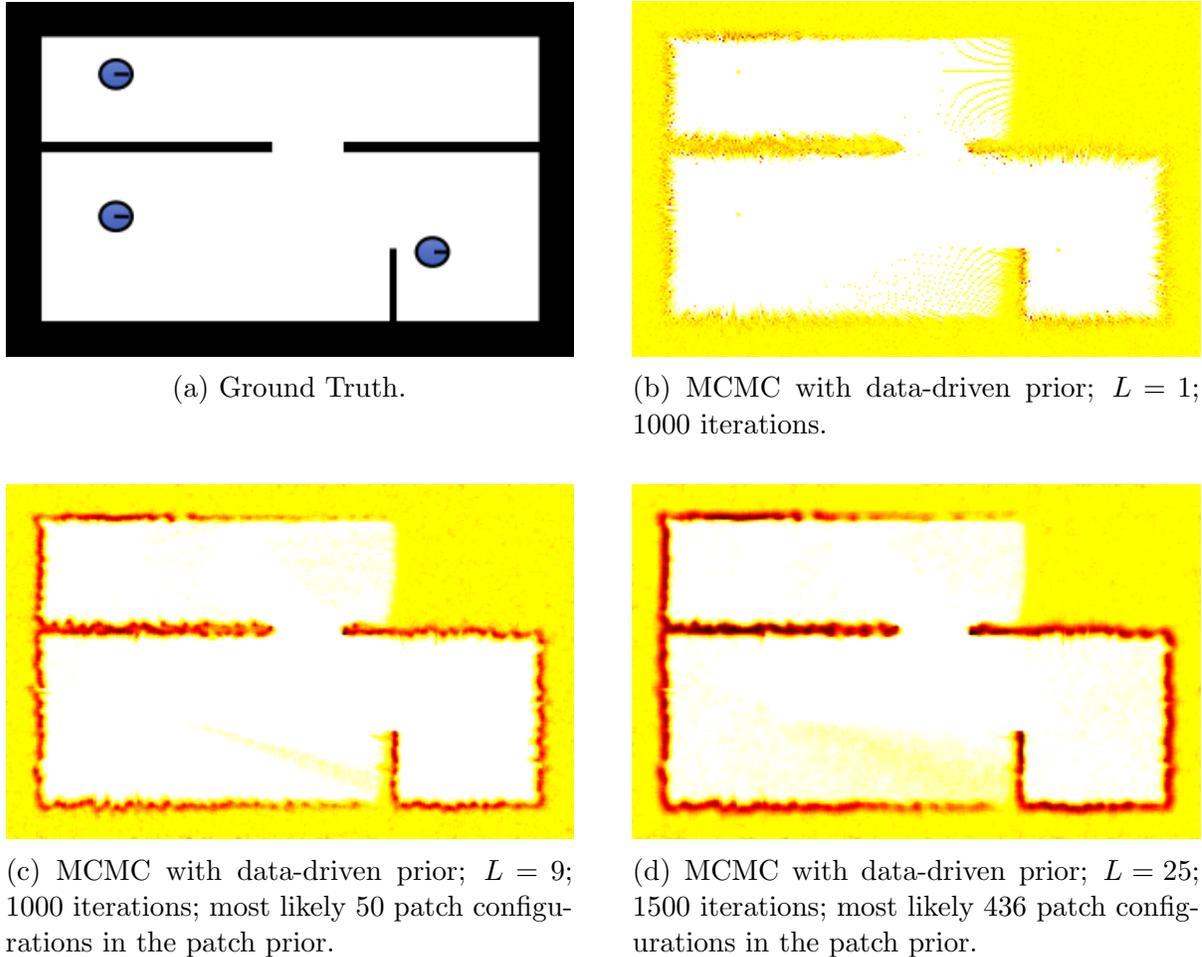


Figure 5.12: 2D example illustrating the resulting OG maps from the same measurements taken from three robot positions. The MCMC algorithm uses a data-driven prior to correlate prior cell occupancy. These correlations in the prior allow the algorithm to better estimate unmapped cells and draw patch samples that are likely in the prior. The legend used is shown in Figure 1.10g.

the wall” that we observed in the higher order MRFs in Section 5.2.1. Most notably, only considering a subset of likely patch configurations has a substantial computational saving (e.g., a 5×5 patch goes from computationally intractable to being computed faster than a 3×3 patch) and an accurate data-driven prior yields a more realistic map estimate.

5.2.3 Data-Driven Patch Prior with Border

In Section 5.2.1, we highlighted the benefit of correlating the prior patch probability to its neighbouring cells, specifically to those in the Markov blanket. Then in Section 5.2.2, we applied a data-driven prior but assumed that the patch prior is independent of other

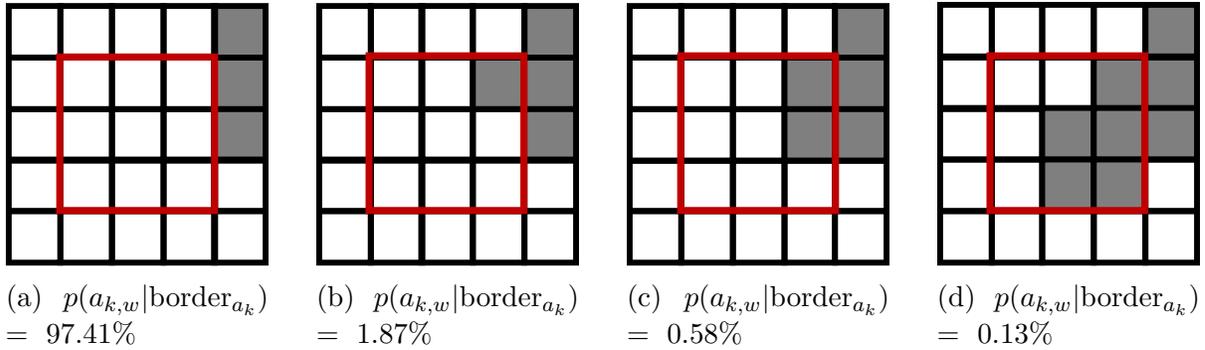


Figure 5.13: An example of four 3×3 patches and their likelihood given the same 16 border cells. Conditioning the patch prior on the border results in a significant computational savings because so few (often one) patch priors are nonzero. The four 3×3 patches depicted (in red) are the only four sampled from the data with the given border.

cells in the map. We will now explore combining these two concepts with a data-driven patch prior that is correlated to other cells in the map. The MCMC Gibbs sampling formulation in Equation (5.1) allows the patch prior to be correlated to all other cells in the map, $p(a_{k,w}|m_{-a_k})$. Correlating to every other cell in the map would be computationally intractable, but having the patch prior correlated to a Markov blanket around the patch yields much of the benefit as we have shown throughout this thesis that cell correlations are greatest between neighbouring cells. Therefore, in this section, we define a Markov blanket around each patch as those cells that are adjacent to the patch, which we refer to as the *border* cells. Using the same 5×5 patch samples from Section 5.2.2, we are able to compute the probability of a 3×3 patch configuration, given the configuration of 16 border cells of that patch. Figure 5.13 shows an example 16-cell border and four possible 3×3 patch configurations within that border. Similarly, we are able to compute the probability of a cell, $L=1$, given the configuration of its 8 border cells.

Figure 5.14 highlights additional statistics that can be computed from the 10^{10} samples of 5×5 patches presented in Section 5.2.2. Specifically, Figure 5.14a illustrates that only 865 unique 16-cell borders are ever sampled; much less than the 2^{16} possible borders. Two of the values shown are greater than the y-axis displayed. In particular, 71.98% of the borders are fully unoccupied and 18.92% of the borders are fully occupied. These percentages are the same as the fully unoccupied and fully occupied 5×5 patches observed in Section 5.2.2; therefore only these two patch configurations have these particular borders. The corollary to this is that, for both of these cases, if these 16 border cells are observed, then there is only one possible 3×3 patch observed in the data for the interior cells.

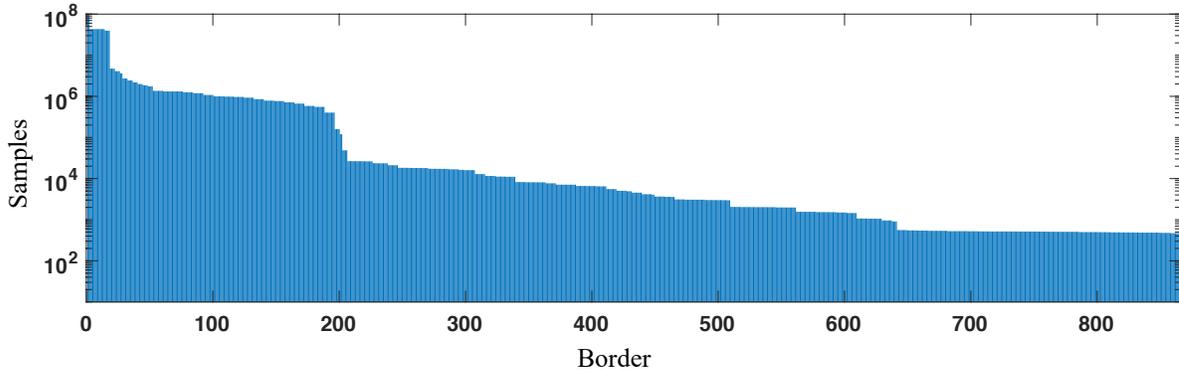
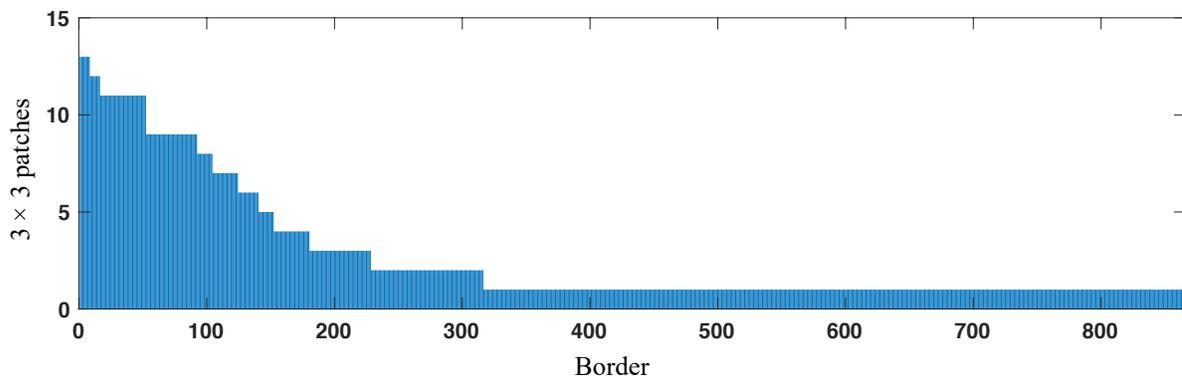
(a) Histogram of 5×5 patches sampled, sorted by 16-cell border.(b) Number of unique 3×3 patches, given 16-cell border.

Figure 5.14: Sub-sampling the 10^{10} patches sampled from data. The 5×5 patches can be grouped by their 16-cell border to highlight additional sparsity in 3×3 patch prior when the border is known.

Therefore, Figure 5.14b shows the number of unique 3×3 patches for a given 16-cell border. As observed with the fully unoccupied and fully occupied borders, the majority of borders have only one unique 3×3 patch configuration and the maximum is 13. This is significantly less than the 124 patch configurations observed in the data or even the 50 configurations that represent 99.9% of the samples. Therefore, conditioning the patch prior on the border in Equation (5.1), yields a significant computational efficiency as very few patch configurations have a nonzero likelihood given the border. In fact, because the patch prior is zero for all but one patch configuration in most cases, the measurements, $p(z_n | x_n, m_{-a_k}, a_{k,w})$, have no effect on those cases and the MCMC algorithm will sample the one patch configuration with a nonzero prior probability in Algorithm 2. This is yet another computational efficiency of conditioning the patch prior on the border. In fact, in our 2D experiments we observed that for $\sim 57\%$ of MCMC samples there was only one

possible 3×3 patch configuration in the prior, given the border. This empirical evidence correlates with the observation from Figure 5.14b that most 16-cell borders have only one corresponding 3×3 patch configuration.

To further highlight the additional sparsity achieved by conditioning the 3×3 patch on its border, Figure 5.15 shows that the average probability of the most likely 3×3 patch given its border is 86.5% and that the four most likely patches account for 99.7% of the probability mass. Figure 5.15 also illustrates the maximum and minimum prior probability of accounting for additional 3×3 patch configurations and we observe that after the first 7 configurations, the highest prior patch probability is 0.40%. Therefore, conditioning the 3×3 patch prior on its border cells, reduces the computation to at most 13 patch configurations, but most of the benefit can be achieved by only computing 7 or even 4 patch configurations.

An example 3×3 patch prior given its 16-cell border is shown in Figure 5.13. Given the border depicted, the fully unoccupied 3×3 patch has a prior probability of 97.41%. In contrast, this same patch represents 74.3% of patches when not conditioned on the border. More revealing is that the fully occupied patch has a 21.1% prior probability when independent of the border, but is zero when conditioned on the border shown in Figure 5.13. This observation is true for many common patch configurations. For example, the common 3×3 patch depicted in Figure 5.11a also has zero likelihood when conditioned on the border shown in Figure 5.13. Furthermore, note that the patches depicted in Figure 5.13 have their occupied cells neighbouring the occupied cells in the border, as we would expect from real maps. The data-driven prior from Section 5.2.2,

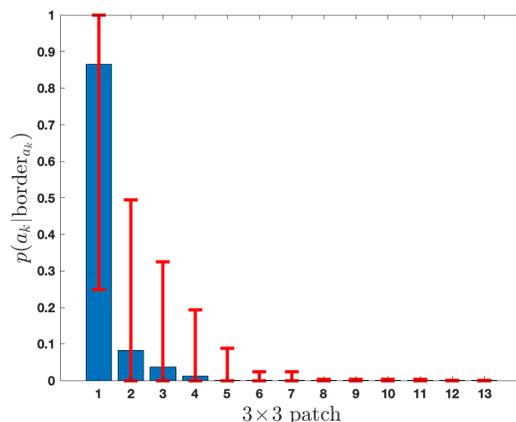


Figure 5.15: Average probability of a 3×3 patch, given its 16-cell border. This analysis of the 10^{10} patch samples highlights the sparsity in the prior when the border is known.

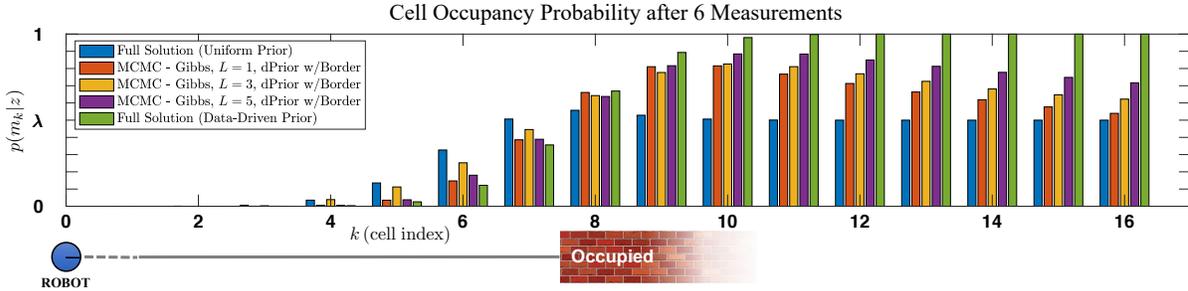


Figure 5.16: Occupancy grid after six measurements. The robot is at $k = 0$, the true first occupied cell is at $k = 8$ (therefore $f = 8$). The full Bayesian solution is illustrated with a uniform prior and a data-driven prior. The MCMC algorithms draw 10^5 map samples, using varying patch sizes. The data-driven patch prior is conditioned on bordering cells and is thus able to estimate the marginal cell occupancy better than a data-driven patch prior that assumes the patch is independent of its bordering cells.

that is independent of the border, assumes that symmetric variants of the same patch are equally likely, but Figure 5.13 shows that the symmetric variants of the patches depicted are zero (with the exception of Figure 5.13a because of its rotational and symmetric invariance). Thus, in addition to the computational efficiency, conditioning on the patch border enables a more accurate prior estimate of the patch.

The 10^{10} patches sampled from prior maps can also be used to determine the probability of a cell, $L = 1$, given its 8 bordering cell. This analysis shows that only 84 of the possible $2^8 = 256$ borders are observed in the data. Furthermore, in only 40 of these unique 8-cell borders are both occupied and unoccupied cells observed for $L = 1$. Therefore, the remaining 44 borders observed will be perfectly confident in the occupancy of the cell being sampled by the MCMC algorithm.

Approximating the prior in Equation (5.1) by conditioning the patch prior on bordering cells as opposed to all cells outside of the patch, $p(a_{k,w} | m_{-a_k}) \approx p(a_{k,w} | \text{border}_{a_k})$, is seen to be effective for the 1D toy example in Figure 5.16. In 1D, each patch has only two border cells – one on either side of the patch. The figure highlights that as the patch size, L , increases, the MCMC algorithm is able to better estimate the full Bayesian solution using data-driven prior. By comparing the marginal cell occupancy in Figure 5.16 of the MCMC algorithm with a patch size of $L = 1$, to an order 1 MRF in Figure 5.7, we observe that the data-driven patch prior that is conditioned on bordering cells better represents the transition from unoccupied to occupied. Specifically, the occupancy probability of cell $m_{k=7}$ is lower and cell $m_{k=8}$ is higher when using the data-driven approach.

Extending this algorithm to 2D using a border-based data-driven prior yields the results illustrated in Figure 5.17. Similar to the higher-order MRF prior, we see that unknown cells on the far side of the wall tend to be occupied as opposed to the cell prior. We also observe that the algorithm is able to estimate unknown cells between range measurements as unoccupied. In comparing the results in Figure 5.17 to the data-driven prior that was independent of the border in Figure 5.12, we see that incorporating the border yields straighter walls that resemble the ground truth map, given the same noisy measurements. Although Figure 5.17b shows some of the “eating-into-walls” issues that were highlighted previously with higher-order MRFs in Figure 5.8d, increasing the size of the patch prior alleviates this issue as seen in Figure 5.17c. Therefore, it appears that a binary (cell) prior can become overconfident, but a patch prior allows for more configurations to overcome this issue.

Table 5.1 highlights the computational benefit of only computing the probability of patch configurations that are likely given the bordering cells. These results were computed for the simple 2D example shown throughout this thesis thus far and depicted in Figure 5.17a. The results in the table were computed using a laptop computer running Matlab on Ubuntu with 32GB of RAM and a 2.70GHz Intel i7-6820HQ CPU. For a patch size of $L=1$, the uniform prior is computed fastest because a uniform prior removes the prior addition term, $l(m_k)$, in the log-odds domain as we saw in Equation (4.3). However, Equation (4.2) demonstrates that a data-driven prior introduces this additional term and therefore adds computation time. A border-based data-driven prior adds additional computational cost to look up a unique prior value based on the 8 border cells. Therefore, neither data-driven prior results in computational efficiency for a patch size of $L = 1$

Table 5.1: Average time to compute one map sample (K/L patches) for various patch sizes, L , and priors. The 2D map has $K = 64\,000$ cells. A data-driven prior is more efficient than a uniform prior because it only computes the probability of likely patch configurations. A border-based data-driven prior is more efficient still because the border reduces the number of likely patch configurations in the prior. Only 5×5 patches were sampled from prior maps and therefore cells bordering a patch size of $L = 25$ were not sampled.

	Uniform Prior [s]	Data-Driven Prior [s]	Data-Driven Prior with Border [s]
$L=1$	4.4	5.1	6.6
$L=9$	38.3	23.9	2.7
$L=25$	intractable	124.9	

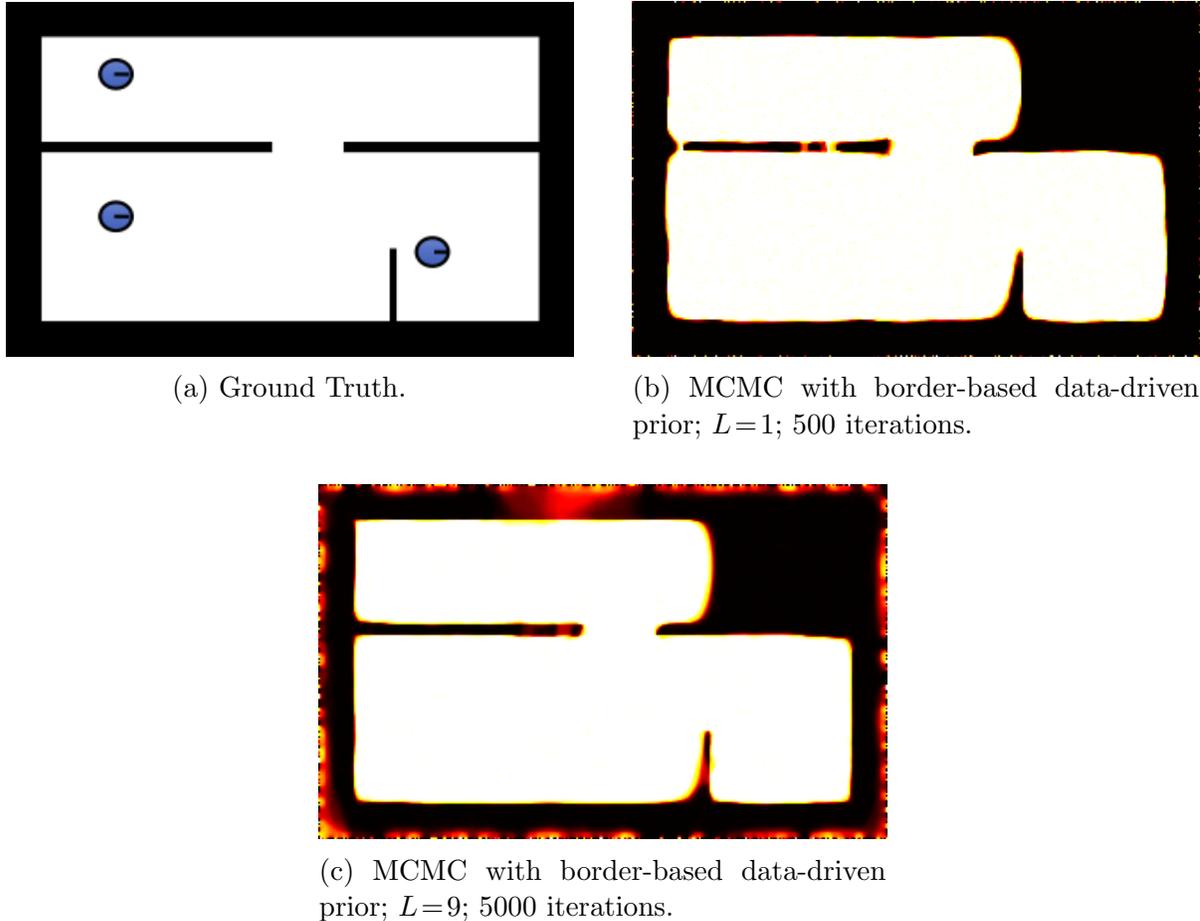


Figure 5.17: 2D example illustrating the resulting OG maps from the same measurements taken from three robot positions. The MCMC algorithm uses a border-based data-driven prior to correlate prior cell occupancy. These correlations in the prior allow the algorithm to better estimate the map and provide computational efficiency. The legend used is shown in Figure 1.10g.

because there are only two possible patch configurations and a data-driven prior does not reduce that. For a patch size of $L=9$, computing a map sample with all $2^9 = 512$ possible patch configurations requires an average of 38.3 seconds, but that time is reduced to 23.9 seconds by only considering the most likely patch configurations representing 99.9% of the 3×3 samples. This computation time is reduced to only 2.7 seconds by only computing patch configurations that are likely given the 16 border cells. We are able to compute a map sample with a border-based data-driven prior with $L=9$ faster than a uniform prior with $L=1$ because we compute K/L patches for each map sample. Therefore, the border-based data-driven prior can be both faster and more accurate for larger patch sizes. For a patch size of $L=25$ the computational savings is the most pronounced, however we

did not compute it for a border-based data-driven prior because we did not sample 7×7 patches to compute the border-based prior values. By only computing the 2329 patches that represent 99.9% of the 5×5 samples, each map sample requires only 124.9 seconds on average. However, computing a map sample with a uniform prior and a patch size of 5×5 is computationally intractable as there are 2^{25} possible patch configurations. We estimate that a single map sample would require approximately $(2^{25}/2329 \times 124.9/3600 =)$ 500 hours.

5.3 Chapter Summary

In this chapter, we analyzed the importance of a realistic prior in OGM. We showed that a realistic prior increases cell correlations between proximate cells, which underscores the importance of keeping local cell correlations. We then extended the MCMC algorithm developed in Chapter 4 to incorporate a more realistic prior using three methods.

First, we used a higher-order MRF representation of occupancy grids that assume neighbouring cell correlations are constant. This method better represents homogeneous regions of the map, but has a smoothing effect that can result in its inability to represent narrow walls or obstacles. Second, we presented a novel data-driven patch prior that enables realistic local cell correlations in the prior for the MCMC Gibbs sampling algorithm and allows the mapping algorithm to estimate unobserved cells that border observed cells. This data-driven approach to the prior yields improved results and also a computational efficiency as many patch configurations are not observed in the prior. Third, Section 5.2.3 extended the sampling method by conditioning the patch prior on the cells bordering the patch being sampled in the MCMC algorithm. This conditioning results in further computational efficiency, the ability to propagate measurements to all unobserved cells and a resulting map that is able to estimate the original map better than existing OGM techniques. Notably, the algorithm presented in Section 5.2.3 is able to draw map samples faster than the MCMC algorithm with a uniform prior, presented in Section 4.1. The algorithms will be evaluated on real data in Chapter 6.

Chapter 6

Experimental Results

This thesis has presented numerous OGM algorithms and used several existing techniques to compare them. A summary of these methods is outlined in Table 6.1, where the novel algorithms presented in this thesis are listed below the double horizontal line. All of these grid-based maps will depend on the size, K , and orientation of the initial grid chosen. The *ground truth* map is the term used to describe the true underlying map based on the grid selected; this is typically only available in simulation. All other techniques seek to estimate the maps based on sensor measurements and are therefore effected by the sensor model, $p(z_n|x_n, m^r) = p(z_n|f_{r,n})$. The full Bayesian solution estimates all 2^K possible maps without assuming cell or measurement independence. Given that this is computationally intractable for realistic maps, other OGM algorithms seek to estimate the OG with various simplifying assumptions. Throughout this thesis, we have illustrated the benefits of the various OGM techniques on a pose-constrained 1D toy problem so they could be compared to the full Bayesian solution, which can be computed for this 1D example. We have also shown anecdotal 2D results on a simple example to illustrate the effectiveness of the algorithms in a higher dimension and to highlight their ability to correlate cells that are not directly measured. In this chapter, we apply the algorithms to realistic 2D data to highlight their effectiveness on simulated datasets in Section 6.1 and real hardware datasets in Section 6.2. The simulated results allow the algorithms to be compared on experiments where the ground truth map is known, the robot pose is precisely known and the measurement model is also known. A more thorough analysis is conducted on the simulated results where the algorithms can be tested on various known environments. The algorithms are also shown to be effective on real hardware data, where the true map is not known and the robot pose and sensor model are estimated.

Table 6.1: Summary of grid maps discussed in this thesis. Algorithms below the double line are novel to this thesis.

Label	Reference	Application	Description	Key Variables
Ground Truth		Reference	True map, often only available in simulation.	
MAP	Section 2.2	Offline	Maximum a posteriori estimate; the most likely map based on data.	
Traditional OGM	Section 2.4	Online	Most commonly used OGM technique that was first presented by Moravec and Elfes (1985)	Δ
Full Bayesian Solution	Section 2.3	Intractable	Computing the probability of every possible OG map	
Pose-Constrained Full Bayesian Solution	Section 2.3.1	1D Reference	Computing the probability of every possible OG map	
p21	Section 2.5	Online	Optimized traditional OGM	Δ
Online Patch Map	Section 3.1	Online	Generalizing traditional OGM to estimate patches of cells	$L, p(a_k z_n, x_n)$
Offline Patch Map	Section 3.2	Offline	Batch algorithm that estimates each patch of cells using a forward sensor model and a reference map	L, \tilde{m}
MCMC w/ Uniform Prior	Section 4.1	Anytime	MCMC Gibbs sampling cells from the full posterior; marginals can be computed from many samples	$p(m_k)$
MCMC w/ Data-Driven Prior	Section 5.2	Anytime	MCMC Gibbs sampling patches, where the patch prior is based on data	$L, p(a_{k,w} m_{-a_k})$

The full Bayesian solution is too computationally expensive to compute for 2D (or 3D) maps. Therefore, we propose thresholding the 2D OG map generated and comparing to the binary ground truth map, which is available in simulation. Grewe et al. (2012) propose an evaluation metric of comparing an OG to a ground truth map by comparing

the percentage of free and occupied cells. However, Georgiou et al. (2017) propose the use of *precision* and *recall* (Davis and Goadrich, 2006) to compare the maps on a cell-by-cell basis, which offers further insights. In this chapter, we also use *precision* and *recall* to evaluate OG maps, but we weight the two equally with an F_1 metric, which will be defined in Section 6.1.4. In contrast, Georgiou et al. use an F_2 metric that favours recall over precision, as opposed to weighting them equally.

6.1 Simulation Results

To compare and contrast the various OGM algorithms in 2D, we collected 100 simulated 2D datasets in Player/Stage (Gerkey et al., 2003): 10 runs on each of 10 realistic maps. The 10 maps are shown in Figure 5.9 and range from cluttered office-type maps to sparsely occupied maps. The maps are 500×500 cells, and each range measurement can map a maximum of $F = 75$ cells with Gaussian noise on each measurement, $p(z_n|f) = \mathcal{N}(f, 3^2)$. A screenshot from the Stage robotics simulator is shown in Figure 6.1 to highlight the relative size of the range measurements on one of the maps. Both the overall size of the map and the maximum range of the sensor are held constant across the 100 simulations. A simple exploration algorithm is used to ensure coverage for each of the datasets. The robot only maps the environment when stopped, as is common with exploration algorithms that seek to produce highly accurate maps (Tong et al., 2012).

Section 6.1.1 highlights the results of the offline patch map algorithm that were first presented in Merali and Barfoot (2012) and described in Chapter 3. The offline patch map estimates the full solution in 2D quite well and is therefore used as a benchmark in Section 6.1.2 to compare the optimized traditional OGM algorithms that were presented in Section 2.5. These results were also published in Merali and Barfoot (2014) to highlight the benefit of optimizing the update function in traditional OGM. The MCMC algorithm presented in Chapter 4 is used on simulated 2D data in Section 6.1.3. Specifically, the results are shown for sampling one cell at a time (as opposed to a patch) as we recognize that the two produce the same result for a uniform prior. These experimental results were first published in Merali and Barfoot (2013). Finally, Section 6.1.4 shows the benefit of incorporating a patch prior on the 2D datasets. These results are based on the algorithms presented in Section 5.2 and are being prepared for publication.

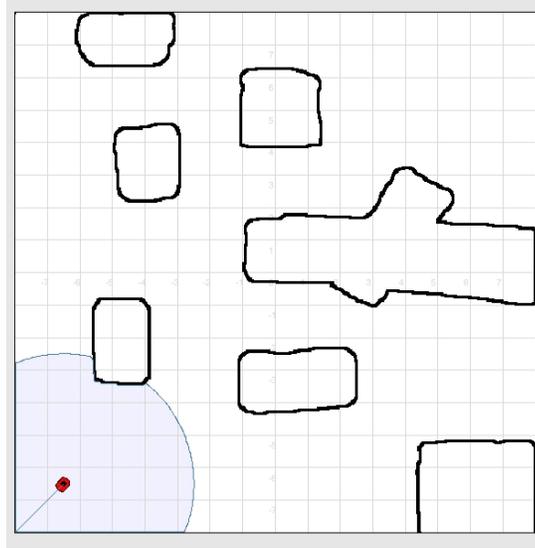


Figure 6.1: A screenshot from the Stage robotics simulator used to collect the 100 simulated datasets. The robot (in red) is shown at the first scan location and uses a stop-and-scan approach and measures 360-degrees around the robot at each scan location. A border of 20 occupied cells was later applied to each dataset; the map depicted here is one of ten shown in Figure 5.9.

6.1.1 Patch Map Results

In Chapter 3, we introduced various patch map algorithms and explained that an offline patch map algorithm with a large patch size, $L = 9$, and the MAP estimate for cells outside of the patch, is a great benchmark for comparing the marginal cell occupancy of each cell in the map. In fact, with a uniform patch prior, a patch size larger than $L = 9$ becomes computationally intractable. Therefore, each of the 100 simulated 2D datasets was used to compute a 3×3 patch centered at each of the K cells using the offline patch map method described in Section 3.2, with both the MAP solution and GT used for cells outside of the patch being estimated.

Figure 6.2 illustrates a typical result from one of the 100 datasets and compares the offline patch map result to traditional OGM. In addition, the figure shows three magnifications per map to highlight some anecdotal results. The first magnification illustrates a well mapped boundary where the patch map begins to approach the ground truth map. The patch map is more confident in the cell boundary than the traditional OG map, which could translate to better localization in a grid-based localization problem where the map is given. The second magnification illustrates a poorly mapped boundary, where the patch map remains at the prior as opposed to the overconfident ground truth map.

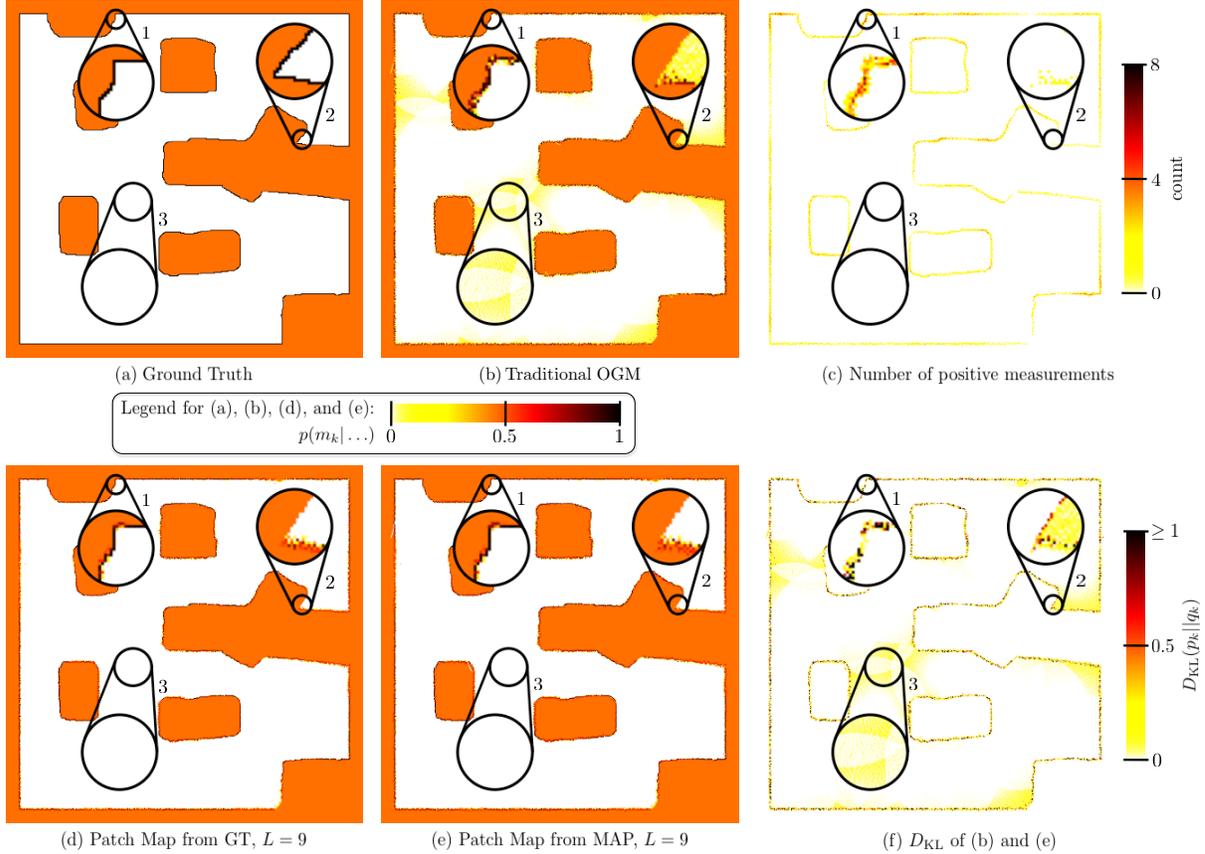


Figure 6.2: Results from one of the 100 2D datasets. Image (a) illustrates the ground truth map, where $p(m_k) \in \{0, \lambda = 0.5, 1\}$. Image (b) shows the result of traditional OGM. Image (c) shows the number of times the range sensor reflected off a cell, m_k , in the dataset. Image (d) illustrates the patch map using ground truth, $p(m_k|z, x, \tilde{m}_{-a})$. Image (e) shows the patch map using the MAP estimate. Image (f) displays the D_{KL} between (b) and (e), where the values have been thresholded at $D_{\text{KL}}(p_k||q_k) = 1$ for the purpose of this illustration. The circles in the six images are magnifications. The first magnification shows that the patch map approaches the ground truth map in well-mapped areas. The second magnification shows that patch map remains at the prior for unmapped cells, unlike the ground truth map. The third magnification shows that the traditional OGM algorithm is not confident about the unoccupied space, but the patch map is. Each cell in the third magnification received few (between one and three) measurements.

Finally, the third magnification highlights that traditional OGM is not as confident as the patch map method in poorly mapped unoccupied areas. Depending on the occupancy threshold, traditional OGM could consider this area occupied, thus hindering a path planning (or even localization) algorithm. Furthermore, we observe that the patch map generated using the MAP estimate in Figure 6.2(e) is nearly indistinguishable from the patch map generated using the GT map in Figure 6.2(d). In fact, 97.4% of the cells

have exactly the same probability, $D_{\text{KL}} = 0$, and an additional 1.7% have a divergence of less than 0.001, $D_{\text{KL}} < 0.001$. Thus, we conclude that in the absence of the ground truth map, the MAP estimate is suitable for achieving nearly the same patch map. Note that Figure 6.2(c) only counts the number of times a cell reflected a range measurement (positive information), not the number of times a measurement passed through the cell without being reflected (negative information). The patch map and traditional OGM algorithms use the negative information to decrease the probability that cells are occupied. Finally, Figure 6.2(f) illustrates the D_{KL} between each cell in the map generated by traditional OGM (shown in Figure 6.2(b)) and the offline patch map using the MAP solution (shown in Figure 6.2(e)). This figure highlights that traditional OGM diverges most near the boundaries between unoccupied and occupied areas of the map and the three magnifications highlight that traditional OGM fails to capture the uncertainty in the map in both sparsely mapped and heavily mapped areas.

Since the full solution could not be computed for the simulated 2D maps with $K = 250\,000$ cells, the benchmark used in 2D is the offline patch map computed using the MAP estimate for \tilde{m}_{-a} and overlapping patches of size $L = 9$. For comparison, the patch map is computed for $L \in \{1, 5, 9\}$ in the configurations shown in Figure 3.1. The occupancy grids were also computed using the traditional OGM algorithm described in Section 2.4. Table 6.2 summarizes the results by comparing the D_{KL} between the maps produced by various OG mapping algorithms and the benchmark map. Table 6.2 highlights that the MAP estimate and the ground truth map diverge a great deal from the benchmark. In addition, the table shows that using the MAP estimate or ground truth (GT) for \tilde{m}_{-a} makes little difference. Finally, the table shows that a patch size of $L = 9$ or even $L = 5$ realizes the majority of the benefit. Therefore, with a uniform patch prior, there is merit to computing the patch map for a patch size greater than $L = 1$, but most of the benefit is realized with only slightly larger patches. This conclusion aligns with the mutual information study in Section 2.6 that mutual information is greatest between neighbouring cells and approaches zero as cells are farther apart.

6.1.2 Optimizing Traditional OGM

Section 6.1.1 showed that the offline patch map algorithm is able to accurately capture the residual uncertainty in 2D occupancy grids. Furthermore, Section 2.5 illustrated that traditional OGM can be optimized to capture the residual uncertainty in the map,

Table 6.2: The average D_{KL} (from 100 2D datasets) between maps produced by various OG mapping algorithms and an offline patch map using the MAP estimate, a uniform patch prior, and $L=9$

	D_{KL}
MAP Estimate	6.065×10^5
Ground Truth	8.259×10^4
Traditional OGM	7.476×10^3
Patch map from GT, $L=1$	7.518×10^3
Patch map from MAP, $L=1$	8.814×10^3
Patch map from GT, $L=5$	4.715×10^3
Patch map from MAP, $L=5$	3.542×10^3
Patch map from GT, $L=9$	1.321×10^3

without sacrificing the fast computation time that allows traditional OGM to run online. In this section, we show that the optimization performed in 1D is effective in 2D and use the offline patch map as a benchmark because the full Bayesian solution cannot be tractably computed for these large 2D maps.

Given the optimized update terms for traditional OGM shown in Section 2.5, the new update terms can be used online. Therefore, the online computation time is the same as traditional OGM methods. For comparison, we use the update term presented by Hähnel (2004) and used throughout this thesis as the traditional update term. For each of the simulated datasets, we are able to compute the OG using traditional methods and the optimized update terms presented in Table 2.1. As a benchmark, we use the offline patch map with overlapping 3×3 patches centered at each cell and compute the D_{KL} for each cell as compared to this benchmark. Table 6.3 shows the average D_{KL} of the 100 experiments for each set of parameters compared the to the benchmark map. For reference, a map that exactly matches the benchmark would have $D_{\text{KL}}=0$, and an entirely unknown map where each cell has a value of $p(m_k)=0.5$ would have $D_{\text{KL}}=1.25 \times 10^5$, given the benchmark maps. Note that each simulated map has $K=2.5 \times 10^5$ cells and therefore dividing the mean D_{KL} values in Table 6.3 by this value of K yields the average D_{KL} per cell. Therefore the p21 algorithm has an average $D_{\text{KL}}=0.0158$ per cell, whereas traditional OGM has an average $D_{\text{KL}}=0.0584$ per cell. As expected, we see that the optimized parameters of Moravec (1988) and Marshall and Barfoot (2007) outperform the unoptimized parameters of traditional OGM. Using the unoptimized parameters, the average D_{KL} between the maps produced and the benchmark map is 1.46×10^4 . Also, the update term presented by Borenstein and Koren (1991) is the only one to perform

Table 6.3: Average D_{KL} of 100 experiments between online OG map and offline patch map in 2D

	mean(D_{KL})	min(D_{KL})	max(D_{KL})
Traditional OGM	1.46×10^4	8.91×10^3	2.37×10^4
Borenstein and Koren (1991)	1.25×10^5	1.07×10^5	1.37×10^5
Moravec (1988)	9.54×10^3	6.29×10^3	1.39×10^4
Marshall and Barfoot (2007)	1.11×10^4	7.22×10^3	1.67×10^4
Stachniss (2006) (sonar)	7.57×10^3	4.78×10^3	1.19×10^4
Elfes (1989a)	4.14×10^3	2.58×10^3	6.53×10^3
p21	3.95×10^3	2.62×10^3	5.79×10^3

worse than traditional OGM and similar to an entirely unknown map. This is expected as this update term only adds information to the map where the sensor was reflected and does not decrease the occupancy probability of cells that were measured as unoccupied. Furthermore, Table 6.3 highlights that the parameterization from Section 2.5 with 21 parameters has the lowest average D_{KL} compared to the benchmark map with a value of 3.95×10^3 . Therefore, the new update term better approximates the benchmark map.

Figure 6.3 illustrates the improved performance of the OG mapping algorithm by using the optimized 21-parameter Δ function as opposed to those found in the literature. The figure compares the performance on one of the 2D datasets. Figure 6.3(a) shows the number of positive measurements (i.e., the cell reflected range sensor) in each cell. The three black circles on each image are magnifications. The first magnification shows a heavily-mapped area and the second shows an under-mapped area. Figure 6.3(b) shows the benchmark offline patch map. Figure 6.3(c) shows the OG using a Δ function from the literature (Hähnel, 2004) and Figure 6.3(d) shows the OG using the optimized p21 Δ function. Figure 6.3(e) and (f) show the D_{KL} on a cell-by-cell basis between (c) and (d), respectively, compared to the benchmark map. The many dark cells in Figure 6.3(e) indicate that the OG created using a Δ function from the literature strongly diverges from the benchmark map, especially in heavily-mapped areas. Note that the largest divergence is concentrated in areas where the range sensor returned a reading, as depicted in Figure 6.3(a). The third magnification highlights an unoccupied area, where the optimized Δ function better captures the residual uncertainty. Each of the cells in the third magnification had one to three measurements. It is not surprising that

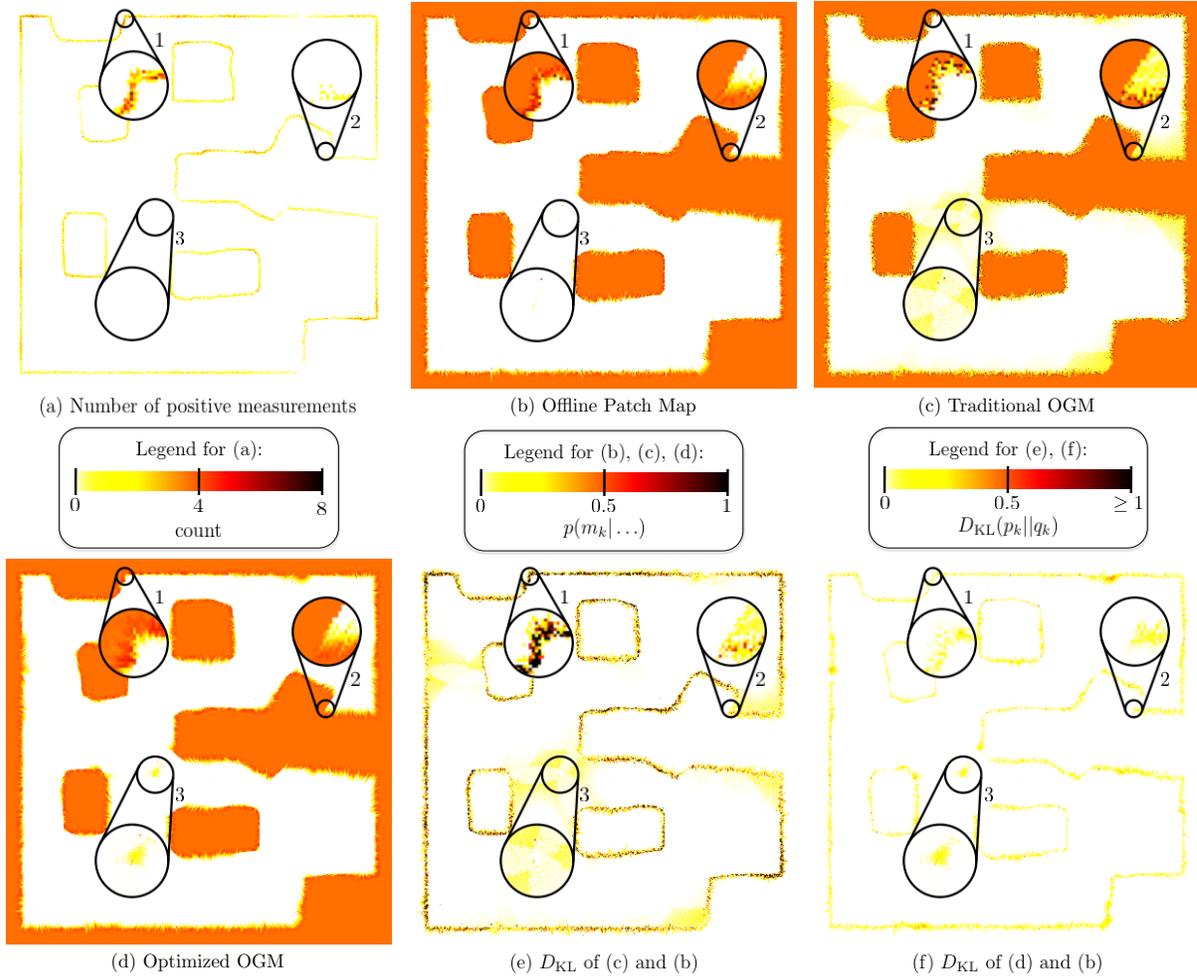


Figure 6.3: Results from one of the 2D datasets. Image (a) shows the number of times the range sensor reflected off a cell, m_k , in the dataset. Image (b) illustrates the benchmark map created using an offline patch map algorithm. Image (c) is the OG created using traditional OGM (Hähnel, 2004). Image (d) is the OG created using the optimized 21-parameter Δ function. Image (e) displays the D_{KL} between (c) and (b) and image (f) displays the D_{KL} between (d) and (b). In both images (e) and (f) the values have been thresholded at $D_{\text{KL}}(p_k || q_k) = 1$ for the purpose of this illustration. The circles in the six images are magnifications. The first magnification shows that traditional OGM is overconfident in well-mapped areas. The second magnification shows that the optimized parameters outperform traditional OGM, even on walls with few measurements. The third magnification shows that the optimized parameters also estimate the unoccupied cells more accurately. Each cell in the third magnification received few measurements.

the optimized p21 Δ function outperforms the others in unoccupied areas because, as Section 2.5 highlighted, this parameterization reduces the occupancy probability of cells in the unoccupied areas more than other Δ functions. Therefore, we have shown that the optimized Δ function can be used online in a traditional OGM algorithm to better capture the residual uncertainty in the map without sacrificing computation time.

6.1.3 MCMC with a Uniform Prior

The offline patch map algorithm is an effective benchmark for comparing marginal cell occupancy probabilities in OGM. However, in Chapter 4, we presented a MCMC Gibbs sampling algorithm that is capable of computing the cell marginals in addition to other statistics about the map as it draws samples from the full posterior. Furthermore, the MCMC algorithm is an *anytime* algorithm as it can provide an estimate quickly, but the estimate continues to improve as more samples are drawn.

To demonstrate that the MCMC algorithm can be computed for large maps where the full solution is computationally intractable, we use the MCMC algorithm with a uniform prior and $L=1$ to sample from the full posterior for all 100 simulated datasets. These experiments highlight the importance of selecting a suitable prior probability for each cell in the occupancy grid. When a uniform cell prior of $p(m_k) = 0.5$ is used, the MCMC algorithm required in excess of 200 iterations to reach a stationary distribution. By lowering the prior to a more realistic value of $p(m_k) = 0.15$, the algorithm converges in less than 20 iterations, as illustrated in Figure 6.4. The MCMC experiments also show that by starting the algorithm with a thresholded OG map developed using traditional OGM, a stationary distribution is reached on the second iteration.

The experiments show that the MCMC Gibbs sampling method can be used for large maps with $K = 2.5 \times 10^5$ cells. Each iteration of the MCMC algorithm takes approximately 10.3 seconds in our unoptimized Matlab implementation on a laptop computer with 32GB

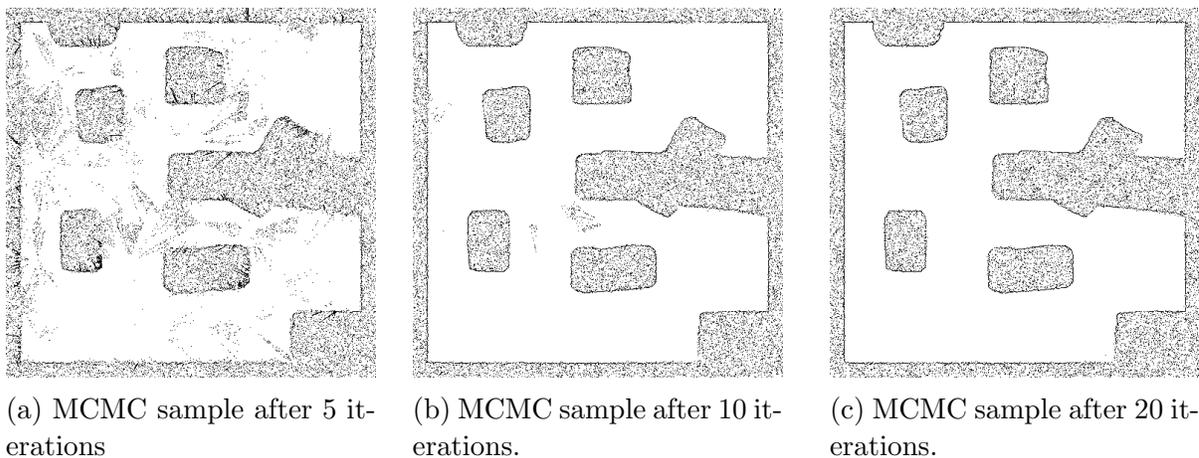


Figure 6.4: MCMC map samples after 5, 10, and 20 iterations, when starting from a cell prior of $p(m_k) = 0.15$. MCMC algorithms typically disregard early samples as the algorithm has not yet reached a stationary distribution. In this case, a stationary distribution is reached in less than 20 samples.

of RAM and a 2.70GHz Intel i7-6820HQ CPU. One key benefit of the MCMC algorithm is that the OG estimate becomes more accurate as more samples are drawn. The average D_{KL} between the MCMC method and the ground truth map for the 100 experiments is 8.8×10^5 . The average D_{KL} between the MCMC method and traditional OGM for the 100 datasets is 6.3×10^3 . This result indicates that traditional OGM more closely models the MCMC method. Since the ground truth map does not model uncertainty and does not depend on the measurements, this result is intuitively correct.

Figure 6.5 shows typical results from one of the 100 datasets. The three magnifications in the figure highlight some anecdotal results. The first magnification shows an obstacle boundary that received several measurements. In this case, the MCMC method approaches the ground truth map. The MCMC method is more confident about the obstacle boundary than traditional OGM, which should translate to better localization in a grid-based localization problem where this map is used. Conversely, the second magnification highlights an obstacle boundary that received few or no measurements. The MCMC algorithm remains at the prior for cells with no measurements, unlike the ground truth map that does not depend on the measurements. Hence, if the ground truth map were used as a benchmark, it would not be realistic to expect an OGM algorithm to estimate cells with no measurements. Lastly, the third magnification shows an unoccupied area that received few (between one and three) measurements. The traditional OGM algorithm does not reduce the occupancy probability of these cells to near-zero as the MCMC algorithm does. Therefore, depending on the user-defined occupancy threshold, the traditional OGM algorithm may consider this area occupied, which would hinder path planning or localization algorithms that the map may be used for. The differences between the algorithms are further illustrated by the D_{KL} plots in Figure 6.5 (e) and (f). With a constant cell prior, the cells are only correlated by the measurements and not the prior. Therefore, cells that are not mapped remain at the cell prior.

6.1.4 MCMC with a Data-Driven Patch Prior

The MCMC algorithm is able to draw samples from the full posterior. However, in Chapter 5 we showed that the algorithm can be improved by incorporating a data-driven patch prior and that the prior is more effective when it is conditioned on the cells bordering the patch. Therefore, in this section we analyze the 100 simulated datasets using the MCMC algorithm with a data-driven patch prior as described in Section 5.2.

As shown in Chapter 4 and 5, a data-driven cell prior allows the MCMC algorithm to better estimate the occupancy of cells, especially those with few or even no measurements. Therefore, to show the effectiveness of a data-driven patch prior, we compare the maps produced to the ground truth map that was used to create the dataset.

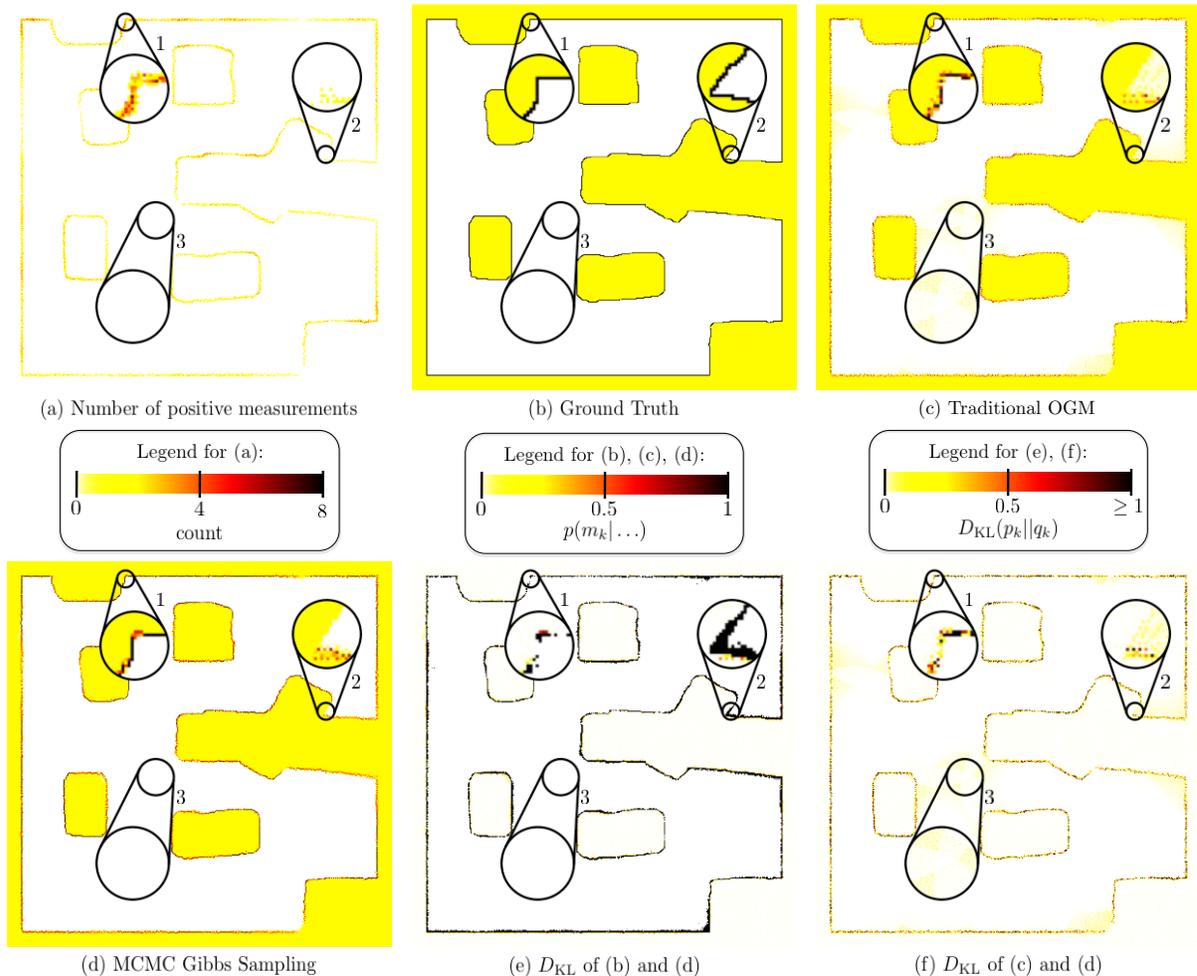


Figure 6.5: Results from one of the 2D datasets. Image (a) shows the number of times the range sensor reflected off a cell, m_k , in the dataset. Image (b) illustrates the ground truth map, where $p(m_k) \in \{0, \lambda = 0.15, 1\}$. Image (c) shows the result of traditional OGM. Image (d) illustrates the result of the MCMC Gibbs sampling algorithm. Image (e) displays the D_{KL} between (b) and (d), where the values have been thresholded at $D_{\text{KL}}(p_k||q_k) = 1$ for this illustration. Similarly, image (f) shows the D_{KL} between (c) and (d). The circles in the six images are magnifications. The first magnification shows that the MCMC method approaches the ground truth map in well-mapped areas. The second magnification shows that unmapped cells in (d) remain at the prior, unlike the ground truth map. The third magnification shows that traditional OGM is not confident about the unoccupied space, but the MCMC algorithm is; each cell in the third magnification received few (between one and three) measurements.

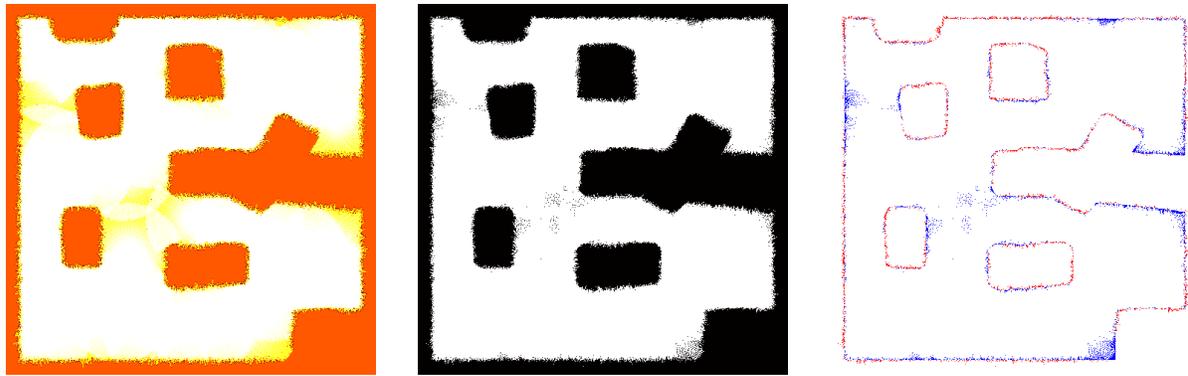
Each ground truth map is binary (each cell is occupied or unoccupied), as shown in Figure 5.9. However, this thesis and much of the OGM literature has considered the *unknown* cells inside objects in the environment to be at the cell prior, $p(m_k)$, because these cells cannot be mapped by a range sensor and if cells are considered independent in the prior, then the OGM algorithms cannot estimate these cells beyond the cell prior. However, by correlating cells with a patch prior, we are better able to estimate the occupancy of these cells, especially when estimating the patch prior given the cells that border it. Therefore, in this section, we compare the results of the MCMC OGM algorithms to the binary ground truth map. We use the metrics of *precision* and *recall* to evaluate the maps against the ground truth map, as is common for evaluating binary classification algorithms. Occupancy grids are often thresholded at some marginal cell probability in order to produce a binary map that can then be used for localization, path-planning and other purposes (e.g., Nagla et al., 2012). We therefore illustrate the effectiveness of the algorithms on a Precision-Recall plot that shows the effectiveness of the algorithm on these two metrics for various marginal cell probability threshold values. To generate these metrics, each cell in the OG map is compared to the ground truth map and those that match are marked as true positives (TP). However, cells that are estimated to be occupied, but are unoccupied in the ground truth are marked as false positive (FP) and cells that are estimated to be unoccupied, but are occupied in the ground truth are marked as false negative (FN). The two metrics are therefore defined as (Davis and Goadrich, 2006),

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Therefore, if the OGM algorithm estimates all cells as occupied, then no false negatives will be detected and the algorithm will have perfect recall but poor precision. Consequently, it is important to have a balance of these two factors and this is often measured by a weighted sum of the two known as a F_1 score,

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (6.1)$$



(a) Traditional OGM

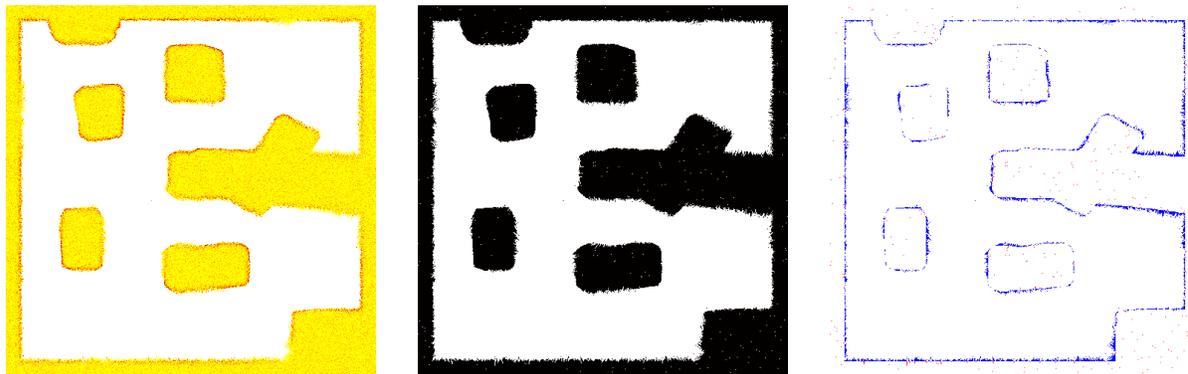
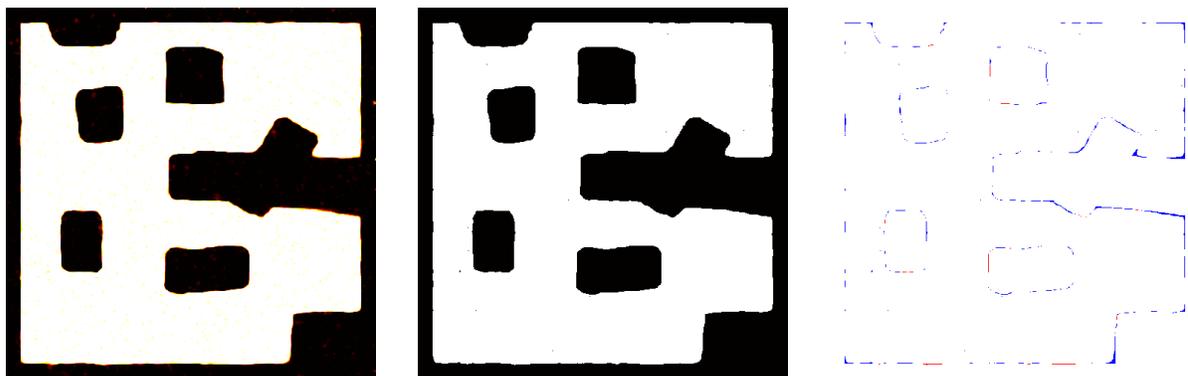
(b) MCMC with a data-driven patch prior; $L=1$; 100 iterations(c) MCMC with a border-based data-driven patch prior; $L=9$; 100 iterations

Figure 6.6: Three different OGM algorithms are used to estimate the occupancy grid from the same data. The first column illustrates the OG produced using the legend in Figure 1.10g, the second column is the best thresholded map (best F_1 score), and the third column only shows the cells that differ from the ground truth map (FP in blue and FN in red). Algorithms that better estimate the GT map (shown in Figure 5.9j), show fewer cells in the third column.

Figure 6.6 shows the map generated by various algorithms on one of the datasets for each of (i) the original OG, (ii) the best thresholded map, and (iii) the FP/FN map. Specifically, the false positives are shown in blue and the false negatives are shown in red. In Figure 6.6a, we see that traditional OGM does well at estimating the map, but not necessarily near the walls where the map transitions from unoccupied to occupied. Furthermore, the false positives in the unoccupied space pose a challenge for localization and path planning algorithms. For example, a path planning algorithm would avoid planning a path through the area with false positives to avoid a perceived collision. A localization algorithm would have difficulty using the map for localization in the area with false positives in the unoccupied space as it would expect a range sensor to reflect off of those cells, but it would not. Figure 6.6b shows the result of the MCMC algorithm with a data-driven prior after 100 iterations. This algorithm better estimates the residual uncertainty in the map and is therefore able to produce a more accurate thresholded map as well. Not only does this map exhibit less false negatives (red), but it also notably does not exhibit the false positives (blue) in the free space, thus allowing a path planning algorithm to plan a path in this space and localization algorithms to better localize against the map. Finally, Figure 6.6c highlights that incorporating a border-based data-driven patch prior into the MCMC algorithm yields significantly better results as the thresholded map very nearly resembles the ground truth map, shown in Figure 5.9j. This result is evident by the few false positives and false negatives shown in the third image. This high fidelity estimate of the ground truth map could be used to improve path planning and localization algorithms.

To highlight that the results in Figure 6.6 are robust to different threshold values, Figure 6.7 performs a similar analysis on the three algorithms at three different threshold values: (i) 0.1, (ii) 0.2, and (iii) 0.6. Traditional OGM estimates all *unknown* cells at the cell prior and therefore does dramatically worse when a threshold above this value is chosen – as seen in the third image of Figure 6.7a. The MCMC algorithm with $L=1$ also performs poorly for *unknown* cells when a high cell threshold is used. However, this algorithm performs better than traditional methods for cells that are mapped. Finally, the MCMC algorithm with a border-based data-driven patch prior and $L=9$ outperforms the other algorithms for all three threshold values, as indicated by the few FP and FN cells shown in Figure 6.7c. Therefore Figure 6.7 highlights that this MCMC method is robust to varying threshold values and the absence of blue and red cells in Figure 6.7c demonstrates that the algorithm accurately estimates the ground truth map.

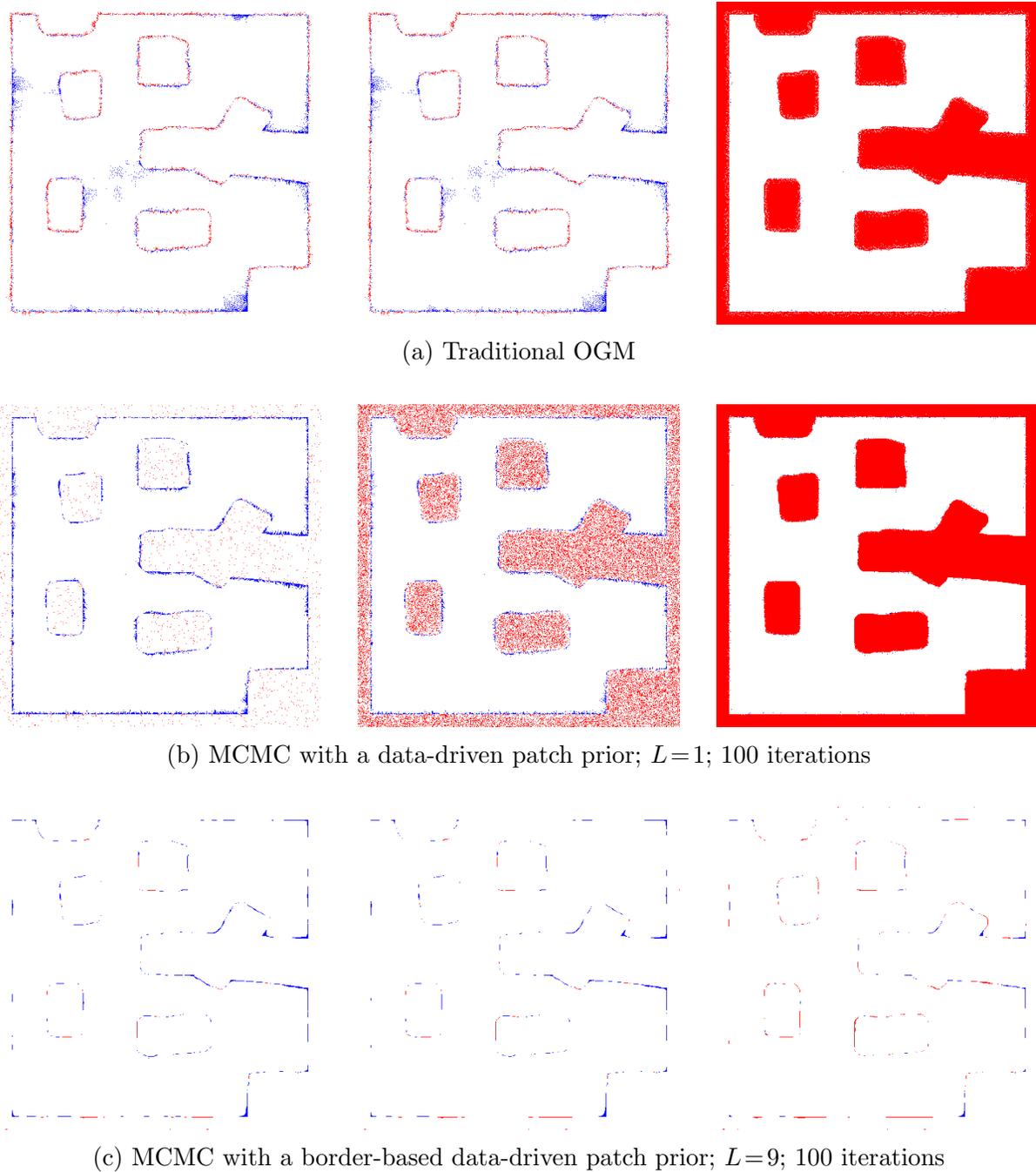


Figure 6.7: The OG is computed using three different algorithms, then thresholded at three values and compared to the ground truth map (shown in Figure 5.9j). Maps in each column represent a different threshold value of 0.1, 0.2, and 0.6, respectively. The correctly estimated cells are not displayed, but the false positives are displayed in blue and the false negatives in red. Therefore, the fewer cells displayed, the more accurately the algorithm estimates the ground truth map. Traditional OGM estimates all *unknown* cells at the cell prior and thus experiences a sharp decline in the estimate as a threshold greater than the cell prior is selected. The MCMC algorithm with a border-based data-driven prior outperforms the other two algorithms for all three thresholds.

Table 6.4: Precision and Recall of 100 simulated datasets. Comparing traditional OGM to MCMC OGM with a data-driven prior.

	Traditional OGM	Data-driven prior, $L=1$	Data-driven prior, $L=9$	Border-based Data-driven prior, $L=9$
Max Precision	0.9717	0.9571	0.9913	0.9976
Max F_1 Score	0.9328	0.9460	0.9458	0.9820
Threshold for max F_1	0.20	0.10	0.10	0.07
Precision @ max F_1	0.9310	0.9026	0.9032	0.9694
Recall @ max F_1	0.9346	0.9939	0.9925	0.9950

To further illustrate the strength of the border-based data-driven patch prior MCMC algorithm, Figure 6.8 illustrates the P-R curves over all 100 simulated datasets, and Table 6.4 summarizes the highest precision, recall, and F_1 score achieved for each method in the analysis. Furthermore, Figure 6.9 plots the F_1 score over all 100 datasets for various threshold values. This analysis was conducted by estimating the maps for all 100 simulated datasets and varying the threshold value for each algorithm’s estimate of the marginal cell occupancy for all $500 \times 500 \times 100$ cells. Not only does the border-based algorithm have the highest precision and F_1 score, but Figure 6.9 highlights that it is robust to the threshold as there is little uncertainty in the marginal occupancy probability of each cell. Figure 6.9 also highlights that all of the algorithms perform poorly for very low threshold values. However, we note that traditional OGM has a large discontinuity at the cell prior because all unmapped cells are estimated at the cell prior. Figure 6.8 shows that traditional OGM can have higher precision than the MCMC algorithms that do not incorporate a patch border. In fact, the plot is somewhat deceiving as both MCMC algorithms have a higher maximum F_1 score as shown in Table 6.4, but they can do worse on precision for very high threshold values (i.e., > 0.97) as traditional OGM can be overconfident in the marginal cell occupancy probability. Note that the values in Table 6.4 for the data-driven prior for $L=1$ and $L=9$ are quite close (except for maximum precision) and we believe that this slight difference will continue to decrease with more iterations of both MCMC algorithms. However, Figure 6.9 shows that a larger data-driven patch size outperforms a smaller one on F_1 score for most threshold values. Overall, the MCMC algorithm with a border-based data-driven patch prior outperforms the other algorithms in all measures. This result is highlighted in Table 6.4 and by the fact that it dominates the other algorithms in Figure 6.8.

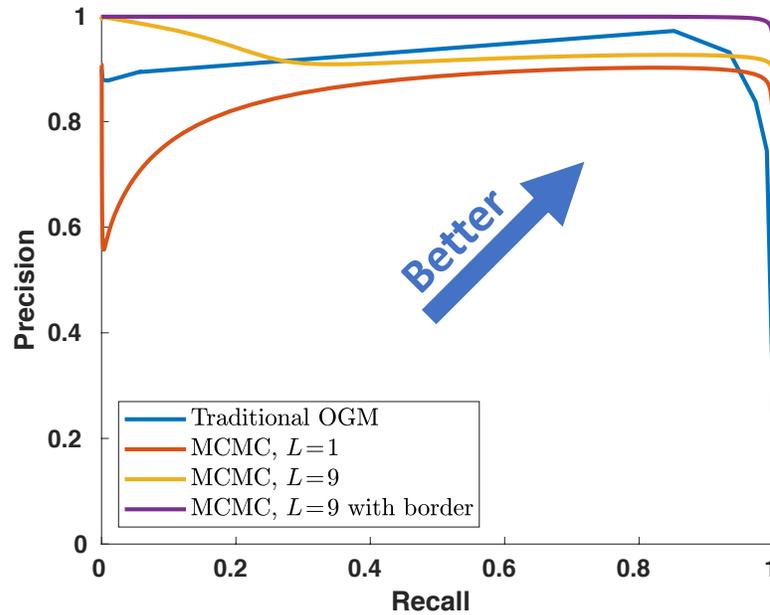


Figure 6.8: Precision-Recall curve for 100 simulated datasets, where each MCMC algorithm is averaged over 100 map samples. The MCMC algorithm with a border-based data-driven patch prior outperforms the other algorithms for all thresholds.

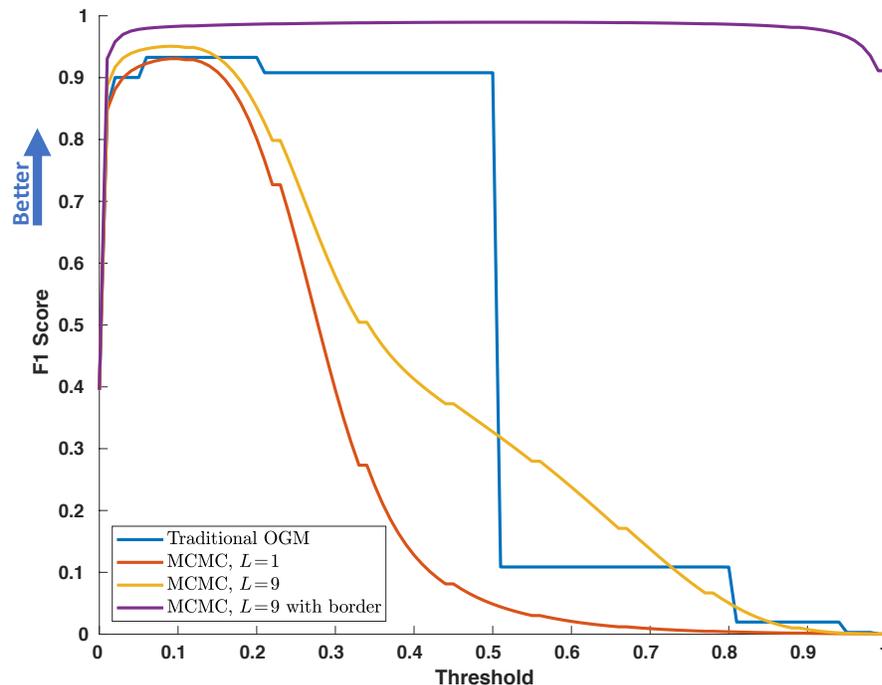


Figure 6.9: F_1 score for 100 simulated datasets as a function of cell threshold value. The MCMC methods are able to achieve a higher F_1 score than traditional OGM for certain cell occupancy probability threshold values. The MCMC algorithm that uses a border-based data-driven patch prior is very robust to the threshold value selected as it has a high F_1 score for nearly all cell threshold values.

Treating OGM as a binary classification problem and analyzing the precision and recall of an OG highlights one effective use of capturing cell correlations in OGM. However, by thresholding the map on a cell-by-cell basis, all mutual information between cells is lost. As highlighted in Chapter 4, the MCMC algorithm is also a powerful tool to estimate other statistics, such as the first-occupied-cell along a measurement ray. In higher dimensions, this can be generalized to estimating the occupancy of a group of cells. For example, a path planning algorithm may seek to determine the joint probability that all cells along a path are unoccupied or a semantic mapping algorithm may seek to determine the occupancy probability of a group of cells to determine the size or shape of an area. The MCMC algorithm for OGM enables these and many more applications that should be studied further. However, the improved binary classification is one illustration of the effectiveness of this approach. It is worth noting that many of the cells will not have mutual information with neighbouring cells and most of the mutual information exists where the range sensor is reflected – often the most critical part of the map to estimate. Dhiman (2019) also observed this result and suggested having the MCMC algorithm focus on sampling cells that reflect the measurement four times more frequently than other cells. He also observed that this leads to faster convergence of the MCMC algorithm. This and other extensions of the research presented in this thesis are discussed further in Section 7.2, but the analysis presented in this section highlights some of the benefits of capturing the cell correlations in both the measurements and the prior.

6.2 Hardware Results

Section 6.1 showcased various OGM algorithms presented in this thesis on realistic data from simulated datasets. Some of these algorithms use a data-driven patch prior which was derived from drawing patch samples from the 10 maps used to generate the simulated datasets. The set of patch priors is the same throughout this thesis. In this section, we show that the mapping algorithms can be applied to real data and that the data-driven patch priors can be applied to maps that were not sampled to generate the patch priors. To demonstrate this, we use publicly available data of the Intel Research Lab in Seattle, Washington (Howard and Roy, 2003) and the MIT Stata Center in Cambridge, Massachusetts (Fallon et al., 2013). These datasets were chosen because they were created by densely mapping a confined area using a laser range finder on a mobile robot. The dense laser data is useful for creating a benchmark map and can be subsampled to high-

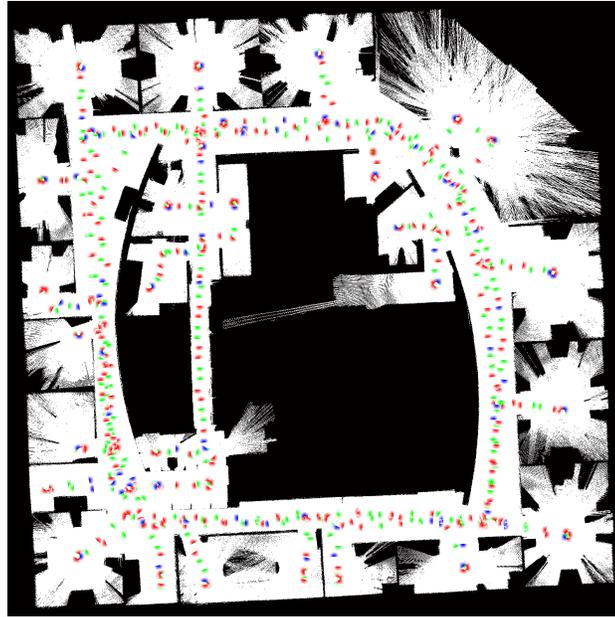
light that the algorithms are able to estimate the benchmark with less data. O’Meadhra et al. (2019) used a similar approach to highlight the benefit of their algorithm on sparse data. This approach was not required with the simulated data in Section 6.1 because the ground truth map was known. Both datasets also use the dense data to provide accurate estimates of the robot’s pose with each laser scan.

Given the two datasets, we use all of the data within each set to generate a benchmark OG for each dataset. The benchmark map is created using traditional OGM and then each cell in the map is thresholded at an occupancy probability of 0.20 because, as we saw in Table 6.4, this value yields a high F_1 score for traditional OGM. Figure 6.10 shows the two benchmark maps, where occupied cells are shown in black and the remaining cells are unoccupied. The two maps in Figure 6.10 also include additional information that will be discussed in the following sections. Section 6.2.1 will discuss the results of the Intel Lab dataset in detail and Section 6.2.2 will examine the results of the MIT Stata dataset. Both sections will show the effectiveness of the MCMC algorithm for OGM with a border-based data-driven patch prior.

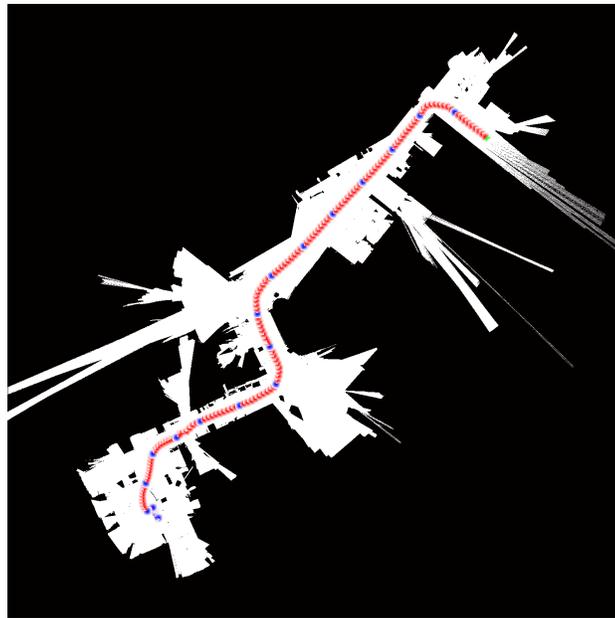
6.2.1 Intel Lab Dataset

In the Intel Lab dataset, the location is estimated by a SLAM algorithm using the dense laser data. The dataset uses a laser range finder that can measure up to 80 meters with ± 15 mm accuracy and measures every 1.0° for a 180° field-of-view. The Intel Lab dataset had many measurements that were reported as *maximum range*, which were considered erroneous given that the experiment was conducted indoors, where the hallways were no more than 20 meters long. As a consequence, any measurements longer than 20 meters were ignored in this dataset. Figure 6.11a shows the OG map generated using traditional OGM and all of the range measurements in the dataset and Figure 6.11b shows the OG after omitting the *maximum range* measurements. Figure 6.11b appears to better represent the environment and we therefore omit the *maximum range* measurements for any further analysis using the Intel Lab dataset. Given the occupancy grid map estimate using traditional OGM shown in Figure 6.11b, we threshold the map to obtain the benchmark map shown in Figure 6.11c; note that this is the same result shown in Figure 6.10a.

A densely mapped environment does not have much uncertainty in the OG map and therefore provides an analog for the ground truth map. To showcase the algorithms



(a) Benchmark OG map for the Intel Lab dataset



(b) Benchmark OG map for the MIT Stata dataset

Figure 6.10: The binary benchmark OG map used for the two hardware datasets, overlaid with the first 10 cells mapped by each measurement ray from the three experiments on each map. For the benchmark map, occupied cells are show in black and all other cells are unoccupied. The most dense experimental dataset uses all of the rays depicted, the second most dense uses only those depicted in red and blue and the least dense uses only the rays depicted in blue.

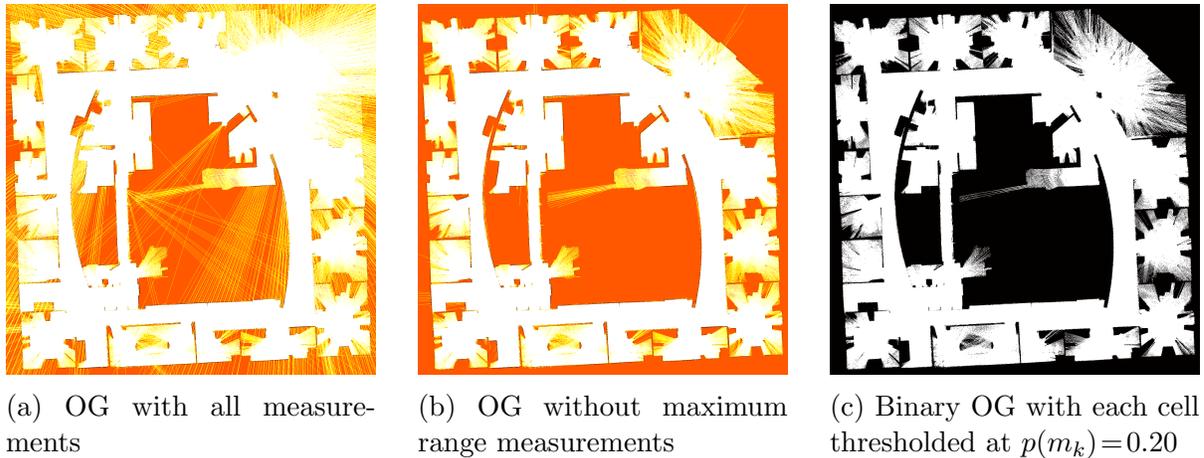


Figure 6.11: OG maps generated using traditional OGM with the Intel Lab dataset. Figure (a) shows the OG map produced by using all of the data. The dataset contains many maximum range measurements, so figure (b) shows the resulting OG map after removing those. Finally, figure (c) shows the result after thresholding each cell at $p(m_k) = 0.20$. The thresholded map is used as the benchmark as other OGM algorithms seek to estimate it with fewer measurements. The legend used is shown in Figure 1.10g.

developed in this thesis, we use a subset of the data in each experiment to approximate a sparsely mapped environment. Note that such sparsity would exist in data collected at longer range (e.g., outdoors) or from a fast moving vehicle (e.g., a car or plane), or from a lower cost sensor. Although this sparsity would highlight the benefit of the algorithms presented in this thesis, the same sparsity could result in decreased localization accuracy and may not provide a reference map similar to that obtained from a dense dataset. Therefore, we use the dense data to retain the localization benefit, but subsample the measurements from the hardware datasets to highlight the strength of the MCMC OGM algorithms with a data-driven patch prior, presented in Section 5.2.

Table 6.5 outlines how much of the Intel Lab dataset was used for the three experiments conducted on this dataset. All of the data was used to develop the benchmark map, which was then used to evaluate the map generated by each of the subsequent experiments. All three experiments only use half of the range measurements per robot position, thus leaving 2.0° between measurements in the 180° field-of-view of the sensor. The first experiment uses all of the robot poses in the dataset, the second experiment uses half of the robot poses and the third experiment uses only 10% of the robot poses. Each of the experiments was conducted on a map size of $30\text{m} \times 30\text{m}$ and used a grid resolution of 2.0cm per cell. This is a considerably finer resolution than is typical in the literature, where authors typically use a grid resolution of $5\text{-}25\text{cm}$ (e.g., Danescu et al.,

Table 6.5: Experiments conducted on the Intel Lab dataset (Howard and Roy, 2003). Each map measures $30\text{m} \times 30\text{m}$ with a grid resolution of 2.0cm for a total of $K = 2.25 \times 10^6$ cells. The benchmark has a 1.0° increment between measurements, whereas each experiment has a 2.0° increment between measurements.

	Number of Poses	Measurements per Pose	Total Measurements, N	Portion of Cells Mapped
Benchmark	910	180	163 800	59.63%
Experiment 4	910	90	81 900	55.78%
Experiment 5	455	90	40 950	50.27%
Experiment 6	91	90	8190	28.22%

2011; Schaefer et al., 2018b). A finer grid resolution enables a more detailed map of the environment. Given the grid resolution and measurements, Table 6.5 also indicates the portion of cells in the map with at least one range measurement. This ratio highlights the sparsity of mapped cells with each experiment.

Figure 6.10a displays the binary benchmark map for the Intel Lab dataset by showing the occupied cells in black and the remaining cells are unoccupied. However, the white unoccupied cells are also overlaid with the first 10 cells mapped by each measurement ray in the experiments. Each cluster of measurement rays originates from a robot pose. All of the measurement rays shown in Figure 6.10a were used for Experiment 1. Only the red and blue rays were used for Experiment 2 and only the blue rays were used for Experiment 3.

Similar to Section 6.1.4, for each experiment, traditional OGM was compared to MCMC with a data-driven prior with a patch size of $L=1$, $L=9$ and with a border for $L=9$ as well. However, as these experiments intentionally examine sparse measurements in the map and the previous experiments in simulation used an exploration algorithm to ensure full coverage of the map being explored, the sparse measurements revealed a challenge with the MCMC technique depicted in Figure 6.12. As stated previously, the MCMC algorithm starts with a thresholded version of traditional OGM to reduce or eliminate the *burn-in* or convergence period typically required for MCMC algorithms. However, Figure 6.12 highlights that the MCMC algorithm with $L=1$ appears to diverge from the correct solution as more samples are obtained. Upon further investigation, this appears to be a result of (a) the sensor model and (b) sampling a single cell at a

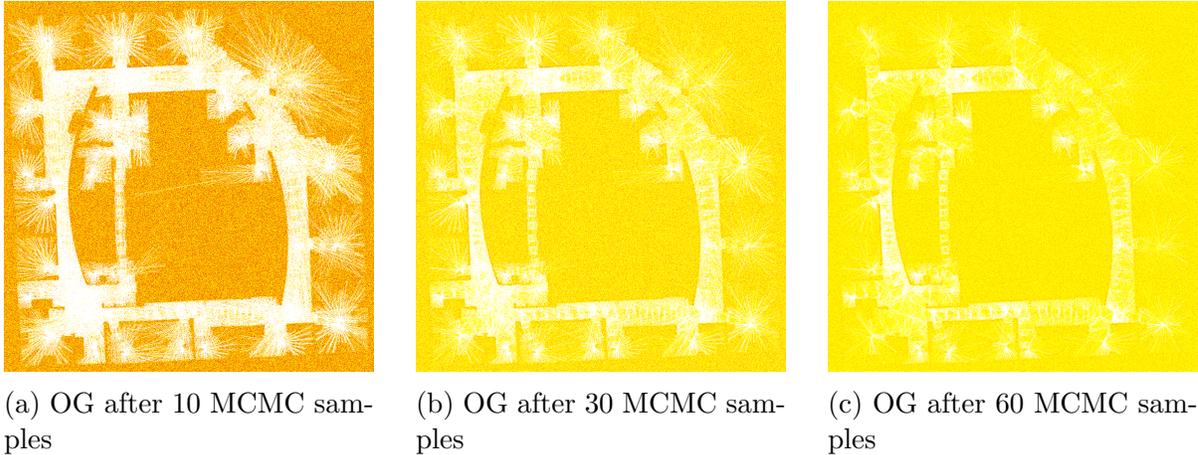


Figure 6.12: OG maps generated using the MCMC algorithm with $L=1$ and 10% of the measurements in the Intel Lab dataset. The resulting OG map is shown after drawing (a) 10, (b) 30, and (c) 60 samples from the posterior. However, because the measurement model used depends on the first-occupied-cell, f , and the algorithm only samples one cell at a time, the result is that many measurements are negated by the “flipped” cell in a sparsely mapped area. A larger patch size with a data-driven prior can help overcome this limitation. The legend used is shown in Figure 1.10g.

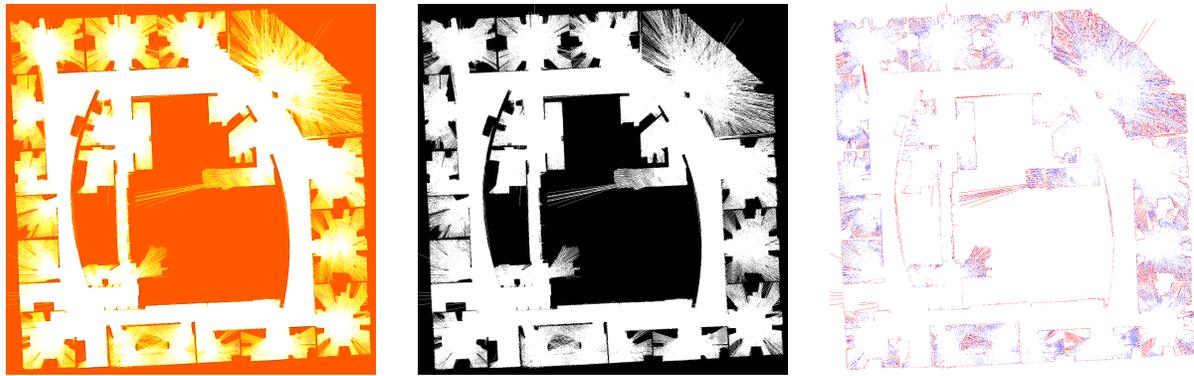
time in the MCMC algorithm. The sensor model relies on the previous iteration of the map to determine the first occupied cell, f . However, a single incorrectly *flipped*¹ cell along the measurement ray yields a significantly worse range estimate for that sensor measurement. However, if that measurement was the only measurement for some cells, then the cells beyond that *flipped* cell will essentially draw from the prior and several more unoccupied cells will be sampled as occupied, thus compounding the problem. This issue was not observed previously because cells had multiple measurements that would prevent unoccupied cells from reverting to the prior if one range measurement was impacted by the randomness of the sampling algorithm. We believe that a different sensor model could overcome this issue. However, we also found that sampling patches as opposed to cells also helped to overcome this issue, as a patch is more likely to be measured by multiple measurements. This effect was also observed for a patch size of $L=9$ for very sparse measurements, but it was not observed when the patch prior was dependent on the cells bordering the patch. Therefore, we have omitted the results of the MCMC algorithm with $L=1$ from this section. Furthermore, we have recognized that an additional benefit of conditioning the patch prior estimate on bordering cells is to overcome this issue.

¹We use the term *flipped* to describe a cell being sampled as occupied that was previously sampled as unoccupied, or vice versa.

Figure 6.13 shows the result of the Experiment 1 on the Intel Lab dataset. Each of the three algorithms depicted performs well against the benchmark map. However, Figure 6.13c shows that the MCMC algorithm using a border-based data-driven patch prior shows several false negatives (red) in undermapped areas. We believe that the MCMC algorithm is actually correct in its estimate of these cells, but they were not represented in the benchmark map since it is not a true ground truth map. Figure 6.14 shows the result of Experiment 2 on the Intel Lab dataset. We observe that traditional OGM has several false positives as expected because cells are more sparsely mapped in this experiment and traditional OGM methods are not able to estimate unmapped cells beyond the cell prior. In contrast, the MCMC algorithm with a patch size of $L=9$ is able to estimate many of the cells between range measurements as unoccupied. Therefore, we not only see fewer false positives than traditional OGM, but also some false negatives that we believe are in fact correct because the benchmark map was not able to accurately capture their occupancy. Finally, Figure 6.14c depicts the MCMC algorithm with $L=9$ and a border-based data-driven patch prior. This algorithm shows very similar results to Experiment 1. Therefore, we find that fewer measurements (i.e., increased sparsity) have little effect on this algorithm’s ability to estimate the benchmark map.

The Experiment 3 results on the Intel Lab dataset are shown in Figure 6.15. The sparsity in this experiment is evident from the map generated by traditional OGM in Figure 6.15a. The resulting thresholded map highlights the occupied cells between range measurements. The thresholded map in Figure 6.15a would be virtually unusable for most path planning and localization algorithms. In fact, the data in Experiment 3 is so sparse that we begin to see the challenges in Figure 6.15b with $L=9$ that were observed in Figure 6.12 with less sparse data and $L=1$. Namely, in patches with very few measurements, and a sensor model that is dependent on the first-occupied-cell, the MCMC algorithm is unable to converge to the correct solution and reverts to sampling from the prior. However, Figure 6.15c highlights that by conditioning the patch prior on bordering cells, the MCMC algorithm is able to converge and sample from the full posterior. The result is a map that clearly indicates the hallways and most of the rooms. The map generated best estimates the benchmark map and would be usable for path planning and localization algorithms. Moreover, the map may also be useful for semantic mapping algorithms that seek to label the hallways and rooms.

We quantify the results of the three experiments by computing the F_1 score, presented in Equation (6.1), that combines the precision and recall of the experiments as



(a) Traditional OGM

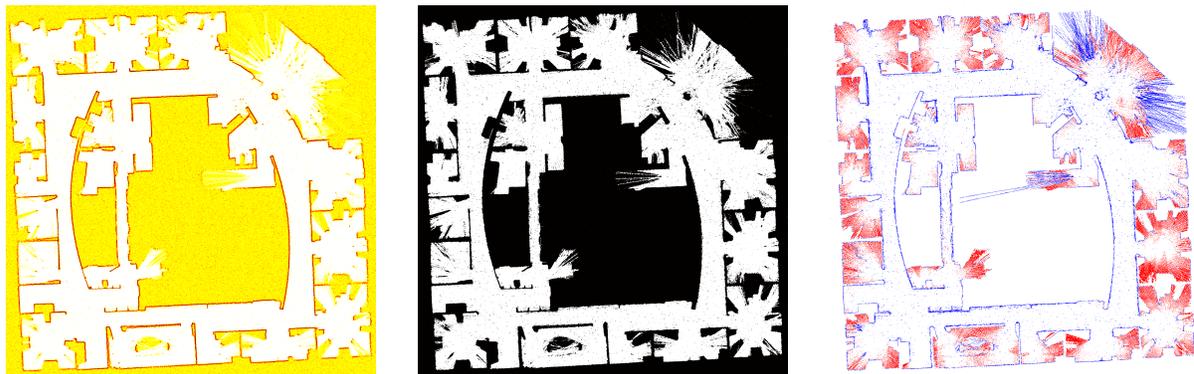
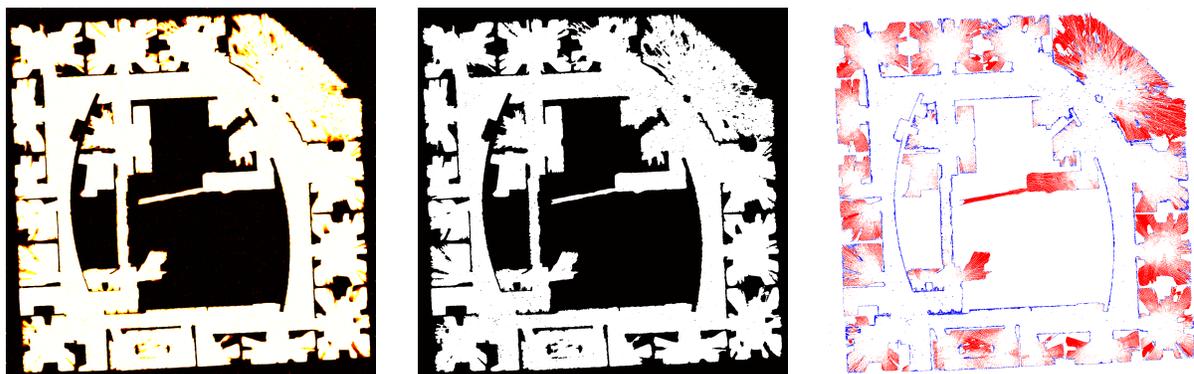
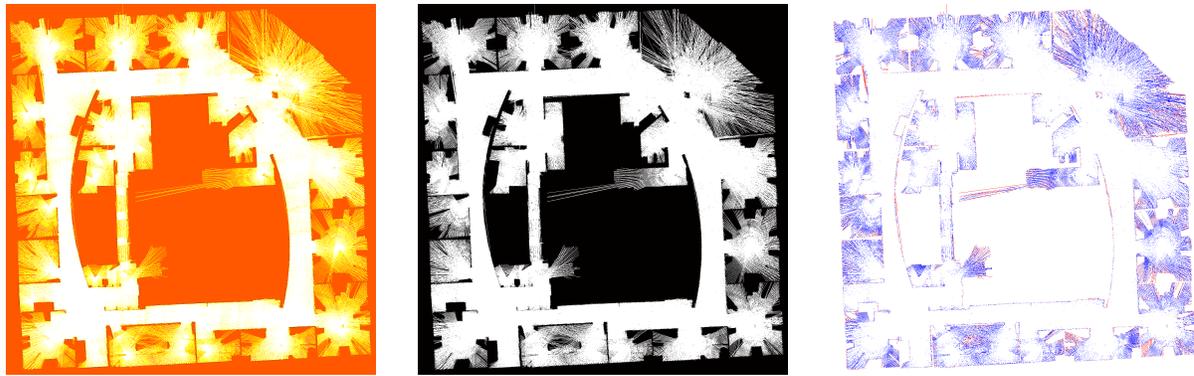
(b) MCMC with data-driven patch prior; $L=9$; 100 iterations(c) MCMC with data-driven patch prior with border; $L=9$; 170 iterations

Figure 6.13: Three different OGM algorithms are used to estimate the occupancy grid from Experiment 1 on the Intel Lab dataset. The first column illustrates the OG produced using the legend in Figure 1.10g, the second column is the best thresholded map (best F_1 score), and the third column only shows the cells that differ from the benchmark map (FP in blue and FN in red). Algorithms that better estimate the benchmark map, shown in Figure 6.10a, show fewer cells in the third column.



(a) Traditional OGM

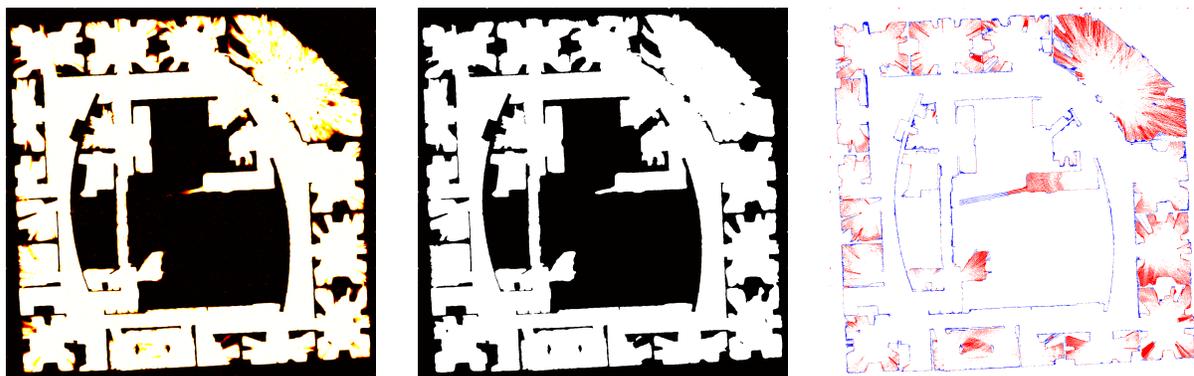
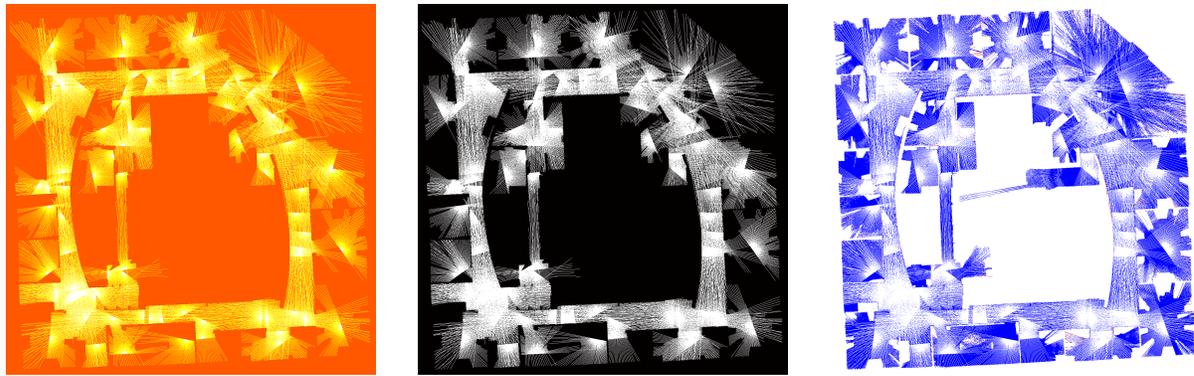
(b) MCMC with data-driven patch prior; $L=9$; 100 iterations(c) MCMC with data-driven patch prior with border; $L=9$; 200 iterations

Figure 6.14: Three different OGM algorithms are used to estimate the occupancy grid from Experiment 2 on the Intel Lab dataset. The first column illustrates the OG produced using the legend in Figure 1.10g, the second column is the best thresholded map (best F_1 score), and the third column only shows the cells that differ from the benchmark map (FP in blue and FN in red). Algorithms that better estimate the benchmark map, shown in Figure 6.10a, show fewer cells in the third column.



(a) Traditional OGM

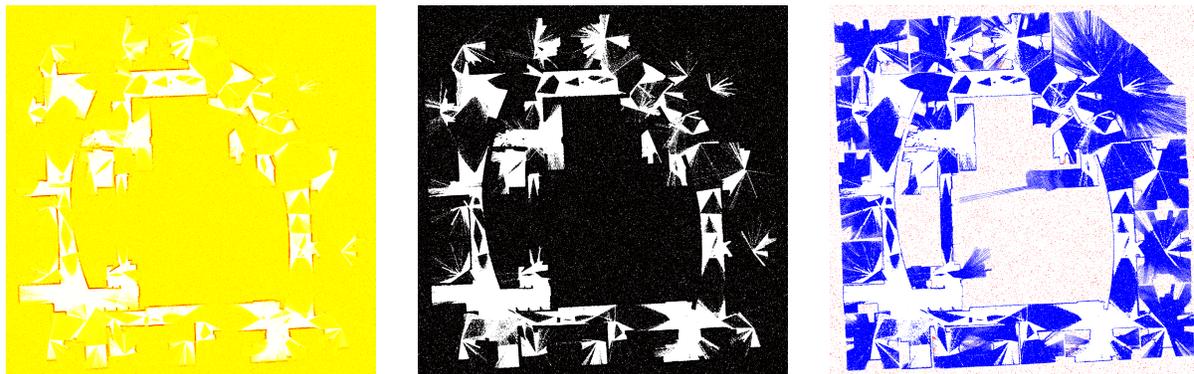
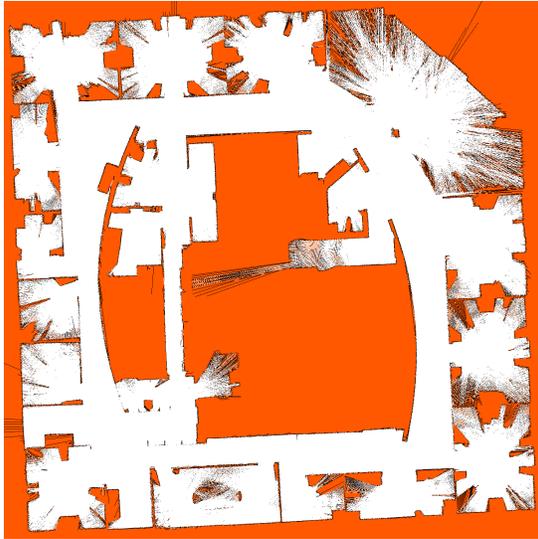
(b) MCMC with data-driven patch prior; $L=9$; 350 iterations(c) MCMC with data-driven patch prior with border; $L=9$; 200 iterations

Figure 6.15: Three different OGM algorithms are used to estimate the occupancy grid from Experiment 3 on the Intel Lab dataset. The first column illustrates the OG produced using the legend in Figure 1.10g, the second column is the best thresholded map (best F_1 score), and the third column only shows the cells that differ from the benchmark map (FP in blue and FN in red). Algorithms that better estimate the benchmark map, shown in Figure 6.10a, show fewer cells in the third column.



(a) Benchmark map with only mapped cells.



(b) Benchmark map with all cells within 10 cells of a mapped cell.

Figure 6.16: Benchmark map for Intel Lab dataset where only cells near measured cells are included – white cells are unoccupied and black cells are occupied. The cells depicted in orange are omitted when comparing maps to the benchmark as they do little to highlight the effectiveness of the various mapping algorithms. Figure (a) shows only the mapped cells as part of the benchmark, whereas (b) includes cells that are within 10 cells of a mapped cell. We use (b) to compute the F_1 to include more occupied cells.

compared to a benchmark map. The benchmark map presented in Figure 6.10a is used to compute the F_1 score; however, certain cells are omitted from the computation because the bounding box chosen can affect the result. Many cells in the benchmark map are far from any sensor measurements and therefore each of the algorithms considered is likely to estimate these accurately and therefore inflate the F_1 score of each algorithm. If a benchmark map is dominated by these cells, then the differences between the F_1 scores is diminished. Figure 6.16 shows two possible benchmark maps considered. Figure 6.16a shows only the measured cells considered (in black or white) and cells with no range measurements omitted from the benchmark and shown in orange. This map would use only 59.63% of the cells; however, it would favour mapping algorithms that bias cells as unoccupied since each range measurement can map several cells as unoccupied and only one as occupied. Figure 6.16b shows an alternative benchmark map that includes all cells that are within 10 cells of a cell mapped by a range measurement. This benchmark includes 73.04% of the cells and seeks to strike a balance between considering all cells in the bounding box and only those mapped by a range measurement. Therefore, we use

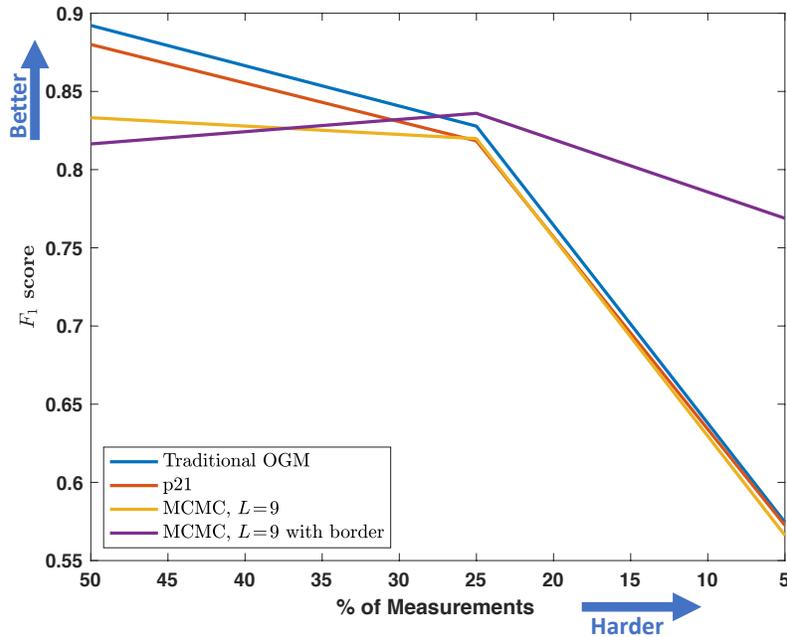


Figure 6.17: F_1 scores for Intel Lab dataset experiments with 50%, 25% and 5% of measurements used to create the benchmark map. All of the OGM algorithms do well when the area is densely mapped, but the MCMC algorithms that use a data-driven patch prior are able to better estimate the map with sparse measurements.

the black and white cells show in Figure 6.16b to compute the F_1 score of the various algorithms for each experiment on the Intel Lab dataset.

Figure 6.17 summarizes the results from the Intel Lab experiments by plotting the F_1 score of each OGM algorithm against a decreasing number of range measurements (i.e., each of the three experiments). As shown in Table 6.5, each experiment considers fewer range measurements by decreasing the number of robot poses considered for each experiment. Therefore, the decrease in measurements is correlated to an increased sparsity of cells that are mapped. The results of Experiment 1 show that each of the algorithms is able to estimate the benchmark map well with a dense set of measurements. Specifically, traditional OGM is seen to outperform the others on Experiment 1 since it was used to develop the benchmark map. However, as the number of measurements decreases and sparsity increases, the MCMC patch algorithms clearly do better as they incorporate measurements from neighbouring cells and a data-driven patch prior. Furthermore, the MCMC algorithm with a border-based patch prior outperforms the other algorithms when the data is particularly sparse in Experiment 2 and 3. Again, we believe that this

Table 6.6: Average time to compute one MCMC map sample (K/L patches) for experiments on the Intel Lab dataset, where $L = 9$ and $K = 2.25 \times 10^6$. A border-based data-driven patch prior is more efficient because the border reduces the number of likely patch configurations in the prior.

	Data-driven prior [s]	Border-based, Data-driven prior [s]
Experiment 1	6433	1518
Experiment 2	3023	723.6
Experiment 3	306.1	166.8

algorithm would outperform in all instances if a ground truth map was available as we saw in Section 6.1.4. Figure 6.17 also shows the performance of the p21 algorithm from Section 2.5 for reference. The p21 algorithm performs similarly to traditional OGM on this metric as it is the same algorithm with a different Δ function. The optimized p21 algorithm is seen to perform slightly worse on the F_1 metric, but this is likely because traditional OGM was used to generate the benchmark map.

Table 6.6 highlights the computation time required to draw one map sample using the MCMC algorithm on each of the Intel Lab experiments. The results in the table were computed using a laptop computer running Matlab on Ubuntu with 32GB of RAM and a 2.70GHz Intel i7-6820HQ CPU. Furthermore, each map sample drew 2.25×10^6 cells and did not omit cells that were not used to compute the F_1 score. Each sampling algorithm could be sped up by only sampling mapped cells or cells in close proximity to mapped cells as shown in Figure 6.16. Table 6.6 shows that the border-based patch prior is able to draw each MCMC sample faster than the algorithm that does not depend on the border. Coupled with the fact that this method also performs the best on the F_1 metric, we recommend this algorithm over others. Traditional OGM, particularly with an optimized Δ function, is useful for a fast initial estimate and works well as the first sample in an MCMC algorithm. However, the MCMC algorithm is able to achieve a better result by drawing additional samples from the full posterior. The MCMC algorithm with a border-based data-driven patch prior will perform particularly well in sparsely mapped environments. Even in densely mapped environments, this algorithm generates a map that is more robust to a cell threshold value. In addition, the MCMC algorithm has the ability to compute more than the cell marginals. For example, the samples can be used to estimate the joint probability of several cells.

6.2.2 MIT Stata Dataset

The MIT Stata Center dataset (Fallon et al., 2013) contains extensive 2D LiDAR data in an indoor environment where the authors manually calibrated the robot poses, such that the laser scans match the floor plans of the building. The authors of the MIT data were able to achieve 2-3cm accuracy for each robot pose estimate. This dataset uses a laser range finder that is capable of measuring up to 30 meters with ± 50 mm accuracy and measures every 0.25° for a 260° field-of-view. This dataset is significantly more dense than the Intel Lab dataset and therefore only one experiment from the dataset was used – specifically the dataset labeled “2012-04-03-07-56-24_part4_floor2.gt.laser”.

Table 6.7 outlines how much of the MIT Stata dataset was used for the three experiments on this dataset. All of the data was used to develop the benchmark map shown in Figure 6.10b and the resulting map from each experiment was subsequently compared to this benchmark. All three experiments use only 10% of the range measurements per robot position, thus leaving 2.5° between measurements in the 260° field-of-view of the sensor. Experiment 4 uses all of the robot poses in the dataset, Experiment 5 uses 10% of the robot poses and Experiment 6 uses only 1% of the robot poses. Each of the experiments was conducted on a map size of $50\text{m} \times 50\text{m}$ and used a grid resolution of 3.3cm per cell. Again, this fine grid resolution enables a more detailed map of the environment. Given the grid resolution and measurements, Table 6.7 also indicates the portion of cells in the map with at least one range measurement. This ratio highlights the sparsity of mapped cells with each experiment.

Figure 6.10b illustrates the benchmark map used for experiments on the MIT Stata dataset. The robot poses in this dataset are quite close together. The sparse measurement rays used for Experiment 6 are shown in blue. Experiment 5 uses these in addition to those shown in red. Finally, Experiment 4 uses the rays shown in blue, red and green; however, the green rays are hardly visible as they are overlaid by the others.

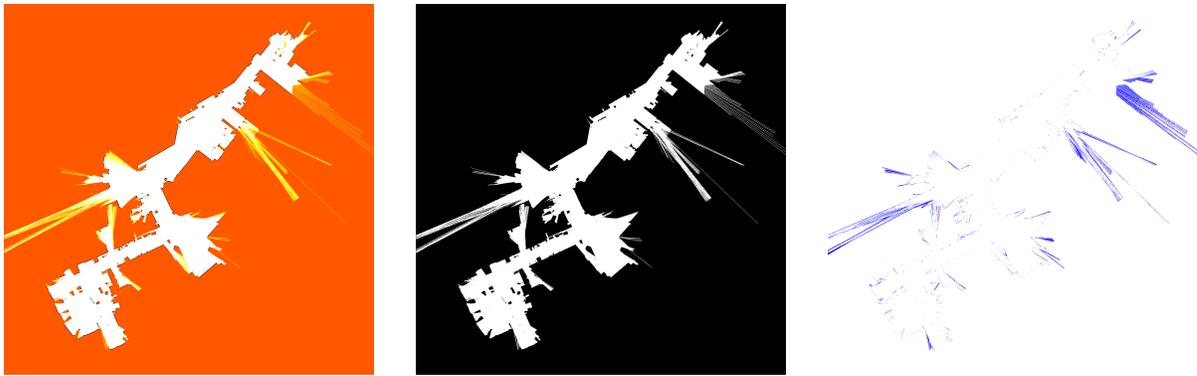
Figure 6.18 shows the result of Experiment 4 on the MIT Stata dataset. Each of the three algorithms depicted performs well against the benchmark map. Note that the MIT Stata dataset has better coverage of the map than the Intel Lab dataset and therefore the benchmark map generated is more complete. Therefore, we note that the MCMC algorithms represent the benchmark well and do not have the false negatives that were seen in Figure 6.13.

Table 6.7: Experiments conducted on a portion of the MIT Stata dataset (Fallon et al., 2013). Each map measures $50\text{m} \times 50\text{m}$ with a grid resolution of 3.3cm for a total of $K = 2.25 \times 10^6$ cells. The benchmark has a 0.25° increment between measurements, whereas each experiment has a 2.5° increment between measurements.

	Number of Poses	Measurements per Pose	Total Measurements, N	Portion of Cells Mapped
Benchmark	1977	1040	2 056 080	16.21%
Experiment 4	1977	104	205 608	14.58%
Experiment 5	198	104	20 592	11.74%
Experiment 6	20	104	2080	5.23%

Figure 6.19 shows the result of Experiment 5 on the MIT Stata dataset. We observe that traditional OGM has several false positives as expected because cells are more sparsely mapped in this experiment and traditional OGM methods are not able to estimate unmapped cells beyond the cell prior. In contrast, the MCMC algorithm with a patch size of $L = 9$ is able to correctly estimate many of the cells between range measurements as unoccupied. Therefore, we see fewer false positives in Figure 6.19b than traditional OGM in Figure 6.19a. Figure 6.19c illustrates the results of a third algorithm, the MCMC algorithm with $L = 9$ and a border-based, data-driven prior. This algorithm shows very similar results to Experiment 4. Therefore, we find that fewer measurements (i.e., increased sparsity) have little effect on this algorithm’s ability to estimate the benchmark map.

Figure 6.20 shows the results from Experiment 6, the final experiment on the MIT Stata dataset. The sparsity in this experiment is evident from the map generated by traditional OGM in Figure 6.20a. The resulting thresholded map highlights the occupied cells between sequential range measurements. This map would be virtually unusable for most path planning and localization algorithms. The sparsity of the measurement data also causes some issues with the MCMC algorithm with a data-driven prior as seen in Figure 6.20b. Specifically, the MCMC algorithm is unable to estimate the benchmark solution in some areas due to the sparsity of the measurements. However, Figure 6.20c highlights that by conditioning the patch prior on bordering cells, the MCMC algorithm is better able to estimate the benchmark map. The result is a map that clearly indicates the hallways and most of the rooms. The map generated by this algorithm is the best estimate of the benchmark map and would be usable for path planning and localization algorithms. Moreover, the map may also be useful for semantic mapping algorithms that seek to label the hallways and rooms.



(a) Traditional OGM

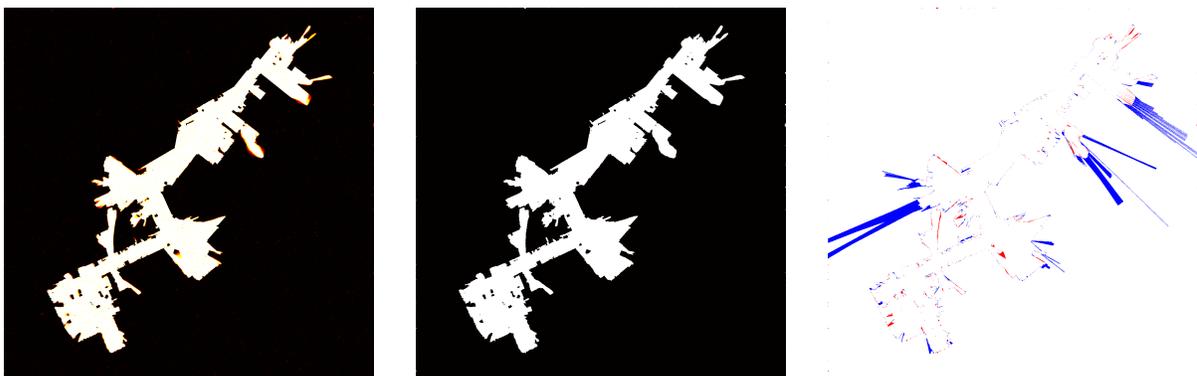
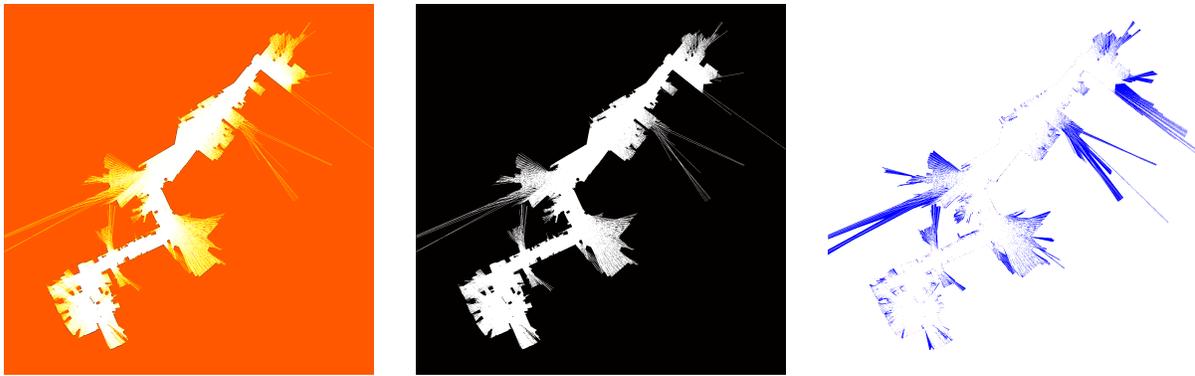
(b) MCMC with data-driven patch prior; $L=9$; 100 iterations(c) MCMC with data-driven patch prior with border; $L=9$; 300 iterations

Figure 6.18: Three different OGM algorithms are used to estimate the occupancy grid from Experiment 4 on the MIT Stata dataset. The first column illustrates the OG produced using the legend in Figure 1.10g, the second column is the best thresholded map (best F_1 score), and the third column only shows the cells that differ from the benchmark map (FP in blue and FN in red). Algorithms that better estimate the benchmark map, shown in Figure 6.10b, show fewer cells in the third column.



(a) Traditional OGM

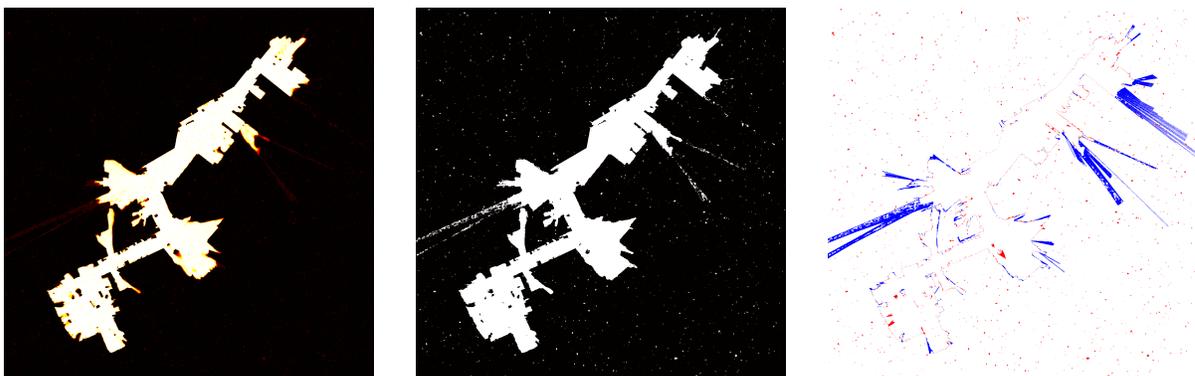
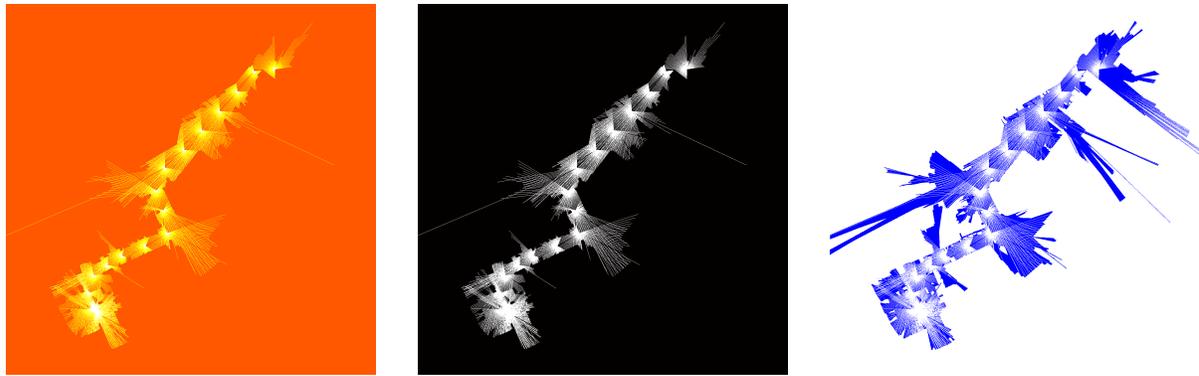
(b) MCMC with data-driven patch prior; $L=9$; 100 iterations(c) MCMC with data-driven patch prior with border; $L=9$; 180 iterations

Figure 6.19: Three different OGM algorithms are used to estimate the occupancy grid from Experiment 5 on the MIT Stata dataset. The first column illustrates the OG produced using the legend in Figure 1.10g, the second column is the best thresholded map (best F_1 score), and the third column only shows the cells that differ from the benchmark map (FP in blue and FN in red). Algorithms that better estimate the benchmark map, shown in Figure 6.10b, show fewer cells in the third column.



(a) Traditional OGM

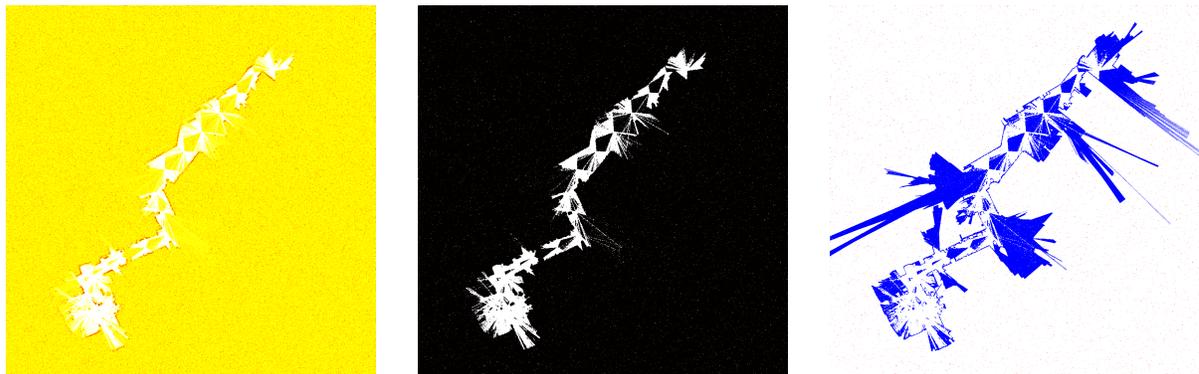
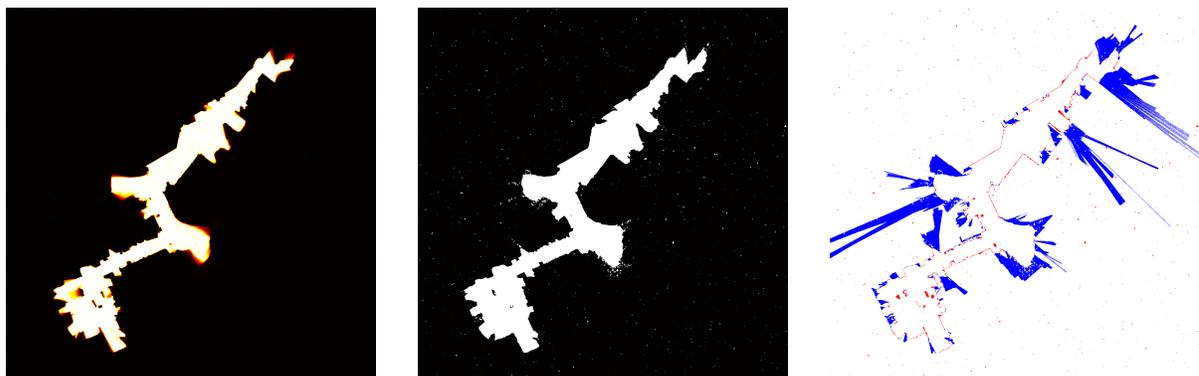
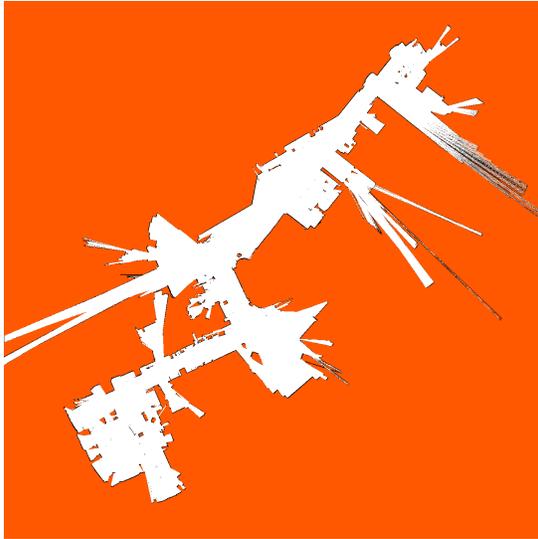
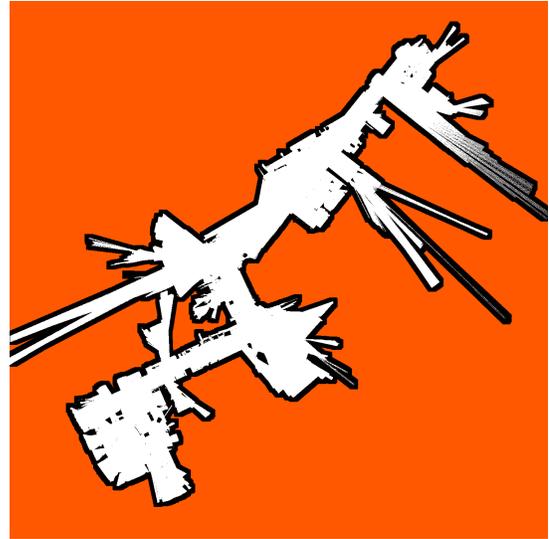
(b) MCMC with data-driven patch prior; $L=9$; 100 iterations(c) MCMC with data-driven patch prior with border; $L=9$; 1220 iterations

Figure 6.20: Three different OGM algorithms are used to estimate the occupancy grid from Experiment 6 on the MIT Stata dataset. The first column illustrates the OG produced using the legend in Figure 1.10g, the second column is the best thresholded map (best F_1 score), and the third column only shows the cells that differ from the benchmark map (FP in blue and FN in red). Algorithms that better estimate the benchmark map, shown in Figure 6.10b, show fewer cells in the third column.



(a) Benchmark map with only mapped cells.



(b) Benchmark map with all cells within 10 cells of a mapped cell.

Figure 6.21: Benchmark map for MIT Stata Center dataset where only cells near measured cells are included – white cells are unoccupied and black cells are occupied. The cells depicted in orange are omitted when comparing maps to the benchmark as they do little to highlight the effectiveness of the various mapping algorithms. Figure (a) shows only the mapped cells as part of the benchmark, whereas (b) includes cells that are within 10 cells of a mapped cell. We use (b) to compute the F_1 to include more occupied cells.

Similar to Section 6.2.1, we evaluate the three experiments by computing the F_1 score, presented in Equation (6.1), that combines the precision and recall of the experiments as compared to a benchmark map. The benchmark map presented in Figure 6.10b is used to compute the F_1 score; however, certain cells are omitted from the computation because the bounding box chosen can affect the result. Figure 6.21 shows two possible benchmark maps, where Figure 6.21a shows only the measured cells in black or white and cells with no range measurements shown in orange. The measured cells represent only 16.21% of the cells in the map, so there is a greater need to omit cells in the F_1 computation than in the Intel Lab experiments. Figure 6.21b shows an alternative benchmark map that includes all cells that are within 10 cells of a cell mapped by a range measurement and includes 24.09% of the cells. To evaluate the F_1 score on the MIT Stata Center experiments, we use the occupied (black) and unoccupied (white) cells show in Figure 6.21b, and omit those shown in orange.

Figure 6.22 summarizes the results from the MIT Stata experiments by plotting the F_1 score of each OGM algorithm against the decreasing number of measurements used in

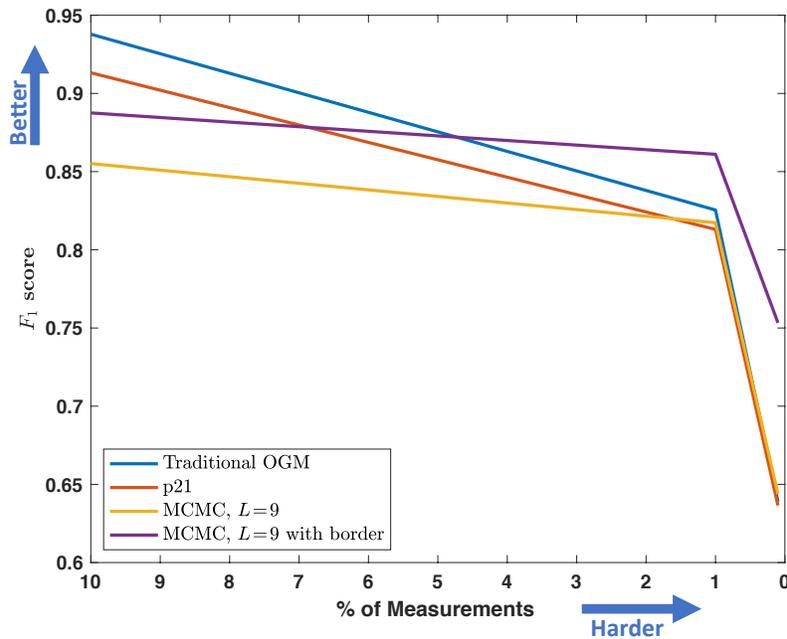


Figure 6.22: F_1 scores for MIT Stata dataset experiments with 10%, 1% and 0.1% of measurements used to create the benchmark map. All of the OGM algorithms do well when the area is densely mapped, but the MCMC algorithms that use a data-driven patch prior are able to better estimate the map with sparse measurements.

each experiment. Considering fewer robot poses and therefore fewer measurements results in more sparsely mapped cells. Each of the algorithms is able to estimate the map well with a dense set of measurements as seen in Experiment 4 with 10% of the measurements. Traditional OGM is seen to perform best on this experiment because this algorithm was used to create the benchmark map. However, as the number of measurements decreases and the cells are more sparsely measured, the MCMC patch algorithms clearly do better as they incorporate measurements from neighbouring cells and a data-driven patch prior. Specifically, Experiment 5 uses only 1% of the data used to generate the benchmark map and the F_1 score of the MCMC algorithm with a border-based data-driven prior is largely unaffected in contrast to the other algorithms that perform worse with the sparse data in Experiment 5. This algorithm continues to outperform the others on the F_1 metric as the data is reduced to 0.1% of range measurements in Experiment 6 (i.e., only 20 poses). Figure 6.22 also shows the result of using the optimized Δ function, p21, described in Section 2.5. As seen with the Intel Lab dataset, this algorithm performs similarly to traditional OGM on the F_1 metric.

Table 6.8: Average time to compute one MCMC map sample (K/L patches) for experiments on the MIT Stata dataset, where $L = 9$ and $K = 2.25 \times 10^6$. A border-based data-driven patch prior is more efficient because the border reduces the number of likely patch configurations in the prior.

	Data-driven prior [s]	Border-based, Data-driven prior [s]
Experiment 4	4158	1734
Experiment 5	557.2	201.2
Experiment 6	44.71	33.14

Furthermore, each map sample drew 2.25×10^6 cells and did not omit cells that were not used to compute the F_1 score. Each sampling algorithm could be sped up by only sampling mapped cells or cells in close proximity to mapped cells as shown in Figure 6.16. Table 6.6 shows that the border-based patch

Table 6.8 summarizes the computation time for the MCMC algorithms. These results were computed using a laptop computer running Matlab on Ubuntu with 32GB of RAM and a 2.70GHz Intel i7-6820HQ CPU. Similar to Table 6.6, these results are from sampling 2.25×10^6 cells and did not omit cells that were omitted in computing the F_1 score. Each sampling algorithm could be sped up by only sampling mapped cells or cells in close proximity to mapped cells as shown in Figure 6.21. Table 6.8 highlights that the border-based prior yields a significant computational savings and Figure 6.22 shows that it also yields superior performance. We therefore recommend this algorithm over other OGM algorithms, particularly in sparsely mapped areas. The MCMC methods are considered *anytime* as they continue to improve their estimate with additional computation to draw more samples from the posterior. Furthermore, the MCMC algorithms enable OGM to estimate more than marginal cell probabilities, such as the joint probability of several cells.

6.3 Chapter Summary

In this chapter we showed the benefit of several novel OGM techniques developed in this thesis on both simulated and hardware data. Using 100 simulated datasets we performed an in-depth analysis of many of these novel algorithms. Section 6.1.1 showed that the novel offline patch map algorithm is able to be computed for realistic maps, where the full Bayesian solution would be computationally intractable. Section 6.1.2 then used

the offline patch map as a basis to compare the optimized update terms developed in Section 2.5. Therefore, we showed that OGM with the optimized update terms can be computed online and is better able to estimate the offline patch map, as compared to traditional OGM.

Section 6.1.3 showed that the MCMC Gibbs sampling algorithm is able to draw samples from the full posterior without making a cell independence assumption between cells. Therefore, this novel anytime algorithm is able to estimate cell marginals as well as other statistics about the full posterior. Given more computation time, the algorithm will continue to draw more samples from the posterior and improve its estimate. Section 6.1.4 highlighted that the MCMC algorithm is able to use the novel data-driven patch prior and sample patches from the posterior. This algorithm is shown to perform so well on the simulated data that the results are nearly indistinguishable from the ground truth map.

Section 6.2 tests the MCMC Gibbs sampling algorithm with a data-driven patch prior on real hardware data. Section 6.2.1 shows the results on the Intel Lab dataset and Section 6.2.2 on the MIT Stata Center dataset. In both cases, the hardware data is subsampled to highlight that the MCMC algorithm outperforms others when the measurement data is sparse. In particular, we show the benefit of a border-based data-driven patch prior for the MCMC algorithm and demonstrate that it is able to provide the best estimate of the map for sparse measurements. In addition, this algorithm is faster to compute than other variants of the MCMC OGM algorithm. As robots continue to move faster and sensors measure longer distances, the measurement data will become increasingly sparse and thus this algorithm will continue to perform well. Furthermore, as computers increase in speed or as offboard computing is more readily available, the MCMC algorithm will benefit by drawing more samples and producing a more accurate map estimate. The hardware results show that the novel algorithms developed in this thesis are better able to estimate the map and that this result is more pronounced on sparse datasets.

Chapter 7

Conclusion

This thesis has provided further insight into occupancy grid mapping (OGM), which was first presented in 1985 and has been a dominant mapping technique used in robotics since that time. OGM is particularly useful as it estimates the unoccupied space as well as the occupied space and is therefore able to distinguish between unknown and unoccupied areas. We used OGM in our viewpoint-planning work, which we published in Merali et al. (2012), and that research motivated the in-depth analysis of cell correlations in OGM that we present in this thesis.

In Chapter 2, we presented the OGM framework and introduced the notion of a full Bayesian solution for OGM, which all other OGM techniques seek to approximate because it is computationally intractable for realistic map sizes. We first presented this notion in Merali and Barfoot (2012), but extend the idea in Section 2.3.1 to efficiently compute the full Bayesian solution for any 1D map where the robot’s pose is constrained. Section 2.5 then describes a method of variational inference to optimize traditional OGM using the full Bayesian solution in 1D; this work was published in Merali and Barfoot (2014). The result is an algorithm that runs online for any size map as fast as traditional OGM, but better captures the residual uncertainty in the map. Chapter 2 concludes by highlighting the mutual information between cells in the full Bayesian solution that is lost due to the cell independence assumption in traditional OGM. However, we show that the mutual information is highest between neighbouring cells and quickly approaches zero as the cells are farther apart.

Chapter 3 exploits the learning that cell correlations tend to be local by introducing the idea of estimating the occupancy probability of a *patch* of correlated cells as opposed to each cell independently. We first introduced the patch map idea in Merali and Barfoot

(2012), but specifically the offline patch map algorithm presented in Section 3.2. In Section 3.1, we presented a novel algorithm, known as the online patch map algorithm, that extends traditional OGM to estimating patches of correlated cells as opposed to each cell independently. Both patch map algorithms are able to better capture the residual uncertainty in the map, compared to traditional OGM, and especially between cells in the same patch. However, these patch map algorithms suffer from asymmetric cell correlations at patch boundaries.

The issue of patch boundaries was resolved in Chapter 4 by introducing a Markov Chain Monte Carlo (MCMC) Gibbs Sampling algorithm that samples from the full posterior. We first introduced and published this novel MCMC OGM algorithm in Merali and Barfoot (2013) and we generalized it in Section 4.2 to sample patches of cells as opposed to individual cells. This sampling algorithm does not suffer from patch boundaries as patch boundaries are not predefined and any patch of cells can be sampled. Furthermore, the MCMC algorithm does not require a reference map as the offline patch map algorithm does and the MCMC algorithm continues to improve its estimate as more samples are drawn from the posterior. We also showed that the MCMC samples can be used to compute statistics other than cell or patch occupancy probability. For example, the MCMC samples can be used to compute the first-occupied-cell along a measurement ray without making an independence assumption between cells or patches of cells along the measurement ray.

Chapter 5 explores the notion of introducing cell correlation in the prior. Section 5.2.1 explores correlating cells to their neighbouring cells, and Section 5.2.2 explores using a data-driven patch prior. The patch prior (as opposed to a cell prior) is able to exploit structure in the data, such as common patch configurations. Subsequently, Section 5.2.3 combines the two ideas by coupling a data-driven patch prior with a dependence on cells that border the patch. Exploiting this dependence results in further computational efficiency and yields maps that more closely represent the true map from fewer measurements. The work in Chapter 5 is being prepared for publication.

Finally, the algorithms presented in this thesis are analyzed on simulated and hardware datasets in Chapter 6. Specifically, 100 simulated datasets across 10 different maps are analyzed to highlight the benefits of the various algorithms on larger 2D maps. The results reinforce the 1D and simple 2D results presented throughout this thesis. The algorithms are then tested on two hardware datasets and the results are shown on increasingly sparse data to highlight the patch prior’s ability to accurately estimate the

occupancy of cells with few or even no measurements. Chapter 6 also highlights that the MCMC algorithm with a border-based prior is more computationally efficient than other MCMC OGM algorithms.

In conclusion, the optimized Δ function developed in Section 2.5 can better capture the residual uncertainty in the map at no additional computational cost, compared to traditional OGM. The concept of a *patch* of cells is useful because most cell correlations are local – these correlations may come from the prior or the measurements. The MCMC OGM algorithm is a useful method to draw samples from the full posterior and is therefore able to correlate all cells. Furthermore, this method is able to compute statistics beyond the marginal cell occupancy that is commonly used in OGM. The MCMC algorithm benefits further by using a border-based data-driven patch prior that is able to estimate the underlying map with very sparse measurements of the environment. This *anytime* algorithm can start with a map developed using traditional OGM, then continue to improve the map estimate as it draws more samples. Chapter 6 showed that this algorithm is able to produce a map that is nearly indistinguishable from the ground truth map with enough samples.

7.1 Review of Novel Contributions

In summary, this thesis makes the following novel contributions:

- Section 2.3.1 presents the pose-constrained full Bayesian solution for OGM that can be used to evaluate other OGM algorithms in 1D.
- Section 2.5 optimizes the inverse sensor model for traditional OGM, published in Merali and Barfoot (2014).
- Section 3.1 generalizes traditional OGM to an online patch map algorithm that captures mutual information between cells in the patch.
- Section 3.2 presents an offline patch map algorithm that better captures the residual uncertainty in the map and can be used as a benchmark for other OGM algorithms in 2D; published in Merali and Barfoot (2012).
- Chapter 4 introduces a MCMC Gibbs sampling method to provide an *anytime* algorithm for OGM that captures cell correlations; published in Merali and Barfoot (2013).

- Chapter 5 applies a data-driven patch prior to OGM that takes advantage of the sparsity in the patch prior (being prepared for publication).
- Section 5.2.3 extends the data-driven patch prior to include neighbouring cells in the prior, resulting in more sparsity and higher-fidelity maps (being prepared for publication).

Finally, although not discussed in detail in this thesis, the study into the benefits of cell correlations in OGM was motivated by our earlier research on next-best viewpoint-planning using occupancy grids. This early research used traditional OGM techniques to estimate the occupancy probability of each cell in the map. We then used the OG to weigh potential viewpoints based on the expected information gain in the map from each candidate viewpoint. Some of the results of this research were published in Merali et al. (2012).

7.2 Future Work

This thesis has highlighted the benefits of keeping cell correlations in OGM. However, there remain several open problems and extensions of this work. For example, Section 2.6 briefly discussed different sensor models, but this thesis focused on a narrow-beam range sensor with Gaussian noise. The research presented in this thesis can be extended by exploring various sensors and sensor models, including those with a wider sensing frustum. The pattern-matching sensor model may be useful in this regard. The remainder of our recommendations fall into four categories: improving the data-driven prior, applying the data-driven prior to other algorithms, improving the MCMC algorithm, and new applications of this research.

The data-driven prior presented in Chapter 5 can be improved in a number of ways. For instance, the prior samples drawn were 5×5 cells, which allows the MCMC algorithm to sample 3×3 patches if the full 16 cell border is used. Larger patch sizes may improve the result at the cost of more computation. Furthermore, considering fewer or more border cells around the patch may result in faster computation or more accuracy, respectively. We suspect that fewer than 16 border cells (e.g., four corner cells) for a 3×3 patch will yield a similar result with less storage cost. This thesis only considered square patches and there may be merit in exploring other shapes and sizes. For example, using a patch that is the same shape and size of a robot, target, or sensing frustum may be useful. The

data-driven prior should also be extended to 3D, where we suspect further structure in the 3D environment will yield additional computational savings. For example, occupied cells are likely to cluster near the lower portion of 3D patch priors due to gravity and the patch symmetry observed in 2D may not exist in 3D.

In addition to changing the shape and size of the patch prior, the effect of cell resolution requires further investigation for both the prior and map. Furthermore, the samples themselves can be drawn from more diverse prior maps to include a diverse set of patch priors. Alternatively, they can be drawn from more specific prior maps that better represent the application. For example, a warehouse robot may be equipped with map and patch priors from other warehouse environments as opposed to outdoor environments and vice versa. Section 5.2.1 explored higher-order MRFs for OGM, but assumed constant edge weights between neighbouring cells. However, having the measurements affect the edge weights may result in more accurate maps. Specifically, range sensors are able to strengthen the connection between many cells that are observed as unoccupied and decrease the correlation between the cell(s) that reflects the measurement and the cells before it.

The data-driven patch prior presented in Chapter 5 can be applied to OGM algorithms other than the MCMC algorithm. This prior can be applied to the MAP solution presented in Section 2.2 to better estimate the most likely OG, given the data. The data-driven prior would also result in cells past the wall to be estimated as occupied, which was assumed when using the MAP solution for the offline patch map algorithm in Section 3.2. Furthermore, this prior can be applied directly to the patch map algorithms presented in Chapter 3. The online patch map algorithm may be able to compute more accurate OG maps than traditional OGM at a similar computation cost. Further, applying this prior to the offline patch map may also yield a more accurate result and improve the computation speed as we observed in Section 5.2 that the majority of patch configurations are not observed in the data-driven prior. In addition, the patch map algorithms could be extended to shapes other than squares. For example, patches could be linear arrays of cells if cell correlations are seen to be directional in some environments.

In developing the MCMC algorithms for OGM, we observed several key factors that could improve the algorithm. Focusing the MCMC samples to be drawn from areas with the most uncertainty (typically near walls), is one way to improve the efficiency of the algorithm. In fact, Dhiman (2019) made a similar observation and suggests that sampling cells which reflect the measurement four times more than other cells leads to

faster convergence. This implies sampling *unknown* and *unoccupied* areas less as they generally have less uncertainty. Therefore, the MCMC OGM algorithm may also benefit from being extended to a quadtree or octree map representation where cells near obstacles are further discretized and are therefore sampled more per unit volume. In addition, the MCMC algorithm can avoid drawing samples from areas of the map that are reasonably certain. This certainty may be influenced by the number of MCMC samples drawn or the number of measurements. Similar to the observation by Paskin and Thrun (2005), the algorithm may benefit from reducing samples drawn from areas of the map that are more certain. Furthermore, if areas of the map are being sampled disproportionately, it may make sense to only store new samples on each MCMC iteration to conserve storage space.

The MCMC mapping algorithm may be extended to be applied online by focusing the map updates to areas with new measurements. In addition, cooperative robots could be used to each compute a portion of the map using MCMC, each drawing samples from a different part of the map, then sharing their samples to obtain a global perspective of the map. MCMC algorithms parallelize well and thus the algorithms presented in this thesis would benefit from being implemented on a GPU for greater speed.

This thesis has focused on static environments, and the algorithms could be extended to non-static environments. For example, the MCMC algorithms could “forget” measurements that were taken in the distant past. Furthermore, the patch prior could be used to distinguish between the static environment (e.g., buildings, furniture, etc) and dynamic obstacles (e.g, humans, vehicles, etc). In addition, the maps resulting from this work can be used for semantic grid maps, which is an active field of research and commonly start with an OG map (Liu and von Wichert, 2014; Qi et al., 2020). Liu and von Wichert (2014) highlight that their semantic mapping algorithm has difficulty classifying under-mapped rooms as rooms. The results in Section 6.2 highlight that the algorithms presented in this thesis are especially useful for sparsely mapped environments. As mobile robots continue to move faster, grid resolution continues to increase, and range sensors map farther, mapping algorithms will increasingly deal with sparsely mapped environments. Finally, as the computational capabilities of robots continue to improve, the anytime algorithms presented in this thesis will provide better estimates of the otherwise intractable full Bayesian solution for occupancy grid mapping.

Bibliography

- Agha-Mohammadi, A.-A., Heiden, E., Hausman, K., and Sukhatme, G. (2019). Confidence-rich grid mapping. *The International Journal of Robotics Research*, pages 1–23. (ref. pages 9 and 42)
- Andriamahefa, T. R. (2017). *Integer Occupancy Grids: a probabilistic multi-sensor fusion framework for embedded perception*. PhD thesis, Université Grenoble Alpes. (ref. page 8)
- Anguelov, D., Koller, D., Parker, E., and Thrun, S. (2004). Detecting and modeling doors with mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. (ref. page 9)
- Arak, T., Clifford, P., and Surgailis, D. (1993). Point-based polygonal models for random graphs. *Advances in Applied Probability*, 25(2):348–372. (ref. page 11)
- Arak, T. and Surgailis, D. (1989). Markov fields with polygonal realizations. *Probability Theory and Related Fields*, 80:543–579. (ref. page 11)
- Arak, T. V. (1982). On Markovian random fields with a finite number of values. In *4th USSR-Japan Symposium on Probability Theory and Mathematical Statistics*. (ref. page 11)
- Arbuckle, D., Howard, A., and Mataric, M. (2002). Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 409–414. (ref. page 35)
- Avidbots (2020). Neo. <https://www.avidbots.com>. Accessed: 2020-04-10. (ref. page 1)
- Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press. (ref. page 60)
- Bauer, D., Kuhnert, L., and Eckstein, L. (2019). Deep, spatially coherent occupancy maps based on radar measurements. *ArXiv*, abs/1903.12467. (ref. pages 8 and 49)
- Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48(3):259–302. (ref. page 28)

- Biber, P. and Strasser, W. (2003). The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, volume 3, pages 2743–2748 vol.3. (ref. page 10)
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. (ref. page 60)
- Borenstein, J. and Koren, Y. (1991). The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Journal of Robotics and Automation*, 7(3):278–288. (ref. pages 32, 34, 35, 37, 38, 39, 102, and 103)
- Broyden, C. (1970). The convergence of a class of double-rank minimization algorithms. *Journal Inst. Math. Applic.*, 6:76–90. (ref. page 36)
- Cain, C. and Leonessa, A. (2016). FastSLAM using compressed occupancy grids. *Sensors Technologies and Methods for Perception Systems in Intelligent Vehicles*. (ref. page 8)
- Cardinal, D. (2020). The future of sensors for self-driving cars: All roads, all conditions. <https://www.extremetech.com/computing/305691-the-future-of-sensors-for-self-driving-cars-all-roads-all-conditions>. Accessed: 2020-04-10. (ref. page 3)
- Chatila, R. and Laumond, J. (1985). Position referencing and consistent world modeling for mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 138–145. (ref. page 10)
- Collins, T., Collins, J., and Ryan, C. (2007). Occupancy grid mapping: An empirical evaluation. In *Control & Automation, 2007. MED '07. Mediterranean Conference on*, pages 1–6. (ref. page 24)
- Danescu, R., Oniga, F., and Nedeveschi, S. (2011). Modeling and tracking the driving environment with a particle-based occupancy grid. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1331–1342. (ref. page 117)
- Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. *International conference on Machine learning (ICML)*. (ref. pages 15, 98, and 108)
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38. (ref. page 27)
- Dhiman, V. (2019). *Towards Better Navigation: Optimizing Algorithms for Mapping, Localization and Planning*. PhD thesis, University of Michigan, Horace H. Rackham School of Graduate Studies. (ref. pages 9, 32, 42, 43, 59, 114, and 140)

- Dhiman, V., Kundu, A., Dellaert, F., and Corso, J. J. (2014). Modern MAP inference methods for accurate and fast occupancy grid mapping on higher order factor graphs. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2037–2044. (ref. page 59)
- Dragiev, S., Toussaint, M., and Gienger, M. (2011). Gaussian process implicit surfaces for shape estimation and grasping. In *2011 IEEE International Conference on Robotics and Automation*, pages 2845–2850. (ref. page 12)
- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110. (ref. page 15)
- Elfes, A. (1987). Sonar-based real-world mapping and navigation. *Robotics and Automation, IEEE Journal of*, 3(3):249–265. (ref. page 8)
- Elfes, A. (1989a). *Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation*. PhD thesis, Carnegie Mellon University. (ref. pages 8, 9, 14, 31, 35, 38, and 103)
- Elfes, A. (1989b). Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 22(6):46–57. (ref. pages 8 and 79)
- Fallon, M., Johannsson, H., Kaess, M., and Leonard, J. J. (2013). The mit stata center dataset. *The International Journal of Robotics Research*, 32(14):1695–1699. (ref. pages 114, 127, and 128)
- Finkel, R. A. and Bentley, J. L. (1974). Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9. (ref. page 12)
- Fletcher, R. (1970). A new approach to variable metric algorithms. *Computer Journal*, 13:317–322. (ref. page 36)
- Fox, D., Burgard, W., and Thrun, S. (1998). Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207. (ref. page 8)
- Friedman, S., Pasula, H., and Fox, D. (2007). Voronoi random fields: Extracting topological structure of indoor environments via place labeling. In *IJCAI*, volume 7, pages 2109–2114. (ref. page 5)
- Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6:721–741. (ref. pages 59 and 60)
- Georgiou, C., Anderson, S., and Dodd, T. (2015). Constructing contextual slam priors using architectural drawings. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 50–56. (ref. page 73)

- Georgiou, C., Anderson, S., and Dodd, T. (2017). Constructing informative bayesian map priors: A multi-objective optimisation approach applied to indoor occupancy grid mapping. *The International Journal of Robotics Research*, 36(3):274–291. (ref. pages 15, 73, and 98)
- Gerkey, B., Vaughan, R. T., and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *11th International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal. (ref. page 98)
- Goldfarb, D. (1970). A family of variable metric updates derived by variational means. *Mathematics of Computing*, 24:23–26. (ref. page 36)
- Grewe, R., Komar, M., Hohm, A., Lueke, S., and Winner, H. (2012). Evaluation method and results for the accuracy of an automotive occupancy grid. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, pages 19–24. (ref. page 97)
- Guizilini, V. and Ramos, F. (2018). Learning to reconstruct 3d structures for occupancy mapping from depth and color information. *The International Journal of Robotics Research*, 37(13-14):1595–1609. (ref. page 13)
- Guizilini, V., Senanayake, R., and Ramos, F. (2019). Dynamic hilbert maps: Real-time occupancy predictions in changing environments. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4091–4097. (ref. page 13)
- Hähnel, D. (2004). *Mapping with Mobile Robots*. PhD thesis, University of Freiburg, Department of Computer Science. (ref. pages 32, 33, 35, 40, 102, 103, and 104)
- Hastings, W. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109. (ref. pages 59 and 60)
- Hoermann, S., Bach, M., and Dietmayer, K. (2018). Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2056–2063. IEEE. (ref. page 15)
- Homm, F., Kaempchen, N., Ota, J., and Burschka, D. (2010). Efficient occupancy grid computation on the gpu with lidar and radar for road boundary detection. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1006 –1013. (ref. page 8)
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, (34):189–206. (ref. pages 8 and 9)
- Hou, L., Chen, X., Lan, K., Rasmussen, R., and Roberts, J. (2019). Volumetric next best view by 3D Occupancy Mapping using Markov Chain Gibbs Sampler for precise manufacturing. *IEEE Access*, 7:121949–121960. (ref. pages 6, 9, 13, 42, and 59)

- Howard, A. and Roy, N. (2003). The robotics data set repository (radish). (ref. pages 114 and 118)
- Jadidi, M. G., Miro, J. V., and Dissanayake, G. (2018). Gaussian processes autonomous mapping and exploration for range-sensing mobile robots. *Autonomous Robots*, (42):273–290. (ref. page 12)
- Jang, B.-G., Choi, T.-Y., and Lee, J.-J. (2006). Adaptive occupancy grid mapping with clusters. *Artificial Life and Robotics*, 10(2):162–165. (ref. page 15)
- Joubert, D., Brink, W., and Herbst, B. (2015). Pose uncertainty in occupancy grids through monte carlo integration. *Journal of Intelligent & Robotic Systems*, (77):5–16. (ref. page 8)
- Kaufman, E., Lee, T., Ai, Z., and Moskowitz, I. (2016). Bayesian occupancy grid mapping via an exact inverse sensor model. In *American Control Conference (ACC)*, pages 5709–5715, Boston, MA, USA. (ref. pages 31 and 34)
- Kaufman, E., Takami, K., Ai, Z., and Lee, T. (2019). Bayesian mapping-based autonomous exploration and patrol of 3d structured indoor environments with multiple flying robots. *Journal of Intelligent & Robotic Systems*. (ref. pages 13 and 19)
- Kaufman, E., Takami, K., Lee, T., and Ai, Z. (2018). Autonomous exploration with exact inverse sensor models. *Journal of Intelligent & Robotic Systems*, 92(3):435–452. (ref. pages 9 and 13)
- Kluszczynski, R., van Lieshout, M. N. M., and Schreiber, T. (2007). Image segmentation by polygonal markov fields. *Annals of the Institute of Statistical Mathematics*, 59:465–486. (ref. page 11)
- Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press. (ref. page 80)
- Kostavelis, I. and Gasteratos, A. (2015). Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86 – 103. (ref. page 5)
- Kullback, S. (1959). *Information Theory and Statistics*. John Wiley, New York, NY. (ref. page 36)
- Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86. (ref. page 36)
- Lang, T., Plagemann, C., and Burgard, W. (2007). Adaptive non-stationary kernel regression for terrain modelling. In *In Proc. of the Robotics: Science and Systems Conference (RSS)*. (ref. page 11)
- Lee, B., Zhang, C., Huang, Z., and Lee, D. D. (2019). Online continuous mapping using gaussian process implicit surfaces. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6884–6890. (ref. page 12)

- Li, H., Tsukada, M., Nashashibi, F., and Parent, M. (2014). Multivehicle cooperative local mapping: A methodology based on occupancy grid map merging. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2089–2100. (ref. page 15)
- Li, M., Feng, Z., Stolz, M., Kunert, M., Henze, R., and Kucukay, F. (2019). High resolution radar-based occupancy grid mapping and free space detection. In *4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018)*, pages 70–81. Science and Technology Publications. (ref. page 8)
- Li, Y. and Ruichek, Y. (2013). Building variable resolution occupancy grid map from stereoscopic system – a quadtree based approach. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 744–749. (ref. page 8)
- Li, Y. and Ruichek, Y. (2014). Occupancy grid mapping in urban environments from a moving on-board stereo-vision system. *Sensors*. (ref. page 15)
- Limketkai, B., Biswas, R., and Thrun, S. (2003). Learning occupancy grids of non-stationary objects with mobile robots. In Siciliano, B. and Dario, P., editors, *Experimental Robotics VIII*, volume 5 of *Springer Tracts in Advanced Robotics*, pages 222–231. Springer. (ref. page 8)
- Liu, Z. and von Wichert, G. (2014). Extracting semantic indoor maps from occupancy grids. *Robotics and Autonomous Systems*, 62(5):663–674. Special Issue Semantic Perception, Mapping and Exploration. (ref. pages 59 and 141)
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110. (ref. page 5)
- Luo, Z., Mohrenschildt, M. V., and Habibi, S. (2019). A probability occupancy grid based approach for real-time lidar ground segmentation. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13. (ref. page 15)
- Makarenko, A., Williams, S., Bourgault, F., and Durrant-Whyte, H. (2002). An experiment in integrated exploration. *IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS)*, 1:534–539. (ref. page 9)
- Marshall, J. and Barfoot, T. (2007). Design and field testing of an autonomous underground tramping system. In *Int. Conf. on Field and Service Robotics (FSR)*. (ref. pages 35, 78, 102, and 103)
- Martin, M. C. and Moravec, H. P. (1996). Robot evidence grids. Technical Report CMU-RI-TR-96-06, Carnegie Mellon University Robotics Institute, Pittsburgh, PA. (ref. pages 4, 8, and 32)
- Mazuran, M., Diego Tipaldi, G., Spinello, L., Burgard, W., and Stachniss, C. (2014). A statistical measure for map consistency in slam. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3650–3655. (ref. page 15)

- Meagher, D. (1982). Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147. (ref. page 12)
- Merali, R. S. and Barfoot, T. D. (2012). Patch map: A benchmark for occupancy grid algorithm evaluation. *IEEE/RSJ International Conference on Intelligent Robots and System (IROS)*, pages 3481–3488. (ref. pages 17, 20, 36, 47, 52, 54, 55, 85, 87, 98, 136, and 138)
- Merali, R. S. and Barfoot, T. D. (2013). Occupancy grid mapping with Markov Chain Monte Carlo Gibbs sampling. *International Conference on Robotics and Automation (ICRA)*, pages 3168–3174. (ref. pages 19, 20, 36, 59, 63, 68, 85, 98, 137, and 138)
- Merali, R. S. and Barfoot, T. D. (2014). Optimizing online occupancy grid mapping to capture the residual uncertainty. *IEEE International Conference on Robotics and Automation (ICRA)*. (ref. pages 17, 20, 34, 36, 41, 85, 98, 136, and 138)
- Merali, R. S., Tong, C. H., Gammell, J., Bakambu, J., Dupuis, E., and Barfoot, T. D. (2012). 3d surface mapping using a semi-autonomous rover: A planetary analog field experiment. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Turin, Italy. (ref. pages 9, 13, 16, 136, and 139)
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092. (ref. pages 59 and 60)
- Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341. (ref. pages 59 and 60)
- Meyer-Delius, D., Beinhofer, M., and Burgard, W. (2012). Occupancy grid models for robot mapping in changing environments. In *AAAI Conference on Artificial Intelligence*, volume 26. (ref. page 15)
- Milstein, A. (2008). *Improved Particle Filter Based Localization and Mapping Techniques*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada. Computer Science. (ref. page 73)
- Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *International joint conference on Artificial intelligence*, (18):1151–1156. (ref. page 73)
- Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74. (ref. pages 8, 31, 34, 35, 37, 39, 41, 102, and 103)
- Moravec, H. and Blackwell, M. (1992). Learning sensor models for evidence grids. In *CMU Robotics Institute 1991 Annual Research Review*, pages 8–15. (ref. page 14)
- Moravec, H. and Cho, D. W. (1989). A bayesian method for certainty grids. In *AAAI 1989 Spring Symposium Series, Symposium on Mobile Robots*. (ref. page 14)

- Moravec, H. and Elfes, A. E. (1985). High resolution maps from wide angle sonar. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 116–121. (ref. pages 4, 8, 22, 31, 97, and 136)
- Muffert, M. (2018). *Incremental Map Building with Markov Random Fields and its Evaluation*. PhD thesis, University of Bonn. (ref. pages 8, 9, and 79)
- Murray, D. and Little, J. J. (2000). Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8(2):161–171. (ref. pages 32 and 34)
- Nagla, K., Uddin, M., and Singh, D. (2012). Improved occupancy grid mapping in specular environment. *Robotics and Autonomous Systems*, 60(10):1245 – 1252. (ref. pages 22 and 108)
- NASA (2020). Perseverance. <https://mars.nasa.gov/mars2020/>. Accessed: 2020-04-10. (ref. page 2)
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In Brooks, S., Gelman, A., Jones, G., and Meng, X.-L., editors, *Handbook of Markov Chain Monte Carlo*, chapter 5, pages 113–162. Chapman and Hall/CRC. (ref. page 59)
- O’Callaghan, S., Ramos, F., and Durrant-Whyte, H. (2009). Contextual occupancy maps using gaussian processes. In *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*, pages 1054 –1060. (ref. page 11)
- O’Callaghan, S. T. and Ramos, F. T. (2012). Gaussian process occupancy maps. *I. J. Robotic Res.*, 31(1):42–62. (ref. pages 11 and 12)
- Oh, S. and Kang, H. (2016). Fast occupancy grid filtering using grid cell clusters from LIDAR and stereo vision sensor data. *IEEE Sensors Journal*, 16(19):7258–7266. (ref. page 15)
- O’Meadhra, C., Tabib, W., and Michael, N. (2019). Variable resolution occupancy mapping using Gaussian mixture models. *IEEE Robotics and Automation Letters*, 4(2):2015–2022. (ref. pages 10 and 115)
- OTTO Motors (2020). OTTO 220. <https://ottomotors.com>. Accessed: 2020-04-10. (ref. page 1)
- Paskin, M. A. and Thrun, S. (2005). Robotic mapping with polygonal random fields. In Bacchus, F. and Jaakkola, T., editors, *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*. AUAU Press, Arlington, Virginia. (ref. pages 11, 64, and 141)
- Pathak, K., Birk, A., Poppinga, J., and Schwertfeger, S. (2007). 3d forward sensor modeling and application to occupancy grid based sensor fusion. In *IROS*, pages 2059–2064. IEEE. (ref. pages 22 and 42)

- Peasley, B., Birchfield, S., Cunningham, A., and Dellaert, F. (2012). Accurate on-line 3d occupancy grids using manhattan world constraints. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5283–5290. IEEE. (ref. pages 8 and 73)
- Perrollaz, M., Yoder, J., Negre, A., Spalanzani, A., and Laugier, C. (2012). A visibility-based approach for occupancy grid computation in disparity space. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1383–1393. (ref. pages 8 and 22)
- Pfaff, P., Plagemann, C., and Burgard, W. (2007). Improved likelihood models for probabilistic localization based on range scans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2192–2197. (ref. page 8)
- Plagemann, C., Mischke, S., Prentice, S., Kersting, K., Roy, N., and Burgard, W. (2008). Learning predictive terrain models for legged robot locomotion. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3545–3552. (ref. page 11)
- Qi, X., Wang, W., Yuan, M., Wang, Y., Li, M., Xue, L., and Sun, Y. (2020). Building semantic grid maps for domestic robot navigation. *International Journal of Advanced Robotic Systems*, 17(1). (ref. page 141)
- Ramos, F. and Ott, L. (2016). Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730. (ref. pages 12 and 14)
- Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press. (ref. page 11)
- Roborace (2020). Robocar. <https://roborace.com>. Accessed: 2020-04-10. (ref. page 2)
- Saarinen, J., Andreasson, H., Stoyanov, T., Ala-Luhtala, J., and Lilienthal, A. J. (2013a). Normal distributions transform occupancy maps: Application to large-scale online 3d mapping. In *2013 IEEE International Conference on Robotics and Automation*, pages 2233–2238. (ref. page 10)
- Saarinen, J., Andreasson, H., Stoyanov, T., and Lilienthal, A. J. (2013b). Normal distributions transform monte-carlo localization (ndt-mcl). In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 382–389. (ref. page 10)
- Saarinen, J. P., Andreasson, H., Stoyanov, T., and Lilienthal, A. J. (2013). 3d normal distributions transform occupancy maps: An efficient representation for mapping in dynamic environments. *The International Journal of Robotics Research*, 32(14):1627–1644. (ref. page 10)
- Schaefer, A., Buscher, D., Luft, L., and Burgard, W. (2018a). A maximum likelihood approach to extract polylines from 2-D laser range scans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. (ref. pages 9 and 73)

- Schaefer, A., Luft, L., and Burgard, W. (2018b). DCT maps: Compact differentiable lidar maps based on the cosine transform. *IEEE Robotics and Automation Letters*, 3(2):1002–1009. (ref. pages 13 and 118)
- Schiavullo, R. (2018). Mathematical approach and detecting devices for automotive LIDAR. <https://twittertechnews.com/lidar-3d-scanning/mathematical-approach-and-detecting-devices-for-automotive-lidar/>. Accessed: 2020-04-10. (ref. page 4)
- Senanayake, R. and Ramos, F. (2017). Bayesian hilbert maps for dynamic continuous occupancy mapping. In *Conference on Robot Learning*, pages 458–471. (ref. page 13)
- Shankar, K. S. and Michael, N. (2020). MRFFMap: Online probabilistic 3D mapping using forward ray sensor models. In *Robotics: Science and Systems*. (ref. pages 8, 9, and 42)
- Shanno, D. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of Computing*, 24:647–656. (ref. page 36)
- Shrestha, R., Tian, F.-P., Feng, W., Tan, P., and Vaughan, R. (2019). Learned map prediction for enhanced mobile robot exploration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1197–1204. IEEE. (ref. page 9)
- Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2014). *Introduction to Autonomous Mobile Robots, Second Edition*. MIT Press. (ref. page 6)
- Sodhi, P., Ho, B.-J., and Kaess, M. (2019). Online and consistent occupancy grid mapping for planning in unknown environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7879–7886. IEEE. (ref. page 15)
- Song-Chun Zhu, Rong Zhang, and Zhuowen Tu (2000). Integrating bottom-up/top-down for object recognition by data driven markov chain monte carlo. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000*, volume 1, pages 738–745. (ref. page 59)
- Stachniss, C. (2006). *Exploration and Mapping with Mobile Robots*. PhD thesis, University of Freiburg, Department of Computer Science. (ref. pages 32, 34, 35, and 103)
- Stachniss, C. (2009). *Robotic Mapping and Exploration*. Springer Berlin Heidelberg. (ref. page 6)
- Stachniss, C. and Burgard, W. (2003). Exploring unknown environments with mobile robots using coverage maps. *Int. Conf. on Artificial Intelligence (IJCAI)*, 1:1127–1132. (ref. page 9)
- Stoyanov, T., Magnusson, M., and Lilienthal, A. J. (2013). Comparative evaluation of the consistency of three-dimensional spatial representations used in autonomous robot navigation. *Journal of Field Robotics*, 30(2):216–236. (ref. page 10)

- Stoyanov, T., Saarinen, J., Andreasson, H., and Lilienthal, A. J. (2013). Normal distributions transform occupancy map fusion: Simultaneous mapping and tracking in large scale dynamic environments. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4702–4708. (ref. page 10)
- Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71. (ref. pages 14 and 34)
- Thrun, S. (2002). Robotic mapping: A survey. In Lakemeyer, G. and Nebel, B., editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann. (ref. page 14)
- Thrun, S. (2003). Learning occupancy grids with forward sensor models. *Autonomous Robots*, 15:111–127. (ref. pages 15, 22, 25, 27, 29, 42, 55, and 58)
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press. (ref. page 43)
- Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1):99 – 141. (ref. pages 2 and 8)
- Thrun, S., Haehnel, D., Ferguson, D., Montemerlo, M., Triebel, R., Burgard, W., Baker, C., Omohundro, Z., Thayer, S., and Whittaker, W. R. L. (2003). A system for volumetric robotic mapping of abandoned mines. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA '03)*, volume 3, pages 4270 – 4275. (ref. page 10)
- Tipaldi, G. D. (2009). *Looking Inside for Mapping the Outside: Introspective Simultaneous Localization and Mapping*. PhD thesis, University of Rome La Sapienza. (ref. pages 4 and 7)
- Tipaldi, G. D., Meyer-Delius, D., Beinhofer, M., and Burgard, W. (2011). Simultaneous localization and dynamic state estimation in reconfigurable environments. (ref. page 15)
- Tong, C., Barfoot, T. D., and Dupuis, E. (2012). Three-dimensional SLAM for mapping planetary worksite environments. *Journal of Field Robotics*, special issue on “Space Robotics”, 29(3):381–412. (ref. pages 2 and 98)
- Vasto, D. M.-D. D. (2011). *Probabilistic Modeling of Dynamic Environments for Mobile Robots*. PhD thesis, Technische Fakultat Albert-Ludwigs-Universität Freiburg im Breisgau. (ref. page 32)
- Werber, K., Rapp, M., Klappstein, J., Hahn, M., Dickmann, J., Dietmayer, K., and Waldschmidt, C. (2015). Automotive radar gridmap representations. (ref. page 8)
- Weston, R., Cen, S., Newman, P., and Posner, I. (2019). Probably unknown: Deep inverse sensor modelling radar. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5446–5452. (ref. page 8)

- Wurm, K. M., Hennes, D., Holz, D., Rusu, R. B., Stachniss, C., Konolige, K., and Burgard, W. (2011). Hierarchies of octrees for efficient 3d mapping. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA. (ref. pages 7 and 8)
- Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., and Burgard, W. (2010). OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA. (ref. pages 7, 8, and 10)
- Xu, L., Feng, C., Kamat, V. R., and Menassa, C. C. (2019). An occupancy grid mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments. *Automation in Construction*, 104:230–245. (ref. page 32)
- Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. *IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1997. CIRA '97.*, 1:146–151. (ref. pages 6 and 13)
- Yguel, M., Aycard, O., and Laugier, C. (2006). Efficient gpu-based construction of occupancy grids using several laser range-finders. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 105–110. IEEE. (ref. pages 8 and 82)
- Zhi, W., Ott, L., Senanayake, R., and Ramos, F. (2019). Continuous occupancy map fusion with fast bayesian hilbert maps. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4111–4117. (ref. page 13)
- Zhou, J., Ruan, X., Dong, P., and Zhang, J. (2018). Adaptive GridMap building algorithm for mobile robot based on multiway tree. *MATEC Web of Conferences*, 160:06001. (ref. page 8)
- Ziebart, B. D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., and Srinivasa, S. (2009). Planning-based prediction for pedestrians. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936. IEEE. (ref. page 15)