Multi-resolution hybrid mapping for autonomous robots

Leong, Chee Weng

2011

Leong, C. W. (2011). Multi-resolution hybrid mapping for autonomous robots. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/44882>

<https://doi.org/10.32657/10356/44882>

Multi-Resolution Hybrid Mapping for Autonomous Robots

LEONG CHEE WENG

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

2011

Abstract

This thesis examines the fundamentals of some common robotic mapping techniques that will be used in this research work, such as Hough Transformation line extraction technique, the Inverse Sensor Model which is used for occupancy grid mapping and EKF-SLAM which is used for localization. It will also present the concept of *HYbrid Metric Mapping* (HYMMs) technique which the proposed *Multi Resolution Hybrid Mapping* (MRHM) is based on.

This thesis goes on to discuss some technical improvements that can be made to existing robotic mapping techniques. Although the *Inverse Sensor Model* is able to model the environment accurately, there is a drawback which may mis-classify the occupancy status of the grid cells. The fundamentals of the *Inverse Sensor Model* will be examined and its main drawbacks are discussed. An alternative approach is then presented to provide a more accurate occupancy grid mapping algorithm. Also presented in this thesis a technique which is able to reduce the time needed to construct dense grid maps, the *Multi Resolution Hybrid Mapping*. The concept of MRHM, which uses lines as base landmarks to propagate corrections from the feature map to the grid map, is then presented. Finally this thesis presents the simulation and field experiment setups and the obtained results.

Acknowledgements

First and foremost I would like to thank my parents and my family for their unconditional love, generosity and support. Without them, none of the things I have achieved so far would have been possible.

I would next like to thank my initial supervisor, Associate Professors Martin Adams for giving me this opportunity to pursue this research, coaching and providing me with necessary information, reference books and articles to make this research as “painless” as possible. I would also like to extend my thanks to my current supervisor, Assistant Professor Su Rong for his patience, advice and guidance during the last phase of my research. Without his guidance, this research will not be possible. I would also like to thank Professor Wang Dan Wei for including me as a team member of Tech-X Challenge, Singapore. This competition had been a real eye-opener for me. Many thanks to my fellow team members as well for gifting me with many fond memories. I also wish to extend my appreciation to Mr. Tan Chai Soon and Ms. June Lee of the Autonomous Robotics Research Laboratory. Without their technical assistance and support, my research would not have been going smoothly.

To Dr. John Mullane, Sok Chhay, Akshay Rao for so many sensical and nonsensical, research and non-research related discussions. These discussions provide me with many a joke and help kept my sanity in check. Thanks to many other friends who help take my mind off this research, allowing me to have a breather.

Finally, to my fiancée, Ice, for her unconditional love and patience and for being the pillar of support. Encouraging me through all the ups and downs during my academic journey. Thank you.

Contents

Abstract	i
Acknowledgements	ii
Contents	iv
List of Figures	vi
1 Introduction	1
1.1 Motivation	1
1.2 Objective	4
1.3 Contributions	5
1.4 Organisation	6
2 Related Work	9
2.1 Literature Review	9
3 Detailed Review on Some Robotic Mapping Paradigms	17
3.1 Introduction	17
3.2 Extraction Of Hough Line For EKF-SLAM	20
3.2.1 Hough Transformation	20
3.2.2 Hough Lines For EKF-SLAM	22
3.2.3 Hough Transformation Error Analysis	25

3.3	EKF-SLAM Based On Hough-Line Features	31
3.3.1	The Prediction Stage	32
3.3.2	Update Stage	36
3.3.3	Adding New Features	39
3.4	Occupancy Grid Mapping	42
3.4.1	Occupancy Grid Definitions	43
3.4.2	Occupancy Grid Mapping Algorithm - The Inverse Sensor Model .	46
3.4.3	Multi Resolution Occupancy Grid Mapping	53
3.5	Fundamentals of HYbrid Metric Maps (HYMMs)	57
3.6	Conclusion	60
4	Technical Improvements of Robotic Mapping Paradigms	61
4.1	Introduction	61
4.2	Improved Inverse Sensor Model	62
4.2.1	Standard Inverse Sensor Model - The Drawback	62
4.2.2	Concept Of Improved Inverse Sensor Model	66
4.3	Multi-Resolution Hybrid Mapping	68
4.4	Conclusion	71
5	Simulations and Experiments - Setups and Results	72
5.1	Simulation Setups	72
5.2	Simulation Results	75
5.2.1	EKF-SLAM Simulation Results	75
5.2.2	Improved Inverse Sensor Model Simulation Results	79
5.2.3	Multi Resolution Occupancy Grid Mapping Simulation Results . . .	81
5.3	Experiment Setup	90
5.4	Experiment Results	91
5.4.1	EKF-SLAM Experiment Results	91

5.4.2	Multi Resolution Occupancy Grid Mapping Experiment Results . .	94
5.5	Multi Resolution Hybrid Mapping Setup and Results	99
5.6	Conclusion	104
6	Conclusion and Recommendations	105
6.1	Conclusion	105
6.2	Recommendations and Future Work	106
	Bibliography	108

List of Figures

2.1	Example of a feature map.	12
2.2	Example of an occupancy grid map.	12
3.1	Line extraction using Hough transform.	21
3.2	Relationship of a feature between the local vehicle frame and the global navigation frame.	23
3.3	Huge parametric errors derived when transforming features from local frame to global navigation frame due to vehicle position.	28
3.4	Vehicle Kinematics.	33
3.5	Content of the EKF predicted covariance matrix.	41
3.6	2-D occupancy grid area of interest	47
3.7	1-Dimensional occupancy grid map.	48
3.8	2-Dimensional occupancy grid map.	51
3.9	Zoom-in view of a 2-Dimensional occupancy grid map.	51
3.10	Comparison of low resolution and high resolution occupancy grid map. . .	54
3.11	Comparison of cells and area tessellated by different resolution grid maps. .	55
3.12	Definitions of grid cells used in MROGM.	57
3.13	Definition of relative features in point-feature based hybrid mapping. . . .	58
3.14	Definition of relative features in point-feature based hybrid mapping. . . .	59
4.1	Instance where standard Inverse Sensor Model do not work.	64

4.2	Zoom-in view on one instance where standard Inverse Sensor Model did not work.	64
4.3	Comparison of performance of standard Inverse Sensor Model.	65
4.4	Solution to overcome the drawback of standard <i>Inverse Sensor Model</i>	67
4.5	Definition of relative features in line-feature based hybrid mapping.	69
5.1	User interface of the simulation program	73
5.2	Window showing occupancy map constructed during simulation.	74
5.3	Line-features based EKF-SLAM simulation result.	76
5.4	Zoomed view of EKF-SLAM simulation starting and ending point.	77
5.5	Zoomed view of EKF-SLAM simulation starting and ending point.	77
5.6	Comparison of simulation positional error: x-direction.	78
5.7	Comparison of simulation positional error: y-direction.	78
5.8	Comparison of simulation positional error: error magnitude.	79
5.9	Comparison of grid maps constructed using the standard <i>Inverse Sensor Model</i> and the <i>Improved Inverse Sensor Model</i>	80
5.10	Overview of simulation environment.	82
5.11	Modeling of environment using low resolution grid map.	84
5.12	Modeling of environment using high resolution grid map.	85
5.13	Modeling of environment using multi resolution grid map.	86
5.14	Cumulative time required to construct occupancy grid maps using different mapping paradigms.	87
5.15	Situation where MROGM is the least efficient.	87
5.16	Low resolution grid mapping of simulation environment.	88
5.17	High resolution grid mapping of simulation environment.	89
5.18	Multi resolution grid mapping of simulation environment.	89
5.19	Cumulative time spent by various grid mapping paradigm during simulation.	90

5.20	View of carpark to the EKF-SLAM and MROGM algorithms.	91
5.21	GPS ground truth around MAE car park.	92
5.22	EKF-SLAM results for experiment in MAE car park.	93
5.23	Zoomed view of EKF-SLAM results for experiment in MAE car park. . . .	93
5.24	Close up view of EKF-SLAM results at starting / ending point.	94
5.25	Low resolution grid map representation of MAE car park.	96
5.26	High resolution grid map representation of MAE car park.	96
5.27	Multi resolution grid map representation of MAE car park.	97
5.28	Cumulative time required to construct MAE car park grid maps of different resolution.	97
5.29	Window panels causing discrepancy between grid map and actual environ- mental structure.	98
5.30	Window panels causing discrepancy between grid map and actual environ- mental structure.	98
5.31	<i>Multi Resolution Hybrid Mapping</i> simulation setup.	99
5.32	Propagation of MRHM correction from feature map to grid map through simple concatenation.	100
5.33	Grid map constructed without implementation of MRHM.	102
5.34	Grid map constructed with implementation of MRHM.	102
5.35	Comparison of grid maps constructed with and without implementation of MRHM.	103
5.36	Zoomed comparison of grid maps constructed with and without implemen- tation of MRHM.	103

Chapter 1

Introduction

This chapter describes the motivation behind this research as well as outlining the objectives and contributions. A detailed overview of this report is also be presented in this chapter.

1.1 Motivation

Autonomous robotics has been an area of active research for the past few decades, propelled by the increasing interest from educational institutes, military and industrial parties. One who is unfamiliar with autonomous robotics, may ask: “What is an autonomous robot? What is it capable of doing to generate such huge interest?”. An ideal autonomous robot possesses the necessary abilities to complete any given tasks without human intervention. To put it simply, an autonomous robot has the ability to make the necessary decisions under different situations in order to accomplish its given tasks, without humans having to provide instructions. Thus autonomous robots could replace humans in order to perform high risk missions or tasks which are beyond human

capabilities such as deep sea or space exploration for scientific research.

With the idea of replacing humans by machines to carry out dangerous missions, in 2004, the Defense Advanced Research Projects Agency (DARPA) of the US military organised the first DARPA Grand-Challenge [14]. In this challenge, the entry vehicles were required to autonomously navigate through 150 miles of interstate routes in the Mojave desert, however no team successfully completed the designated route. Similarly the 2005 Grand-Challenge [15] was also held in the Mojave desert but in this round, the vehicles were required to drive a 132 mile, off-road course with rugged terrain, tunnels and sharp turns. “Stanley” from Stanford University was the winner of this challenge with a total time of 6 hours 54 minutes. In the third challenge in 2007 known as the Urban-Challenge [16], vehicles were required to navigate an obstacle course in an urban setting, interacting with other traffic while obeying existing traffic rules. “Boss” from Carnegie Mellon University was the first vehicle to complete the entire course. Together with some other autonomous robotics competitions organised around the world, such as *RoboCup*, it is evident that robotic autonomy can soon be achieved.

In order to accomplish tasks such as those mentioned above, an autonomous robot needs to answer three basic question: “Where am I?”, “Where am I going to?” and “How do I get there?” [33]. The latter two questions can be solved by *navigation* and *path planning*, however they can only be solved *after* answering the first question, “Where am I?”. The answer to this fundamental question is *localisation*. Localisation, or position estimation, is to determine *where* one is with respect to the environment or initial position. This can be achieved either with or without using exteroceptive sensors.

Exteroceptive-free methods requires the use of proprioceptive sensors such as wheel encoders and/or inertial measurement units (IMUs) to measure the motion of the

robot and derive its current state with respect to its initial state [7][25][39]. But these estimated states suffer from errors, due to sensor noise and drifts, and these accumulated errors can quickly become unbounded over long periods of time. The Global Positioning System (GPS) is able to provide independent positional estimates of the robot's location by means of *trilateration* using GPS satellites and is therefore often incorporated into autonomous robots to bound the above mentioned errors [52]. However, GPS data is prone to signal outages and multi-path interference in urban environments due to tall trees and buildings and it cannot be used in indoor environments. Hence it is also not an entirely reliable solution. Furthermore, localisation through exteroceptive-free methods does not provide information that relates the robot to its environment.

The Simultaneous Localization and Mapping (SLAM) framework allows autonomous robots to concurrently build a map of the environment while localising their current position with respect to this map, something which an exteroceptive-free localisation technique is incapable of. The foundations of this framework were first conceived by Smith et al. [47]. Since then many researchers have based their work on SLAM, using different mapping technique such as feature maps [2][21][32][33], topological maps [5][8] and grid maps [18][19][43]. In recent years, many authors began combining different mapping paradigms, also known as *Hybrid Mapping*, such as topological maps with grid maps [37][51]. However, this approach is mainly for exploration, mapping and navigation and normally adopted in highly structured indoor environments, where topological nodes are used to represent large empty spaces in the environment. Another *Hybrid Mapping* methodology introduced recently combines feature maps with grid maps for localisation and mapping in large scale outdoor environments [31]. In this approach, point-feature maps were used to estimate the platform's position while multiple local-frame occupancy grid maps were used to model the environment based on the estimated vehicle position. In addition, the point-feature maps were also used to correct the positions of the grid

maps such that a more accurate dense map can be constructed. This approach makes use of each mapping paradigm's advantage to overcome each other's drawback to achieve an optimal localisation and mapping process.

1.2 Objective

Feature-based SLAM is capable of achieving reliable estimates of the robot's position with the help of the concurrently built map and this map only stores features which depend on the feature models being used. In a physical environment, the surroundings can only be accurately represented in a feature map with multiple feature models. However, the management of all the feature models needed to build an accurate map of the environment is intangible. Furthermore, maintaining all correlations between different types of features in the SLAM framework will require large amount of resources. Hence, in general practice only one feature model is used for concurrent localisation and map building [13][33][41][57]. In these cases, the maps built will not be able to represent the environment in details, as information that does not fit into the feature model would be excluded. For example, given a line-feature based map, as the name implies, only line-features will be stored in this map, data points which did not contribute to the classification of line-features, such as cylinders, arcs or points, will not be stored in the map. Similarly for a centroid-based point feature map where the center-of-mass of a cluster of points is used as a feature, objects such as walls or pillars may not be classified as features and therefore will not be included in the maps. In environments where there are different types of features, these feature maps will not be able to describe the surrounding accurately as the number of feature models used is limited. Therefore, occupancy grid maps are often used to build a map of the environment as they are capable of modeling

the surrounding with a high resolution, providing valuable information required for navigation and path-planning. The main drawback of occupancy grid mapping is the large amount of cells required to describe the environment which can be in the range of tens of thousands. Thus, building a high-resolution grid map may require significant computational effort when calculating the occupancy probability for every single cell in the map.

And as introduced earlier, the new hybrid mapping approach proposed by Nieto et. al [31] is especially useful when a situation requires the robot to localise itself with high efficiency while concurrently building a high resolution map of the environment [30]. However, in a highly structured environment such as an urban setting or an indoor environment, line features can be more easily extracted than point features. Hence, using line features as base landmarks will be more appropriate in such situations.

The objective of this thesis is to examine the current formulation of hybrid mapping as well as the fundamentals of robotic mapping so that amendments can be made with the aim of improving the overall efficiency and accuracy. It is assumed in this research work that only one feature model, the line-feature model, will be used to reduce the complexity of implementing the proposed solution.

1.3 Contributions

The key contributions of this thesis are as follows:

1. A solution is provided to overcome the main drawback of the standard *Inverse Sensor Model* grid mapping algorithm, which is unable to perform optimally when constructing low-resolution maps. Instead of solely relying on means of boundaries

to determine the occupancy status of these cells, this approach rounds off range finder data to the nearest grid cell's center-of-mass and determines which grid cells are occupied .

2. The formulation of Multi-Resolution Hybrid Mapping (MRHM) is presented in this thesis. In the original approach, point feature maps were used to update the platform's position and provide corrections to the grid map. The MRHM uses line feature maps instead, which can be easily extracted from highly structured environments. Furthermore, MRHM makes use of *Multi Resolution Occupancy Grid Mapping* (MROGM) which improves the efficiency when constructing high resolution grid maps.
3. The full EKF-SLAM, Multi-Resolution Occupancy Grid Mapping and Multi-Resolution Hybrid Mapping algorithms are implemented solely by the author except an existing subroutine which the EKF-SLAM calls on, the Hough lines extraction algorithm¹. These implementations are used to verify existing algorithms and to test the new MRHM algorithm. A simulation environment was also implemented to test the performance of the algorithms mentioned. This simulation allows users to define certain parameters such as vehicle speed, turn rate and the environment, to test the algorithms under different configurations.

1.4 Organisation

This chapter discussed the motivation and objectives of the thesis and briefly presented issues pertaining to current mapping techniques. The contributions of this

¹Hough Lines Extraction algorithm courtesy of Mr. Sok Chhay, NTU

thesis were also highlighted. The rest of the thesis is organised as follows:

Chapter 2 introduces the related work. A brief introduction to the Simultaneous Localization and Mapping (SLAM) framework is provided, as well as a review of how various authors applied this framework to their autonomous robotic research. A discussion on various types of mapping paradigms used to model the environment to aid localisation is also presented. These paradigms include feature maps and occupancy grid maps. Normally, feature-maps built via a SLAM framework use a single feature model only, as discussed earlier. As such, the resulting map will not be able to represent the environment in high detail. Grid maps are able to model the environment in high detail but the amount of computational resources needed can be large. Hence other researchers made use of each mapping paradigm's advantages and proposed a new hybrid mapping paradigm by using feature maps for position estimation and occupancy grid maps (based on the estimated position of the vehicle) to model the environment. However, the traditional approach to build occupancy grid maps still requires a large amount of computational resources, especially for large occupancy grid maps. In order to reduce the computational cost for building these grid maps, other researchers implemented quadtree/octree partitioning processes to build their occupancy grid maps. This chapter will end with a short introduction to the proposed *Multi-Resolution Hybrid Mapping* (MRHM) method, including a discussion on how MRHM draws on the advantages of hybrid mapping and quadtree/octree partitioning processes to create a more computationally efficient hybrid mapping approach.

Chapter 3 reviews the some of the current robotic mapping paradigms that will be used in the proposed MRHM. This includes a detailed discussion of the line extraction algorithm used, the *Hough Transformation*. Follow-on is the detailed formulation of the EKF-SLAM algorithm that will be used to provide recursive estimation of the platform's

position. Also included in this chapter is the presentation of the currently used grid mapping technique, the standard *Inverse Sensor Model*. It will also include discussions on the *Multi-Resolution Occupancy Grid Mapping*. Finally, this chapter will discuss the fundamentals of the *HYbrid Metric Mappings* (HYMMs).

Chapter 4 provides discussion on some of the technical improvements that can be made to enhance the overall system performance. The standard *Inverse Sensor Model* presented in Chapter 3 might not be able to appropriately classify the occupancy status of the low resolution grid cells. In this chapter, an alternative solution, the *Improved Inverse Sensor Model*, is proposed to classify the cell's occupancy status accurately. This approach is able to reduce the time needed to construct a multi resolution grid map when compared to the standard high resolution grid mapping approach. Considering that lines can be much more easily extracted as features in a highly structured environment (such as indoors), it is propose to use line features to replace base landmarks in the HYMMs, which together with the MROGM forms a new approach, the *Multi Resolution Hybrid Mapping* (MRHM).

Chapter 5 presents both the simulation and experimental results. This includes results obtained from the MROGM module, the *Improved Inverse Sensor Model* and the MRHM module. A discussion on the results obtained will also be presented.

Chapter 6 summarises and concludes the work presented in this thesis and provides a discussion on avenues of future work.

Chapter 2

Related Work

In this chapter, related work will be presented and discussed in details. Ideas and concepts will be drawn to formulate the general idea for the proposed work in this thesis.

2.1 Literature Review

For the development of autonomous robotics, accurate position estimation is the fundamental ability which the platform must possess as compared to other abilities, such as navigation and path planning, to achieve autonomy. Without accurate positional estimates, autonomous robots will not be able to navigate properly through the environment and will have great difficulty in reaching its intended goal. During the early stages of autonomous robotics research, attempts to solving the localization problem focussed on the the usage of proprioceptive sensors such as odometry sensors and/or Inertial Measurement Units (IMUs). These sensors measure the relative motion of the platform, in terms of velocities, accelerations and/or body rotations, and by means of integrating these data, the positions and the orientations of the platform can be obtained. This

is also commonly known as the *dead-reckoning* approach [56]. However the inherent uncertainties such as drifts and Gaussian noise which exist within these sensors causes unbounded errors in the estimated vehicle position and orientation, especially when integration is performed on these data over a period of time.

The *Global Positioning System* (GPS) is commonly used to bound the errors mention above [34], however this is not an optimal solution as GPS suffers from outage, especially in urban environments where there are tall buildings and trees which block the signal paths between the GPS antenna and the satellites in constellation. In addition, GPS signals are unavailable in an indoor environment. An autonomous robot may be required to work in both indoor and outdoor environments, if GPS signals are not available at all times to bound the errors, the platform's state information may drift and this will result in mission failure.

The *Simultaneous Localisation and Mapping* (SLAM) framework makes use of the information from the robotic platform's immediate surrounding to identify suitable features to build a map which can aid in refining the dead-reckoning based positional estimates. The SLAM framework has been around for more than two decades since it was first conceived by Smith et al. [47]. It has since became the focus of research for many authors for mobile land platforms [26][38][59] as well as airborne [36] and underwater platforms [44]. One of the more popular approaches to solve the SLAM problem is by incorporating the use of *Kalman Filter* (KF). This approach is able to provide recursive vehicle and map state estimation as well as uncertainty estimates in the vehicle and the stored feature locations. However the KF assumes linear dynamics and linear measurement models for both vehicle and landmark estimation, which is often unrealistic in realistic environments. To overcome this difficulty, *Extended Kalman Filter* (EKF) have been used instead [41]. The EKF takes the partial derivatives (the Jacobian)

of the predicted state to generate covariance estimates in non-linear systems.

As the name implies, SLAM is capable of estimating the mobile platform's position and building a map concurrently. For EKF-SLAM, similar to the standard KF-SLAM, there is a *prediction stage* and an *update stage*. The prediction stage is responsible for estimating the platform's state based on the state transition model which describes the platform's movement. The update stage is then responsible for refining this estimate by using exteroceptive information, usually obtained via range finding sensors or visual sensors. Landmarks or features can be extracted from this information and stored to build a map which the mobile platform will refer to for the sake of refining positional estimates [40][48]. Maps built using this approach are also known as feature maps. One of the main drawbacks of using feature-based maps is that these maps contain only objects which have met the feature criteria, such as the point features as indicated by blue circles in Figure 2.1(b). The rest of the metric information of the surrounding which does not fit into the feature model used will be excluded from this map. When comparing this feature-based map to the actual environment shown in Figure 2.1(a), where the yellow points are the raw laser range finder scans of the environment, it is obvious that a lot of environmental information is being excluded.

Another common approach to map building is by representing the environment with occupancy grid maps. A map of desired size will first be defined together with the grid cells, of desired resolution, which will tessellate the entire map. The cells in this occupancy grid map are normally initialised to be of unknown state (where the occupancy probability of the cell being empty or occupied are equal). As the vehicle moves through the environment, the occupancy probability value of these cells will be updated using the exteroceptive information. As these cells are being updated, they

¹Images courtesy of Dr. John Mullane, NTU

(a) Overview of Testing ground

(b) Example of a feature map representation

Figure 2.1: Example of a feature map. Feature maps are not able to model the environment in high detail and only areas being classified as features are represented in this map.¹

(a) Overview of Testing ground

(b) Example of an occupancy grid map representation

Figure 2.2: Example of an occupancy grid map. The obstacles in the environment are indicated in yellow and red while empty spaces are indicated in blue. It can be seen that this map is able to model the environment in great detail, clearly indicating where the obstacles are.¹

can be used to refine the predicted positional estimates of the vehicle [9][23]. Figure 2.2 shows the results of an occupancy grid map built using the same data set obtained from the experiment conducted in Figure 2.1. The occupancy probability value for each of cell in this high resolution grid map increases as the colours intensifies from blue to red. The green areas indicates spaces that have not been explored, hence the cells that tessellate these areas hold an equal probability of being occupied or empty. When comparing the two metric maps built (between the feature-based map to the occupancy grid map), it is clear that the occupancy grid map provides a richer representation of the environment than the feature map, indicating clearly where each of the objects (red and yellow areas) are and where the empty spaces (shaded in blue) are the in the map, making grid maps a more suitable choice for navigation and path-planning [18].

In a survey conducted by Chen [12], it was noted that although grid maps are simpler to construct and maintain, and most importantly able to model the environment in great detail, the computation time and resources needed to generate such a map can be huge. A feature-based map, although highly dependent on the environment for the successful extraction of features, and its incapability to model the environment in great detail, is more suitable for mobile platform's self-localisation. Acknowledging the limitations of each mapping paradigm, Guivant [30] and Nieto et al. [31] proposed a new approach to hybrid mapping. In conventional hybrid mapping, topological maps were built based on grid maps generated from previous experiments as an offline process to be deployed during the next experiment. The approach of topological maps is to represent an empty space with a node and these maps are usually meant for path planning and navigation [12]. The proposition of "HYbrid Metric Mapping" (HYMMs) by [31] is a new methodology for hybrid mapping which involves combining the feature-based mapping paradigm with occupancy grid mapping. As mentioned earlier, the main advantage of a feature map based localisation process is that this approach is *not* computationally

intensive yet able to produce accurate positional estimates. For an occupancy grid map, it is capable of modeling the environment in high detail. By combining the advantages of these two mapping techniques, highly accurate positional estimates can be obtained along with a highly detailed map of the surroundings without compromising the computational time, an issue normally associated with localisation based on occupancy grid mapping [54].

As mentioned earlier, an occupancy grid map may consist of a large amount of cells which tessellate the desired space. Attached to each cell is an occupancy probability value which defines the probability of an object (or part of an object) existing in the space that this cell represents. In order to produce an occupancy grid map, the occupancy probability value for every single cell defined in this map must be calculated. Although [31] was able to produce dense maps of the environment, their approach was based on limiting the area of cells required to have their occupancy probability calculated by setting up boundaries using SLAM features. However, if the number of cells is significantly large (due to high-resolution cells), the overall computational cost will still increase. Fairfield [42], whose work was based on 3-D SLAM, employed an octree data structure in his work to reduce the computational cost of copying and storing of maps for localisation and map building. This approach, is similar to quadtree mapping for 2 dimensional space. For quadtree mapping, each parent cell will be partitioned into four children cells if required [49]. These children cells can be further partitioned until the required resolution is met. For octree mapping, each cube is partitioned into eight higher resolution cubes (hence the name octree) if necessary and be further partitioned to a desired resolution. If a parent cell is considered to be occupied and it is partitioned into eight higher resolution cells, eight computational process will be needed to compute the occupancy probability value for each of these child-cells. However if a parent-cell is considered to be empty, no further partition will be needed and only one occupancy probability computational process will be required, thus saving time and memory resources.

Although quadtree and octree data structures are useful in reducing computational cost, if a high resolution map is required, it implies that iterative partitioning processes will be needed on the required set of parent-cells. This process of iterative partitioning on multiple cells may cause the map management to eventually become intangible. Other than map management issues, online cell partitioning may use up a lot of resources as well. It is unknown how many partitioning processes will be required on a single parent-cell before obtaining the desired child-cell size. If multiple iterative partitioning processes were required (on a huge set of parent-cells), these processes will use up huge amount of resources as well.

The main contribution of this thesis, the *Multi-Resolution Hybrid Mapping* (MRHM), is an extension of the work from [31]. For MRHM, line-features are used in place of point-features as base landmarks to refine the local occupancy grid maps such that a more accurate representation of the environment can be achieved. There were two main stages to MRHM, with the first stage concentrating on the vehicle *Position Estimation* based on a line-feature map while the second stage focusses on *Multi-Resolution Occupancy Grid Mapping* (MROGM). This MRHM framework takes advantage of the efficiency of feature map based localisation while the occupancy grid mapping provides a high-resolution map of the environment. An alternative approach to the standard occupancy grid mapping, the MROGM, based on the concept of the quadtree/octree data structure is proposed. The MROGM technique attempts to reduce the computational time needed to build an occupancy grid map. For MROGM, it involves the use of two (or more if necessary) grid maps with different resolutions, where one is a higher resolution grid map and the other one a lower resolution grid map. As discussed earlier, quadtree/octree data structure will partition occupied parent cells into higher resolution child-cells of desired resolution (iterative partitioning may be necessary). Based on this concept, the occupancy proba-

bility calculation is conducted for the low resolution grid map first. If a space is deemed occupied in the low resolution grid map, the occupancy probability calculation of the high resolution grid cells, which tessellates the same space, will be needed. For spaces that are deemed empty, the high resolution cells can sample their occupancy probability values from their respective low resolution cells.

Chapter 3

Detailed Review on Some Robotic Mapping Paradigms

3.1 Introduction

The idea of *Hybrid Mapping*, which involves combining different types of mapping techniques to improve the overall efficiency and accuracy of robotic mapping, has been introduced in recent years. Hybrid maps often appear in the form of combining grid maps with topological based maps and are often used for navigation [1][27], path planning and obstacle avoidance [58]. However, high resolution grid mapping is computationally expensive, often requiring huge amounts of computational resources and time. In order to reduce the computational cost, various authors have used the quad-tree decomposition in their robotic navigation research work [11][28][35]. However, the online partitioning of cells is computationally expensive as well and the map management may become intractable if extensive partitioning processes are carried out. Hence a more efficient

method of high resolution grid mapping approach will be presented in this chapter.

In addition to the inherent computational issues, another drawback in constructing grid maps is that it is often assumed that the autonomous platform's position is well known [55]. In practise, the estimation (based on dead reckoning) of the platform's position is often erroneous and requires special position estimation refining processes, such as Kalman Filter / Extended Kalman Filter or SLAM, to correct the estimation. These refining processes will only take place when observations data, such as GPS positional data for KF/EKF and features for SLAM, are available. Hence a grid map constructed in the absence of these observations data may not be accurate. Even when observations data are available at a later stage, the grid map will not be corrected as there are no correlations between the grid map and the vehicle's iteratively corrected position. Some groups of researchers proposed the use of feature maps from SLAM to correct the grid maps recursively such that an accurate and high resolution representation of the environment can be constructed [30][31]. This chapter presents the fundamentals of the current robotic mapping paradigm that will be used in the proposed Multi-Resolution Hybrid Mapping (MRHM).

The second section discusses the process of extracting Hough lines from laser range data set to be used as features in the EKF-SLAM mentioned earlier. In an urban setting, most of the objects in the environment are structurally built with long straight walls. Hence extracting lines from these environments for EKF-SLAM features will be much easier than other types of features such as point clusters.

The third section presents a detailed formulation of the localisation component of the MRHM. One of the fundamental requirements of robotic mapping is the knowledge of the platform's location over time, or localisation. SLAM is often chosen as the localisation

paradigm as it provides recursive estimation of the platform’s location with respect to features around its immediate environment while maintaining cross-correlations between estimated landmarks [41]. Moreover, the feature maps built are able to “provide compact representations in the case of large open spaces with few obstacles” as discussed in [17].

Occupancy grid maps, on the other hand, are commonly employed to model the environment in greater detail, which may include objects that are not classified and stored as features by the EKF-SLAM feature maps during localization. However as discussed earlier, the overall performance of the system suffers as a result of the enormous amount of resources and computational time required by the standard grid mapping to construct and maintain a high resolution map. Hence the fourth section presents the proposed solution, the *Multi-Resolution Occupancy Grid Mapping* (MROGM), which can improve the overall efficiency of occupancy grid mapping.

The fifth section presents the work of HYbrid Metic Mappings (HYMMs) by Nieto et. al [31]. The proposed MRHM is an extension of this hybrid mapping approach. In their original work, they use point-cluster features as base landmarks which the small sized, high resolution grid maps are referenced to. This section looks into their original formulation to understand the working principles behind HYMMs.

The last section discusses the Multi Resolution Occupancy Grid Mapping (MROGM) paradigm. This paradigm can potentially reduce the time needed to model the environment with occupancy grid maps. The basis of this approach is the quadtree mapping technique. Instead of relying on an online partitioning process, this approach makes use of two (or more if necessary) grid maps of different resolution to model the environment.

3.2 Extraction Of Hough Line For EKF-SLAM

This section examines how lines are extracted from laser range finder data to be used as features in EKF-SLAM. It will include the principles of the line extraction technique used, the *Hough Transformation*, and the formulation of the equation sets that transform the line features between the global navigation frame and the local body frame.

3.2.1 Hough Transformation

Feature maps are often used to determine a vehicle's location as they are capable of providing accurate positional estimates yet consume relatively few resources, such as computational time and memory space, as compared to occupancy grid localisation. Lines have been widely used as features for SLAM by various authors, especially in highly structured urban environments where lines can be easily extracted from the surroundings [2][21][32].

One of the line feature extraction technique often used is the *Hough Transformation*. This technique is based on a voting procedure where a local maxima in the accumulator space will be generated. A line can be represented in the Cartesian coordinate system as $y = mx + c$, however vertical lines in this coordinate system may give rise to computational errors of the parameters m and c when $m \rightarrow \infty$ or $c \rightarrow \infty$, therefore measurements in this coordinate system are not optimal solutions. On the other hand, the Hough transform represents the lines in the Polar coordinate system with two parameters, ρ and θ , where ρ describes the perpendicular distance of the line feature to a certain point of reference, usually the origin of the reference frame, and θ describes

the angle created by the perpendicular joining a line to the point of reference and the positive x-axis of the reference frame.

Figure 3.1: Line extraction from a sample data set using Hough transform. The lines are represented in a Polar coordinate system described by ρ and θ .

In order to build the accumulator table for lines extraction, the Hough transformation will gradually increase the values of the two polar parameters of a line, θ and ρ such that every line that exists in the polar space will be generated. For each generated line, the total number of data points from the data set which fit the line model will be used to fill up the accumulator table [50]. The local maxima obtained from this table will indicate the largest amount of data points which fit that particular θ and ρ line model. With this information, lines can be extracted from a set of data points in space as shown in Figure 3.1. This approach is able to extract lines more consistently as it is a sequential search based on a voting procedure. Consistency is an important factor for SLAM as this framework requires *re-observation* of features to perform optimally.

3.2.2 Hough Lines For EKF-SLAM

In recent years, the Hough transform has been implemented in many forms to extract lines from data generated from range sensors to perform SLAM [13][57]. In this section, a discussion on transforming Hough line features between the sensing (local) frame and global (estimated mapping) frame will be presented. The converse operation will be discussed as well since it is required in order to predict line feature observations from the previously estimated map, for the purpose of data association. The derivation of the transformation equations are based on work by [2].

Transformation From Local Frame To Global Navigation Frame

As the vehicle moves, new features may be observed by on-board range finding sensors and these observed features are referenced to the vehicle's local body frame. It is necessary to add these newly observed features to the estimated map in the global reference frame for SLAM, hence the transformation of these features from the local frame to the global navigation frame is necessary. It is assumed throughout this report that the extracted line features, such as walls, do not move when referenced to the global navigation frame (from now on, the reference frame will be referred to as the global navigation frame), and only the vehicle moves.

Figure 3.2 shows a line-feature related to the local vehicle frame and the global navigation frame. In this figure, the global navigation frame is indicated by the subscript G and the body frame is indicated by the subscript L . $\hat{x}_v$, $\hat{y}_v$ and $\hat{\theta}_v$ are the current positional and directional estimates of the vehicle. The distance between the origin, O ,

Figure 3.2: Relationship of a feature between the local vehicle frame and the global navigation frame.

and the vehicle's current position is given by $D = \sqrt{\hat{x}_v^2 + \hat{y}_v^2}$ and α , which is the heading of the vehicle with respect to the global origin. The estimated range and bearing of the i^{th} line feature with respect to the global navigation frame and the local body frame are given by $\hat{\mathbf{m}}_{Gi} = [\hat{\rho}_{Gi} \ \hat{\theta}_{Gi}]^T$ and $\hat{\mathbf{z}}_i = [\hat{\rho}_{Li} \ \hat{\theta}_{Li}]^T$ respectively. The conversion of the observed features in the local frame to the global frame is then given by,

$$\begin{aligned}
 \hat{\mathbf{m}}_{Gi} &= \mathbf{g}_i(\hat{\mathbf{x}}_v, \mathbf{z}_i) \\
 \begin{bmatrix} \hat{\rho}_{Gi} \\ \hat{\theta}_{Gi} \end{bmatrix} &= \begin{bmatrix} g_1(\hat{\mathbf{x}}_v, \mathbf{z}_i) \\ g_2(\hat{\mathbf{x}}_v, \mathbf{z}_i) \end{bmatrix} \\
 &= \begin{bmatrix} \hat{\rho}_{Gi} + D \cos(\alpha - \hat{\theta}_{Gi}) \\ \hat{\theta}_{Li} + \hat{\theta}_v \end{bmatrix} \\
 &= \begin{bmatrix} \hat{\rho}_{Gi} + D(\cos(\alpha) \cos(\hat{\theta}_{Gi}) + \sin(\alpha) \sin(\hat{\theta}_{Gi})) \\ \hat{\theta}_{Li} + \hat{\theta}_v \end{bmatrix} \\
 &= \begin{bmatrix} \hat{\rho}_{Li} + \hat{x}_v \cos(\hat{\theta}_{Gi}) + \hat{y}_v \sin(\hat{\theta}_{Gi}) \\ \hat{\theta}_{Li} + \hat{\theta}_v \end{bmatrix} \tag{3.1}
 \end{aligned}$$

where $\mathbf{g}_i(\cdot)$ is the transformation equation vector for transforming features from the local frame to the global navigation frame where,

$$\alpha = \text{atan2}(y_v, x_v)$$

and,

$$\begin{aligned} -\pi &< \hat{\theta}_{Li} \leq \pi \\ -\pi &< \hat{\theta}_{Gi} \leq \pi \\ -\pi &< \hat{\theta}_v \leq \pi \end{aligned}$$

In Eqn 3.1, $\hat{\mathbf{x}}_v$ is the EKF-SLAM vehicle state vector. The angles mentioned above are in radians and have positive values in the anti-clockwise direction. Upon transforming the line-feature parameters from the global navigation frame to the body frame, it can then be used for feature association in the EKF-SLAM process. It is important to note that $\hat{\theta}_{Gi}$ would need to be derived first using equation $g_2(\cdot)$ as this parameter is required by equation $g_1(\cdot)$. With the derivation of the local-to-global frame transformation equations completed, the transformed line parameters can now be stored in EKF-SLAM map estimates.

Transformation From Global Navigation Frame To Local Frame

As well as transforming Hough features from the local frame to the global reference frame, the converse is also necessary for generating predicted observations in the local frame, from previous map estimates in the global frame. The extracted features in the vehicle's local frame need to be associated with previously stored features to refine positional estimates. Existing features are usually stored with reference to the global navigation frame. However features observed by the range finder sensors are usually in

the body frame. Hence conversion of existing features from the global reference frame to body frame is necessary. The transformation equations can be derived from Figure 3.2 by bringing the global navigation frame parameters, such as $\hat{\rho}_{Gi}$ and $\hat{\theta}_{Gi}$, to the left-hand-side and the local frame parameters ($\hat{\rho}_{Li}$ and $\hat{\theta}_{Li}$) to the right-hand-side of the equation. Hence the transformation equations for i^{th} features from are given by,

$$\begin{aligned}
 \hat{\mathbf{z}}_i &= \mathbf{h}_i(\hat{\mathbf{x}}_v, \hat{\mathbf{m}}_{Gi}) \\
 \begin{bmatrix} \hat{\rho}_{Li} \\ \hat{\theta}_{Li} \end{bmatrix} &= \begin{bmatrix} h_{i1}(\hat{\mathbf{x}}_v, \hat{\mathbf{m}}_{Gi}) \\ h_{i2}(\hat{\mathbf{x}}_v, \hat{\mathbf{m}}_{Gi}) \end{bmatrix} \\
 &= \begin{bmatrix} \hat{\rho}_{Gi} - D \cos(\alpha - \hat{\theta}_{Gi}) \\ \hat{\theta}_{Gi} - \hat{\theta}_v \end{bmatrix} \\
 &= \begin{bmatrix} \hat{\rho}_{Gi} - D(\cos(\alpha) \cos(\hat{\theta}_{Gi}) + \sin(\alpha) \sin(\hat{\theta}_{Gi})) \\ \hat{\theta}_{Gi} - \hat{\theta}_v \end{bmatrix} \\
 &= \begin{bmatrix} \hat{\rho}_{Gi} - \hat{x}_v \cos(\hat{\theta}_{Gi}) - \hat{y}_v \sin(\hat{\theta}_{Gi}) \\ \hat{\theta}_{Gi} - \hat{\theta}_v \end{bmatrix} \tag{3.2}
 \end{aligned}$$

where $\mathbf{h}_i(.)$ is the transformation equation vector for transforming features from the global navigation frame to the local frame. The rest of the notations have the same meaning as described in Chapter 3.2.2. The derived predicted observations, $\hat{\mathbf{z}}_i$, can now be used for data association in the EKF-SLAM process.

3.2.3 Hough Transformation Error Analysis

Noise induced errors in the Hough lines with respect to the local reference frame unavoidable. It is important to determine the effect of these errors on the lines registered in the global map estimates and vice versa. In this section, it will be shown that the resulting errors in the Hough parameters in the global/local frame due to given Hough

parameter errors in the local/global frame depends on the position of the vehicle with respect to the global navigation frame.

Given an error in the vehicle's positional estimate, $\delta \hat{\mathbf{x}}_v$, and an error in the global Hough parameters, $\delta \hat{\mathbf{m}}_{Gi}$, then Eqn. 3.1 can be re-written as

$$\hat{\mathbf{m}}_{Gi} + \delta \hat{\mathbf{m}}_{Gi} = \mathbf{h}_i((\hat{\mathbf{x}}_v + \delta \hat{\mathbf{x}}_v), (\hat{\mathbf{z}}_i + \delta \hat{\mathbf{z}}_i)) \quad (3.3)$$

and Eqn. 3.2 as

$$\hat{\mathbf{z}}_i + \delta \hat{\mathbf{z}}_i = \mathbf{h}_i((\hat{\mathbf{x}}_v + \delta \hat{\mathbf{x}}_v), (\hat{\mathbf{m}}_{Gi} + \delta \hat{\mathbf{m}}_{Gi})) \quad (3.4)$$

respectively, where $\delta \hat{\mathbf{x}}_v = [\delta \hat{x}_v \ \delta \hat{y}_v \ \delta \hat{\theta}_v]^T$ and $\delta \hat{\mathbf{m}}_{Gi} = [\delta \hat{\rho}_{Gi} \ \delta \hat{\theta}_{Gi}]$ are the errors in the vehicle state vector and feature parameters in the global navigation frame respectively. Hence from the chain rule of differentiation, the equation relating the error from the global navigation frame to the local body frame based on Eqn. 3.3 and Eqn. 3.4 can be written as

$$\begin{aligned}
\delta \hat{\mathbf{m}}_i &= \frac{\partial \mathbf{g}_i}{\partial \hat{\mathbf{x}}_v} \delta \hat{\mathbf{x}}_v + \frac{\partial \mathbf{g}_i}{\partial \hat{\mathbf{z}}_i} \delta \hat{\mathbf{z}}_i \\
\begin{bmatrix} \delta \hat{\rho}_{Gi} \\ \delta \hat{\theta}_{Gi} \end{bmatrix} &= \begin{bmatrix} \cos(\hat{\theta}_{Gi}) & \sin(\hat{\theta}_{Gi}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta \hat{x}_v \\ \delta \hat{y}_v \\ \delta \hat{\theta}_v \end{bmatrix} + \\
&\quad \begin{bmatrix} 1 & -\hat{x}_v \sin(\hat{\theta}_{Gi}) + \hat{y}_v \cos(\hat{\theta}_{Gi}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta \hat{\rho}_{Li} \\ \delta \hat{\theta}_{Li} \end{bmatrix} \\
&= \begin{bmatrix} \delta \hat{x}_v \cos(\hat{\theta}_{Gi}) + \delta \hat{y}_v \sin(\hat{\theta}_{Gi}) \\ \delta \hat{\theta}_v \end{bmatrix} + \\
&\quad \begin{bmatrix} \delta \hat{\rho}_{Li} - \delta \hat{\theta}_{Li} (\hat{x}_v \sin(\hat{\theta}_{Gi}) - \hat{y}_v \cos(\hat{\theta}_{Gi})) \\ \delta \hat{\theta}_{Li} \end{bmatrix} \tag{3.5}
\end{aligned}$$

and

$$\begin{aligned}
\delta \hat{\mathbf{z}}_i &= \frac{\partial \mathbf{h}_i}{\partial \hat{\mathbf{x}}_v} \delta \hat{\mathbf{x}}_v + \frac{\partial \mathbf{h}_i}{\partial \hat{\mathbf{m}}_{Gi}} \delta \hat{\mathbf{m}}_{Gi} \\
\begin{bmatrix} \delta \hat{\rho}_{Li} \\ \delta \hat{\theta}_{Li} \end{bmatrix} &= \begin{bmatrix} -\cos(\hat{\theta}_{Gi}) & -\sin(\hat{\theta}_{Gi}) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \delta \hat{x}_v \\ \delta \hat{y}_v \\ \delta \hat{\theta}_v \end{bmatrix} + \\
&\quad \begin{bmatrix} 1 & \hat{x}_v \sin(\hat{\theta}_{Gi}) - \hat{y}_v \cos(\hat{\theta}_{Gi}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta \hat{\rho}_{Gi} \\ \delta \hat{\theta}_{Gi} \end{bmatrix} \\
&= \begin{bmatrix} -\delta \hat{x}_v \cos(\hat{\theta}_{Gi}) - \delta \hat{y}_v \sin(\hat{\theta}_{Gi}) \\ -\delta \hat{\theta}_v \end{bmatrix} + \\
&\quad \begin{bmatrix} \delta \hat{\rho}_{Gi} + \delta \hat{\theta}_{Gi} (\hat{x}_v \sin(\hat{\theta}_{Gi}) - \hat{y}_v \cos(\hat{\theta}_{Gi})) \\ \delta \hat{\theta}_{Gi} \end{bmatrix} \tag{3.6}
\end{aligned}$$

respectively. An investigation was carried out to analyze the impact of Hough parameter errors in the local/global frame had on the transformed feature parameters in the global/local frame.

Figure 3.3: Huge parametric errors derived when transforming features from local frame to global navigation frame due to vehicle position.

Figure 3.3 shows a situation where an actual line feature, indicated by thick solid line, with parameters $\hat{\mathbf{z}}_i = [\hat{\rho}_{Li} \ \hat{\theta}_{Li}]^T$ is being detected by the vehicle in the local body frame. However due to noise, this feature parameters are given as $(\hat{\mathbf{z}}_i + \delta\hat{\mathbf{z}}_i) = [\hat{\rho}_{Li.err} \ (\hat{\theta}_{Li} + \delta\hat{\theta}_{Li})]^T$. This feature is then being transformed and stored in the global referenced map using Eqn. 3.3. The feature parameters stored in the global reference map are derived to be $(\hat{\mathbf{m}}_{Gi} + \delta\hat{\mathbf{m}}_{Gi}) = [\hat{\rho}_{Gi.err} \ (\hat{\theta}_{Gi} + \delta\hat{\theta}_{Gi})]^T$. From the figure, it can be seen visually that there is a significant difference between the actual feature distance, $\hat{\rho}_{Gi}$, and the transformed feature distance with parametric error, $\hat{\rho}_{Gi.err}$. It can be expected that this difference will increase if the vehicle moves further up along the y-axis direction (due to the widening of gaps between the solid line and the dotted line observed visually).

Assuming that errors does not exist in the vehicle state estimates such that $\delta\hat{x}_v = \delta\hat{y}_v = \delta\hat{\theta}_v = 0$, and the local feature parameter $\delta\hat{\rho}_{Li} \approx 0$, such that

$$\begin{bmatrix} \hat{\mathbf{x}}_{v.err} \\ \hat{\mathbf{z}}_{i.err} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{z}}_i \end{bmatrix} + \begin{bmatrix} \delta\hat{\mathbf{x}}_v \\ \delta\hat{\mathbf{z}}_i \end{bmatrix}$$

$$\begin{bmatrix} \hat{x}_{v.err} \\ \hat{z}_{v.err} \\ \hat{\theta}_{v.err} \\ \hat{\rho}_{Li.err} \\ \hat{\theta}_{Li.err} \end{bmatrix} = \begin{bmatrix} \hat{x}_v \\ \hat{z}_v \\ \hat{\theta}_v \\ \hat{\rho}_{Li} \\ \hat{\theta}_{Li} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \delta\hat{\theta}_{Li} \end{bmatrix}$$

then Eqn. 3.5 can be re-written as

$$\begin{aligned}
\begin{bmatrix} \delta \hat{\rho}_{Gi} \\ \delta \hat{\theta}_{Gi} \end{bmatrix} &= \begin{bmatrix} \cos(\hat{\theta}_{Gi}) & \sin(\hat{\theta}_{Gi}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \\
&\quad \begin{bmatrix} 1 & -\hat{x}_v \sin(\hat{\theta}_{Gi}) + \hat{y}_v \cos(\hat{\theta}_{Gi}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \delta \hat{\theta}_{Li} \end{bmatrix} \\
&= \begin{bmatrix} -\delta \hat{\theta}_{Li} (\hat{x}_v \sin(\hat{\theta}_{Gi}) - \hat{y}_v \cos(\hat{\theta}_{Gi})) \\ \delta \hat{\theta}_{Li} \end{bmatrix} \tag{3.7}
\end{aligned}$$

From Eqn. 3.7, the transformed feature parametric error $\delta \hat{\rho}_{Gi}$ depends on the position of the vehicle as well as the globally referenced feature bearing, $\hat{\theta}_{Gi}$. From both Figure 3.3 and Eqn. 3.7, it can be seen that as the vehicle moves further away from the point of reference, the parametric error of the transformed feature, especially the error in feature distance, will increase as well. The cases where there are minimum error occurs only when the vehicle lies along the direction of the reference frame axes (either $\hat{x}_v = 0$ and/or $\hat{y}_v = 0$) and/or when $\sin(\hat{\theta}_{Gi}) = 0$ or $\cos(\hat{\theta}_{Gi}) = 0$.

Using the same assumption mentioned above to perform analysis on the converse of transformation of features, Eqn. 3.6 is re-written as follows

$$\begin{aligned}
\begin{bmatrix} \delta \hat{\rho}_{Li} \\ \delta \hat{\theta}_{Li} \end{bmatrix} &= \begin{bmatrix} -\cos(\hat{\theta}_{Gi}) & -\sin(\hat{\theta}_{Gi}) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \\
&\quad \begin{bmatrix} 1 & \hat{x}_v \sin(\hat{\theta}_{Gi}) & -\hat{y}_v \cos(\hat{\theta}_{Gi}) \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \delta \hat{\theta}_{Gi} \end{bmatrix} \\
&= \begin{bmatrix} \delta \hat{\theta}_{Gi} (\hat{x}_v \sin(\hat{\theta}_{Gi}) - \hat{y}_v \cos(\hat{\theta}_{Gi})) \\ \delta \hat{\theta}_{Gi} \end{bmatrix} \tag{3.8}
\end{aligned}$$

Similar to Eqn. 3.7, the parametric error of the feature distance upon transformation from local frame to global navigation frame is a function of both the vehicle position with respect to the point of reference as well as the globally referenced bearing of the estimate feature.

Hence it can be conclude that the position of the vehicle together with the globally reference bearing of the feature contributes to the resulting errors in the feature Hough parameters in the global/local frame due to Hough parameter errors in the local/global frame. As the *observation prediction* stage of EKF-SLAM depends on these feature parameters, these errors, especially when the vehicle is far from the point of reference, could affect the performance of *data association* and may lead to divergence of SLAM.

3.3 EKF-SLAM Based On Hough-Line Features

An EKF-SLAM solution for the generation of positional estimates of both the vehicle

and the features can be categorized into three main stages, *The Prediction Stage*, *The Update Stage* and finally *Adding New Features*. In this section, a detailed formulation of EKF-SLAM estimation will be presented. The design of the kinematic models and the EKF-SLAM are based on [2][24][41].

3.3.1 The Prediction Stage

The first stage of EKF-SLAM is the *Prediction Stage*. In this stage, predictions of the current vehicle and feature state based on previous state information are made. The prediction includes the *predicted state vector*, $\hat{\mathbf{x}}(k|k-1)$, and *predicted covariance matrix*, $\mathbf{P}(k|k-1)$. The predicted state vector, $\hat{\mathbf{x}}(k|k-1)$, contains two different types of information, the vehicle state, $\hat{\mathbf{x}}_v(k-1|k-1)$, and the feature locations, $\hat{\mathbf{m}}$. At time k , $\hat{\mathbf{x}}_v(k-1|k-1)$ can be written as,

$$\hat{\mathbf{x}}_v(k|k) = \begin{bmatrix} \hat{x}_v(k|k) & \hat{y}_v(k|k) & \hat{\theta}_v(k|k) \end{bmatrix}^T \quad (3.9)$$

where $\hat{x}_v(k|k)$ and $\hat{y}_v(k|k)$ are the position estimates and $\hat{\theta}_v(k|k)$ is the orientation estimate of the vehicle with respect the global navigation frame. The platform's kinematic model shown in Figure 3.4, which is based on [41] and, can be written as,

$$\hat{\mathbf{x}}_v(k|k-1) = \mathbf{f}(\hat{\mathbf{x}}_v(k-1|k-1), \mathbf{u}(k)) \quad (3.10)$$

$\mathbf{u}(k) = \begin{bmatrix} V(k) & \dot{\theta}_v(k) \end{bmatrix}^T$ is defined as the platform's control input. $V(k)$ and $\dot{\theta}_v(k)$ are the vehicle's forward velocity and turn rate respectively. A detailed formulation of Eqn. 3.10 can be written as follows,

$$\begin{aligned}
\begin{bmatrix} \hat{x}_v(k|k-1) \\ \hat{y}_v(k|k-1) \\ \hat{\theta}_v(k|k-1) \end{bmatrix} &= \begin{bmatrix} f_1(\hat{x}_v(k-1|k-1), u_x(k)) \\ f_2(\hat{y}_v(k-1|k-1), u_y(k)) \\ f_3(\hat{\theta}_v(k-1|k-1), u_\theta(k)) \end{bmatrix} \\
&= \begin{bmatrix} \hat{x}_v(k-1|k-1) + \Delta T V(k) \cos(\hat{\theta}_v(k-1|k-1) + \Delta T \dot{\theta}_v(k)) \\ \hat{y}_v(k-1|k-1) + \Delta T V(k) \sin(\hat{\theta}_v(k-1|k-1) + \Delta T \dot{\theta}_v(k)) \\ \hat{\theta}_v(k-1|k-1) + \Delta T \dot{\theta}_v(k) \end{bmatrix} \\
&= \begin{bmatrix} \hat{x}_v(k-1|k-1) + \Delta x_v(k) \\ \hat{y}_v(k-1|k-1) + \Delta y_v(k) \\ \hat{\theta}_v(k-1|k-1) + \Delta \theta_v(k) \end{bmatrix} \tag{3.11}
\end{aligned}$$

where ΔT is the sampling time interval.

Figure 3.4: Kinematics of the vehicle model. The turn rate of the vehicle, $\dot{\theta}_v$, can be obtained from a single-axis gyro by placing it on the center of the rear wheel axes as indicated by the solid dot in the figure.

The other type of information stored in the state vector, as mentioned earlier in this section, are the estimated feature locations, $\hat{\mathbf{m}}(k|k)$. For notation simplicity, $\hat{\mathbf{m}}(k|k)$ will be denoted as $\hat{\mathbf{m}}$ and be referred to as *feature locations* from now on. For line-feature based SLAM, each line feature, $\hat{\mathbf{m}}_i$, is described by two parameters - range, $\hat{\rho}$, and bearing,

$\hat{\theta}$ and they are stored in a vector as follows,

$$\hat{\mathbf{m}}_i = \begin{bmatrix} \hat{\rho}_{Gi} & \hat{\theta}_{Gi} \end{bmatrix}^T \quad (3.12)$$

where the subscript G indicates that the features stored are referenced to the global navigation frame and i refers to the i^{th} feature.

As the global feature locations are independent of vehicle movement, the predicted feature locations, $\hat{\mathbf{m}}(k|k-1)$, with N features at time k can be written as,

$$\begin{bmatrix} \hat{\mathbf{m}}_1(k|k-1) \\ \hat{\mathbf{m}}_2(k|k-1) \\ \hat{\mathbf{m}}_3(k|k-1) \\ \vdots \\ \hat{\mathbf{m}}_i(k|k-1) \\ \vdots \\ \hat{\mathbf{m}}_N(k|k-1) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{m}}_1(k-1|k-1) \\ \hat{\mathbf{m}}_2(k-1|k-1) \\ \hat{\mathbf{m}}_3(k-1|k-1) \\ \vdots \\ \hat{\mathbf{m}}_i(k-1|k-1) \\ \vdots \\ \hat{\mathbf{m}}_N(k-1|k-1) \end{bmatrix} \quad (3.13)$$

and therefore the full $(3 + 2N) \times (1)$ predicted state vector, $\hat{\mathbf{x}}(k|k-1)$, using Eqns. 3.11 and 3.13 can be written as,

$$\begin{bmatrix} \hat{x}_v(k|k-1) \\ \hat{y}_v(k|k-1) \\ \hat{\theta}_v(k|k-1) \\ \hat{\mathbf{m}}_1(k|k-1) \\ \hat{\mathbf{m}}_2(k|k-1) \\ \hat{\mathbf{m}}_3(k|k-1) \\ \vdots \\ \hat{\mathbf{m}}_i(k|k-1) \\ \vdots \\ \hat{\mathbf{m}}_N(k|k-1) \end{bmatrix} = \begin{bmatrix} \hat{x}_v(k-1|k-1) \\ \hat{y}_v(k-1|k-1) \\ \hat{\theta}_v(k-1|k-1) \\ \hat{\mathbf{m}}_1(k-1|k-1) \\ \hat{\mathbf{m}}_2(k-1|k-1) \\ \hat{\mathbf{m}}_3(k-1|k-1) \\ \vdots \\ \hat{\mathbf{m}}_i(k-1|k-1) \\ \vdots \\ \hat{\mathbf{m}}_N(k-1|k-1) \end{bmatrix} + \begin{bmatrix} \Delta x_v(k) \\ \Delta y_v(k) \\ \Delta \theta_v(k) \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (3.14)$$

By using both Eqns. 3.11 and 3.14, the full EKF-SLAM estimation equations can be

written as,

$$\hat{\mathbf{x}}(k|k-1) = [\mathbf{f}^T(\hat{\mathbf{x}}_v(k-1|k-1), \mathbf{u}(k)) \hat{\mathbf{m}}^T(k-1|k-1)]^T \quad (3.15)$$

$$\begin{aligned} \mathbf{P}(k|k-1) = & \nabla_{\mathbf{x}}\mathbf{f}(\hat{\mathbf{x}}(k-1|k-1)) \mathbf{P}(k-1|k-1) \nabla_{\mathbf{x}}\mathbf{f}^T(\hat{\mathbf{x}}(k-1|k-1)) + \\ & \nabla_{\mathbf{u}}\mathbf{f}(\hat{\mathbf{x}}(k-1|k-1)) \mathbf{Q}(k) \nabla_{\mathbf{u}}\mathbf{f}^T(\hat{\mathbf{x}}(k-1|k-1)) \end{aligned} \quad (3.16)$$

where $\mathbf{P}(k-1|k-1)$ is the system covariance matrix at time k while $\nabla_{\mathbf{x}}\mathbf{f}(\hat{\mathbf{x}}(k-1|k-1))$ and $\nabla_{\mathbf{u}}\mathbf{f}(\hat{\mathbf{x}}(k-1|k-1))$ are the Jacobian matrices of the process model with respect to the vehicle state and control input respectively. For notational simplicity, they will be denoted as $\nabla_{\mathbf{x}}\mathbf{f}$ and $\nabla_{\mathbf{u}}\mathbf{f}$ respectively. $\mathbf{Q}(k)$ is the process noise covariance matrix.

When deriving the predicted covariance matrix, $\mathbf{P}$, for the state vector, the EKF requires the process model to be linearised about the vehicle's location. The linearisation is performed by taking the Jacobian of the process model with respect to the state and control input thus obtaining $\nabla_{\mathbf{x}}\mathbf{f}$ and $\nabla_{\mathbf{u}}\mathbf{f}$, from 3.16, respectively. The $(3+2N) \times (3+2N)$ $\nabla_{\mathbf{x}}\mathbf{f}$ matrix can be described as,

$$\begin{aligned} \nabla_{\mathbf{x}}\mathbf{f} &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k-1|k-1)} \\ &= \left[\begin{array}{ccccccccc} 1 & 0 & -\Delta y_v(k) & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \Delta x_v(k) & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{array} \right]_{\mathbf{x}=\hat{\mathbf{x}}(k-1|k-1)} \end{aligned} \quad (3.17)$$

where $\Delta x_v(k)$ and $\Delta y_v(k)$ are described in Eqn 3.11. Similarly $\nabla_{\mathbf{u}}\mathbf{f}$ is a $(3+2N) \times (2)$ Jacobian matrix of the process model however it is with respect to the control input and

it is written as,

$$\begin{aligned} \nabla_{\mathbf{u}} \mathbf{f} &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k-1|k-1)} \\ &= \left[\begin{array}{cc} \frac{\Delta x_v(k)}{V} & -\Delta T \Delta y_v(k) \\ \frac{\Delta y_v(k)}{V} & \Delta T \Delta x_v(k) \\ 0 & \Delta T \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \end{array} \right] \bigg|_{\mathbf{x}=\hat{\mathbf{x}}(k-1|k-1)} \end{aligned} \quad (3.18)$$

and

$$\mathbf{Q}(k) = \begin{bmatrix} \sigma_V^2 & 0 \\ 0 & \sigma_{\dot{\theta}_v}^2 \end{bmatrix} \quad (3.19)$$

where σ_V^2 and $\sigma_{\dot{\theta}_v}^2$ are the variances of the velocity and turn rate of the vehicle respectively.

3.3.2 Update Stage

The second stage of the EKF-SLAM is the *Update Stage*. In this stage, information from the observed features via exteroceptive sensors is used to refine the predictions made in the *Prediction Stage* to arrive at a more accurate estimate of both the vehicle and feature states. The *predicted observation* of the i^{th} feature at time k , $\hat{\mathbf{z}}_i(k|k-1)$, and the full *innovation covariance matrix*, $\mathbf{S}(k)$, which includes terms for all features, can be described as follows,

$$\hat{\mathbf{z}}_i(k|k-1) = \mathbf{h}_i(\hat{\mathbf{x}}_v(k|k-1), \hat{\mathbf{m}}_i(k|k-1)) \quad (3.20)$$

$$\begin{aligned} \mathbf{S}(k) = & \nabla_{\mathbf{x}}\mathbf{h}(\hat{\mathbf{x}}(k|k-1))\mathbf{P}(k|k-1)\nabla_{\mathbf{x}}\mathbf{h}^T(\hat{\mathbf{x}}(k|k-1)) + \\ & \mathbf{R}(k) \end{aligned} \quad (3.21)$$

where $\mathbf{h}_i(\cdot)$ is the predicted observation transformation equation as presented in Section 3.2.2. $\nabla_{\mathbf{x}}\mathbf{h}$ is the Jacobian matrix of the predicted observation transformation equations while $\mathbf{R}(k)$ is the observation noise. The range and bearing parameters of the i^{th} predicted observation, $\hat{\mathbf{z}}_i$, are written as,

$$\hat{\mathbf{z}}_i = \begin{bmatrix} \hat{\rho}_{Li} & \hat{\theta}_{Li} \end{bmatrix}^T \quad (3.22)$$

From Eqn. 3.21, $\mathbf{R}(k)$ is the measurement noise covariance matrix and $\nabla_{\mathbf{x}}\mathbf{h}(\hat{\mathbf{x}}(k|k-1))$ is the Jacobian of the full measurement model, $\mathbf{h}(\hat{\mathbf{x}}(k|k-1))$, with respect to the state vector, $\hat{\mathbf{x}}$. From now on $\mathbf{h}(\hat{\mathbf{x}}(k|k-1))$ and $\nabla_{\mathbf{x}}\mathbf{h}(\hat{\mathbf{x}}(k|k-1))$ will be denoted as $\mathbf{h}(\cdot)$ and $\nabla_{\mathbf{x}}\mathbf{h}$ respectively. The full measurement model is formed using the transformation equation vectors for each predicted feature. With N predicted features, $\mathbf{h}(\cdot)$ is written as,

$$\mathbf{h}(\cdot) = \begin{bmatrix} h_1(\cdot) \\ h_2(\cdot) \\ h_3(\cdot) \\ \vdots \\ h_i(\cdot) \\ \vdots \\ h_N(\cdot) \end{bmatrix} \quad (3.23)$$

therefore $\nabla_{\mathbf{x}}\mathbf{h}$, with N features, can be written as,

$$\begin{aligned}
\nabla_{\mathbf{x}} \mathbf{h} &= \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)} \\
&= \left[\begin{array}{c} \nabla_{\mathbf{x}} \mathbf{h}_1 \\ \nabla_{\mathbf{x}} \mathbf{h}_2 \\ \nabla_{\mathbf{x}} \mathbf{h}_3 \\ \vdots \\ \nabla_{\mathbf{x}} \mathbf{h}_i \\ \vdots \\ \nabla_{\mathbf{x}} \mathbf{h}_N \end{array} \right] \bigg|_{\mathbf{x}=\hat{\mathbf{x}}(k|k-1)} \quad (3.24)
\end{aligned}$$

The Jacobian of each feature, $\nabla_{\mathbf{x}} \mathbf{h}_i$, is derived by differentiating the measurement model with respect to the predicted vehicle state, $\hat{\mathbf{x}}_v$, and predicted feature location, $\hat{\mathbf{m}}_i$. $\nabla_{\mathbf{x}} \mathbf{h}_i$ can be written as,

$$\begin{aligned}
\nabla_{\mathbf{x}} \mathbf{h}_i &= \left[\begin{array}{cc} \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_v} & \frac{\partial \mathbf{h}_i}{\partial \hat{\mathbf{m}}_i} \end{array} \right] \\
&= \left[\begin{array}{ccccc} \frac{\partial h_{i1}}{\partial \hat{x}_v} & \frac{\partial h_{i1}}{\partial \hat{y}_v} & \frac{\partial h_{i1}}{\partial \hat{\rho}_v} & \frac{\partial h_{i1}}{\partial \hat{\rho}_{Gi}} & \frac{\partial h_{i1}}{\partial \hat{\theta}_{Gi}} \\ \frac{\partial h_{i2}}{\partial \hat{x}_v} & \frac{\partial h_{i2}}{\partial \hat{y}_v} & \frac{\partial h_{i2}}{\partial \hat{\rho}_v} & \frac{\partial h_{i2}}{\partial \hat{\rho}_{Gi}} & \frac{\partial h_{i2}}{\partial \hat{\theta}_{Gi}} \end{array} \right] \\
&= \left[\begin{array}{ccccc} -\cos(\hat{\theta}_{Gi}) & -\sin(\hat{\theta}_{Gi}) & 0 & 1 & (\hat{x}_v \sin(\hat{\theta}_{Gi})) - \hat{y}_v \cos(\hat{\theta}_{Gi}) \\ 0 & 0 & -1 & 0 & 1 \end{array} \right] \quad (3.25)
\end{aligned}$$

After deriving $\hat{\mathbf{z}}(k|k-1)$ and $\mathbf{S}(k)$, the measurement of features within the vehicle's vicinity is performed. Each observed feature, $\mathbf{z}_j(k)$, is then associated with the predicted features stored in $\hat{\mathbf{z}}(k|k-1)$ using suitable data association methods. If associations between observations and predicted features are successful, the innovation for each feature, $\tilde{\mathbf{z}}_i(k|k-1)$, is calculated using,

$$\tilde{\mathbf{z}}_i(k|k-1) = \mathbf{z}_j(k) - \hat{\mathbf{z}}_i(k|k-1) \quad (3.26)$$

In situations where observed features are not associated with any predicted features, they will be classified as newly observed features and will be stored in the state vector

at the *Adding New Features* stage. If M observations are associated with M predicted features, a full innovation vector, $\tilde{\mathbf{z}}(k|k-1)$, can be defined as,

$$\tilde{\mathbf{z}}(k|k-1) = \begin{bmatrix} \tilde{\mathbf{z}}_1(k|k-1) \\ \tilde{\mathbf{z}}_2(k|k-1) \\ \tilde{\mathbf{z}}_3(k|k-1) \\ \vdots \\ \tilde{\mathbf{z}}_i(k|k-1) \\ \vdots \\ \tilde{\mathbf{z}}_M(k|k-1) \end{bmatrix} \quad (3.27)$$

Thus, with the formulations above, the update equations of the EKF-SLAM state can then be written as follows,

$$\mathbf{W}(k+1) = \mathbf{P}(k|k-1) \nabla_{\mathbf{x}} \mathbf{h}^T(k) \mathbf{S}^{-1}(k+1) \quad (3.28)$$

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{W}(k+1) \tilde{\mathbf{z}}(k|k-1) \quad (3.29)$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k+1) \nabla_{\mathbf{x}} \mathbf{h}(k) \mathbf{P}(k|k-1) \quad (3.30)$$

where $\mathbf{W}(k)$ is the Kalman gain. As mentioned earlier, the EKF-SLAM algorithm is a recursive process to constantly provide estimates on the platform's position. If there are new features being observed, it is then necessary to store them into the state vector before proceeding with the next EKF cycle. The addition of new features will be discussed in the next section.

3.3.3 Adding New Features

As the vehicle maneuvers, new features will be detected and it is necessary include and store them so that the vehicle makes optimal use of these features. As these new features are observed in the local frame, hence, transformation to the global navigation

frame will be needed and the equations are given as,

$$\hat{\mathbf{m}} = \mathbf{g}(\hat{\mathbf{x}}_v, \mathbf{z}) \quad (3.31)$$

where $\mathbf{g}(\cdot)$ is the function for transforming the observed features in the local frame to the global navigation frame as presented in Chapter 3.2.2. After conversion, these features can then be stored into the feature map. It is necessary to update both the state vector, $\hat{\mathbf{x}}(k|k)$, and the covariance matrix, $\mathbf{P}(k|k)$ as well. Given that a single new feature, $\hat{\mathbf{m}}_{N+1}(k)$ was observed, then $\hat{\mathbf{x}}(k|k)$, with N features already stored, can be updated as,

$$\hat{\mathbf{x}}(k|k) = \begin{bmatrix} \hat{\mathbf{x}}_v(k|k) \\ \hat{\mathbf{m}}_1(k|k) \\ \hat{\mathbf{m}}_2(k|k) \\ \hat{\mathbf{m}}_3(k|k) \\ \vdots \\ \hat{\mathbf{m}}_i(k|k) \\ \vdots \\ \hat{\mathbf{m}}_N(k|k) \\ \hat{\mathbf{m}}_{N+1}(k|k) \end{bmatrix} \quad (3.32)$$

A new row and column were added, as shown by the right-most column and bottom-most row in Figure 3.5, to the covariance matrix, $\mathbf{P}(k|k)$. This is to describe the uncertainty of the new feature's location as well as cross-correlations between this new feature and other existing features.

The $\mathbf{P}_{v,v}$ sub-matrix in Figure 3.5 is a 3×3 matrix which contains the covariance of the vehicle state (x_v , y_v and θ_v). $\mathbf{P}_{\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_1}$ and $\mathbf{P}_{\hat{\mathbf{m}}_{N+1}, \hat{\mathbf{m}}_{N+1}}$ are 2×2 covariance matrices of the first feature, $\hat{\mathbf{m}}_1$, and the newest feature added, $\hat{\mathbf{m}}_{N+1}$, respectively. $\mathbf{P}_{\hat{\mathbf{m}}_1, v}$ and $\mathbf{P}_{\hat{\mathbf{m}}_{N+1}, v}$ are the cross-correlation matrices between the features ($\hat{\mathbf{m}}_1$ and $\hat{\mathbf{m}}_{N+1}$) and the platform state respectively. $\mathbf{P}_{\hat{\mathbf{m}}_{N+1}, \hat{\mathbf{m}}_1}$ is the cross-correlation between features $\hat{\mathbf{m}}_1$ and

	$P_{v,v}$		$P_{v,\hat{m}_1}$	$\dots$		$P_{v,\hat{m}_{N+1}}$	
	$P_{\hat{m}_1,v}$		$P_{\hat{m}_1,\hat{m}_1}$	$\dots$		$P_{\hat{m}_1,\hat{m}_{N+1}}$	
	$\vdots$		$\vdots$	$\ddots$		$\vdots$	
	$P_{\hat{m}_{N+1},v}$		$P_{\hat{m}_{N+1},\hat{m}_1}$	$\dots$		$P_{\hat{m}_{N+1},\hat{m}_{N+1}}$	

Figure 3.5: This figure shows the contents of the estimated covariance matrix, $\mathbf{P}$.

$\hat{\mathbf{m}}_{N+1}$. Similarly, there will be cross-correlation between this new feature with all the existing features stored. For simplicity, these cross-correlations are not shown in the figure. The rest of the cross-correlation sub-matrices are given as,

$$\begin{aligned}\mathbf{P}_{v,\hat{m}_i} &= \mathbf{P}_{\hat{m}_i,v}^T \\ \mathbf{P}_{\hat{m}_i,\hat{m}_j} &= \mathbf{P}_{\hat{m}_j,\hat{m}_i}^T\end{aligned}$$

The uncertainty matrix of the new feature, $\mathbf{P}_{\hat{m}_{N+1}}$, is the combination of uncertainty in the vehicle state and the measurement noise and is given by,

$$\begin{aligned}\mathbf{P}_{\hat{m}_{N+1}}(k|k) &= \nabla_x \mathbf{g}(k|k) \mathbf{P}_v(k|k) \nabla_x \mathbf{g}(k|k)^T + \\ &\quad \nabla_z \mathbf{g}(k|k) \mathbf{R}(k) \nabla_z \mathbf{g}(k|k)^T\end{aligned}\tag{3.33}$$

where,

$$\nabla_x \mathbf{f}(k|k) = \frac{\partial}{\partial \mathbf{x}_v} (\mathbf{g}(\hat{\mathbf{x}}_v, \mathbf{z}))$$

$$\nabla_z \mathbf{f}(k|k) = \frac{\partial}{\partial \mathbf{z}} (\mathbf{g}(\hat{\mathbf{x}}_v, \mathbf{z}))$$

and finally, $\mathbf{P}_{\hat{m}_{N+1},v}$ and $\mathbf{P}_{\hat{m}_{N+1},\hat{m}_1}$ are given by,

$$\mathbf{P}_{\hat{m}_{N+1},v} = \nabla_x \mathbf{g}(k|k) \mathbf{P}_v(k|k) \quad (3.34)$$

$$\mathbf{P}_{\hat{m}_{N+1},\hat{m}_1} = \nabla_x \mathbf{g}(k|k) \mathbf{P}_{v,\hat{m}_1}(k|k) \quad (3.35)$$

In this EKF-SLAM approach, lines were chosen as features as they can be easily extracted from structured urban environments. A detailed formulation on how these lines are used as features was presented which included how features were transformed between different reference frames and how newly observed features are incorporated into the state vector.

3.4 Occupancy Grid Mapping

As discussed earlier, a feature-based map only contains features that fit into its defined feature models and will exclude any other types of features in the environment. A feature-based map will therefore unlikely to be able to represent the environment in high detail. Occupancy grid mapping, also known as evidence grids, is another mapping paradigm which uses an array of cells to describe the environment. This technique was first conceived by Moravec and Elfes in the 1980s [23]. It has since been widely used in both 2D [18][53] and 3D [42] modeling of the environment as well as for navigation

[10] and path planning [4]. The main advantage of using occupancy grid maps is they are capable of modeling objects of any shape in the environment, depending on the size of the cells, as well as producing highly detailed maps. The derivations of the full posterior for the grid map and its log-odds representation are based on works by [6][46][53]

3.4.1 Occupancy Grid Definitions

Given an occupancy map defined as $\mathbf{M}$, each cell in this map can then be defined as $m_{x,y}$ where the subscripts x and y refer to the coordinates of the cell with respect to the global navigation frame. Each cell in this map can then be defined accordingly by the given data set to be empty or occupied. However, if a grid map has 10000 grid cells, the number of unique map configurations that $\mathbf{M}$ can represent is 2^{10000} [55]. This makes the calculation of posterior probability for each map that can be represented unmanageable.

To sidestep this problem, the occupancy probability for every grid cells in the map is calculated instead. The occupancy probability is a value in the interval $[0, 1]$. A cell with a value, $p(m_{x,y}) = 1$ denotes with complete certainty that this cell is occupied and $p(m_{x,y}) = 0$ denotes with complete certainty that this cell is unoccupied (or empty). Assuming that the position of the platform is known, the full posterior of the occupancy grid map can be calculated as $p(\mathbf{M}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k})$ where $\mathbf{x}_{0:k}$ and $\mathbf{z}_{1:k}$ are the platform's position and the set of range sensor (such as sonar, radar or laser range finder) measurements up to time k respectively. However, in the original formulation as discussed in [3][4], a zero-order Markov random field for the map was assumed so that each individual cell can be estimated as an independent random variable. By applying Bayes rule to recursively estimate the full posterior of the map, it can be updated as shown by [53] using the following,

$$p(\mathbf{M}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}, \mathbf{M})p(\mathbf{M}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})} \quad (3.36)$$

Hence the posterior probability for each cell, $m_{x,y}$, can be formulated as,

$$p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}, m_{x,y})p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}. \quad (3.37)$$

One of the common assumptions made in occupancy grid mapping is that the world is *static* [46]. This assumes that given knowledge of the map $\mathbf{M}$, past sensor readings are conditionally independent from the current set of readings as well as from the platform's past position for any point in time k ,

$$p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}, \mathbf{M}) = p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{M}) \quad (3.38)$$

or for each cell as,

$$p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}, m_{x,y}) = p(\mathbf{z}_k|\mathbf{x}_k, m_{x,y}). \quad (3.39)$$

Hence Eqn. 3.37 can be further simplified as,

$$p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, m_{x,y})p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})} \quad (3.40)$$

and by applying Bayes rule to the term $p(\mathbf{z}_k|\mathbf{x}_k, m_{x,y})$, yields,

$$p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) = \frac{p(m_{x,y}|\mathbf{x}_k, \mathbf{z}_k)p(\mathbf{z}_k|\mathbf{x}_k)p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{p(m_{x,y})p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})} \quad (3.41)$$

The posterior probability that a cell is empty can be found through an analogous derivation of Eqn. 3.41. This emptiness will be denoted as $\bar{m}_{x,y}$,

$$p(\bar{m}_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) = \frac{p(\bar{m}_{x,y}|\mathbf{x}_k, \mathbf{z}_k)p(\mathbf{z}_k|\mathbf{x}_k)p(\bar{m}_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{p(\bar{m}_{x,y})p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})} \quad (3.42)$$

It is possible to eliminate some of the terms when dividing Eqn. 3.41 by Eqn. 3.42 as,

$$\frac{p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k})}{p(\bar{m}_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k})} = \frac{p(m_{x,y}|\mathbf{x}_k, \mathbf{z}_k)p(\bar{m}_{x,y})p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{p(\bar{m}_{x,y}|\mathbf{x}_k, \mathbf{z}_k)p(m_{x,y})p(\bar{m}_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})} \quad (3.43)$$

Eqn. 3.43 is a form of odds ratio and when defining $p(\bar{m}_{x,y}) = 1 - p(m_{x,y})$ and $p(\bar{m}_{x,y}|\cdot) = 1 - p(m_{x,y}|\cdot)$, where “.” refers to any conditioning variables, it can be re-written as,

$$\frac{p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k})}{1 - p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k})} = \frac{p(m_{x,y}|\mathbf{x}_k, \mathbf{z}_k)}{1 - p(m_{x,y}|\mathbf{x}_k, \mathbf{z}_k)} \frac{1 - p(m_{x,y})}{p(m_{x,y})} \frac{p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{1 - p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})} \quad (3.44)$$

For reasons of numerical instability for probabilities that are near 0 or 1 [6] and for computational efficiency in the recursive calculation of occupancy probabilities [53], Eqn. 3.44 is represented in log-odds expression as,

$$\begin{aligned} \log \frac{p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k})}{1 - p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k})} &= \log \frac{p(m_{x,y}|\mathbf{x}_k, \mathbf{z}_k)}{1 - p(m_{x,y}|\mathbf{x}_k, \mathbf{z}_k)} + \log \frac{1 - p(m_{x,y})}{p(m_{x,y})} + \\ &\quad \log \frac{p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{1 - p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})} \end{aligned} \quad (3.45)$$

or,

$$l_{x,y}^k = \log \frac{p(m_{x,y}|\mathbf{x}_k, \mathbf{z}_k)}{1 - p(m_{x,y}|\mathbf{x}_k, \mathbf{z}_k)} + \log \frac{1 - p(m_{x,y})}{p(m_{x,y})} + l_{x,y}^{k-1} \quad (3.46)$$

where $l_{x,y}^k$ is the log-odds representation of the occupancy probability value for cell x, y at time k given that,

$$l_{x,y}^{k-1} = \log \frac{p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})}{1 - p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})} \quad (3.47)$$

The occupancy probability for each cell can still be easily recovered from the log-odds

by,

$$p(m_{x,y}|\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) = 1 - \frac{1}{\exp(l_{x,y}^k)}. \quad (3.48)$$

During initialization of the grid map, each cell will have a occupancy probability value of $p(m_{x,y}|\mathbf{x}_0, \mathbf{z}_0) = 0.5$ or $l_{x,y}^0 = 0$, which places an equal weight on the probability of the cell being empty or occupied.

3.4.2 Occupancy Grid Mapping Algorithm - The Inverse Sensor Model

With the definitions addressed, in this section the concept of occupancy grid mapping will be presented in details. The standard occupancy grid mapping concept can be divided into two main parts, namely the *Area of Interest* and the standard *Inverse Sensor Model*. The *Area of Interest* defines the area in which the laser range finder sensor is capable of detecting the presence of objects. The *Inverse Sensor Model* is the occupancy grid mapping paradigm used to update the occupancy probability value of the cells that are within the *Area of Interest*.

Area of Interest

Figure 3.6 shows a 2-D occupancy grid map centered around a 360° laser range finder sensor. The position of the sensor is depicted to be within the red cell and its maximum range is encapsulate by the green circle. The area encapsulated is defined as the *Area of Interest* in this report and cells within this area are shaded gray and the rest white. As the name implies, cells that exist within area are of particular interest since objects

Figure 3.6: A typical 2-D occupancy grid map. The maximum range of a 360° scanning exteroceptive range finding sensor is depicted by green circle, the gray cells are the cells of interest.

that exist within this area can be detected by the sensor. These are the cells whose occupancy probability values need to be updated at that instant of the mapping cycle at time k via the standard *Inverse Sensor Model*. The white cells outside of the *Area of Interest* are not within the perceptual view of the range sensor. Therefore it is not possible to determine if any objects exist within these cells. Thus, these cells will retain their previous log-odds values as of time $k-1$ as $l_{x,y}^k = l_{x,y}^{k-1}$.

Operating Principle of the Inverse Sensor Model

The standard *Inverse Sensor Model* was proposed by Thrun [55]. It will match each cell with the nearest (in terms of angular difference) laser range finder data points. Upon a successful match, a ray-tracing method is used to determined whether the cell is

occupied using the matched range data. In the following discussion, the algorithm is for one single row of cells only, however it can be extended to an entire 2-Dimensional grid map by looping the algorithm to include all of the cells.

Figure 3.7: Mapping of object in a 1-Dimensional environment.

Figure 3.7 shows a 1-Dimensional grid map with an object placed within the perceptual sight of the laser range finder sensor. The solid line depicts the line-of-flight of the laser from the sensor to object in space. D_{e_max} and D_{e_meas} indicate the maximum sensing range of the range finder and the measured object range respectively. D_{m_8} is the distance between the sensor and the center of the cell (m_8) in which the object is located while D_{width} is the width of the cell.

Taking reference to the scenario shown in Figure 3.7, the standard *Inverse Sensor Model* employs the ray-tracing method to determine which cells in the grid map fulfill the following condition,

$$D_{m_i} - (D_{width}/2) \leq D_{e_meas} \leq D_{m_i} + (D_{width}/2) \quad \text{for } 1 \leq i \leq 12 \quad (3.49)$$

which indicates that the particular cell is being occupied by an object. Any cells with

range (D_{m_i}) shorter than the measured range (D_{e_meas}) implies that they are empty (etc. cell m_1 to m_7). However, for cells whose distance from the sensor that are longer than the measured range, or these cells are “behind” the object, their occupancy status cannot be determined as the range finder would not be able to detect if any objects exist in the areas (etc. cell m_9 to m_{12}). Hence their occupancy probability will remain the same as their previous values. The algorithm for the standard *Inverse Sensor Model* is shown in Algorithm 1.

```

1 Inputs:  $D_{m_i}, D_{e\_meas}, D_{e\_max}, l_{x,y}$ 
2 Constants:  $l^{Unknown}, l^{Occ}, l^{Emp}$ 
3 begin
4   if  $(D_{m_i} - D_{width}/2) > \min(D_{e\_max}, D_{e\_meas})$  then
5      $\text{return } l_{x,y} + l^{Unknown}$ 
6   else if  $(D_{e\_meas} < D_{e\_max}) \ \&\& \ (|D_{m_i} - D_{e\_meas}| < D_{width}/2)$  then
7      $\text{return } l_{x,y} + l^{Occ}$ 
8   else if  $(D_{m_i} + D_{width}/2) < \min(D_{e\_max}, D_{e\_meas})$  then
9      $\text{return } l_{x,y} + l^{Emp}$ 
10  end
11 end

```

Algorithm 1: Standard Inverse Sensor Model algorithm.

$l^{Unknown}$, l^{Occ} and l^{Emp} are the log-odds representation of the belief of the probability that the cells are in *unknown*, *occupied* and *empty* state based on the instantaneous sensor (sonars, radars or laser range finders) readings respectively. For this thesis, the values of the above mentioned variables are 0, 1 and -1 respectively. The condition in Line 4 of the algorithm will search for cells such as m_9 to m_{12} as shown in Figure 3.7. The first conditional parameter, $(D_{m_i} - D_{width}/2)$, determines the distance between the sensor and the particular cell’s edge which is nearest to the sensor. This distant will then be compared with the second conditional parameter to determine if the cell is “behind” the object or if the cell is outside of the *Area of Interest*. Given that $D_{e_meas} < D_{e_max}$, this indicates that the laser range finder detects an object within its exteroceptive field

of view, hence the measured distance is shorter than the maximum sensing range. If $(D_{m_i} - D_{width}/2) > D_{e_meas}$, this indicates that the cell is “*behind*” the object and the occupancy status of this cell cannot be determined. Given the same scenario again but $D_{e_max} < D_{e_meas}$, implies that the measured distance exceeds the maximum sensing range capability of the sensor and there are no objects existing within the exteroceptive view of the sensor. Hence, when $(D_{m_i} - D_{width}/2) > D_{e_max}$, these cells exist outside the *Area of Interest* of the grid map. It is impossible to determine the occupancy status of the grid cells in these two categories, hence these cells will retain their previous occupancy probability values. Line 6 search for cells such as m_8 . The first conditional comparison $(D_{e_meas} < D_{e_max})$ determines if the measured range, D_{e_meas} , is shorter than, D_{e_max} , which indicates that the sensor has detected an object within its field of view. The second conditional comparison $(|D_{m_i} - D_{e_meas}| < D_{width}/2)$ determines if the object is within the space that the cell is representing. If these two conditions are fulfilled, it can then be said that an object does indeed exist in that particular cell. The condition in Line 8 is similar to that in Line 4 except that it searches for cells which are “*in front*” of the object.

This concept of the standard *Inverse Sensor Model* in 1-Dimensional mapping can be extended to 2-Dimensional mapping with extra criteria to select the most appropriate sensor data to match with. Figure 3.8 shows a simple 2-Dimensional occupancy grid map with a small section of laser range data points. Dotted lines were drawn to join the laser data points to the position of the sensor to highlight the direction of each data point with respect to the sensor’s location.

The particular cell of interest for the following discussion is bound by the black solid lines as shown in Figure 3.9. The red points (with red dotted lines) indicate that they are not within the space of the interested cell. The yellow dot indicates the cell’s center-of-mass while the dotted yellow line indicates the direction of the cell with respect

Figure 3.8: Figure illustrating a simple 2-Dimensional grid map with a small section of laser range finder data points.

Figure 3.9: A zoomed view into the cell of interest. The yellow dot depicts the center of the cell in interest while the blue and cyan dots indicates the position where the laser range finder detects some objects.

to the sensor. The length of this yellow line would be equivalent to $D_{m_{x,y}}$ (or D_{m_i} in a 1-Dimensional grid map context as discussed earlier). The criteria for the standard 2-Dimensional *Inverse Sensor Model* is to match the data point which has the least directional angular distance to the direction of the cell in interest. In the case shown in Figure 3.9 there is a match between the cell (yellow dot) and the data point with the least angular distance (blue dot). Although the cyan dots are also within the cell, they will not be considered as their directional angular difference are much larger than the blue dot. The selection criteria is performed before Algorithm 1 and the full 2-Dimensional standard *Inverse Sensor Model* algorithm is shown below,

```

1 Inputs:  $m_x, m_y, v_x, v_y, v_\theta, D_z, D_{e\_max}, l_{x,y}$ 
2 Constants:  $l^{Unknown}, l^{Occ}, l^{Emp}$ 
3 begin
4   // to determine the index of the laser data with minimum angular difference
5    $D_{m_{x,y}} = \sqrt{(m_x - v_x)^2 + (m_y - v_y)^2}$ 
6    $\phi = \text{atan2}((m_y - v_y), (m_x - v_x)) - v_\theta$ 
7    $z_{index} = \text{argmin}|\phi - \theta_z|$ 
8
9   // Standard Inverse Sensor Model occupancy probability calculation process
10  if  $(D_{m_{x,y}} - D_{width}/2) > \min(D_{e\_max}, D_{z_{index}})$  then
11    | return  $l_{x,y} + l^{Unknown}$ 
12  else if  $(D_{z_{index}} < D_{e\_max}) \ \&\& \ (|D_{m_{x,y}} - D_{z_{index}}| < D_{width}/2)$  then
13    | return  $l_{x,y} + l^{Occ}$ 
14  else if  $(D_{m_{x,y}} + D_{width}/2) < \min(D_{e\_max}, D_{z_{index}})$  then
15    | return  $l_{x,y} + l^{Emp}$ 
16  end
17 end

```

Algorithm 2: 2-D standard Inverse Sensor Model algorithm.

In Algorithm 2, x_v, v_y and v_θ are the vehicle position and orientation. m_x and m_y are the position of the cell of interest while z_{index} is an index that refers to the range point that has the least directional angular distance. By executing the algorithm from Line 5 to Line 7, the appropriate laser point to be used in the standard *Inverse Sensor Model* can be determined.

3.4.3 Multi Resolution Occupancy Grid Mapping

Although occupancy grid maps have been widely used in robotic mapping to build a highly-detailed map of the environment, in a study described in [12], the authors noted that occupancy grid maps, although easy to construct and maintain, consume a large amount of computational time for large-scale environments. In this section, the concept of the *Multi Resolution Occupancy Grid Mapping* (MROGM), which is a simplified version of the quadtree grid mapping [49], will be presented. It aims to model the environment with high resolution efficiently while avoiding complicated map management.

The idea of *Multi Resolution Occupancy Grid Mapping* (MROGM) is motivated by the work of Burlet [28], Chan [11], Cheol-Taek [35], Fairfield [42] and Gerhard [20] on constructing multi-resolution grid maps. [11][35] also proposed multi-resolution grid maps for robot navigation while [42] proposed 3-D octree occupancy grid map in underwater environments. The concept behind these techniques is the partitioning of selected cells into smaller sized cells to increase the resolution. Furthermore, in the original discussion in [42], it was mentioned that “Large contiguous portions of an evidence grid are either empty, occupied, or unknown, and can be efficiently represented by a single octnode - truncating all the children which would have the same value”. This implies that if the low-resolution parent grid cell is deemed to have the same occupancy probability value throughout its defined space, its children cells that tessellate the same space would have the same occupancy probability value as well.

High Resolution Environmental Modeling Through MROGM

Figure 3.10: a) shows an example of a low resolution grid map of an environment. The equivalent high resolution grid map is shown in b).

Figure 3.10 shows standard occupancy grid maps with different resolutions. Figure 3.10(a) shows a lower resolution grid map to that in Figure 3.10(b), however Figure 3.10(b) presents a more detailed map, indicating more precisely which space in the environment is occupied. This is achieved by using a smaller grid cell size (higher resolution) to describe a spatial space. However, by inspecting the areas which are empty (indicated in the figures by blue cells), it can be realized that it is unnecessary to model these areas with a high resolution map. The low resolution map can represent these areas very accurately as well.

Figure 3.11 shows the two different resolution grid maps from Figure 3.10 superimposed. In this figure only a small part of the high resolution grid map is superimposed onto the low resolution map to avoid confusion. The cells belonging to the high resolution grid map are being defined by the green lines while the low

Figure 3.11: Comparison of cells and area tessellated by different resolution grid maps. Although high resolution maps are needed to model the environment accurately, however empty spaces can still be described by low resolution maps without losing any important information.

resolution grid cells are defined by the black lines. From earlier discussion, only the high resolution grid maps are able to model the environment in high detail, describing exactly where the obstacles are (dark red cells) in the surrounding. However when comparing the empty spaces (dark blue cells), it can be seen that these spaces can actually be described by low resolution grid map cells without losing any critical information. Furthermore, the occupancy probability values for these spaces, be it that described by either the low resolution cells or high resolution cells are the same. Hence this implies that a low resolution grid map can be used to describe empty spaces in the environment, while high resolution cells are needed to describe the rest of the space.

The approach of MROGM utilizes the efficiency of low resolution grid maps in representing empty areas and that of high detail modeling of the environment via a high resolution grid map. MROGM will search for areas that need to be modeled in greater

detail (by referring to the rest of the areas that are *not empty* as indicated by the low resolution grid map) and perform the necessary grid mapping processes on these areas of the high resolution map. This approach will help reduce the overall computational time needed to maintain an equally detailed map.

Time Efficiency of MROGM

As presented earlier, MROGM needs at least two occupancy grid maps (or more, depending on applications) with different resolutions where the low resolution map is being used as the reference map and the high resolution map is the actual grid map built. Given that the time needed for the *Improved Inverse Sensor Model* to calculate the occupancy probability value of a grid cell is ΔT_{ISM} , the total time needed to maintain a grid map during one mapping cycle is therefore,

$$\Delta T_{ISM}(m \times n) \quad (3.50)$$

where m and n are the number of rows and columns of cells in the grid map respectively. Assuming only two maps are being used, an assumption made for the rest of the thesis, the times required to maintain an equivalent low resolution, T_{LR} , and the high resolution map, T_{HR} , taking reference to Eqn. 3.50, are given by

$$T_{LR} = \Delta T_{ISM}(m_1 \times n_1), \quad (3.51)$$

and

$$T_{HR} = \Delta T_{ISM}(m_1 n_1 \times m_2 n_2) \quad (3.52)$$

respectively. m_1 and n_1 are the number of rows and columns of cells in the low resolution grid map while m_2 and n_2 are the number of rows and columns of cells *in* a single low resolution cell. This definition is further explained in Figure 3.12. Based on the MROGM approach and the derivations above, the time required for each MROGM grid mapping maintenance cycle, T_{MROGM} , is givens as,

$$T_{MROGM} = \Delta T_{ISM}(m_1 n_1 + c_o(m_2 n_2)) \quad (3.53)$$

where c_o is the number of cells in the low resolution MROGM reference map that are not empty. In normal applications, $T_{MROGM} \ll T_{HR}$.

Figure 3.12: Definitions of grid cells used in MROGM computation time derivation.

3.5 Fundamentals of HYbrid Metric Maps (HYMMs)

The *HYbrid Metric Maps* (HYMMs) was originally proposed by Nieto et. al[31]. As compared to traditional hybrid mapping with both feature maps and occupancy grid maps, this approach allows for the propagation of corrections in the base landmarks to the local grid maps. In their original work, point-features were used as vertex points to partition what are known as the *Local Triangular Regions* (LTRs). These regions are

then represented by local small sized high resolution grid maps where each local grid map will be taking reference to a particular point feature base landmark.

Figure 3.13: Definition of relative features in point-feature based hybrid mapping.

Figure 3.13 shows the concept of HYMMs using LTRs and local grid maps. The blue solid dots indicate the vertex point features which were used to define the LTRs. The green solid dots indicate all the other point features which were not used as vertex points. Any objects or features which exist within these regions will be represented in each LTR's respective local grid map. In the original discussion, it was mentioned that a "relative representation stores close landmarks in local coordinate frames and then reduces correlation to the rest of the map..." which implies that if the relative features observed by the autonomous platform are close by and these features are referenced to a local coordinate frame, estimation updates of any features within this local coordinate frame are independent from the rest of the map [29].

The relationship between a relative position of a particular point feature and the base landmark, as shown in Figure 3.14, can be derived as

Figure 3.14: The definition of relative features in point-feature based hybrid mapping.

$$\begin{aligned}
 \mathbf{x}_r + \mathbf{x}_b &= \mathbf{x}_v + \mathbf{z} \\
 \mathbf{x}_r &= \mathbf{x}_v + \mathbf{z} - \mathbf{x}_b \\
 &= \mathbf{h}(\mathbf{x}_v, \mathbf{z}, \mathbf{x}_b)
 \end{aligned} \tag{3.54}$$

where $\mathbf{x}_r$, $\mathbf{x}_v$, $\mathbf{z}$ and $\mathbf{x}_b$ represents the position of the relative feature, the platform, the relative feature as observed by the platform and the base landmark respectively. Applying the common assumption used for SLAM that all observations are independent of the vehicle position, the cross-correlation between the relative feature and the vehicle, $\mathbf{P}_{vr}$, is derived as,

$$\begin{aligned}
 \mathbf{P}_{vr} &= \mathbf{P}_{vv} \frac{\partial \mathbf{h}}{\partial \mathbf{x}_v} + \mathbf{P}_{vb} \frac{\partial \mathbf{h}}{\partial \mathbf{x}_b} \\
 &= \mathbf{P}_{vv}(1) + \mathbf{P}_{vb}(-1) \\
 &= \mathbf{P}_{vv} - \mathbf{P}_{vb}
 \end{aligned} \tag{3.55}$$

since $\mathbf{h}(\mathbf{x}_v, \mathbf{z}, \mathbf{x}_b) = \mathbf{x}_v + \mathbf{z} - \mathbf{x}_b$. $\mathbf{P}_{vv}$ and $\mathbf{P}_{vb}$ are the vehicle state covariance and the cross-correlation between the vehicle and the base landmark. Based on Eqn. 3.55, if the magnitudes of $\mathbf{P}_{vv}$ and $\mathbf{P}_{vb}$ are similar in a highly-correlated feature map (usually when SLAM converges over a period of time), it can be then be said that there will almost be no correlation between the vehicle position and the relative feature. This implies that updating of relative features in a particular LTR is an independent process and will not bring about any implications to the rest of the features in other LTRs.

3.6 Conclusion

This chapter presented a detailed review of the current robotic mapping paradigm which the proposed *Multi Resolution Hybrid Mapping* (MRHM) is based on. It included a discussion on line-feature based EKF-SLAM which provides recursive position estimation for the platform as well as the building of feature maps to be used in the *Hybrid Mapping*. A discussion on occupancy grid mapping paradigm was then presented. It described the fundamentals of occupancy grid mapping and examined the operating principles of the grid mapping algorithm, along with the standard *Inverse Sensor Model*. The *Multi Resolution Occupancy Grid Mapping* (MROGM), which is a simplified version of the quadtree mapping technique, is also presented. This is a simple and efficient method that is able to reduce the overall time need to construct multi-resolution grid maps to model the environment. Finally, this chapter examined the concept of *HYbrid Metric Maps* (HYMMs). It showed how the *Local Triangular Regions* (LTRs) are defined and how the correction of relative features within each LTR does not affect the state of other features that exist outside of the LTR.

Chapter 4

Technical Improvements of Robotic Mapping Paradigms

4.1 Introduction

This chapter presents the technical improvements to the current robotic mapping approach as well as proposing an extension to the current HYbrid Metric Mapping (HYMMs) technique. There are two main sections in this chapter.

The first section examines on the drawback that was found in the *Inverse Sensor Model* occupancy grid mapping algorithm. Although it is an efficient algorithm to construct occupancy grid maps, the algorithm may not work as intended when constructing low resolution maps. There may be instances where this algorithm mis-classifies the occupancy status of low resolution cells which may result in an inaccurate map. A solution, the *Improved Inverse Sensor Model* which is suitable for grid maps of all

resolutions, is proposed to overcome this drawback.

The second section presents the Multi Resolution Hybrid Mapping algorithm, which is an extension of the HYbrid Metric Mapping (HYMM) technique. As discussed in earlier chapters, this approach makes use of SLAM point cluster features as point of reference for small local grid maps. However, in a highly structured environment such as office buildings, lines can be more easily extracted and used as features. Hence, the proposed approach, Multi Resolution Hybrid Mapping (MRHM), looks into how line features can be used as base landmarks references for the local grid maps. The grid mapping paradigm used is the MROGM, that is presented in Chapter 3.4.3, which is able to construct high resolution grid maps efficiently.

4.2 Improved Inverse Sensor Model

In this section, the drawback of the standard *Inverse Sensor Model* when constructing low resolution grid maps will be examined in details. A solution which is capable of overcoming this drawback will then be presented.

4.2.1 Standard Inverse Sensor Model - The Drawback

Although the standard *Inverse Sensor Model* is capable of updating the occupancy probability value, there is a drawback which may degrade the effectiveness when using this algorithm, especially for low-resolution grid cells. Drawing a closer inspection to the formulation of the occupancy grid mapping algorithm discussed in Section 3.4.2, it was

shown that a cell was determined to be *occupied*, *empty* or *unknown* by comparing the following two parameters, $D_{m_{x,y}}$ and $D_{z_{index}}$. However, there is an association issue with may result in the erroneous determination of the cell's status.

Figure 4.1 shows an instance where the *Inverse Sensor Model* does not work when constructing a low resolution occupancy grid map. This figure is a replica of the same scene as shown in Figures 3.8 and 3.9 except that in this figure the occupancy probability is being calculated and plotted out. The grid map is plotted using colours where each represents a different degree of occupancy probability value of each cell. The corresponding colour for each of the occupancy probability value is shown on the colour bar on the side with dark red displaying a high probability that the cell is occupied and blue being a low probability of occupancy. In this figure, there are five instances where the standard *Inverse Sensor Model* is not performing as expected, which are highlighted by violet and magenta circles. The following discussion will focus on the misclassification of cell occupancy status circled by violet.

Similar to the previous discussion on the concept of the *Inverse Sensor Model*, the least directional angular distance laser data point (highlighted in brown instead of blue for colour contrast in Figure 4.2) is selected to determine the status of the cell. The cyan points are again not selected due to their larger directional angular distances. However the standard *Inverse Sensor Model* classified the cell as being empty where visual inspection shows clearly that the data points (the brown and cyan points) are well within the cell space.

To highlight the drawback of the standard *Inverse Sensor Model* algorithm when considering low resolution grid maps, Figure 4.3 shows the same section of the occupancy grid map as Figure 4.2 except that the laser range is being adjusted as shown in

Figure 4.1: Instance where standard Inverse Sensor Model do not work as expected when constructing a low resolution occupancy grid map.

Figure 4.2: Zoom-in view on one instance where standard Inverse Sensor Model did not work. By visual inspection, the cell in interest should be occupied, however the standard Inverse Sensor Model classified it as empty.

Figure 4.3: a) The original section of grid map plotted. b) The referenced laser data point falls beyond the boundaries set by the standard *Inverse Sensor Model*. c) The referenced laser data point falls with the boundaries set. d) The referenced laser data point falls before the boundaries set.

Figures 4.3(b), 4.3(c), and 4.3(d). The boundaries set by the standard *Inverse Sensor Model* according to Line 10 ($D_{m_{x,y}} - D_{width}/2$) and Line 14 ($D_{m_{x,y}} + D_{width}/2$) of Algorithm 2 are marked on these figures by magenta lines. It can be seen from Figures 4.3(b) and 4.3(d) that the referenced laser data points fall beyond and before the boundaries respectively. This causes the standard *Inverse Sensor Model* to classify the cell as being “unknown” and “empty” for the respective situations presented. For the situation where the reference point falls within the boundaries as shown in Figure 4.3(c), the algorithm is able to classify the cell’s occupancy probability status correctly as being occupied. This drawback can seriously affect the accuracy of low resolution occupancy grid maps, especially for a Multi-Resolution Occupancy Grid Map (MROGM), which will be discussed in later chapters, where this paradigm relies heavily on low resolution grid maps to improve the time efficiency in building dense grid maps.

4.2.2 Concept Of Improved Inverse Sensor Model

In this section, the approach of the proposed technique to overcome the drawback of standard *Inverse Sensor Model*, termed as the *Improved Inverse Sensor Model*, will be discussed in details. There are two main parts in this section. Part one examines the concept of the *Improved Inverse Sensor Model* while the second part presents the simulation to test the improved grid mapping algorithm as well as a discussion on the results obtained.

Figure 4.4 shows the proposed solution to overcome the drawback of the standard *Inverse Sensor Model*. This is the same simulated scene which generates Figure 4.2 except that the proposed *Improved Inverse Sensor Model* is being implemented. This is to highlight the difference in grid mapping performance between the standard and the

improved grid mapping algorithm.

Figure 4.4: Solution to overcome the drawback of standard *Inverse Sensor Model*.

Instead of relying on one screening process to calculate the occupancy probability values for all the grid cells in the map, the *Improved Inverse Sensor Model* proposes two screening processes to determine the occupancy status of each cell. The first process is to screen out which grid cells contain laser range finder data points, which indicate that these cells are occupied. The first process will round off the coordinates of all the data points (indicated by red dots) to the nearest grid cell center-of-mass. These rounded off data points are indicated by yellow dots in the figure. The cyan lines show the intended convergence of laser range finder data points to the center of respective cells upon rounding off. In accordance with the concept of grid mapping, these cells will be classified as occupied since there are data points within the space defined by them. The second process make use of the standard *Inverse Sensor Model* to classify the occupancy status (empty or unknown) for the rest of the cells.

As mentioned earlier, the algorithm for the *Improved Inverse Sensor Model* is built on top of the *Inverse Sensor Model* which has a total of two screening processes. The algorithm for the *Improved Inverse Sensor Model* is given below in Algorithm 3. A loop, as put forward in Line 5, is used to loop through the entire laser range finder data set to round off every data point to their nearest grid cell center-of-mass. This process is described in Line 6. This rounded off data will be used to search for the appropriate cells and set their occupancy status to occupied, as described in Lines 7 and 8 respectively. The second screening process is as presented by Algorithm 2 in Section 3.4.2. $R_{x,y}$ is the rounded-off value of z_i to the precision of D_{width} .

```

1 Layer 1 screening process of Improved Inverse Sensor Model
2 Inputs:  $z_i, m_{x,y}, l_{x,y}$ 
3 Constants:  $l^{Occ}$ 
4 begin
5   for all  $z_i$  do
6      $R_{x,y} = \text{round}(z_i, D_{width});$ 
7     if occupancy status of  $m_{R_x, R_y}$  has not been determined then
8        $\text{return } l_{x,y} + l^{Occ}$ 
9     end
10  end
11 end

```

Algorithm 3: Improved Inverse Sensor Model algorithm.

4.3 Multi-Resolution Hybrid Mapping

The concept of the *Multi Resolution Hybrid Mapping* (MRHM) is based on HYMMs and MROGM as presented previously in Chapter 3.5. Instead of using point features as base landmarks, the approach of MRHM proposes to use line features as base

landmarks. The high resolution metric mapping is based on MROGM as discussed earlier in Chapter 3.4.3, hence the name *Multi Resolution* and *Hybrid Mapping*. As lines can be easily extracted from highly structured environments such as offices or corridors, they will be more suitable than point features to be used as base landmarks in these situations.

Figure 4.5: Definition of relative features in line-feature based hybrid mapping.

For lines that are extracted using Hough transformation, they can be described by two Hough parameters, the line feature distance, ρ_{Gi} , and the direction of the line, θ_{Gi} , with respect to certain location, usually the origin. Hence, the coordinates of a point in the Cartesian coordinates system, $\mathbf{x}_b$ can be derived from these polar parameters as indicated by the circle in Figure 4.5. This point can then be used as a base landmark in the same approach as presented in the original HYMMs discussion in Chapter 3.5. $\mathbf{x}_r$ is the relative position of the relative feature with respect to the position of the base landmark while $\mathbf{x}_v$ and $\mathbf{z}$ are the platform's position and the landmark's position from the platform's view. The coordinates of $\mathbf{x}_b$ can be derived using

$$\begin{aligned}\mathbf{x}_b &= \begin{bmatrix} x_b \\ y_b \end{bmatrix} \\ &= \begin{bmatrix} \rho_{Gi} \cos(\theta_{Gi}) \\ \rho_{Gi} \sin(\theta_{Gi}) \end{bmatrix}\end{aligned}\tag{4.1}$$

$$\tag{4.2}$$

Thus, the relative feature's position can be determined by

$$\begin{aligned}\mathbf{x}_r + \mathbf{x}_b &= \mathbf{x}_v + \mathbf{z} \\ \mathbf{x}_r &= \mathbf{x}_v + \mathbf{z} - \mathbf{x}_b \\ &= \mathbf{g}(\mathbf{x}_v, \mathbf{z}, \mathbf{x}_b)\end{aligned}\tag{4.3}$$

Assuming that the position of the relative features are independent of the platform's location (as always assumed in SLAM [24]), the cross-correlation between the vehicle and the relative data points is then express as

$$\begin{aligned}\mathbf{P}_{vr} &= \mathbf{P}_{vv} \frac{\partial \mathbf{g}^T}{\partial \mathbf{x}_v} + \mathbf{P}_{vb} \frac{\partial \mathbf{g}^T}{\partial \mathbf{x}_b} \\ &= \mathbf{P}_{vv}(1) + \mathbf{P}_{vb}(-1) \\ &= \mathbf{P}_{vv} - \mathbf{P}_{vb}\end{aligned}\tag{4.4}$$

where $\frac{\partial \mathbf{g}^T}{\partial \mathbf{x}_i}$ is the jacobian matrix of $\mathbf{g}$ with respect to the state $\mathbf{x}_i$. Making the same assumption presented in Chapter 3.5 for point-feature base landmark, there will not be any cross-correlation between the relative features and the vehicle when the magnitude

of $\mathbf{P}_{vv}$ and $\mathbf{P}_{vb}$ are equal. Hence, it is possible to use line features as base landmarks instead of point features as presented in the original discussion for HYMMs.

4.4 Conclusion

In this chapter, technical improvement to some of the existing robotic mapping techniques were presented. The presented *Improved Inverse Sensor Model* seeks to improve the current *Inverse Sensor Model* by accurately classifying the occupancy status of the occupancy grid cells. Finally, the concept of MRHM is presented. MRHM proposes to use line features, rather than point features, as base landmark since lines can easily be extracted from highly structured environments such as corridors or offices. The implementation of algorithms, experimental setup and results will be presented in the next chapter.

Chapter 5

Simulations and Experiments - Setups and Results

In this chapter, the simulation and experimental setups and results obtained will be presented. There are two main sections in this chapter, describing the simulation and experiment. The simulation section presents the simulation setup followed by the results obtained. The experimental section will be presented in the same manner. The last section discusses the *Multi Resolution Hybrid Mapping* (MRHM) preliminary results. The implementation of all the simulation programs and algorithms mentioned, including obtained results, in this report was by the sole effort of author except for the Hough lines extraction algorithm and the field experiment data set¹.

5.1 Simulation Setups

A simulation program was developed to test the algorithms presented in this

¹Hough Transformation lines extraction algorithm and field experiment data set courtesy of Mr. Sok Chhay, NTU

Figure 5.1: This simulation program allows the user to input certain parameters such as the simulation environment and initial platform position through its user interface.

report such as the *line-feature based EKF-SLAM*, the *Multi Resolution Occupancy Grid Mapping*, *Improved Inverse Sensor Model* as well as the proposed *Multi Resolution Hybrid Mapping*. It was developed using C++ programming language together with Qt GUI library under the Linux operating system. Figure 5.1 shows the user interface of the simulation. This simulation program allows the user to load different types of pre-defined environments as well as defining several parameters such as initial position and orientation. It allows the user to select the simulated platform's kinematic speed and the sensor sampling time interval. The platform's speed and its yaw angular rate feedback were corrupted with noise to simulate a real platform system. Some other functions such as zooming and panning functions were also designed to enable the user to focus on particular areas for further visual analysis.

As shown in the above mentioned figure, an environment was loaded for testing

Figure 5.2: This simulation program also allows the user to view the grid map constructed during simulation.

the implemented algorithms. Focusing on the left display window of the first tabbed display, the white lines indicate the obstacles while the gray lines indicate the platform's path. The green dots are the simulated laser data points of the environment. This data was simulated to be corrupted with noise similar to what an actual laser sensor would experience. The right window displays the platform's position ground truth, the positional estimates based on the simple kinematic model as well as the EKF-SLAM positional estimates. Also being displaying in this window are the laser data points and the line-features that were extracted from the data points.

Figure 5.2 shows the grid map constructed during the simulation. This map will be constructed online during the simulation to give the user an immediate visual feedback on the integrity of the grid mapping algorithm implemented. This can be accessed through the second tab of the simulation program. Zooming and panning functions are

also available in this tabbed display.

5.2 Simulation Results

5.2.1 EKF-SLAM Simulation Results

The EKF-SLAM algorithm was implemented and tested using the simulation program mentioned in the previous section. The implementation of EKF-SLAM algorithm, except for the Hough line extraction algorithm, was by the sole effort of the author. Figure 5.3 shows the simulation results of the line-feature based EKF-SLAM. The green, orange and cyan solid points shown in the left display window indicate the ground truth, the position estimation based on simple kinematic model and the EKF-SLAM positional estimate respectively. Figure 5.4 shows a zoomed view of the positional estimates during the simulation. The orange positional estimates were closer to the ground truth as compared to the cyan positional estimates. This implies that the implemented EKF-SLAM algorithm is performing as intended, providing more accurate positional estimates than the simple kinematic model.

One important aspect of SLAM is *loop-closing* as this can determine the performance of the SLAM algorithm [45]. When zooming into the starting/ending point of this simulation, as shown in Figure 5.5, it can be seen that the cyan positional estimates however have started to drift away from the ground truth due to the noise corrupted platform speed and yaw angle feedback. The final orange positional estimate, however, almost overlaps the green dot. Through the EKF-SLAM architecture,

it managed to correct the simulated noise corrupted control inputs to achieve loop-closing.

Figures 5.6, 5.7 and 5.8 show the position estimation errors and magnitude of errors for six simulation experiments represented by different colours. It is shown in all three figures that the positional errors estimated by the line-feature based EKF-SLAM algorithm were significantly smaller than those of the simple kinematic model in every simulation. This shows that, as with the results presented in earlier figures, EKF-SLAM is capable of using lines as features to provide accurate positional estimates.

Figure 5.3: Line-features based EKF-SLAM simulation results.

Figure 5.4: Zoomed view of the EKF-SLAM simulation's starting and ending point.

Figure 5.5: Zoomed view of the EKF-SLAM simulation's starting and ending point.

Figure 5.6: Comparison of simulation positional error: x-direction.

Figure 5.7: Comparison of simulation positional error: y-direction.

Figure 5.8: Comparison of simulation positional error: error magnitude.

5.2.2 Improved Inverse Sensor Model Simulation Results

As discussed in Chapter 3.4.2, the concept of the standard *Improved Inverse Sensor Model* to determine the occupancy status of cells in an occupancy grid map is through the means of using boundaries. However, as presented in Chapter 4.2.1, there are situations where this approach is unable to accurately determine the occupancy status of the cells, especially for low resolution grid maps. Hence, in the same chapter, an improved model, the *Improved Inverse Sensor Model*, was proposed to overcome this drawback. Experiments are carried out to observe the performance of the proposed *Improved Inverse Sensor Model*, as compared to the *Inverse Sensor Model*, when constructing grid maps.

Figures 5.9(a) and 5.9(b) show a side-by-side comparison of the grid maps constructed using these two paradigms. The cyan and brown lines in Figure 5.9(a) indicate the

(a) 2-Dimensional occupancy grid map constructed using the standard *Inverse Sensor Model*.

(b) 2-Dimensional occupancy grid map constructed using the *Improved Inverse Sensor Model*.

Figure 5.9: Comparison of grid maps constructed using the standard *Inverse Sensor Model* and the *Improved Inverse Sensor Model*.

direction and position of the data points residing within the cell of interest. The yellow line indicates the direction and position of the cell's center-of-mass. The magenta dots in Figure 5.9(b) show the position of the rounded-off data points while the cyan dot indicates the platform's position. The side-by-side comparison clearly shows that the grid map constructed by the *Improved Inverse Sensor Model* (Figure 5.9(b)) does not mis-classify the occupancy status of the grid cells as compared to when using the standard *Inverse Sensor Model*. This methodology ensures that all sensor data points are used to determine the occupancy status instead of just relying on the smallest angular-difference data points. Furthermore, it is suitable for constructing grid maps of any resolution. The occupancy status of rest of the cells were classified using the standard *Inverse Sensor Model* as the drawback does not affect the accuracy in determining the status of these cells.

5.2.3 Multi Resolution Occupancy Grid Mapping Simulation Results

Chapter 3.4.3 presents a time efficient approach to constructing grid maps, the *Multi Resolution Occupancy Grid Mapping* (MROGM). This method of grid mapping inherits the dual advantages of efficient mapping through low resolution grid maps and dense metric modeling of the environment by high resolution grid maps. A simple experiment was carried out to construct grid maps of different resolution, using different grid mapping paradigms to analyze the performance of MROGM as compared to the standard grid mapping approach. Experiments using the simulation program as presented in Chapter 5.1 will also be conducted and their results discussed. The *Improved Inverse Sensor Model* algorithm discussed in earlier sections is integrated into the grid mapping algorithms that will be used in this section. It should be noted that for this simulation,

it was assumed that the position estimates were accurate and the grid maps built were based solely based on these position estimates.

(a) An overview of the simulation environment along with the platform's intended path as indicated by the blue lines.

(b) Zoomed view of the simulation environment into a protruding corner meant to be modeled by high resolution grid maps.

Figure 5.10: Overview of simulation environment.

Figure 5.10(a) shows the simulation environment which the grid maps of different resolutions will be modeling. The black solid lines indicate obstacles in the environment. A short protruding corner, as circled by red in Figure 5.10(b), was intentionally included to highlight the dense metric modeling of the environment, which the high resolution maps are capable of. The blue lines in these figures show the platform's path during the simulation.

Figures 5.11 to 5.13 showed grid maps constructed using the standard occupancy grid mapping approach and the *Multi Resolution Occupancy Grid Mapping* (MROGM) approach. It can be seen that these maps generally are able to model the environment, clearly indicating where the obstacles are in the environment. However, it can also be observed that the low resolution grid map (Figure 5.11(b)) was unable to show where the protruding corner was. Instead, it modeled the vicinity as being occupied. The high resolution (Figure 5.12(b)) and multi resolution (Figure 5.13(b)) grid maps, on the other hand, were able to clearly showed where this protruding corner was. When comparing the level of detail between the high resolution grid map and the multi resolution grid map, comparable results are evident.

The cumulative time required to construct occupancy grid maps of different resolution are shown Figure 5.14. As expected, the time required to construct a low resolution grid map is the least. Although MROGM maps have the most amount of cells, the time required is much less than what is required to construct a standard high resolution occupancy grid map. The only time where MROGM is the least efficient was during the first grid mapping cycle, where no MROGM cells were empty as shown in Figure 5.15. Once empty cells are identified, the time required immediately shortens. Hence, the MROGM paradigm is capable of constructing a comparable multi resolution grid map with reduced time.

(a) Overview of environment modeled using low resolution grid map.

(b) Zoomed view of environment modeled using low resolution grid map.

Figure 5.11: Modeling of environment using low resolution grid map.

(a) Overview of environment modeled using high resolution grid map.

(b) Zoomed view of environment modeled using high resolution grid map.

Figure 5.12: Modeling of environment using high resolution grid map.

(a) Overview of environment modeled using multi resolution grid map.

(b) Zoomed view of environment modeled using multi resolution grid map.

Figure 5.13: Modeling of environment using multi resolution grid map.

Figure 5.14: Cumulative time required to construct occupancy grid maps using different mapping paradigms.

Figure 5.15: MROGM is the least efficient only at the first grid mapping cycle.

Figures 5.16 to 5.18 show grid maps of different resolution modeling the simulated environment using the above mentioned grid mapping paradigms. These figures show results which are consistent with those presented earlier in this section whereby the map constructed by MROGM is comparable to the high resolution grid map, modeling the environment in high detail. Furthermore, Figure 5.19 also shows that the time required by MROGM to construct its grid map is significantly less than the standard high resolution grid map throughout the entire simulation.

Figure 5.16: Low resolution grid mapping of the simulation environment.

Figure 5.17: High resolution grid mapping of the simulation environment.

Figure 5.18: Multi resolution grid mapping of the simulation environment.

Figure 5.19: Cumulative time spent by the various grid mapping paradigms during simulation.

5.3 Experiment Setup

An experiment was carried out to test the EKF-SLAM and MROGM algorithms. The experiment was conducted in the vicinity of the Mechanical and Aerospace Engineering (MAE) car park, Nanyang Technological University (NTU), as shown in Figure 5.20. The structured buildings around this car park made this vicinity an ideal location for line-feature based EKF-SLAM.

The starting/ending point is indicated in the figure by the red solid dot while the path which the vehicle took is defined by the blue lines. The yellow lines indicate possible line-features that can be used by the EKF-SLAM algorithm. Two laser range finders were mounted on top of a saloon car to retrieve environmental data to be used by EKF-SLAM algorithm. GPS was also mounted to provide positional ground truth information. For this experiment, the vehicle drove around the car park three times (three loops) while the laser range finders collected data. Ground truth information for the experiment was taken

Figure 5.20: View of carpark to the EKF-SLAM and MROGM algorithms.

before the commencement of the experiment and it is shown as cyan points in Figure 5.21.

5.4 Experiment Results

In this section, results obtained from the experiment will be presented. This section consists of two main parts. The first part presents the results of line-feature based EKF-SLAM while the second part presents the MROGM results. The EKF-SLAM and MROGM algorithms were implemented as off-line processes as the data were collected before the development of these algorithms.

5.4.1 EKF-SLAM Experiment Results

The results obtained from the line-feature based EKF-SLAM algorithm is shown in

Figure 5.21: GPS ground truth around MAE car park.

Figure 5.22. The EKF-SLAM and the dead-reckoning position estimation is indicated by the green and white points respectively. It can be clearly seen (Figure 5.23) that the dead-reckoning positional estimates drifted and was not able to close the loop, however, the positional estimates provided by the EKF-SLAM indicates that the loop has been closed.

When zooming into the starting and ending point of the experiment, it is shown in Figure 5.24 that error in the EKF-SLAM estimate of the vehicle's final position was approximately 2 meters while the dead-reckoning estimation error was 16m. This field result agrees with the simulation results presented earlier showing that EKF-SLAM algorithm is able constantly provides a more accurate estimate than dead-reckoning.

Figure 5.22: EKF-SLAM results for the experiment in MAE car park.

Figure 5.23: Zoomed view of the EKF-SLAM results for experiment in MAE car park.

Figure 5.24: Close up view of the EKF-SLAM results at starting / ending point.

5.4.2 Multi Resolution Occupancy Grid Mapping Experiment Results

It should be noted that the positional information provided by the EKF-SLAM algorithm, as discussed in previous section, was assumed to be accurate and used directly to construct the grid maps presented here. A low resolution grid map representation of the MAE car park is shown Figure 5.25. This figure is able to show where the walls and trees are (refer to Figure 5.20) in the environment. However, when comparing to Figure 5.26, the latter was able to outline the trees and pillars in the vicinity, showing the environment in greater detail.

When examining the map constructed by MROGM (Figure 5.27) together with the

high resolution map, although some parts of the MROGM map displayed low resolution modeling of the environment, the two maps were comparable to each other. The time required to construct these three maps by different grid mapping paradigms are shown in Figure 5.28. Conforming to the concept of MROGM, as well as the simulation results discussed in Chapters 3.4.3 and 5.2.3 respectively, MROGM spent much less time than what was required to construct a high resolution map, and was able to construct a comparable multi-resolution grid map.

Although it was mentioned earlier that the experiment was conducted in a car park with buildings in the vicinity (refer to Figure 5.20), the grid maps, as shown above, indicate there to be open spaces. This phenomena was due to the glass window panels as circled in red in Figure 5.29. These window panels reflect the ranging lasers away, causing the sensor to interpret there to be free space in that the direction. This can be observed from the initial grid map constructed and shown in Figure 5.30. The yellow dots indicate the maximum detection range of the laser range finder, which also represents the cells along that direction to be free space. When comparing this figure to Figure 5.29, it can be observed that the yellow dots, circled by cyan, are ranging laser points that were reflected off from the window panels. However, this reflection of ranging laser points it does not affect the performance of the mapping paradigm used to construct the grid maps as presented earlier in this chapter.

Figure 5.25: Low resolution grid map representation of the MAE car park.

Figure 5.26: High resolution grid map representation of the MAE car park.

Figure 5.27: Multi resolution grid map representation of the MAE car park.

Figure 5.28: Cumulative time required to construct MAE car park grid maps of different resolution.

Figure 5.29: Window panels causing a discrepancy between the grid map and actual environmental structure.

Figure 5.30: Window panels causing a discrepancy between the grid map and actual environmental structure.

5.5 Multi Resolution Hybrid Mapping Setup and Results

As discussed in Chapter 4.3, the *Multi Resolution Hybrid Mapping* (MRHM) is able to update the information of the local relative features without affecting the rest of the map. This concept is extended to update the information about the position and direction of the local grid map such that a more accurate representation of the environment can be constructed.

Figure 5.31: *Multi Resolution Hybrid Mapping* simulation setup.

A simulation, as shown in Figure 5.31, was developed to investigate the implementation and performance of the MRHM in a simple environment. As the proposed MRHM is based on line-features for localization, a non line-feature obstacle (shown by the inserted

zoomed view), whose information will not be corrected by the EKF-SLAM algorithm, was added to the environment. This is to investigate if the grid map constructed by the MRHM will be able to accurately model this obstacle. In this simulation, only one local grid map will be used to model the environment. This is to reduce the complexity of implementation so that the concept of using line features as base landmarks can be assessed in details.

As only one base landmark is being used in the following simulations, it is not possible to define an LTR which can corrects the grid map by means of redefining the LTR boundaries. Hence, the propagation of correction from the feature map to the grid map is achieved through simple concatenation as shown in Figure 5.32. The dotted line shows the concatenation link between the feature map and the grid map. It is assumed that this link is rigid, hence, any positional corrections made to the base landmark will propagate to the grid map as well.

Figure 5.32: The dotted line indicates the rigid concatenation link between the feature map and the grid map. The feature map corrections are propagate to the grid map via this link.

For comparison purposes, the above mentioned simulation was conducted twice. The

first simulation was conducted *without* the implementation of MRHM algorithm, while the second simulation was conducted *with* the MRHM algorithm. The EKF-SLAM algorithm is used to provide positional estimates while the *Improved Inverse Sensor Model* and MROGM algorithms will be used to construct the grid map in both simulations.

The left display window of Figure 5.33 shows the grid map constructed *without* the implementation of MRHM while Figure 5.34 shows the grid map constructed using MRHM algorithm. In both of these figures, the line-feature map is indicated by the yellow lines overlapping the grid maps. As mentioned earlier, the MRHM seeks to improve the accuracy of the environmental representation through the constructed grid map. A side-by-side comparison of these grid maps shows that the map constructed using the MRHM algorithm is able to represent the environment more accurately. There is much less misclassification of the cell status as circled in red in Figure 5.35. When zooming into the vicinity where the non line-feature obstacle was, as shown circled in blue in Figure 5.36, although it was not conclusive on whether MRHM indeed represented this obstacle in a more accurate manner, the MRHM grid map was indeed able to represent the rest of the environment more accurately.

These preliminary results show that it is possible to use lines as base landmarks to improve the accuracy of the grid maps constructed. However, more simulations will be needed before a conclusive answer can be arrived at on the performance of the MRHM.

Figure 5.33: Grid map constructed without implementation of MRHM.

Figure 5.34: Grid map constructed with implementation of MRHM.

Figure 5.35: Comparison of grid maps constructed with and without implementation of MRHM.

Figure 5.36: Zoomed-in comparison of grid maps constructed with and without implementation of MRHM.

5.6 Conclusion

This chapter presented the simulation setup and results obtained from the implementation of the mapping algorithm as discussed in Chapter 4. It is shown that in all the simulations and experiments conducted, line-feature based EKF-SLAM is able to provide much more accurate positional estimates than dead-reckoning in a highly structured environment.

This chapter also compares the results of the standard *Inverse Sensor Model* and the proposed *Improved Inverse Sensor Model*. It is shown that the *Improved Inverse Sensor Model* grid mapping approach is able to determine the status of the cells more accurately, especially for those with low resolution. This chapter then moved on to present the MROGM paradigm. The results obtained showed that the grid maps constructed using this mapping were comparable to high resolution grid maps constructed using the standard approach. In addition, the cumulative time needed to construct such maps is much less.

Finally the chapter presented the MRHM algorithm results. These preliminary results showed that line-features can indeed be used as base landmarks to aid in the correction of the local grid maps. However more studies will be needed to improve the implementation as well as the efficiency of this approach.

Chapter 6

Conclusion and Recommendations

This chapter summarizes the work presented in this thesis, including the contributions made to improve the efficiency of *Hybrid Mapping*. Recommendations and future work to further improve Hybrid Mapping will also be discussed.

6.1 Conclusion

In recent years, autonomous robotics has been put into the limelight due to escalating interest from both military and industrial parties to develop machines capable of replacing humans to perform highly dangerous or laborious tasks. A fundamental ability required of an autonomous robot is to build a map of its surroundings whilst localizing itself within this map. Such an ability allows an autonomous robot to know where it is with respect to an environment and together with other path planning abilities to be able to determine where it should go. Time efficiency is another important aspect in autonomous robotics as it will determine whether the robot is indeed suitable for autonomous missions.

The preliminary obtained results from the proposed *Multi Resolution Hybrid Mapping* showed that the implementation of *Improved Inverse Sensor Model* is able to improve the accuracy and reduce the time needed to construct multi resolution grid maps. It also shows that line features can be used as base landmarks for corrections be propagated from the feature maps to the occupancy grid maps.

6.2 Recommendations and Future Work

This thesis makes use of Hough-lines as features to refine positional estimates for building multi-resolution occupancy grid maps. However, the preliminary results presented only make use of one line feature as base landmark and one local grid map to model the environment. In order to implement MRHM into field systems, more base landmarks and local grid maps will be needed. Hence, one of the future work will be designing and developing an approach to include more base landmarks and local grid maps, and a decision process to online partition the environments into *Local Trigular Regions* (or LTRs as mentioned in HYMMs) using base landmarks. These LTRs can then also be used to redefine the *Area of Interest* in the MROGM.

Another recommendation is to include both point features and line features into MRHM algorithm. As mentioned in this thesis, MRHM is suitable in structured environments where line features can be easily extracted. However, if the autonomous platform is to transverse from outdoor environment into structured environment, it will be more appropriate if both types of features are being modeled into the algorithm.

As presented in this thesis, the propagation of error corrections is currently unidirec-

tional, i.e., from the feature maps to the grid maps. To achieve the goal of feeding the information from the grid map back to the feature map and close the correction loop, an appropriate grid map matching technique is required. This will further improve the accuracy of both the feature maps and the grid maps. Considering that matching usually requires a large amount of resources, the chosen technique should not compromise the overall efficiency of the MRHM algorithm.

Bibliography

- [1] A. Bandera, C. Urdiales and F. Sandoval, “An hierarchical approach to grid-based and topological maps integration for autonomous indoor navigation”, in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Systems* (IROS’01), pp.883-888, Nov. 2001
- [2] A. Garulli, A. Giannitrapani, A. Rossi, and Vicino, A., “Mobile robot SLAM for line-based environment representation”, in *Proc. of the 44th IEEE Conf. on Decision and Control 2005 and 2005 European Control Conf.* (CDC-ECC’05), pp.2041-2046, Dec. 2005
- [3] A. Elfes, “Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation”, *PhD thesis*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1989
- [4] A. Elfes, “Using occupancy grids for mobile robot perception and navigation”, *Computer*, 22(6):46-57, Jun. 1989
- [5] A. Ranganathan and F. Dellaert, “Inference in the space of topological maps: an MCMC-based approach”, in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Systems* (IROS’04), pp.1518-1523, Sept./Oct. 2004
- [6] A. Souza, A. Santana, R. Britto, L. Goncalves and A. Medeiros, “Representation of Odometry Errors on Occupancy Grids”, in *Proc. of Int. Conf. on Informatics in Control, Automation and Robotics* (ICINCO’08), Funchal, Portugal, May 2008
- [7] B. Barshan and H. Durrant-Whyte, “Inertial Navigation Systems for Mobile Robots”, *IEEE Trans. Robot. Autom.* 11(3):328-342, June 1995
- [8] B. Kuipers, and Byun Y.T. “A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations”, *J. of Robotics and Autonomous Systems*, 8(1-2):47-63, 1991
- [9] B. Schiele, and J.L. Crowley, “A comparison of position estimation techniques using occupancy grids”, in *Proc. of the IEEE Int. Conf. on Robotics and Autom.* (ICRA’94), pp.1628-1634, May 1994

- [10] B.A. Merhy, P. Payeur and E.M. Petriu, "Application of Segmented 2D Probabilistic Occupancy Maps for Mobile Robot Sensing and Navigation", in *Proc. of the IEEE Instrumentation and Measurement Technology Conf (IMTC'06)*, pp.2342-2347, Apr 2006
- [11] Chan R.H.T., Tam P.K.S. and Leung D.N.K., "Robot navigation in unknown terrains via multi-resolution grid maps", in *Proc. of the Int. Conf. on Ind. Electron., Control and Instrum.* (IECON'91), pp.1138-1143, Oct./Nov. 1991
- [12] Chen C. and Cheng Y.H., "Research on Map Building by Mobile Robots", in *Second Int. Symp. on Intell. Inform. Technology Applicat.* (IITA'08), pp.673-677, Dec. 2008
- [13] Choi Y.H., Lee T.K. and Oh S.Y., "A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot", *Autonomous Robot*, 24(1):13-27, 2008.
- [14] DARPA Grand Challenge 2004, Available: <http://www.darpa.mil/grandchallenge04/>
- [15] DARPA Grand Challenge 2005, Available: <http://www.darpa.mil/grandchallenge05/>
- [16] DARPA Grand Challenge 2007, Available: <http://www.darpa.mil/grandchallenge/>
- [17] D. Dufourd, R. Chatila and D. Luzeaux, "Combinatorial maps for simultaneous localization and map building (SLAM)", in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS'04)*, pp.1047-1052, Sept./Oct. 2004
- [18] D. Pagac, E. Nebot and H.F. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles", *IEEE Trans. on Robotics and Autom.*, 14(4):623-629, Aug. 1998
- [19] E. Ivanjko, and I. Petrovic, "Extended Kalman filter based mobile robot pose tracking using occupancy grid maps", in *Proc. of the 12th IEEE Mediterranean Electrotechnical Conf.* (MELECON'04), pp.311-314, May 2004
- [20] G. Kraetzschmar, G. Gassull and K. Uhl, "Probabilistic quadtrees for variable-resolution mapping of large environments", in *Proc. of the 5th IFAC/EURON Symp. on Intell. Autonomous Vehicles*, M. I. Ribeiro and S. J. Victor, *Eds*, Lisbon, Portugal, July 2004
- [21] He F., Fang Y.C., Wang Y.T and Ban T., "Practical feature-based simultaneous localization and mapping using sonar data", in *Proc. of the 27th Chinese Control Conf.* (CCC'08), pp.421-425, Jul. 2008
- [22] Huang G.Q., A.B. Rad and Wong, Y.K., "Online SLAM in dynamic environments", in *Proc. of the 12th Int. Conf. on Advanced Robotics (ICAR'05)*, pp.262-267, Jul. 2005
- [23] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar", in *Proc. of the IEEE Int. Conf. on Robotics and Autom.* (ICRA'85), pp.116-121, Mar. 1985

- [24] H.F. Durrant-Whyte, and T. Bailey, "Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms", *IEEE Robot. Automat. Mag.* 13(2):99-110, Jun. 2006
- [25] H.J. von der Hardt, D. Wolf and R. Husson, "The dead reckoning localization system of the wheeled mobile robot ROMANE", in *Proceedings of the IEEE/SICE/RSJ Int. Conf. on Multisensor Fusion and Integration for Intell. Systems*, pp.603-610, Dec. 1996
- [26] Jung M.J., Myung H., Hong S.G, Park D.R., Lee H.K. and Bang S.W, "Structured light 2D range finder for simultaneous localization and map-building (SLAM) in home environments," in *Proc. of the 2004 Int. Symp. on Micro-Nanomechatronics and Human Science 2004 and The Fourth Symp. Micro-Nanomechatronics for Information-Based Society 2004*, pp.371-376, Oct./Nov. 2004
- [27] J. Blanco, J. Fernandez-Madrigal and J. Gonzalez, "A New Approach for Large-Scale Localization and Mapping: Hybrid Metric-Topological SLAM", in *Proc. of the IEEE/RSJ Int. Conf. on Robots and Autom. (ICRA'07)*, pp.2061-2067, Apr. 2007
- [28] J. Burlet, O. Aycard, and T. Fraichard, "Robust motion planning using Markov decision processes and quadtree decomposition", in *Proc. of the IEEE/RSJ Int. Conf. on Robots and Autom. (ICRA'04)*, pp.2820-2825, Apr./May 2004
- [29] J. Guivant and E. Nebot, "Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms", *IEEE Trans. Robot. Autom.* 19(4):749-755, Aug. 2003
- [30] J. Guivant, E. Nebot, J. Nieto and F. Masson, "Navigation and mapping in large unstructured environments", *Intl. J. of Robotics Research*, 23(4):449-472, Apr. 2004
- [31] J. Nieto, J. Guivant and E. Nebot, "The HYbrid metric maps (HYMMs): a novel map representation for DenseSLAM", in *Proc. of the IEEE Int. Conf. on Robotics and Autom. (ICRA'04)*, pp.391-396, Apr./May 2004
- [32] J.D. Tardos, J. Neira, P. Newman and J.J. Leonard, "Robust mapping and localization in indoor environments using sonar data", *Int. J. Robotics Research*, 21(4):311-330, Apr. 2002
- [33] J.J. Leonard, H.F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot", in *Proc. of the IEEE/RSJ Int. Workshop on Intell. Robots and Systems (IROS'91)*, pp.1442-1447, Nov. 1991
- [34] Kao W.W, "Integration of GPS and dead-reckoning navigation systems", *Vehicle Navigation and Information Systems Conf.*, pp.635-643, Oct. 1991
- [35] Kim C.T. and Lee J.J., "Mobile robot navigation using multi-resolution electrostatic potential field", in *31st Annu. Conf. of the IEEE Ind. Electron. Society 2005 (IECON'05)*, pp.1774-1778, Nov. 2005

- [36] Kim J.H. and S. Sukkarieh, "Robust multi-loop airborne SLAM in unknown wind environments", in *Proc. of the IEEE Int. Conf. on Robots. and Autom. (ICRA'06)*, pp.1536-1541, May 2006
- [37] Ko Bang-Yun, Song Jae-Bok and Lee Sooyong, "Real-time building of a thinning-based topological map with metric features", in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS'04)*, pp.1524-1529, Oct. 2004
- [38] K. Shojaie and A.M. Shahri, "Iterated Unscented SLAM algorithm for navigation of an autonomous mobile robot", in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS'08)*, pp.1582-1587, Sept. 2008
- [39] Liu B.B., M. Adams and J. Ibanez-Guzman, "Multi-aided Inertial Navigation for Ground Vehicles in Outdoor Uneven Environments", in *Proc. of the IEEE/RSJ Int. Conf. on Robots and Autom. (ICRA'05)*, pp.4703-4708, Apr. 2005
- [40] Luo R.C., Lin S.C. and Lai C.C., "Indoor autonomous mobile robot localization using natural landmark," in *Proc. of the 34th Annual Conf. of IEEE on Industrial Electronics (IECON'08)*, pp.1626-1631, Nov. 2008
- [41] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.* 17(3):229-241, June 2001
- [42] N. Fairfield, G. Kantor, and D. Wettergreen, "Real-time slam with octree evidence grids for exploration in underwater tunnels", *J. of Field Robotics*, 24(1-2):3-21, Feb 2007
- [43] O. Ozisik and S. Yavuz, "An occupancy grid based SLAM method", in *Proc. of the IEEE Int. Confe. on Computational Intell. for Measurement Systems and Applicat. (CIMSAS'08)*, pp.117-119, Jul. 2008
- [44] O. Matsebe, S. Holtzhausen, C.M. Kumile and N.S. Tlale, "Modeling the Kinematics of an Autonomous Underwater Vehicle for Range-Bearing Simultaneous Localization and Mapping", in *Proc. of the 15th Int. Conf. on Mechatronics and Machine Vision in Practice (M2VIP'08)*, pp.412-417, Dec 2008
- [45] P.M. Newman and Ho K., "SLAM - loop closing with visually salient features", in *Proc. of the IEEE Int. Conf. on Robotics and Autom. (ICRA'05)*, pp.635-642, Apr 2005
- [46] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots", in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS'02)*, pp.1014-1019, Dec. 2002
- [47] R. Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics", in *Proc. of the IEEE Int. Conf. on Robotics and Autom. (ICRA'87)*, pp.850-850, Mar 1987

- [48] R. Vazquez-Martin, P. Nunez, J.C. del Toro, A. Bandera and F. Sandoval, "Adaptive Observation Covariance for EKF-SLAM in Indoor Environments using Laser Data", in *Proc. of the IEEE Mediterranean Electrotechnical Conf. (MELECON'06)*, pp.445-448, May 2006
- [49] R.A. Finkel, and J.L. Bentley, "Quad trees a data structure for retrieval on composite keys", *J. of Acta Informatica*, 4(1):1-9, 1974
- [50] R.O. Duda and P.E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures", in *Commun. of the ACM Mag.*, 15(1):.11-15, Jan. 1972
- [51] Su L.Y., Yu Y.Q. Chen W., Cao Z.Q. and Tan M., "Extraction of Topological Map Based on the Characteristic Corners from Grid Environment Exploration," in *Proc. of the IEEE Conf. on Robotics, Automation and Mechatronics*, pp.1-6, Dec. 2006
- [52] S. Sukkarieh, E. Nebot and H.F. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications", *IEEE Trans. on Robotics and Autom.*, 15(3):572-578, Jun. 1999
- [53] S. Thrun, "Learning Occupancy Grid Maps with Forward Sensor Models", *J. of Autonomous Robots*, 15(2):111-127, 2003
- [54] S. Thrun, "Robotic mapping: A survey", *Report CMU-CS-02-111*, Computer Science Dept, Carnegie Mellon University, Pittsburgh, PA, 2001
- [55] S. Thrun, W. Burgard and D. Fox, "Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)", *The MIT Press*, 2005
- [56] Xu D., Tan M. and Chen G., "An improved dead reckoning method for mobile robot with redundant odometry information", in *7th Int. Conf. on Control, Automation, Robotics and Vision 2002 (ICARCV'2)*, pp.631-636, Dec. 2002
- [57] Yap T.N. and C.R. Shelton, "SLAM in large indoor environments with low-cost, noisy, and sparse sonars", in *Proc. of the IEEE Int. Conf. on Robotics and Autom. (ICRA'09)*, pp.1395-1401, May 2009
- [58] Yu H.L., R.W. Beard and J. Byrne, "Vision-based local multi-resolution mapping and path planning for Miniature Air Vehicles", in *Proc. of the American Control Conf. (ACC'09)*, pp.5247-5252, June 2009
- [59] Zang Y.T., Yuan K., Zou W. and Hu H.S, "A Two-Step Particle Filter for SLAM of Corridor Environment," in *Proc. of the IEEE Int. Conf. on Inform. Acquisition*, pp.370-375, Aug 2006