ON CONSISTENT MAPPING IN DISTRIBUTED ENVIRONMENTS USING MOBILE SENSORS

A Dissertation

by

ROSHMIK SAHA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2011

Major Subject: Aerospace Engineering

ON CONSISTENT MAPPING IN DISTRIBUTED ENVIRONMENTS USING MOBILE SENSORS

A Dissertation

by

ROSHMIK SAHA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Suman Chakravorty
Committee Members,	Aniruddha Datta
	John L. Junkins
	Nancy Amato
	Raktim Bhattacharya
Head of Department,	Dimitris Lagoudas

August 2011

Major Subject: Aerospace Engineering

ABSTRACT

On Consistent Mapping in Distributed Environments
Using Mobile Sensors. (August 2011)
Roshmik Saha, B.Tech., Indian Institute of Technology, Kharagpur
Chair of Advisory Committee: Dr. Suman Chakravorty

The problem of robotic mapping, also known as simultaneous localization and mapping (SLAM), by a mobile agent for large distributed environments is addressed in this dissertation. This has sometimes been referred to as the holy grail in the robotics community, and is the stepping stone towards making a robot completely autonomous. A hybrid solution to the SLAM problem is proposed based on "first localize then map" principle. It is provably consistent and has great potential for real time application. It provides significant improvements over state-of-the-art Bayesian approaches by reducing the computational complexity of the SLAM problem without sacrificing consistency. The localization is achieved using a feature based extended Kalman filter (EKF) which utilizes a sparse set of reliable features. The common issues of data association, loop closure and computational cost of EKF based methods are kept tractable owing to the sparsity of the feature set. A novel frequentist mapping technique is proposed for estimating the dense part of the environment using the sensor observations. Given the pose estimate of the robot, this technique can consistently map the surrounding environment. The technique has linear time complexity in map components and for the case of bounded sensor noise, it is shown that the frequentist mapping technique has constant time complexity which makes it capable of estimating large distributed environments in real time. The frequentist mapping technique is a stochastic approximation algorithm and is shown to converge to the true map probabilities almost surely. The Hybrid SLAM software is developed in the

C-language and is capable of handling real experimental data as well as simulations. The Hybrid SLAM technique is shown to perform well in simulations, experiments with an iRobot Create, and on standard datasets from the Robotics Data Set Repository, known as Radish. It is demonstrated that the Hybrid SLAM technique can successfully map large complex data sets in an order of magnitude less time than the time taken by the robot to acquire the data. It has low system requirements and has the potential to run on-board a robot to estimate large distributed environments in real time.

To the loving memory of my beloved mother,

who had wished for this day till her last breath.

ACKNOWLEDGMENTS

I would like to take the opportunity to acknowledge life which gives us the opportunity to explore and investigate the unknown. I would like to express my gratitude to all the great minds that have lived on this planet, especially the ones who were not recognized for their contributions.

I would like to express my deepest gratitude towards my advisor, Dr. Suman Chakravorty, for his continued support and guidance without which this work could never have been completed. His contagious passion for research and the insightful discussions has not only made it a great learning experience for me, but also has left a permanent impression for life. I would like to thank my committee members, Dr. Aniruddha Datta, Dr. John L. Junkins, Dr. Nancy Amato and Dr. Raktim Bhattacharya for their critical comments and guidance. I greatly appreciate Dr. John E. Hurtado for agreeing to be a part of my preliminary examination at the last moment. His comments thereof and his valuable discources on dynamics as a wonderful teacher have made a lasting impression on me. I am immensely indebted to Dr. Raktim Bhattacharya for being a source of inspiration and wisdom to fight the challenges of research and beyond. I am grateful to all members of the Aerospace Department for providing such a compassionate and resourceful environment to grow as an individual. I would like to especially thank Karen Knabe for not only answering all my SOS calls with a smile, but also for motivating me and helping me focus on work by assisting with all kinds of department and graduate school formalities.

Special thanks to James Doebbler and Clark Moody for making the experiments at LASR Lab possible. I would like to acknowledge and thank Andrew Howard for collecting and publishing the SDR and USC SAL datasets, and Maxim Batalin for the Intel dataset.

I offer my deepest gratitude to Sandy and Vinny for providing me with the sense of home when I needed it the most. I would like to thank my good friends - Ashivni, Mrinal, Gaurav, Parikshit, Derek, Taylor, Baljeet, Nikhil and Avinash, for making my stay in College Station enriching and memorable. I would like to thank my family for instilling in me the quest of knowledge and for always being so supportive. Over everything else, I would like to remember the loving memories of my mother, Mrs. Shikha Saha, whose departure last year leaves a big void in my life. I will always be grateful to her for giving me this life, teaching me how to live it, and for having confidence in me. Her love, support and encouragement cannot be replaced and have been instrumental in completing this work.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Robotics and artificial intelligence (AI) have received considerable attention in the research community and are fast evolving fields. Mobile robots constitute an important part of the spectrum of robots. The environments these robots work in, and their functions, vary greatly. They could be used in places where the conditions are hazardous or pose a threat to human life, for example, detection of land mines, poisonous gas leakage or fire, and mapping abandoned underground mines or other inaccessible places. They could be used to simply reduce human effort as in the case of service robots in an office environment, or to increase efficiency and reduce human error. The primary focus of research in this area is to minimize human involvement in the working of the robot and to improve efficacy of robot performance and robustness. To ensure that the robot will be performing the tasks efficiently without failure and that the activities are scalable in terms of environment size and duration of robot runs, the underlying algorithms should be analyzed for computational complexity, mathematical consistency and convergence. These basic notions which are vital will be analyzed and addressed for all the algorithms that are developed and used throughout this work.

The three basic capabilities which are fundamental for a mobile robot to become autonomous are mapping, localization and path planning without human intervention. Firstly, for a mobile robot to perform different tasks, it needs to be aware of its surroundings. Many times a priori knowledge of the environment is not available. In such conditions the robot should be able to build the map of its environment based on

---

The journal model is *IEEE Transactions on Automatic Control*.

the information it receives from its exteroceptive sensors. This is called the mapping or map-building problem. This dissertation provides a real time solution to build high fidelity maps autonomously. Secondly, the robot should be aware of its own position and orientation relative to the map of its environment while navigating through its workspace. An accurate Global Positioning System (GPS) would greatly help in this regard but GPS signal is not always available, for instance, indoors and underwater. In other situations it may be jammed, for example, in a war zone. Hence, in the absence of a reliable GPS connectivity, the robot should be able to localize itself based on its map information, surrounding observations and odometry and/or inertial measurement unit(IMU) readings. This is called the localization problem for a mobile robot navigating the environment. In the current work raw odometry readings from the robot motion actuators and observation of the surrounding from exteroceptive sensor(s) are assumed to be present.

If the map is not known a priori, it is easy to see that the mapping and localization problems get coupled as the observation of the surrounding is used for both localization and mapping. This results in a search problem in a high-dimensional space which is computationally expensive to solve. In the robotics community this is called the simultaneous localization and mapping (SLAM) problem and will be addressed in this dissertation. Thirdly, the robot should be able to navigate itself from one point in the map to another, preferably in an optimal fashion. This is referred to as the path planning problem. When the robot dynamics is also considered, it is referred to as the motion planning problem. During the process of map building, path planning becomes more challenging as then it has to deal with the uncertainty in the map. A truly autonomous mobile robot should be able to perform all of the above tasks simultaneously and is sometimes referred to as simultaneous planning, localization and mapping (SPLAM). An hierarchical approach to motion planning

under uncertainty has been presented in [1] and [2] which could be used along with the SLAM solution provided in this dissertation to achieve SPLAM.

This dissertation introduces "Frequentist mapping", a novel stochastic approximation based approach to map large distributed/dense environments. It is called the frequentist approach since it is based on the Law of large numbers. Robotic SLAM for large distributed environments under sensor uncertainty in real time, while maintaining consistency, has remained a formidable problem in the robotics community for about 20 years. The pertinent issue in the state-of-the-art Bayesian approaches to the SLAM problem is that consistency and complexity are at cross purposes. This issue is addressed in the current work by following a "first localize and then map" policy and is called the Hybrid SLAM solution. It is shown how a few prominent sparse features (or landmarks) could be used to localize and get the probability density function (*pdf*) on the robot pose. This is a sufficient statistic for the robot pose and is also called the belief state.

An Extended Kalman Filter (EKF) based algorithm, which is the Bayesian filter under the Gaussian assumption, is used for the localization part. It retains the consistency guarantees while avoiding computation and data association issues, by retaining only sparse features. Given the belief (*pdf*) on the pose of the robot from the EKF, the frequentist algorithm is used to map the distributed workspace. It is inherently immune to the data association problem and is provably strongly consistent. Its complexity is linear in the map components. Further, when the sensor noise is assumed to be bounded, the complexity of the frequentist algorithm is constant time. Thus, the structure of the problem is exploited to give a computationally tractable solution while preserving consistency. Consistent results are provided across simulations, real experiments and standard datasets. Hence the case is made that the proposed method can solve the SLAM problem for very large distributed environments online,

and in a provably consistent fashion.

The remainder of this chapter is arranged as follows: Subsection A presents a brief history of research in stochastic mapping highlighting the important achievements and challenges in solving SLAM. The following references [3, 4, 5] are suggested for the interested reader. In subsection B, the motivation for the current research is developed from the existing challenges and avenues for development. In subection C, the organization of this dissertation is presented.

A. Background

Robotic mapping has been an active research area in robotics and AI for more than two decades now. It deals with the problem of acquiring spatial models of the physical environment using mobile robots. This problem is the stepping stone to make mobile robots autonomous as maps are needed for navigation. This problem has even been referred to as the holy grail by the community [6, 3]. Despite significant progress in this area, it still poses great challenges. Mapping unstructured, dynamic, and large-scale environments still remain open problems at large [5]. Robotic mapping has been referred to as CML or SLAM, which is short for concurrent mapping and localization [7, 8] and simultaneous localization and mapping [9, 10, 11, 12] respectively. The latter became more popular for obvious reasons and is now used quite ubiquitously as a synonym for robotic mapping. The problem of SLAM for an autonomous system is to start at an unknown location in an unknown environment and then to incrementally build a consistent map of this environment while simultaneously determining its location in the map using the current estimate of the map. As can be seen, the robot needs to estimate the spatial map but at the same time needs the surrounding map to localize itself in the map. This leads to a philosophical chicken and egg

problem, which results in very high computational burden. This makes solving the SLAM problem very challenging, especially, implementing it as a real time, consistent, incremental algorithm which scales to large unstructured environments.

The field of mapping can be divided into metric and topological approaches based on the kind of information stored in the map. The metric maps capture the exact geometrical properties of the map while the topological maps simply describe the connectivity of the different places in the map. Some examples of the work done using topological maps are [13, 14, 15, 16, 17, 18] but they are not geared towards providing efficient navigation. In practice, metric maps are finer grained than the topological maps. Though higher resolution comes with higher computational cost, it is better suited for navigation and various robot activities. Topological maps are sometimes used in conjunction with metric maps to improve consistency [19, 20, 21].

Most of the successful SLAM solutions employ some kind of metric map representation. They are either the lower level feature based maps, or the more information rich grid based maps. Feature or landmark based maps are represented by geometrical features. The most commonly used features are points and lines. The mapping algorithm in this case tries to estimate the spatial location of the features or landmarks [11, 22, 23]. The feature based maps do not lend themselves well to unstructured environments since it cannot capture all the details of the map that are important for navigation, path planning etc. The grid based maps are based on the discretization of the entire map into small grids which can contain information of various kinds, the simplest being if the grid is occupied or empty. It could potentially contain other kind of information, for example, texture, gas concentration, luminance or temperature. The important work of Elfes and Moravec [24, 25, 26] introduced the occupancy grid (OG) representation which uses 2D spatial grids to model the occupied and free space in the map. Since then a lot of work has been done using grid based maps

as in [27, 28, 29]. The grid based maps are much more rich in information than the feature based maps and by simply reducing the grid size higher resolution maps can be generated. Many researchers are trying to come up with efficient solutions to SLAM in this context [30, 31]. Recently, a combination of grid based and feature based maps has been used [32]. The Hybrid SLAM solution being proposed also uses both a feature map and a grid based map.

The genesis of probabilistic SLAM may be traced back to the 1986 IEEE Robotics and Automation Conference with contributions from Peter Cheeseman, Jim Crowley, Hugh Durrant- Whyte, Raja Chatila, Oliver Faugeras, Randal Smith, and others [3]. This was the time when researchers in the field of robotics and AI were beginning to employ estimation-theoretic methods to mapping and localization. Over the next few years a series of key papers were published. References [33] and [34] established a statistical basis for describing relationships between the landmarks. Kalman filter based algorithms were used in [35, 36, 37] which would later become one of the most commonly used methods to solve SLAM. This led to the seminal work by Smith et al. [38] which developed the concept of a stochastic map and used the Extended Kalman Filter (EKF) to estimate the state of the stochastic map. They also showed that the landmark errors necessarily get correlated as the robot moves around in the environment due to the common pose error of the robot [39]. It meant that a consistent estimate of the map would require a joint state composed of the pose and all the features and a full state update would be required for each observation. At this time the convergence properties of the estimated map was not studied and it was widely assumed that it would not converge and instead exhibit a random walk like behavior with unbounded error growth. Lack of the convergence knowledge, and given the computational complexity of maintaining and updating the cross correlations, researchers assumed or even forced the correlations between the landmarks to be

minimized or eliminated [40]. The conceptual breakthrough came with the realization that the joint estimate of the map and the robot pose is actually convergent. It was also realized that the correlations between the features in the EKF based SLAM algorithms (EKF-SLAM), which researchers had tried to minimize or eliminate, were integral to the filter convergence. The more the correlations grew, the better the solution of the filter was [41, 42]. Thus the EKF-SLAM was formalized and solved at a conceptual level. Even today, the EKF based methods are the most commonly used methods to solve the SLAM problem and will be used in the dissertation for localization.

Researchers then started to address the issues related to implementation and realization in real world for the EKF based SLAM solutions. The first issue is the computational complexity of the algorithm which scales as the cube of the number of map features when implemented in its naive form. The problem formulation has a peculiar structure where the process model only affects vehicle pose states and the observation model only makes reference to a single vehicle-landmark pair. A wide range of techniques have been developed to exploit this special structure to reduce the computational complexity of EKF-SLAM and can be primarily characterized as being either optimal or conservative. Optimal algorithms reduce the computational complexity using the structure of the problem and still result in the same solution as the basic formulation. Conservative algorithms, also called sub-optimal filters, result in estimates that have higher uncertainty or covariance than the optimal solution because of approximations involved in the method. Algorithms which result in uncertainties less than the optimal solution are called inconsistent and are considered invalid solutions as they lead to divergence.

The covariance prediction, which has a cubic complexity can be reduced to linear complexity in the map features by simply exploiting the structure that the process

model only effects the robot pose states. To limit the complexity in the observation update step which is quadratic, a notion of local maps is used which partitions the map features. For each observation only the features in the local map are updated and the global map is updated only when the robot moves from one local map to another. The local update step is independent of the number of features in the global map. However, it remains quadratic in the number of features in the local map. The global update has a complexity equal to the full state update filter but is done very infrequently. The local updates can be done in the global frame as in compressed EKF (CEKF) [43] or the postponement algorithm [44]. Using a local frame for the local maps has some numerical advantages but transformations would be needed at each time step to get the pose and map estimates in the global frame. This approach is used in the constrained local sub-map filter (CLSF) [45] and the local map sequencing algorithm [46]. This is a significant improvement but it still renders the EKF based SLAM handicapped when faced with a large dense map. Researchers have worked on getting approximate solutions but then the convergence and consistency guarantees of the filter is lost. The important thing to realize here is that the complexity is inherent to the problem formulation and it cannot be solved without neglecting the accuracy and consistency of the filter.

Other issues that researchers tried to address were that of data association and loop closing. Under uncertainty, it is not trivial to associate the observation to the physical object in the environment where the readings originated from. Its complexity grows with the growth in uncertainty. It especially becomes difficult in a cluttered environment or while closing a large loop as the odometry errors tend to accumulate over time [47]. Also, once the update step is complete, it is impossible to revise a wrong association. Wrong data associations are fatal for the EKF and can quickly make the filter diverge. The joint compatibility test [48] gives significant improvement

but adds to the computational complexity.

The other recursive solution to SLAM is based on Rao-Blackwellized particle filters called FastSLAM [49]. These are trajectory based methods that exploit the property of the Bayesian formulation of the SLAM problem that given the robot path, the map feature estimates are independent of each other. FastSLAM is an instance of the RaoBlackwellized particle filter (RBPF) [50], which factors the full SLAM posteriors exactly into a product of a robot path posterior and landmark posteriors, conditioned on the robot path estimate [51]. This factoring of the SLAM posteriors reduce the computational complexity to linear in map components. On the other hand, every particle has an associated map with it and each of them has to be updated in the observation update step of the filter. This adds both to computational effort and space complexity. It also suffers deeply from the particle depletion problem which arises from the re-sampling step of the particle filter and causes it to lose its history. This prevents the update of feature estimates from observations recorded in the past. In the degenerate case, all the weights get reduced to one particle. This particle depletion issue makes FastSLAM inconsistent over time, irrespective of the number of particles [4]. In practice, FastSLAM needs a large number of particles to maintain consistency, as well as good proposal distributions. In essence, the FastSLAM technique replaces the "curse of dimensionality" with the "curse of history". These have been the primary focus of research in this area [31, 27, 52].

Information filter based methods have also been used to solve the SLAM problem [22, 5, 53]. These techniques use the information matrix instead of the covariance and information vector instead of the state vector. The primary motivation behind using the information filter was based on the observation that the information matrix is very sparse and would be amenable to approximation techniques. However, using approximations again leads to the loss of consistency. Secondly, at each time step the

covariance matrix needs to be calculated for the data association step which includes an inverse of the matrix of size of the number of states and adds to the computational complexity tremendously [4]. These are the main issues related to its implementation in the SLAM context.

An alternate family of algorithms called the Expectation Maximization (EM) approach for the SLAM problem is based on Dempster's EM algorithm [54]. In this approach, the map building problem is posed as a constrained, probabilistic maximum-likelihood estimation problem. It involves batch processing of the data collected by the robot to find the most likely map along with the most likely path taken by the robot [8]. However, these techniques are non-recursive in nature and cannot be implemented for online use.

B. Motivation

The various approaches to solve the SLAM problem have been outlined in the previous subsection, with an emphasis on the state-of-the-art Bayesian approaches. It can be seen from this review that any solution to the SLAM problem is faced with several difficult issues. Though significant progress has been made in the last two decades, a real time solution to the SLAM problem for large, unstructured and dense environments has remained elusive. The major issues and motivations have been enumerated below:

- *Computational Complexity:* The study of computational complexity is of primary consequence for exploring the potential of any numerical solution to be applicable to large scale applications. It becomes all the more important for real time applications as it can soon render most solutions incapable for applications of practical significance. The issue with the Bayesian approaches is

that complexity and consistency are at cross purposes. Approximations which help in reducing the complexity of the solution but takes away the consistency guarantees does not ultimately help as it means that over time the map might not be consistent. So it is important to formulate a consistent solution whose complexity is such that the system requirements of processing power and memory requirements are within reasonable limits.

It becomes increasingly challenging when the map building is for the purposes of navigation in a dense environment as navigation requires rich information about the environment. A gridded map is suitable for navigation but can lead to unreasonable computational burden on the underlying mapping algorithm. An area of one square kilometer with a very humble grid size of one meter translates to 10^6 map components. Even for an algorithm with linear complexity along with the real time constraints quickly becomes challenging for an increasing environment size. Given that SLAM is a difficult problem to solve, it is easy to see that a solution to be applicable to large dense environments it has to be independent of the global environment size and should be amenable to partitioning or local solution. An algorithm whose solution space considers only the local neighborhood and only occasionally needs to update some global parameters while being mathematically consistent is an ideal candidate for the issue at hand.

- *Consistency:* It is important for a stochastic method to be consistent as otherwise the solution might diverge from the actual solution with time and lead to catastrophic failures. Mathematical consistency of the mapping algorithm gives us more assurance of getting a consistent map over long durations of robot run given that the underlying assumptions are not violated. For a SLAM solution

to be scalable in time it needs to satisfy this property. Hence, approximations which take away the consistency guarantees need to be avoided.

- *Loop Closing:* The loop closing problem is the data association problem for the case when the mobile robot is in a region that has been previously visited by it. It is called the loop closing problem as in many cases there are loops in the environment which leads the robot to come back to a place it has already visited at least once during the map building exercise. As pointed out earlier, data association is not trivial under uncertainty and becomes especially challenging in this case because of the accumulation of odometry errors along the loop for a dense environment. Though batch validation with JCBB or CCDA help us achieve reliable data association, they add significant computational burden for cluttered environment. Using batch validation under a sparse feature assumption can give reliable results without having an adverse affect on the computational complexity of the SLAM solution.

- *Real Time:* A truly autonomous system should be able to create the map of its surrounding on the fly as it receives sensor observations. The current estimate of the map is used for navigation and hence grid based maps are preferred over feature maps. To develop a real time SLAM solver capable of running on board the robot and creating the map of a dense environment has remained a formidable challenge. The requirement of running on board the robot puts limitations on the available computational resources. Having the minimum system requirements is one of the top priorities while developing such a software. Developing the software in the C language is a judicious choice as it has very low runtime overhead, provides low level access and is arguably the best choice for performance critical applications. It is probably the closest one can get

to assembly language performance without actually writing code in assembly language.

C. Organization of the Dissertation

The dissertation is organized in such a way that the presented material is accessible to a larger audience and also complete in its own right. To this end, some of the technical details and proofs are covered in the appendices which the interested reader can refer to as needed. *The rest of the document is organized as follows:*

- The problem statement addressed in this dissertation is stated in Chapter II. This chapter clarifies the scope of the dissertation. The solution approach is then described to provide an insight into the work that was conducted as part of this dissertation.

- The Bayesian filtering approach to SLAM is reviewed in Chapter III along with the EKF which is a tool to solve the Bayesian filtering problem under the Gaussian noise assumption. The compressed EKF (CEKF) is also discussed which provides significant computational benefits to the Bayesian SLAM solution.

- In Chapter IV, the novel frequentist mapping technique based on stochastic approximation is introduced which provides an efficient solution for distributed mapping. First, it is derived for the case of known sensor location followed by the case when the sensor location is not perfectly know and is defined by its *pdf*. The consistency of the frequentist method can be rigorously proved and is included as an appendix to the dissertation to improve readability.

- In Chapter V, the hybrid solution to SLAM (Hybrid SLAM) is presented which is based on the principle of first localize and then map. It utilizes the Bayesian

filter for localization using a feature based map. The frequentist mapping technique is employed for estimating the dense environment using an occupancy grid map. The Hybrid SLAM is developed for the case of mapping the distributed environment using a 2-dimensional laser range finder. The various mathematical models used, for example, the sensor model and the feature models are outlined. The software developed to provide a real time mapping solution using this hybrid technique is also discussed.

- The results of applying hybrid SLAM to large distributed environments are presented and discussed in Chapter VI to demonstrate the efficacy and potential of the developed method. Firstly, simulation results are presented for maps containing multiple loops. It is demonstrated that the method is capable of successfully closing large loops which is one of the challenges faced in SLAM. It is followed up with application of Hybrid SLAM in real experiments showing that Hybrid SLAM can successfully handle real environments and also to provide an end to end experience of the Hybrid SLAM software developed as a part of this work. The experiments were performed in the corridors of the Houston building in the West Campus of Texas A&M University using an iRobot Create fitted with a Hukoyo URG laser sensor. Lastly the results are provided for a few datasets from the Radish website ([55]) proving the efficacy of the developed methods/software to handle large complex datasets. These results will also help to provide a benchmark against other SLAM solutions as they are widely used as a kind of standard in the SLAM community. The results are discussed in depth for better understanding of the implications of the findings.

- Finally, in Chapter VII, conclusions are drawn from the presented work and the contributions of the dissertation are noted. The scope and direction of future

research to exploit the full potential of the current work is also briefly described.

- In appendix A, the consistency of the frequentist mapping algorithm is proved under very mild assumptions using stochastic approximation techniques.

- In appendix B, the observation model involved in the frequentist mapping approach is derived for the case of laser range finders used to build a 2-D OG map.

- The feature extraction algorithm used for extracting line and point features from laser range data is discussed in appendix C. These line and point features are used by the EKF as the map components.

CHAPTER II

PROBLEM STATEMENT AND SOLUTION APPROACH

A. Introduction

In this chapter, formal statements of problems considered in this dissertation is presented. A step by step approach taken to solve the problems are then described.

B. Problem Statement

This dissertation addresses the mapping problem for distributed environments by a mobile robot using its on-board exteroceptive sensors. As mapping needs the reference frame from which the observations are made, the position of the robot also needs to be determined. The position of the robot is also needed for navigation. This is called the localization problem. In the absence of a GPS, the observation of the environment features helps in estimating the position of the robot. As both problems needs to be addressed in tandem, it is referred to as SLAM in the robotics literature. Distributed and dense are used interchangeably throughout the text.

Problem B.1. Simultaneous Localization and Mapping (SLAM)

Consider a mobile robot which is capable of recording its odometry readings and is fitted with an exteroceptive sensor, like a laser range finder, capable of observing the surrounding. The robot is capable of navigating the environment for the purpose of exploring either with the help of an on-board controller or somebody remotely controlling it.

Such a mobile robot is to start in an unknown location in an unknown environment and needs to incrementally build a consistent map of the environment using its exteroceptive sensors and to simultaneously localize itself in the map based on the

current map estimates and sensor readings.
The goal of this problem is to get a dense map of the surrounding at each time step and also the position of the robot.

Problem B.2. Localization using sparse features

Consider the same robot as described in problem B.1. Such a robot is to start in an unknown location in an unknown environment and needs to consistently estimate its current position (localize) at every time step using its odometry readings and observations from the exteroceptive sensors.

To successfully localize the robot a map of the surrounding needs to be built. The map created in this step is only for the purpose of localization and does not need a dense representation as it would not be used for navigation.

Problem B.3. Mapping distributed environment

Consider the same robot as described in problem B.1. Given a consistent estimate of the robot position and orientation (its pdf), the dense environment needs to be estimated incrementally at each time step.

The goal of this problem is to construct a dense representation of the map which can be used for navigation or other robot activities where rich information about the environment is required.

Problem B.4. Software development

Software needs to be developed for solving the localization and distributed mapping problem incrementally in real time. The software should include a simulation environment where various mapping and localization algorithms can be tested before deploying to real life solutions. It should be able to handle real data from exploration

experiments by various mobile robots and should seamlessly integrate with the commonly used log file formats in the robotics community. The software should also have the potential to run on-board a robot in real time using minimum system requirements.

Note: The applications presented in this dissertation only uses one specific kind of exteroceptive sensor, namely laser range finders. Yet, the problem statements are defined for the general case of any exteroceptive sensor because the solutions developed in this dissertation still hold for the general case. Some other examples of exteroceptive sensors commonly used in robots are sonars, radars and cameras. While developing the theory, no special assumptions are made which are only applicable to laser range finders. For localization and mapping using other kinds of exteroceptive sensors, the observation model for the same will need to be derived. The observation model for the laser range finder is derived in appendix B which can be used as a guide.

C. Solution Approach

1. Hybrid SLAM

The problem statements are formulated in such a way that the solution to problem B.2 (localization) and problem B.3 (mapping) can be combined to get a solution to problem B.1 (SLAM) which is the central goal for robotic mapping. A hybrid methodology to the SLAM problem is thus proposed that is based on the "first localize and then map" approach. It uses a Bayesian filter for localization and a frequentist mapping algorithm for dense mapping and is thus called the Hybrid SLAM solution. The problem is formulated in such a way that the mapping algorithm is independent of the localization. This is achieved through the introduction of a sufficient statistic called the 'belief state' which is simply the *pdf* of the robot pose and is computed

from the localization algorithm. This *pdf* is then used by the frequentist method to map the rest of the environment. The Hybrid SLAM solution is consistent and its frequentist part has constant time complexity. Thus, the proposed methodology has the potential to map large distributed environments incrementally in real time.

2. Localization

In this hybrid solution, the localization is accomplished by using a feature based EKF algorithm which is the Bayesian filter for Gaussian systems. The only requirements on the localization algorithm is that it should produce a consistent pose estimate for the robot efficiently. This is accomplished by employing a feature based EKF-SLAM algorithm which uses a very sparse set of prominent features in the map. The sparseness of the features helps us to curb the issues of computational complexity and data association in EKF-SLAM. The benefits of using EKF for localization and tracking is well known and is the reason for it being the most extensively used solution to SLAM [3].

3. Mapping

Given the belief on the robot pose from the localization step, the distributed environment is estimated using a novel stochastic approximation based algorithm which we call the frequentist mapping approach since it is based on the law of large numbers. The frequentist mapping approach has been formulated, developed and demonstrated in this dissertation. It uses OG maps introduced by Elfes where the OG is specified in terms of the probability of a grid being occupied [25]. However, the process of occupancy probability estimation is completely different. Instead of the Bayesian update used by Elfes, we use a frequentist approach based on the principle of counting. The frequentist approach developed here has a solid foundation in stochastic

approximation literature and is provably consistent. It also gives us constant computational complexity in the map components which is a tremendous improvement over the existing methods and thus is applicable to very large environments.

Consistency of the mapping algorithm : Rigorous proofs are given to show the consistency of the novel frequentist mapping algorithm under relatively mild assumptions. Given a consistent estimate of the robot pose, the mapping algorithm converges to the true map probabilities almost surely. Please refer to Appendix A for the proof. It is also shown that the complexity of the mapping algorithm is linear in the map components.

4. Software Development

Software is developed implementing the Hybrid SLAM solution which can handle real sensory data from mobile robots using a laser sensor to incrementally build the map of the environment. All software development is done in the C programming language as it is arguably the best platform to develop real time application which can run on limited system resources. A modular notion is followed to develop the software to keep it versatile, so that individual modules can be replaced without affecting other parts of the software. The main modules are simulation, feature extraction, localization and mapping. The software developed provides an efficient solution to the SLAM problem.

For doing research in this field it is very important to have a test bed on which new algorithms can be tested or already existing methods can be compared. The software developed provides such a test bed for testing localization and mapping algorithms. In foresight, due to the nature of the Hybrid solution, any localization algorithm which gives a consistent estimate of the robot pose could be potentially used. The SLAM solver is designed in such a way that a new localization module

could be plugged in without changing any other part of the code. The new localization module can choose to use the current feature extraction and data association modules or can bring in its own required modules. Hence an extensible framework is provided for further development of localization and mapping algorithms.

D. Applications

- *Simulations :* The efficacy of the proposed solution is first demonstrated in simulations where the robot mounted with a laser range sensor is able to map large distributed maps containing multiple loops which is considered a difficult problem in the community. It is also shown that the EKF remains consistent using a sparse set of line and/ or point features. This is a criteria that has to be satisfied by the localization part. It is a requirement for the frequentist mapping algorithm for consistently map the distributed environment.

- *Experiments :* It is shown that the proposed solution is able to effectively map the corridors of the Houston Building in the west campus of Texas A&M University. An iRobot Create fitted with a Hukoyo URG laser sensor was used for these experiments. The experimental results show that Hybrid SLAM is able to build a consistent map using the inaccurate odometry readings of the iRobot and a cheap and noisy laser range finder.

- *Datasets :* The Hybrid SLAM is also tested on the standard datasets used in the SLAM community to show the capabilities of the current implementation. The first step in this direction is taken by generating the dense map with the data collected form the SAL building at University of Southern California (USC). Next, the raw sensor data acquired from the Intel Laboratory in Hillsboro, Oregon is solved using Hybrid SLAM. This dataset is much larger compared to

the USC SAL dataset. Finally, the data collected from the SDR site B is used to create the dense map of the explored region. This is one of the most complex dataset in the Radish website which includes the robot visiting several rooms. The exploration time, i.e., the time taken by the robot to collect this data is close to 1 hour.

All the three datasets were collected using a Pioneer robot fitted with a SICK LMS 200 laser range finder. All these datasets are real experimental data collected by a remotely controlled robot in respective indoor environments as mentioned above. The robot is remotely guided to explore the environment and the raw laser readings along with the odometry readings are recorded in a log file. These log files are published on the Radish website for researchers in the SLAM community to test their algorithms on.

CHAPTER III

THE BAYESIAN APPROACH TO SLAM

The Bayesian formulation of the SLAM problem is considered in this section. The Bayesian state estimation framework for a stochastic dynamical system is introduced, focusing in particular on the *filtering problem*. The different formulations to solve the filtering problem is then described. In particular, the Bayes filter and the Kalman filter are reviewed. The Extended Kalman Filter is also studied as it has been one of the most widely used tools to solve the SLAM problem using feature based maps. In EKF-SLAM, the state vector is composed of the robot pose and the map features as parameters and a joint estimate of robot pose and map features is obtained at each time step. Thus a recursive solution to SLAM is obtained.

A. Introduction

The goal of a probabilistic filter for a dynamic system is to estimate a distribution of the possible system state x_t, given the control input $u_{0:t-1}$ and the observation history $z_{0:t}$. It is denoted by $p(x_t/z_{0:t}, u_{0:t-1})$. Several on-line and off-line techniques have been proposed for solving the filtering problem [56, 57, 58, 59]. Most of them rely on the assumption that the underlying process being observed is a Markov process. A Markov process is defined as a random process where the future does not depend on the past given the current state. In filtering, the current measurements are assumed to be independent of the past observations given the current state, i.e.,

$$p(z_t/z_{0:t-1}, x_t) = p(z_t/x_t). \tag{3.1}$$

In the context of the SLAM problem, the map features are generally included in

the state vector as parameters, θ, which are fixed, i.e., $\dot{\theta} = 0$. This imposes an obvious restriction that the environment should not contain any moving objects unknown to the robot. In practice, the methods proposed have been shown to work under small violations. In case of large violation, the observations are filtered in order to skip the sensor readings generated from dynamic objects before feeding it into the filtering algorithm.

B. The Bayesian Filter

The Bayesian filtering problem is formalized which provides solution to problem B.2. Let $p(z/x)$ be the given observation model, i.e., the *pdf* of the measurement z given that the system is in state x. Let $p(x_{t+1}/x_t, u_t)$ be the evolution model or motion model, i.e. the *pdf* of the state at time $t+1$ given the state and the control in the previous time step. The recursive form of the Bayesian filter which is used to estimate the current state at all times can be derived as follows:

$$\begin{aligned} p(x_t/z_{0:t}, u_{0:t-1}) &= \frac{p(z_t/x_t)p(x_t/z_{0:t-1}, u_{0:t-1})}{p(z_t/z_{0:t-1}, u_{0:t-1})} \\ &= \eta p(z_t/x_t) \int p(x_t/x_{t-1}, u_{t-1}) p(x_{t-1}/z_{0:t-1}, u_{0:t-2}) dx_{t-1}, \end{aligned} \tag{3.2}$$

where η is a normalization factor ensuring that Eq.3.2 correctly represents a probability distribution. The filtering equation in this form is used by the EKF-SLAM to give a recursive solution to SLAM.

Usually the evaluation of Eq. 3.2 is done in two steps : *prediction* and *update*. In the prediction step, the predicted prior probability is calculated, i.e, $p(x_t/z_{0:t-1}, u_{0:t})$, using the process model $p(x_t/x_{t-1}, u_t)$ and the posterior distribution from the last step, $p(x_{t-1}/z_{0:t-1}, u_{0:t-2})$. In the update step, the last observation z_t is incorporated in the predicted prior to get the posterior distribution on the states. Referring to

Eq. 3.2, the prediction step consists of calculating the integral term (predicted prior) and the update step is performed by weighing the predicted prior by the observation likelihood, $p(z_t/x_t)$.

Bayes filter in this form is exact and can be applied to estimate any dynamical system for which the Markov assumption holds. The issue with this exact formulation is that it involves integrations over the state space which makes it computationally intractable for high dimensional systems. In many cases the state space is high dimensional. For instance, in the SLAM problem, the number of states is the sum of the robot pose states and the map features which can easily be of the order of thousands or millions for large dense maps. For this reason approximate solutions are needed.

C. Kalman Filter

The Kalman filter provides a closed form solution to the Bayesian filtering equations under the assumptions of linearity and gaussian noise. Both the evolution model and observation model are assumed to be linear and affected by additive gaussian noise. Under these assumptions it provides a recursive solution to state estimation which minimizes the mean of the squared error for any system governed by the following linear stochastic difference equation

$$x_t = Ax_{t-1} + Bu_{t-1} + w_{t-1}, \tag{3.3}$$

with a measurement model that is

$$z_t = Hx_t + v_t \tag{3.4}$$

where $w_t \sim N(0, Q_t)$ and $v_t \sim N(0, R_t)$ are zero mean gaussian random variables that represent the process and measurement noise respectively. The Kalman filter update has time complexity that is cubic in the number of states and is updated by the following iterative equations:

Prediction step:

$$\hat{x}_t^- = A\hat{x}_{t-1} + Bu_{t-1} \tag{3.5}$$

$$P_t^- = AP_{t-1}A' + Q_t \tag{3.6}$$

Measurement update step:

$$K_t = P_t^- H_t'(H_t P_t^- H_t' + V_t R_t V_t')^{-1} \tag{3.7}$$

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - h(\hat{x}_t^-, 0)) \tag{3.8}$$

$$P_t = (I - K_t H_t)P_t^- \tag{3.9}$$

For the case of mobile robots, both the process model and observation model are non-linear and are of the following form:

$$x_t = f(x_{t-1}, u_{k-1}, w_{t-1}), \tag{3.10}$$

$$z_t = h(x_t, v_t). \tag{3.11}$$

Please see Chapter V for examples of process models and observation models. The Extended Kalman Filter (EKF), which is the non-linear extension of the KF is now employed to generate the state estimates recursively [60]. The process noise and the observation noise is still assumed to be gaussian, i.e, $w_t \sim N(0, Q_t)$ and $v_t \sim N(0, R_t)$. The EKF works similar to the KF by performing local linearization of the state transition function $f(.)$ and observation model $h(.)$. The EKF algorithm can be expressed as follows.

EKF time update equations:

$$\hat{x}_t^- = f(\hat{x}_{t-1}, u_{k-1}, 0), \tag{3.12}$$

$$P_t^- = A_t P_{t-1} A_t' + W_t Q_t W_t'. \tag{3.13}$$

EKF measurement update equations:

$$K_t = P_t^- H_t' (H_t P_t^- H_t' + V_t R_t V_t')^{-1}, \tag{3.14}$$

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - h(\hat{x}_t^-, 0)), \tag{3.15}$$

$$P_t = (I - K_t H_t) P_t^-. \tag{3.16}$$

where A and W are the Jacobian matrix of partial derivatives of f with respect to x and ω respectively. H and V are the Jacobian matrix of partial derivatives of h with respect to x and v, i.e.,

$$\begin{aligned} A_{[i,j]}^t &= \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{t-1}, u_{t-1}, 0), \\ W_{[i,j]}^t &= \frac{\partial f_{[i]}}{\partial \omega_{[j]}}(\hat{x}_{t-1}, u_{t-1}, 0), \\ H_{[i,j]}^t &= \frac{\partial h_{[i]}}{\partial x_{[j]}}(\hat{x}_t^-, 0), \\ V_{[i,j]}^t &= \frac{\partial h_{[i]}}{\partial v_{[j]}}(\hat{x}_t^-, 0). \end{aligned} \tag{3.17}$$

In EKF-SLAM, the state vector $x = [s, \theta]$, where s is the robot pose states, and θ is the collection of all map features. The map features are considered parameters, i.e., $\theta_{t+1} = \theta_t$. The EKF estimates the joint distribution of the robot pose states and the map features at each time step. As mentioned earlier, updating the joint distribution is essential to consistently estimate the map and is the motivation behind using EKF in this fashion. However, the size of the state vector in this form is equal to the

sum of the robot pose states and all the parameters defining the features of the map which could easily be of the order of thousands for a large map. This leads to a high dimensional problem and is apparent that complexity and consistency are at cross purposes. The computational complexity is at least quadratic in the map features without using the notion of local maps. While using local maps, the complexity is quadratic only in the local map features and independent of the total map features. However, the global map update, which is not done very frequently, might remain a bottleneck for real time applications as it has complexity equal to the full state update step of the EKF. We address the issue of complexity by only keeping very sparse features. This helps to keep the computational burden tractable.

Features can be any geometrical object in the environment that can be represented by a parametric model and whose parameters can be extracted from the observations made by the sensor. For the case of a laser range finder, the typical features used are point and line features. An algorithm to extract point and line features from raw laser range readings is described in appendix C. The EKF needs to associate each feature observed with the features in its state vector before calculating the posterior. This is called the data association problem. Once the data association is accomplished, the joint state posterior is calculated. For the case of sparse features, the data association step is considerably simpler than for the case of dense features and is another motivation for us to keep a sparse map.

The steps followed by an EKF-SLAM algorithm to estimate the joint posterior each time sensor readings are received can be summarized as follows:

- Extract features from raw sensor data.
- Obtain the predicted prior estimates using Eq. 3.12- 3.13.
- Associate the current feature readings to features already existing as part of the

EKF states.

- Calculate the posterior estimates using Eq. 3.14 - 3.16.
- Include newly observed features into the EKF states.

The key limitations in the use of the EKF lies in the underlying assumptions it makes on the system, namely linearizable models and Gaussian noise. In practical situations, mild violations to the assumptions lead only to a loss of optimality. Linearization errors generally introduce systematic errors in the estimate.

D. Summary

Despite its limitations, the EKF is one of the most widely used tools in estimation and SLAM due to its simplicity, optimality, tractability and robustness. Moreover, when the underlying approximations hold, it exhibits a strong convergence rate when compared with other filtering techniques.

CHAPTER IV

THE FREQUENTIST MAPPING APPROACH

A. Introduction

A novel frequentist mapping approach is proposed in this section which is aimed at providing an efficient solution to the dense mapping problem. The mapping problem at hand is formulated as a stochastic approximation problem where the environment is modeled as a stationary, spatially uncorrelated random process whose stationary probabilities are fixed but unknown, and have to be estimated by an autonomous system moving in the environment and making observations with its sensors. The mapping problem is then solved using stochastic approximation techniques resulting in independent approximations for the various environment components as opposed to the Bayesian formulation where the components get correlated. This significantly reduces the computational burden of the mapping algorithm compared to the Bayesian techniques. First the environmental model for the map is defined. Next, the frequentist algorithm is developed for the case when the pose of the robot/sensor is perfectly known. Subsequently, this assumption is relaxed and the frequentist mapping algorithm is extended to the case when the robot pose is uncertain and is specified by a probability density function (*pdf*) or "belief" on the robot pose. The frequentist mapping method is strongly consistent and converges to true map probabilities with probability one. Please refer to Appendix A for the proof. For the purpose of clarity, the frequentist mapping algorithm will be derived for the case of occupancy grid map, which is a special case for the algorithm. The formulation for the general case is straight forward and is given in [61].

B. Environment Model

Consider a single autonomous agent and let its state/pose be denoted by the variable s, and let the state of the environment be denoted by the variable $Q = \{q_1, \cdots, q_M\}$, where M is the number of grids in the map and $q_k \in \{O, E\}$[1], representing that the grid can either be occupied or empty. The probability of a grid being occupied or empty will be denoted as $p^*(q_k = O)$ and $p^*(q_k = E)$ respectively. The environment is assumed to be stationary and uncorrelated, i.e.,

$$p^*(Q) = \prod_{i=1}^{M} p^*(q_i), \tag{4.1}$$

where $p^*(Q)$ represents the probability of the realization Q of the environment, and $p^*(q_i)$ represents the probability of the realization q_i for the i^{th} grid of the map. It can be anticipated that a large part of most environments can be modeled in this fashion. Any deterministic environment trivially satisfies the above assumption. The probability of observing the i^{th} grid in the state $\hat{q}_i$, where $\hat{q}_i \in \{O, E\}$, and given that it is observed from the pose s, is given by:

$$p(\hat{q}_i/s) = \sum_z p(\hat{q}_i/s, z)p(z/s) = \sum_z 1(\hat{q}_i/s, z)p(z/s), \tag{4.2}$$

where z is one of the finitely many observations that are possible from state s, $p(z/s)$ denotes the likelihood of making the observation z given that the observation is made from state s, and $1(\hat{q}_i/s, z)$ is the indicator function for the i^{th} component of the map being observed in the state $\hat{q}_i$ given that the observation is z and is made from state

[1]For the general case, $q_k \in \{e_1, \cdots, e_D\}$

s. Eq.4.2 is short hand for the following two equations:

$$p(\hat{q}_i = O/s) = \sum_z 1(\hat{q}_i = O/s, z)p(z/s),$$
$$p(\hat{q}_i = E/s) = \sum_z 1(\hat{q}_i = E/s, z)p(z/s).$$

The observation likelihood $p(z/s)$ can be obtained in terms of the map probabilities using the law of total probability as:

$$p(z/s) = \sum_{q_1 \cdots q_M} p(z/s, q_1, \cdots, q_M)p^*(q_1) \cdots p^*(q_M). \tag{4.3}$$

The likelihood $p(z/s, q_1, \cdots q_M)$ denotes the probability of an observation z given the particular realization of the environment $Q = \{q_1, \cdots, q_M\}$, and can be extracted from the physics and noise characteristics of the sensor model and the map probabilities. For any grid being observed from state s, the observation will either be that the grid is occupied or that it is empty. In the case that a laser range finder is being used, the observation z is of the form that k^{th} grid is occupied and the grids between the pose and the k^{th} grid are empty. In that case, $p(z/s) = p(\hat{q}_k = O/s)$ and can be calculated by

$$\begin{aligned} p(z/s) &= \sum_{q_k} p(\hat{q}_k/s, q_k)p^*(q_k) \\ &= p(\hat{q}_k = O/s, q_k = O)p^*(q_k = O) + p(\hat{q}_k = O/s, q_k = E)p^*(q_k = E), \end{aligned} \tag{4.4}$$

where, $p(\hat{q}_k = O/s, q_k = O)$ and $p(\hat{q}_k = O/s, q_k = E)$ are obtained from Eq. B.3 and B.4 in appendix B. Plugging Eq. 4.3 into Eq. 4.2 implies:

$$\begin{aligned}
p(\hat{q}_i/s) &= \sum_z p(\hat{q}_i/s, z) \sum_{q_1 \cdots q_M} p(z/s, q_1 \cdots q_M) p^*(q_1) \cdots p^*(q_M) \\
&= \sum_{q_1 \cdots q_M} \sum_z p(\hat{q}_i/s, z) p(z/s, q_1 \cdots q_M) p^*(q_1) \cdots p^*(q_M) \\
&= \sum_{q_1 \cdots q_M} p(\hat{q}_i/s, q_1 \cdots q_M) p^*(q_1) \cdots p^*(q_M) \\
&= \sum_{q_i} p^*(q_i) \sum_{q_1 \cdots q_{i-1} q_{i+1} \cdots q_M} p(\hat{q}_i/s, q_1 \cdots q_M) p^*(q_1) \cdots p^*(q_{i-1}) p^*(q_{i+1}) \cdots p^*(q_M) \\
&= \sum_{q_i} p(\hat{q}_i/s, q_i) p^*(q_i) \qquad (4.5)
\end{aligned}$$

Eq. 4.5 is called the observation equation and can also be obtained by simply expanding the left hand side by the law of total probability as follows:

$$p(\hat{q}_i/s) = \sum_{q_i} p(\hat{q}_i/q_i, s) p^*(q_i)$$

where,

$$p(\hat{q}_i/s, q_i) = \sum_{q_1 \cdots q_{i-1} q_{i+1} \cdots q_M} p(\hat{q}_i/s, q_1 \cdots q_M) p^*(q_1) \cdots p^*(q_{i-1}) p^*(q_{i+1}) \cdots p^*(q_M) \quad (4.6)$$

The observation equation can be written in compact matrix form as:

$$\hat{P}_i(s) = A_i^*(s) P_i^* \qquad (4.7)$$

where,

$$A_i^*(s) = \begin{bmatrix} p(\hat{q}_i = O/s, q_i = O) & p(\hat{q}_i = O/s, q_i = E) \\ p(\hat{q}_i = E/s, q_i = O) & p(\hat{q}_i = E/s, q_i = E) \end{bmatrix}, \qquad (4.8)$$

$$\hat{P}_i(s) = [p(\hat{q}_i = O/s), p(\hat{q}_i = E/s)]' \tag{4.9}$$

is the observation vector for the i^{th} grid, and

$$P_i^* = [p^*(q_i = O), p^*(q_i = E)]' \tag{4.10}$$

are the true map probabilities. The matrix $A_i^*(s)$ is the true observation model of the i^{th} component when observed from pose s, which means that the observation model is formed by using the true map probabilities $P^* = \{P_1^*, \cdots, P_M^*\}$. This is the fundamental observation equation for the mapping problem.

Note that, in general, the observation matrix for the i^{th} map component, $A_i^*(s)$, depends on the location of the sensor s, the true map probability for the i^{th} component, P_i^*, as well as the map probabilities for the other map components. In practice, the probabilities of a map component are affected only by its neighboring components. The actual number that affect the component depends on the noise characteristics of the sensor. Given an accurate sensor such as a laser range finder, the number is extremely small while for noisy sensors such as Sonar, the number is much higher. The observation model can be obtained given the physics and the statistical noise characteristics of the sensor, the sensor location and the map probabilities. Please see appendix B to see how this is done for a laser range finder. The above formulation of the observation model is very general and can be applied to most sensing scenarios. In certain special cases, the observation models $A_i^*(s)$ for the i^{th} map component may be independent of the rest of the map components, for instance, in sparse maps. In such cases the analysis is much easier and the conditions required for consistency are weaker. This special case was considered in [1].

C. Mapping with known Sensor Location

Given the time varying pose/ state of the sensor, $\{s_1, \cdots, s_t\}$, and the noisy sensor readings from these poses, $\{z_1, \cdots, z_t\}$, the problem of mapping under known sensor location is to construct a consistent estimate of the map probabilities, P_i^*.

Please recall that the observation model for a particular map component is given by Eq.4.7. **This equation is the fundamental equation for the frequentist approach and provides an avenue for estimating the true map probabilities** P_i^*. Suppose repeated observations are made of the i^{th} map component from pose s, and the number of times the i^{th} component is observed in its various states is counted. Then a consistent estimate of the observation probability vector $\hat{P}_i(s)$ can be constructed by using Eq. 4.2 as follows:

$$\hat{P}_i(s) = E_z[\mathbf{1}(\hat{q}_i/s, z)] \equiv E_z[\mathbf{c}_i(s, z)] = \lim_N \frac{1}{N} \sum_{t=1}^{N} \mathbf{1}(\hat{q}_{i,t}/s, z_t). \tag{4.11}$$

where, given an observation z, $\mathbf{c}_i(s, z) = \mathbf{1}(\hat{q}_i/s, z) = [1(\hat{q}_i = O/s, z), 1(\hat{q}_i = E/s, z)]'$ is the observation vector (this notation is chosen to help extend the formulation to the case with uncertain pose). The above equation is correct due to the Law of Large Numbers and we can approximate the expected value of $c_i(s, z)$ by the average of the result obtained from a large number of observations $\mathbf{1}(\hat{q}_i/s, z)$. Then, using the knowledge of $A_i^*(s)$, the true environmental probabilities P_i^* can be obtained as follows:

$$P_i^* = A_i^*(s)^{-1} \hat{P}_i(s). \tag{4.12}$$

Next, the assumption that the observations are made from the same pose s is relaxed. Now, the observations can be made from the time varying poses $\{s_t\}$, with true observation models $A_i^*(s_t)$. The true map probabilities, P_i^*, can be recovered

asymptotically by keeping a count of the observations of the i^{th} component in its various states, and a time average of the observation models as follows:

$$P_i^* = \bar{A_i}^{-1} \hat{P}_i, \tag{4.13}$$

$$\hat{P}_i = \lim_N \frac{1}{N} \sum_{t=1}^{N} c_i(s_t, z_t), \tag{4.14}$$

$$\bar{A}_i = \lim_N \frac{1}{N} \sum_{t=1}^{N} A_i^*(s_t). \tag{4.15}$$

where $\hat{P}_i = [p(\hat{q}_i = O), p(\hat{q}_i = E)]'$ is the observation vector formed by counting the frequency of observing the i^{th} grid as occupied and empty during the course of the mapping process and is interpreted as a probability, and $\bar{A}_i$ is the time averaged observation model for the i^{th} grid. In order to derive the above expressions, the frequency of the robot being in a state s is also interpreted as a probability, $p(s)$. Then, using the law of total probability and the fact $p^*(q_i/s) = p^*(q_i)$ we get:

$$p(\hat{q}_i) = \sum_s p(\hat{q}_i/s)p(s) = \sum_s \sum_{q_i} p(\hat{q}_i/q_i, s)p^*(q_i/s)p(s) = \sum_{q_i} \left[\sum_s p(\hat{q}_i/q_i, s)p(s) \right] p^*(q_i). \tag{4.16}$$

Thus, in matrix form:

$$\hat{P}_i = \bar{A}_i P_i^* \tag{4.17}$$

If the state s_t evolves according to a Markov Chain, and given that the chain is ergodic (i.e., converges to a stationary distribution for all initial distributions), the left hand side $\hat{P}_i$ in the above equation is given by the time average in Eq. 4.14, and the average observation matrix $\bar{A}_i$ is given by the time average in Eq. 4.15. Hence, the heuristically derived estimation equations for the time varying pose case are mathematically justified. Work based on the frequentist approach with no pose

uncertainty has been published in earlier work [62, 63, 64].

The estimation equation for recovering the map probabilities, Eq. 4.13, is an asymptotic equation, i.e., it is true only as $N \to \infty$. However, a recursive estimator for the map probabilities is needed for every time step N with the guarantee that the map probabilities converge as $N \to \infty$. This is to say that the equation $\hat{P}_i = \bar{A}_i P_i^*$ needs to be progressively solved for estimates of P_i^* at every time step. Lets call the estimate of P_i^* at time step t as $P_{i,t}$. Given $\hat{P}_i$, $\bar{A}_i$, and that $-\bar{A}_i$ is Hurwitz, a recursive way to solve the equation is as follows:

$$P_{i,t+1} = P_{i,t} + \gamma(\hat{P}_i - \bar{A}_i P_{i,t}). \tag{4.18}$$

To see why this is the case, note that for small γ, the above equation is the forward Euler approximation of the ordinary differential equation (ODE):

$$\dot{P}_i = \hat{P}_i - \bar{A}_i P_i. \tag{4.19}$$

Transforming the co-ordinates such that P_i^* is the origin, it is easy to see that P_i^* is the unique, global, exponentially stable equilibrium of the ODE, if $-\bar{A}_i$ is Hurwitz and full-rank. However, at any finite time during the algorithm's progress, the asymptotic values of $\hat{P}_i$ and $\bar{A}_i$ are not available. Thus, at time t, given the pose of the sensor s_t and the reading z_t, $\hat{P}_i$ and $\bar{A}_i$ are approximated by their one step noisy estimates, i.e.,

$$\hat{P}_i \approx c_i(s_t, z_t), \qquad \bar{A}_i \approx A_i(s_t). \tag{4.20}$$

Utilizing the above approximations in Eq. 4.18, and given that $A_i^*(s_t)$ is positive definite (which is true under mild conditions), the true map probabilities P_i^* can be estimated at each time step t using the following recursion.

Estimator E1:

$$P_{i,t} = \Pi_{\mathcal{P}}\{P_{i,t-1} + \gamma_t(c_i(s_t, z_t) - A_i^*(s_t)P_{i,t-1})\}, \tag{4.21}$$

where $\mathcal{P}$ represents the space of probability vectors in $\Re^D$, and $\Pi_{\mathcal{P}}(.)$ denotes a projection onto this compact set. The projection operator is needed since the map probability estimates $P_{i,t}$ need to be probability vectors and the recursion above need not result in a probability vector. The sequence $\{\gamma_t\}$ is usually of the form $at^{-\alpha}, \alpha \leq 1$, where a and α are learning rate parameters and standard for any stochastic approximation algorithm. The "noisy" algorithm above is a Stochastic Approximation algorithm and its convergence to the true map probabilities can be shown using techniques from Stochastic Approximation [65, 66]. Please see appendix A for details. Note that stochastic approximation algorithms as above are used in Q-learning, neural networks and system identification [67].

However, there remains the problem of using the "true" observation models $A_i^*(s)$ in order to form the estimates. This is unreasonable since the true map probabilities that are unknown. However, the current map probability estimates, $P_i(t)$, can be used, to form the observation models $A_i(s)$ as an approximation of the true observation models $A_i^*(s)$. These models can be inferred from the model of the particular type of sensor being used for sensing the environment [25]. The observation model $A_i(s)$ for the case of laser range finder in the OG case is derived in appendix B.

Remark A few details regarding the projection operator $\Pi_{\mathcal{P}}$ is provided in this remark. Note that $p(q_i = O) + p(q_i = E) = 1$ and that both terms are positive since they are probabilities. Then, one of the probabilities can be eliminated by replacing $p(q_i = E)$ by $[1 - p(q_i = O)]$ and having the constraint that $0 < p(q_i = E) < 1$. Let the probability that the i^{th} grid is occupied, $p(q_i = O) = p_i^o$. Then, the above

algorithm reduces to the following for the OG case:

$$P_{i,t+1} = \Pi_{\mathcal{P}}[P_{i,t} + \gamma_t\{c_i(s_t, z_t) - p(\hat{q}_i = O/q_i = O, s_t)p^o_{i,t} - p(\hat{q}_i = 0/q_i = E, s_t)(1 - p^o_{i,t})\}], \tag{4.22}$$

where $c_i(s_t, z_t) = 1(\hat{q}_{i,t} = O/s_t, z_t)$, and $\Pi_{\mathcal{P}}[.]$ represents projection onto the interval $[0, 1]$.

Another way to recursively estimate P^*_i from the equation $\hat{P}_i = \bar{A}_i P^*_i$ is as follows.

Estimator E2:

$$P_{i,t} = \underset{P \in \mathcal{P}}{\operatorname{argmin}} ||\bar{A}_{i,t}P - \bar{c}_{i,t}|| \tag{4.23}$$

$$\bar{A}_{i,t} = (1 - \gamma_t)\bar{A}_{i,t-1} + \gamma_t A_i(s_t) \tag{4.24}$$

$$\bar{c}_{i,t} = (1 - \gamma_t)\bar{c}_{i,t-1} + \gamma_t c_i(s_t, z_t) \tag{4.25}$$

where γ_t is a deterministic sequence with $\sum_t \gamma_t = \infty$ and $\sum_t \gamma^2_t < \infty$. It can be easily seen that $\bar{A}_{i,t} \to \bar{A}_i$ as $t \to \infty$, and $\bar{c}_{i,t} \to \hat{P}_i$ as $t \to \infty$. It can be shown that $P_{i,t} \to P^*_i$ almost surely, i.e., with probability one.

D. Mapping under Sensor Location Uncertainty

We relax the assumption that the robot pose s_t is known perfectly and instead, assume that we are given the time varying belief on the pose, $\{b_1(s), \cdots, b_t(s)\}$, and the noisy sensor readings from these belief states at each time step, $\{z_1, \cdots, z_t\}$. The mapping problem under sensor position uncertainty is to construct a consistent estimate of the components of the map probabilities, P^*_i. The uncertainty in the pose is called the belief and is defined by the given *pdf*.

It is immediately apparent that there is an inherent "data association" problem with the mapping problem in this scenario. The pose of the robot/sensor is now

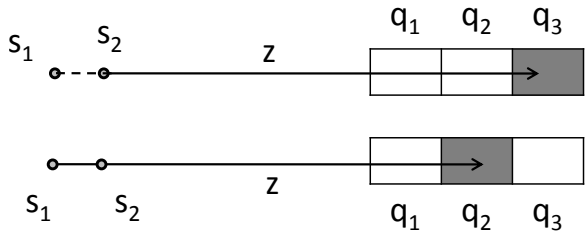


Fig. 1. The problem of data association

uncertain, which means that the robot could be in different positions, with some probability defined by the given *pdf*, from where it made the observation. Given the range observation z, the observation $\hat{q}_i$ of an environmental component q_i, could be occupied or empty depending on the position of the sensor from where the observation was made. For example, consider the simple situation illustrated in Fig.1 where the robot pose is uncertain and could be in pose s_1 with probability p_1 and in s_2 with probability $(1 - p_1)$ and makes a range observation z. The dark boxes in the figure depict occupied grids and the white boxes show that the grid is empty, i.e., $1(\hat{q}_2 = O/s_1, z) = 1$ and $1(\hat{q}_2 = O/s_2, z) = 0$.

However, since we have uncertainty in the location of the sensor, we cannot be sure as to whether the reading $\hat{q}_2$ is empty or occupied. Hence, an observation $\hat{q}_i$, can no longer be certain and the uncertainty needs to be captured in the mapping technique. This may be done as follows.

Given the uncertainty in the pose of the robot $b(s)$ and the reading of the environment z, the observation of the i^{th} component of the environment $\hat{q}_i$ is derived by using the rules of conditional probability, Bayes rule and the law of large numbers:

$$\begin{aligned} p(\hat{q}_i/b) &= \sum_z p(\hat{q}_i/b, z)p^*(z/b) = \sum_z \sum_s p(\hat{q}_i/s, b, z)p(s/b, z)p^*(z/b) \\ &= \sum_z \left[\sum_s 1(\hat{q}_i/s, z) \frac{p^*(z/s)b(s)}{p^*(z/b)} \right] p^*(z/b) = E_z[c_i(b, z)] \end{aligned} \tag{4.26}$$

where $c_i(b,z) = \sum_s 1(\hat{q}_i/s,z)\frac{p^*(z/s)b(s)}{p^*(z/b)}$, $p^*(z/b) = \sum_s p^*(z/s)b(s)$ is the factor used to normalize $c_i(.)$, and $p^*(z/s)$ is the true likelihood of the observation z given that it is made from pose s. In order to derive the above expression, note that using the theorem of total probability, $p(\hat{q}_i/b,z) = \sum_s p(\hat{q}_i/s,b,z)p(s/z,b)$. By using Bayes rule $p(s/z,b) = \frac{p^*(z/s)b(s)}{p^*(z/b)}$, and using the fact that $p(z/s,b) = p(z/s)$, and $p(\hat{q}_i/s,b,z) = \mathbf{1}(\hat{q}_i/s,z)$, Eq. (4.26) above follows. As in the perfect pose information case, to get a consistent estimate of the probability of observing state $\hat{q}_i$ given the belief state $b(s)$, an average is taken over all observations z. This can be formed by a time average due to the law of large numbers:

$$\hat{P}_i(b) = [p(\hat{q}_i = O/b), p(\hat{q}_i = E/b)]' = E_z[\mathbf{c}_i^*(b,z)] = \lim_N \frac{1}{N}\sum_{t=1}^{N} \mathbf{c}_i^*(b,z_t). \tag{4.27}$$

Note that the above probabilistic description of the observation solves the "data association" problem. As the pose from which the observation is made is no longer certain, so are the observations. Thus, an uncertainty or probability is now associated with each observation that the i^{th} map component is occupied or empty. The probability of observing the map component q_i at level $\hat{q}_i$, given the belief on the pose $b(s)$ is also given by,

$$p(\hat{q}_i/b) = \sum_s p(\hat{q}_i/s)b(s) = \sum_s \sum_{q_i} p(\hat{q}_i/q_i,s)p^*(q_i)b(s) \tag{4.28}$$

Thus, the observation equation for the frequentist mapping problem under pose uncertainty can be written in compact matrix form as follows:

$$\hat{P}_i(b) = A_i^*(b)P_i^*, \tag{4.29}$$

where

$$A_i^*(b) = \sum_s A_i^*(s)b(s). \tag{4.30}$$

Note here that this equation is exactly analogous to the frequentist mapping Eq. 4.7, wherein the exact pose knowledge s has been replaced by the belief on the pose of the robot $b(s)$. The observation model $A_i^*(s)$ is replaced by the expected observation model with the expectation being taken with respect to the belief on the pose of the robot. Thus, similar to the case with perfect pose information, if the robot was to remain in the belief state $b(s)$ and make repeated observations of the i^{th} component of the environment, the left hand side of Eq. 4.29, $\hat{P}_i(b)$, could be recovered by averaging the (probabilistic) observations of the i^{th} component, $c_i(b, z_t)$, using Eq. 4.27. Hence, the true environmental probabilities may be recovered asymptotically by inverting Eq. 4.29. Generalizing the situation to the case when there is a time-varying belief on the pose of the robot, $b_t(s)$, the true environmental probabilities can be estimated recursively using the following generalization of frequentist estimator E1.

Estimator E3:

$$P_{i,t} = \Pi_{\mathcal{P}}\{P_{i,t-1} + \gamma_t(c_i^*(b_t, z_t) - A_i^*(b_t)P_{i,t-1})\}, \tag{4.31}$$

As in the pure mapping case, the variables $c_i^*(b_t, z_t)$ and $A_i^*(b_t)$ are dependent on the true map probabilities P_i^*. Hence the estimator is actually run by using the current estimate of the true observation models/ observation likelihood. In other words, the above algorithm is run using $c_i(b_t, z_t, P_t)$ and $A_i(b_t, P_t)$, where the current estimate of the map probabilities P_t is used, instead of the true map probabilities P^*, in Eq. (4.26) to form $c_i(b_t, z_t, P_t)$, and in Eqs. (4.29)-(4.30) to form $A_i(b_t, P_t)$. The map probabilities could also be recursively estimated similar to estimator E2.

Estimator E4:

$$P_{i,t} = \underset{P \in \mathcal{P}}{\operatorname{argmin}} ||\bar{A}_{i,t} P - \bar{c}_{i,t}|| \tag{4.32}$$

$$\bar{A}_{i,t} = (1 - \gamma_t)\bar{A}_{i,t-1} + \gamma_t A_i(b_t) \tag{4.33}$$

$$\bar{c}_{i,t} = (1 - \gamma_t)\bar{c}_{i,t-1} + \gamma_t c_i(b_t, z_t) \tag{4.34}$$

E. Complexity Study

In this subsection, the time complexity of the frequentist mapping technique as derived in estimator 3 is studied. The problem size for the frequentist mapping is the number of map components, say n. First, the assumption is made that the pose estimate is spread over a finite space. This assumption is required to discretize the space around the mean of the pose *pdf* to calculate the belief states. For a consistent filter, the uncertainty is bounded and hence the number of belief states under uniform grids is also bounded, say m. For a large map, $m << n$. Then, from Eqn. 4.30 and 4.31, time complexity of the frequentist mapping technique under pose uncertainty is : $O(m + n) = O(n)$, as $m << n$. Hence, for a consistent pose estimate, the frequentist mapping algorithm has linear time complexity.

Further, it is assumed that the sensor noise is bounded is bounded in the discretized world. If the noise model for a sensor follows a Gaussian distribution, this is to say that the probability density is negligible beyond a certain number of grids. This is a very standard assumption for any real world sensor. Without satisfying this assumption, a sensor can never successfully give meaningful observations. For the frequentist method this translates to saying that the observation model for each grid is only affected by a fixed number of map components around it, say k. It follows that for each observation, only a fixed number of map components (k) needs to be

updated. Thus the time complexity of frequentist mapping (equation 4.31), under bounded sensor noise, is constant time.

This is a powerful result and plays a crucial part in the success of the frequentist mapping algorithm. It is a major improvement over other mapping algorithms most of which suffer from the "curse of dimensionality". A constant time complexity is as good as it can get and truly makes an algorithm applicable to very large scale application. A small caveat that is worth noting is that the time complexity is only a measure of the processor clock cycles that will be needed to solve a problem. The total time required to solve a problem also depends on the space complexity as accessing and reading from memory is an expensive task. The space complexity of the frequentist mapping technique is linear in the map components as information for each grid is stored separately. Linear space complexity is generally considered good under most circumstances. To reduce the space complexity much more complicated data representation will have to be used and it is expected that it will add to the time complexity of the problem.

F. Summary

In this subsection the frequentist mapping technique was first derived for the case of known sensor location and then extended to the case when the sensor location is uncertain. Given the sensor location estimate and the current observations, the frequentist technique provides a way to recursively estimate the environment components. A gridded map is especially suited for this technique and will be used in the Hybrid SLAM solution.

The data association problem under uncertainty is addressed by associating a probability with each observation and hence avoiding the high computational cost

Table I. Mapping with known sensor position.

Estimator	$P_{i,t} = \Pi_{\mathcal{P}}\{P_{i,t-1} + \gamma_t(c_i(s_t, z_t) - A_i(s_t)P_{i,t-1})\}$
Observation vector	$\mathbf{c}_i(s_t, z_t) = [1(\hat{q}_i = O/s_t, z_t), 1(\hat{q}_i = E/s_t, z_t)]'$
Observation matrix	$A_i(s_t) = \begin{bmatrix} p(\hat{q}_i = O/s_t, q_i = O) & p(\hat{q}_i = O/s_t, q_i = E) \\ p(\hat{q}_i = E/s_t, q_i = O) & p(\hat{q}_i = E/s_t, q_i = E) \end{bmatrix}$

Table II. Mapping under sensor location uncertainty.

Estimator	$P_{i,t} = \Pi_{\mathcal{P}}\{P_{i,t-1} + \gamma_t(c_i(b_t, z_t) - A_i(b_t)P_{i,t-1})\}$
Observation vector	$\mathbf{c}_i(b_t, z_t) = \eta \sum_s \mathbf{1}(\hat{q}_i/s, z)p(z_t/s)b(s)$
Observation matrix	$A_i(b_t) = \sum_s A_i(s)b(s)$
Observation uncertainty	$p(z_t/s) = \sum_{q_k} p(\hat{q}_k/s, q_k)p(q_k)$, where $1(\hat{q}_k = O/s, z_t) = 1$

generally associated with it. It can be said that the frequentist method is inherently immune to the data association problem as it is solved in the problem formulation step. The estimation equations for the frequentist mapping technique are summarized for the case of known sensor location and the case of sensor location uncertainty in Table I and Table II respectively.

It was also shown that under the bounded sensor noise assumption, the frequentist method has constant time complexity which is highly desirable and makes the frequentist method applicable to very large environments.

Note: The indicator function, $1(\hat{q}_k = O/s, z_t) = 1$, represents that the range measurement z_t from pose s corresponds to the reading that the k^{th} grid is occupied. For example, as in Fig. 1, $k = 2$ if the robot pose, $s = s_1$, and $k = 3$ if the robot pose, $s = s_2$. Please see Appendix B for detailed derivations of how the above estimation

algorithms can be implemented for OG maps using a laser range finder.

CHAPTER V

HYBRID SLAM

A. Introduction

In this section, the Hybrid SLAM solution is introduced which is based on the first localize and then map philosophy. It uses EKF or CEKF for localization uses point or line features. The distributed environment is mapped using the frequentist technique developed in Chapter IV. The frequentist technique produces an OG map in which the environment is discretized into grids and the mapping algorithm estimates the probability that each grid is occupied. Thus a solution to problem B.1 is developed for large distributed environments.

In the rest of the section the mathematical models used in both the EKF and dense mapping are defined. Algorithm details are also provided to give a better idea about the working of the proposed method and also makes it easier to implement the algorithm. Lastly, we describe the setups under which the algorithm was tested and the results obtained from the various exploratory runs made by the robots.

B. Hybrid Methodology

The robot starts in an unknown location in an unknown environment and the task is to generate an occupancy grid map for the environment with the help of the on-board laser sensor readings. The global frame of reference is fixed as the local reference frame of the robot at the initial time since the absolute position of the robot is unknown. The mapping algorithm generates an estimated map in this global frame. The robot is guided by a remote controller operated by a human to explore the environment that needs to be mapped. This is a widely used procedure in the SLAM community.

While exploring, the robot keeps a log file to store all the details of the run, i.e., the odometry information and the laser readings along with the corresponding time stamps. The odometry information in the log file is generated by integrating the control inputs over the time interval between successive readings and completely neglects the effect of noise. As can be envisioned, these odometry readings have a high error content and drift far away from the real position of the robot over time. In the current implementation, the log file generated by the robot is later fed into the Hybrid SLAM algorithm which sequentially reads in the log file assuming it is coming from the robot as it is exploring and builds the map incrementally. It should be noted that no batch processing of the data is done and the entire mapping process can as well be run on-line while the robot is exploring. The implementation change that will be needed is for the robot to be able to pipe the data into the mapping software.

The laser range finder installed on the robot logs all the range and bearing information it gets at different time instants into the same log file described earlier. These laser observations are used to both localize and build the dense map. The sensor characteristics are assumed to be given. These are generally provided by the manufacturer. In some cases they are also calibrated by researchers, for instance the Hokuyo URG laser sensor [68]. The sensor characteristics that are required as input parameters are: a) noise parameter for range readings and b) the maximum and minimum reliable range readings that the laser range finder can make. The control inputs to the robot can also be recovered from the log file. The approach of "first localize and then map" is then applied to build a consistent map of the environment or the explored area. The Bayesian formulation is used to localize the robot pose with a feature based EKF algorithm and the frequentist mapping technique developed is used to build the dense map. Thus the full SLAM solution is obtained and hence it

is called the Hydrid SLAM solution.

The EKF based method used for localization uses point and line features. The important point to be noted is that only the few most reliable features are used to localize the robot to keep various issues related to the EKF implementation at bay. The robot location is defined by (x_t, y_t, ϕ_t), the i^{th} point feature by (x_t^i, y_t^i) and the j^{th} line feature by (r_t^j, θ_t^j) at time t. Note that (r, θ) are the line parameters in the line model $\rho cos(\theta - \alpha) - r = 0$, where (ρ, α) is a point on the line expressed in polar coordinates. The states of the EKF consists of the robot states and all the point and line features in the map. When the robot starts, the state vector only contains the robot location and incrementally includes new features as it encounters them while exploring new areas. The features are extracted from the laser readings and those are the inputs to the EKF instead of the raw laser readings. The EKF gives the joint estimate of mean and covariance for all the states at each time step. As the distribution is assumed Gaussian, it can be easily marginalized to get the mean and covariance of the robot states, (x_t^r, P_t^r), denoted by the superscript. In fact the mean for individual states remains the same and the covariance matrix can be formed merely by picking up the corresponding elements from the joint state covariance matrix. At each time step, in this case whenever a laser reading is received, the EKF is updated to generate (x_t^r, P_t^r). This mean and covariance estimate along with the laser range readings is used by the frequentist mapping part to construct the dense map. A feature map can be generated from the EKF states but it does not contain as much information as the dense grid map associated with the frequentist map building.

The frequentist mapping algorithm developed earlier is used for the dense mapping part where the environment is divided into small uniform occupancy grids. The belief state/*pdf* of the robot pose is defined by the mean and covariance, (x_t^r, P_t^r), which is given by the localization algorithm preceding the frequentist mapping step.

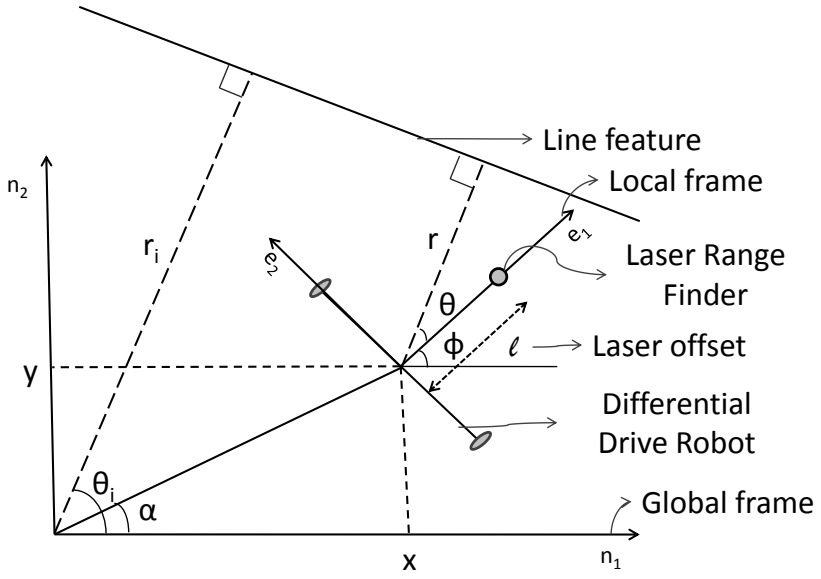


Fig. 2. Global and local coordinate frame definitions

The belief state is also discretized in all the dimensions of the robot states and the probability that the robot is in each grid is calculated by integrating the *pdf* over the domain of the corresponding grid. Each observed grid and its nearby grids in the map are updated for each laser range reading according to Estimator 3. Given the maximum range reading that the laser can make reliably, all readings greater than this are discarded. Under uniform grids and bounded sensor noise this means that an upper bound can be formed on the number of grids that needs to be updated after each observation. Hence, it follows that the frequentist mapping approach has constant time complexity in the map size. In this fashion, an estimated occupancy grid map is generated for the dense map at each time step when a laser observation is made.

C. Process Model

The robot kinematics is assumed to be that of a differential drive since a wide variety of robots comes fitted with a differential drive, for instance the Pioneer robot, which was used to record several data sets available online through the Radish repository. Also, the iRobot Create has a differential drive which was used to record real data in the Houston Building at Texas A&M University. The control input to the differential drive are the angular velocities to its two wheels which can be recovered from the log file. The vehicle kinematics model is given by:

$$\begin{aligned} \dot{x} &= \frac{R}{2}(u_r + u_l + w_r + w_l)\cos\phi, \\ \dot{y} &= \frac{R}{2}(u_r + u_l + w_r + w_l)\sin\phi, \\ \dot{\phi} &= \frac{R}{L}(u_r - u_l + w_r - w_l), \end{aligned} \tag{5.1}$$

where R is the radius of the wheels, L is the length of the wheel base, u_r and u_l are the velocity input to the right and left wheel corrupted by zero mean Gaussian noise, w_r and w_l respectively. A discrete time model from these equations are obtained as follows:

$$\begin{aligned} x(t+1) &= x(t) + \frac{R}{2}(u_r + u_l)\cos\phi(t)\Delta t + \frac{R}{2}(\omega_1 + \omega_2)\cos\phi(t)\Delta t, \\ y(t+1) &= y(t) + \frac{R}{2}(u_r + u_l)\sin\phi(t)\Delta t + \frac{R}{2}(\omega_1 + \omega_2)\sin\phi(t)\Delta t, \\ \phi(t+1) &= \phi(t) + \frac{R}{L}(u_r - u_l)\Delta t + \frac{R}{L}(\omega_1 - \omega_2)\Delta t, \end{aligned} \tag{5.2}$$

where ω_1 and ω_2 are the noise components of the input velocities. The map features/landmarks are assumed to be stationary. The i^{th} point feature is define as (x_t^i, y_t^i) and the j^{th} line feature as (r_t^j, θ_t^j). The landmark process model for the i^{th}

point feature is,

$$x^i_{t+1} = x^i_t,$$
$$y^i_{t+1} = y^i_t \tag{5.3}$$

and the process model for j^{th} line feature is,

$$r^j_{t+1} = r^j_t,$$
$$\theta^j_{t+1} = \theta^j_t \tag{5.4}$$

Thus, the state transition function needed by the EKF for the prediction step is defined.

D. Observation Models

1. Observation Model for Line Features

In this subsection the observation model used in the observation update step in EKF is defined for line features. Let the mean pose estimate for the robot be (x, y, ϕ). Let l be laser sensor position offset distance in the x direction from the center of the robot. Let the mean pose estimate for the robot be (x, y, ϕ). Let l be laser sensor position offset distance in the x direction from the center of the robot (see Fig. 2). Let (r, θ) be the line parameters observed by the robot for a line using the line model

$$\rho cos(\theta - \alpha) - r = 0, \tag{5.5}$$

where (ρ, α) is a point on the line expressed in polar coordinates.

Let (r_i, θ_i) be the parameters of the corresponding line parameters in the global reference frame which is a part of the EKF state vector. Note that (r, θ) is in the

laser frame . Then the observation model for the line is as follows:

$$r = r_i - lcos(\theta - \phi) - dcos(\theta_i - \alpha) + v_1$$
$$\theta = \theta_i - \phi + v_2 \tag{5.6}$$

where, $d = \sqrt{x^2 + y^2}$, $\alpha = tan^{-1}(y/x)$, v_1 and v_2 are zero mean Gaussian random variables representing the additive noise. If $r < 0$ then $r = -r$ and $\theta = \theta + \pi$.

2. Observation Model for Point Features

In this subsection the observation model used in the observation update step in EKF is defined for point features. Let (x_1, y_1) be the observation of the point feature in the laser frame. Let (x_i, y_i) be the cartesian coordinates of the point feature in the global reference which is part of the EKF state vector. Then the observation model for the point feature is defined as follows:

$$x_1 = x + lcos(\phi) + rcos(\phi + \theta) + v_1$$
$$y_1 = y + lsin(\phi) + rsin(\phi + \theta) + v_2 \tag{5.7}$$

where $r = \sqrt{x_1^2 + y_1^2}$, $\theta = tan^{-1}(y_1/x_1)$, v_1 and v_2 are zero mean Gaussian random variables representing the additive noise.

Please refer to appendix C on how to extract these point and line features from laser range data.

E. Summary

The Hybrid SLAM algorithm can be summarized as follows.

- **Initialization:**
  - Initialize the EKF assuming the initial robot pose as origin.

  - Initialize each grid in the OG map with occupancy probability 0.5.

- **Localization using EKF:**

  - Extract point and/or line features from laser range readings.
  - Generate the joint posterior of EKF states using Eqs. 3.12-3.16.
  - Marginalize the joint distribution (posterior) over the map features to get the belief on the robot pose.

- **Dense mapping using the proposed frequentist approach:**

  - Get the belief on robot pose from the localization step.
  - Update the OG grids following the estimation equations for frequentist mapping given in Table II.

CHAPTER VI

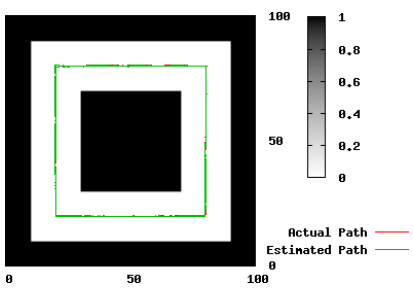
RESULTS AND DISCUSSIONS

A. Introduction

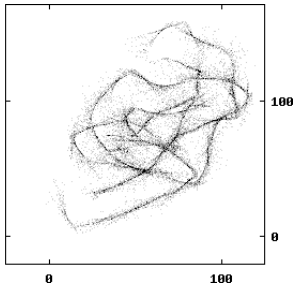
The results of applying Hybrid SLAM to large distributed environments are presented in this chapter to demonstrate the efficacy and potential of the developed methods. Firstly, simulation results are presented for maps containing multiple loops. It is followed up with application of Hybrid SLAM in experiments for checking its capabilities for real world application. It is one of the end to end scenarios that the Hybrid SLAM software has been designed to handle. The experiments were performed in the corridors of the Houston building in the West Campus of Texas A&M University using an iRobot Create fitted with a Hukoyo URG laser sensor. Lastly the results are provided for a few datasets from the Radish website ([55]) to show the efficacy of the developed methods/software to handle large complex datasets. These results will also help to provide a benchmark against other SLAM solutions as they are widely used as a kind of standard in the SLAM community. The results are followed by discussion to put the results in perspective.

B. Simulation Results

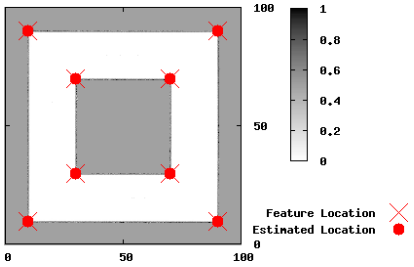
Simulations are performed where a mobile robot explores a previously unknown corridor like environment and the hybrid methodology is used to solve the mapping problem. Two kinds of 1-D laser sensor with different noise characteristics were considered: a) a noisy sensor with noise covariance $\sigma_r = 0.2m$ and b) an accurate laser sensor such as a SICK laser range sensor with $\sigma_r = 0.01m$. The maximum range of the sensors is assumed to be 40 meters. Just to give a context, the SICK LMS laser



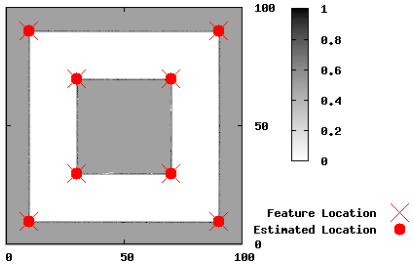
(a) Original map



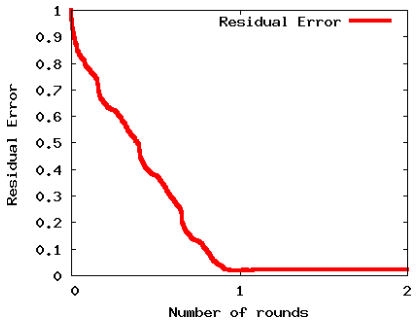
(b) Raw odometry map



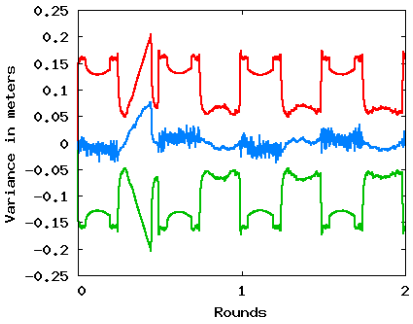
(c) Final map after one round



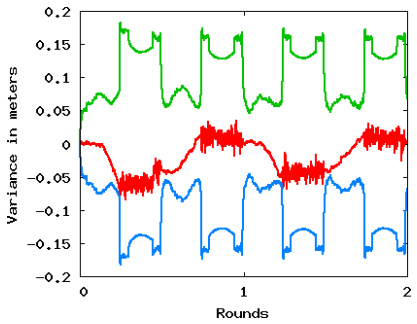
(d) Final map after two rounds



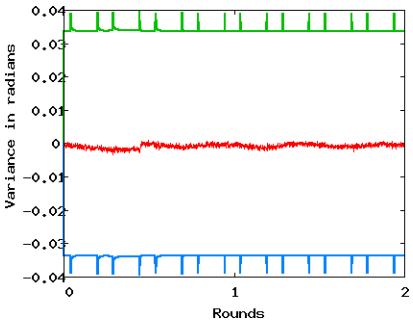
(e) Residual error plot



(f) x error and 3σ envelop

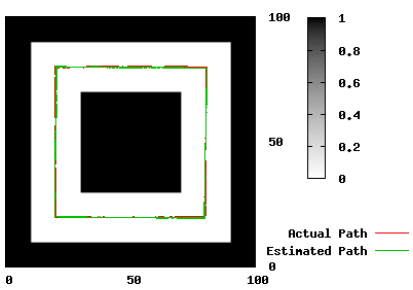


(g) y error and 3σ envelop

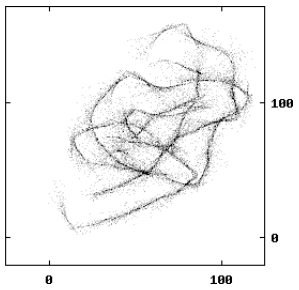


(h) ϕ error and 3σ envelop

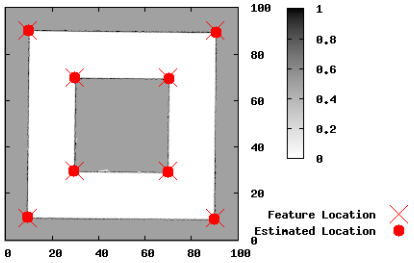
Fig. 3. Simulation results for Map 1 with accurate Laser range sensor



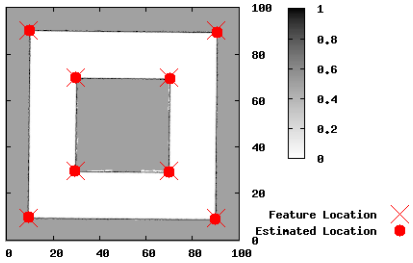
(a) Original map



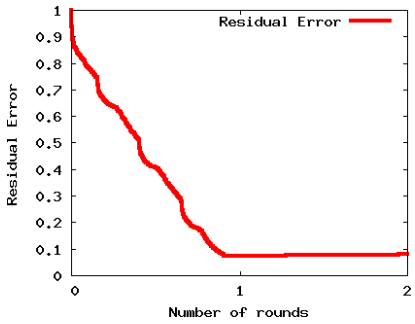
(b) Raw odometry map



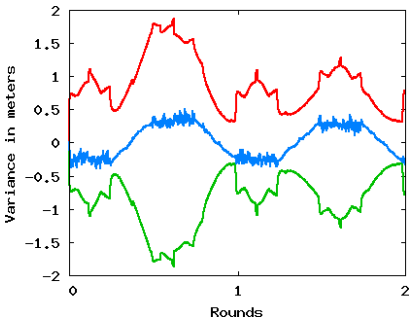
(c) Final map after one round



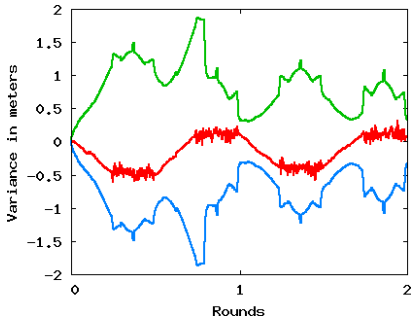
(d) Final map after two rounds



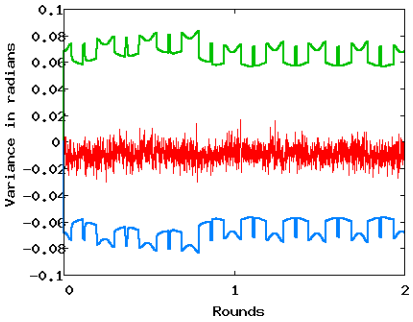
(e) Residual error plot



(f) x error and 3σ envelop

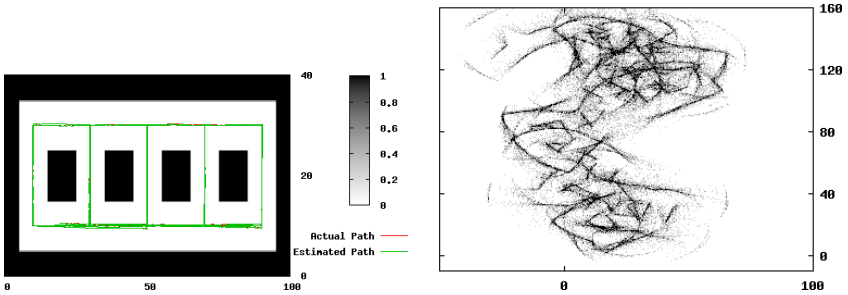


(g) y error and 3σ envelop

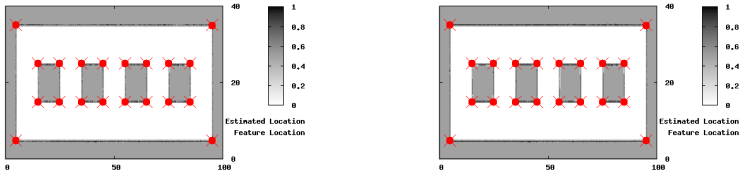


(h) ϕ error and 3σ envelop

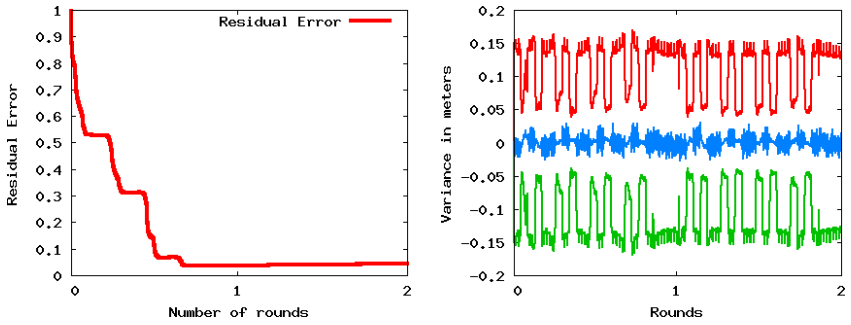
Fig. 4. Simulation results for Map 1 with noisy Laser range sensor



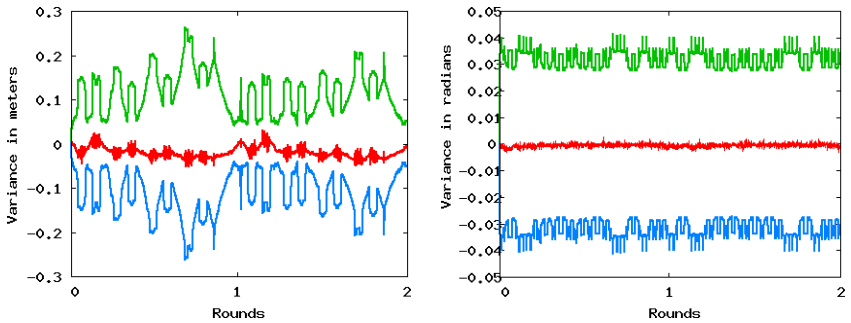
(a) Original map (b) Raw odometry map



(c) Final map after one round (d) Final map after two rounds

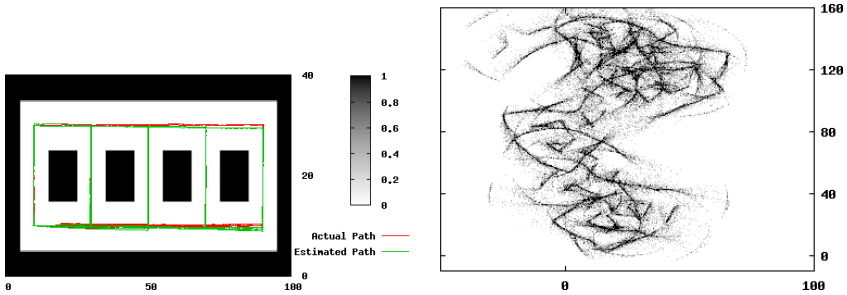


(e) Residual error plot (f) x error and 3σ envelop

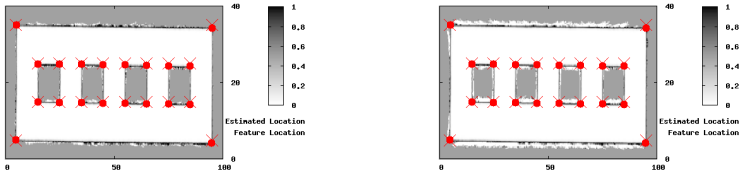


(g) y error and 3σ envelop (h) ϕ error and 3σ envelop

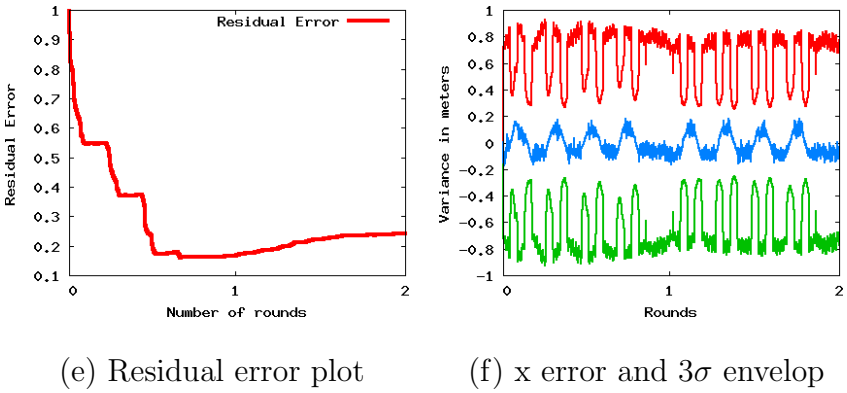
Fig. 5. Simulation results for Map 2 with accurate Laser range sensor



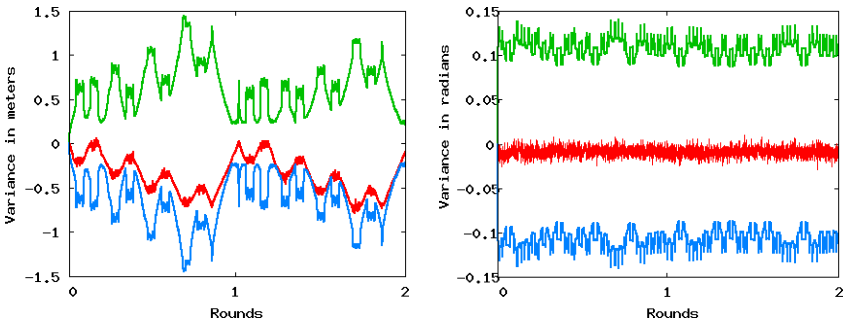
(a) Original map (b) Raw odometry map



(c) Final map after one round (d) Final map after two rounds



(e) Residual error plot (f) x error and 3σ envelop



(g) y error and 3σ envelop (h) ϕ error and 3σ envelop

Fig. 6. Simulation results for Map 2 with noisy Laser range sensor

sensors have a range from 60 to 80 meters on the average. A feature based EKF is employed for localization. The features used are point features which represents corners (intersection of line segments) in the environment and are assumed to be reliably extracted from the laser range data. The feature extraction algorithm as described in appendix C can be used to extract line segments from raw laser scan data to get the point features in case of experiments. Given the *pdf* of the robot pose estimate from the localization step, the frequentist mapping algorithm is used to estimate the distributed map of the environment. Figs. 3-6 show the results of our simulation experiments. Two different maps are considered in these experiments. The first map (Map 1) consists of a large cyclic corridor of side 100 meters while Map 2 is a long hallway (100m x 40m) with 4 cyclic corridors. A total of 2 laps of each map is made. The total length of the runs was approximately 1 km for each Map 1 and Map 2. Each of these maps are explored using both the noisy, as well as the accurate laser range sensor.

Figs. 3-4 represent the results for Map 1 while Figs. 5-6 represent the results for Map 2. In each of these figures, sub-figure (a) shows the original map along with the actual as well as the estimated robot trajectory, sub-figure (b) shows the raw odometry data, sub-figures (c) and (d) show the estimated maps, along with the features and their estimates, after the completion of one and two rounds respectively. In sub-figure (e), we show the total error in the map as a function of the number of rounds the robot makes and is defined as the sum of the absolute values of the component-wise error between the estimated map and the true map divided by the total number of grids. This may be interpreted as the fraction of the map that has not converged. Sub-figures (f)-(h) show the error in the estimates of the x, y, and ϕ co-ordinates of the robot along with their associated 3σ uncertainty bounds.

These figures give us an idea as to how well the algorithm is performing and

also give us valuable practical insight into the algorithm. The reason we chose these examples is because of the well-known challenge the maps with multiple cycles pose to SLAM algorithms. The raw odometry maps (sub-figure (b) in the plots) show how the map would look like if no localization was performed and the mapping was accomplished by using the robot pose as calculated by integrating the odometry readings. It is evident that pure odometry is not enough for accomplishing the mapping and localization tasks. The results show that the hybrid algorithm is able to map a large area with multiple cycles without any problems even though the sensors are noisy. Sub-figures (c), (d) in the plots show us qualitatively that the maps were successfully estimated.

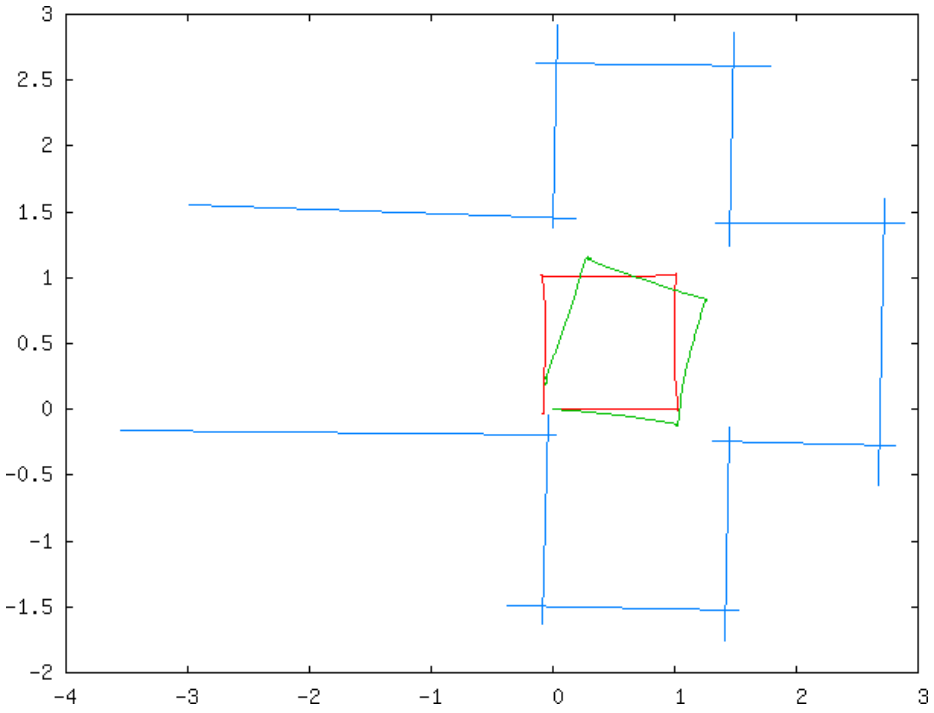
As expected, the quality of the estimated map, as in the sharpness of the edges, was better for the accurate sensor than for the noisy sensor. From sub-figure (e) in the plots it can be seen that the error is approximately in the range of 0.10-0.25 for the noisy sensor while it is less than 0.05 for the accurate sensor. Thus, the map estimates improve quantitatively for more accurate sensor and reinforces the intuitive idea that much larger environments can be mapped with accurate sensors. From sub-figure (e), it is clear that the estimate of the map practically converges within one round. This is the case since the number of observations of any component that is in the robots field of view is high enough during the first round. It can also be seen from the total map error plots (sub-figure (e)) that the mapping algorithm seems to converge exponentially fast. It is left as future work to investigate if any rigorous claims on convergence rate can be made for the frequentist mapping technique.

The algorithm had no problems in closing large loops as the ones shown here and we did not have to make any heuristic corrections when such a loop was closed. In fact, the size of the map, or the number of cycles in it, is really never a problem for this method till the EKF remains consistent. In sub-figures (f)-(h), the true errors in

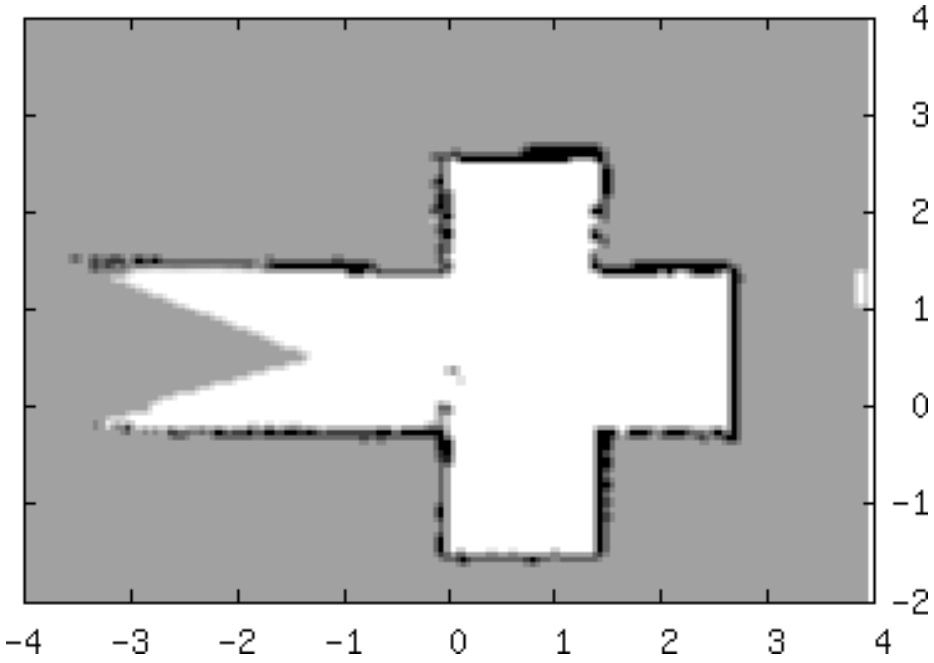
the estimates of the pose of the robot remain within the 3σ uncertainty bounds and show that the EKF used for the Bayesian sub-problem does indeed remain consistent. This is not a rigorous test of the consistency of an EKF but is routinely used by the filtering community as a test for the consistency of a filter. The hybrid algorithm has also been tested on several other maps with satisfactory results to check the robustness of the method in practice. We note here that in Fig. 6, sub-figure (g), the error in the estimates in the y direction is very close to the 3σ boundary and thus, the EKF starts to become inconsistent. This results in the entire hybrid algorithm becoming inconsistent and is evidenced in the relatively high error (≈ 0.25) in the figure when compared to the other maps (< 0.1). Thus, maintaining the consistency of the Bayesian filter is key to the overall performance of the hybrid scheme. It should also be noted that the noise characteristics for the noisy sensor is much higher than what is seen in practice for laser range finders. This shows that the Hybrid SLAM is robust and has potential for real life application.

C. Experimental Results from LASR Lab

The Land Air and Space Robotics (LASR) Laboratory in the Houston Building was used to run experiments inside the Houston building using an iRobot Create fitted with a Hukoyo URG laser sensor. The robot was also fitted with a Dell Mini laptop running on Ubuntu. The Player software was used to control the robot and log the laser and odometry data. The robot was remotely controlled by an operator using the Player software running on another laptop. The control signals were transmitted through wireless intranet. The instance of the Player software running on the Dell Mini attached to the top of the robot received the control inputs and recorded the odometry and laser range data to a local log file. The final step of the experiment is

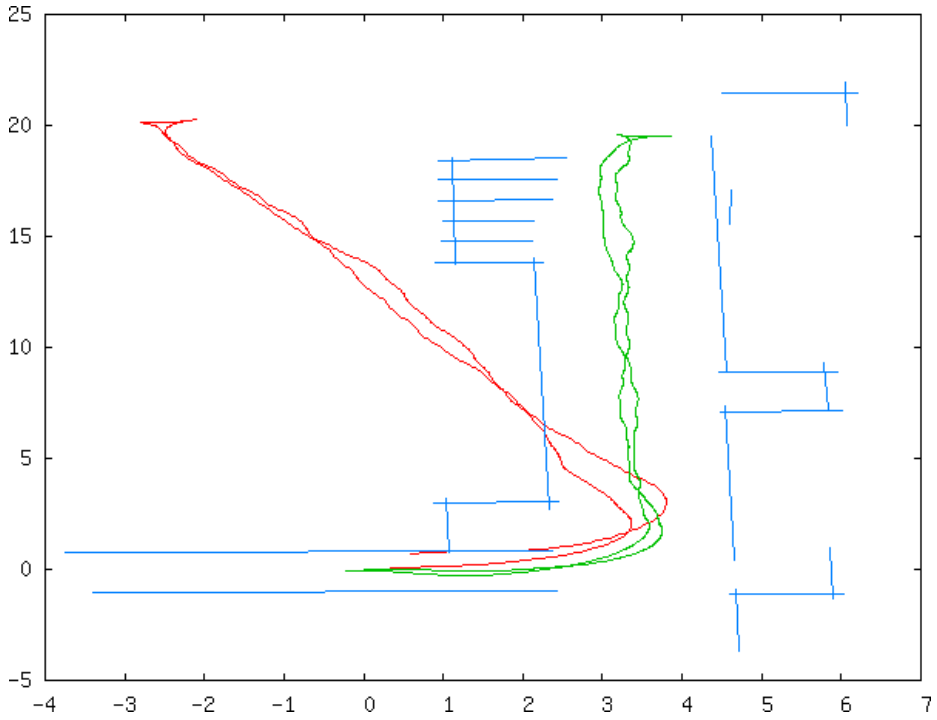


(a) Feature map

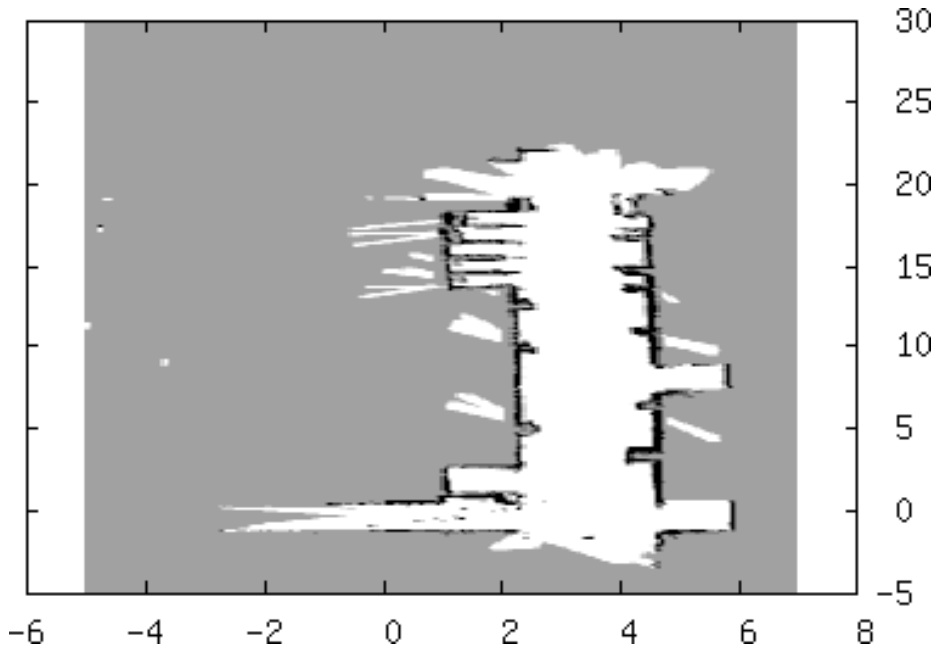


(b) Occupancy grid map

Fig. 7. Experimental result for Run 1

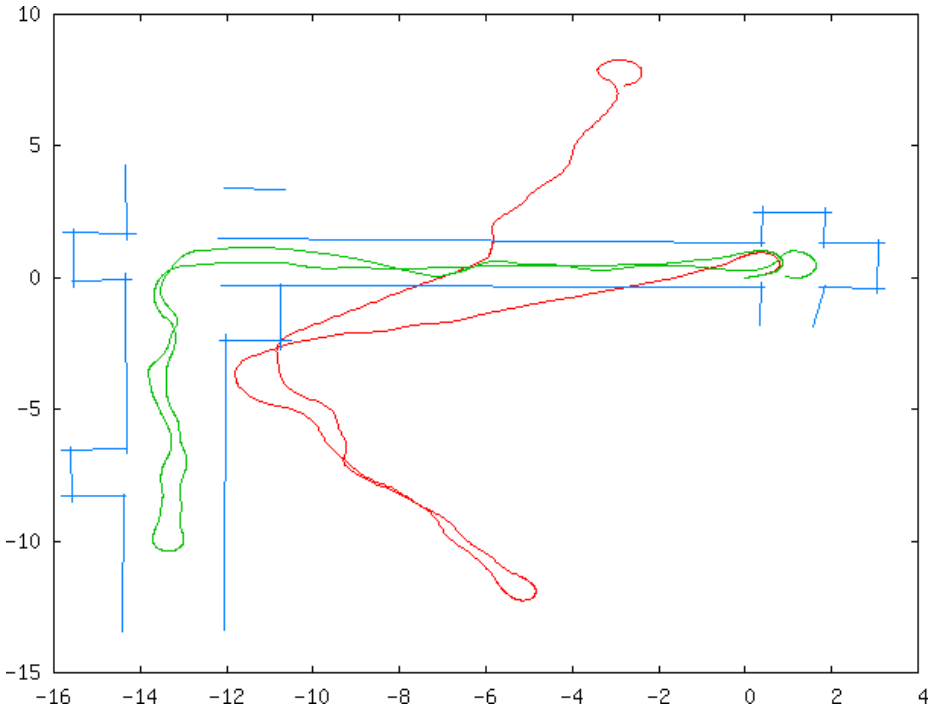


(a) Feature map

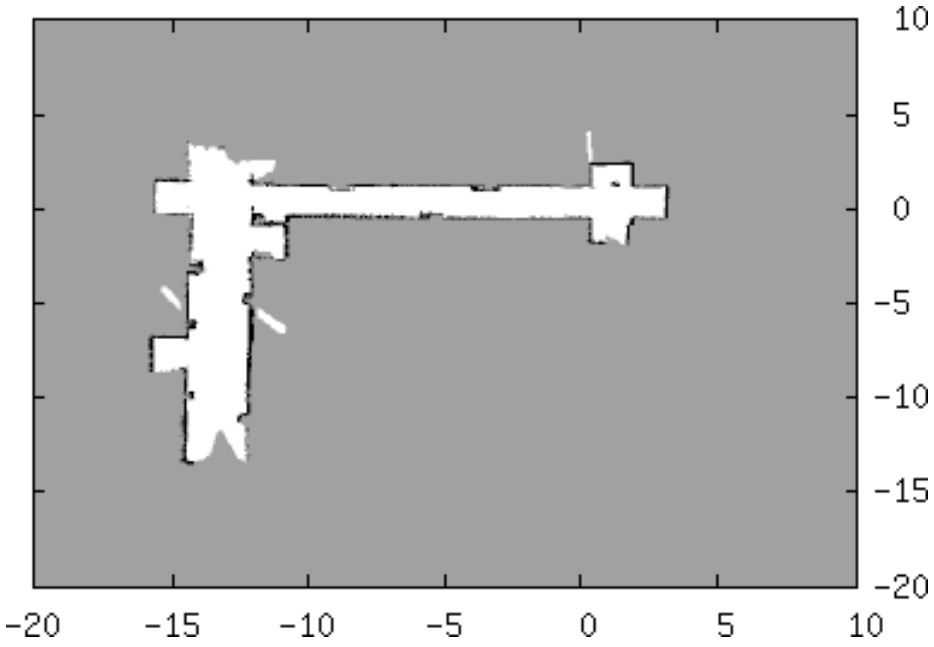


(b) Occupancy grid map

Fig. 8. Experimental result for Run 2

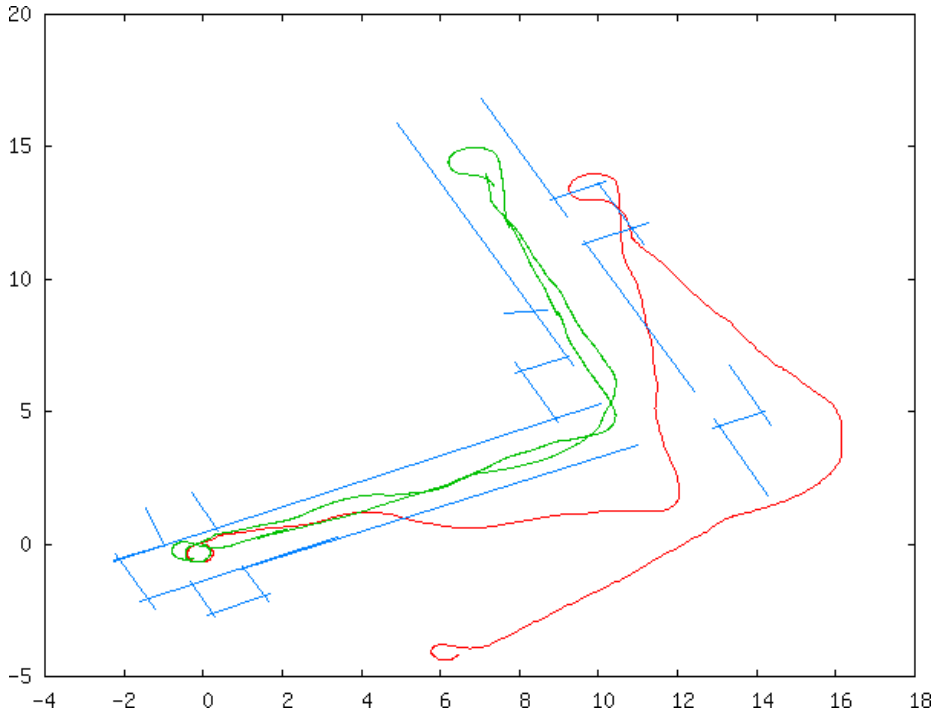


(a) Feature map

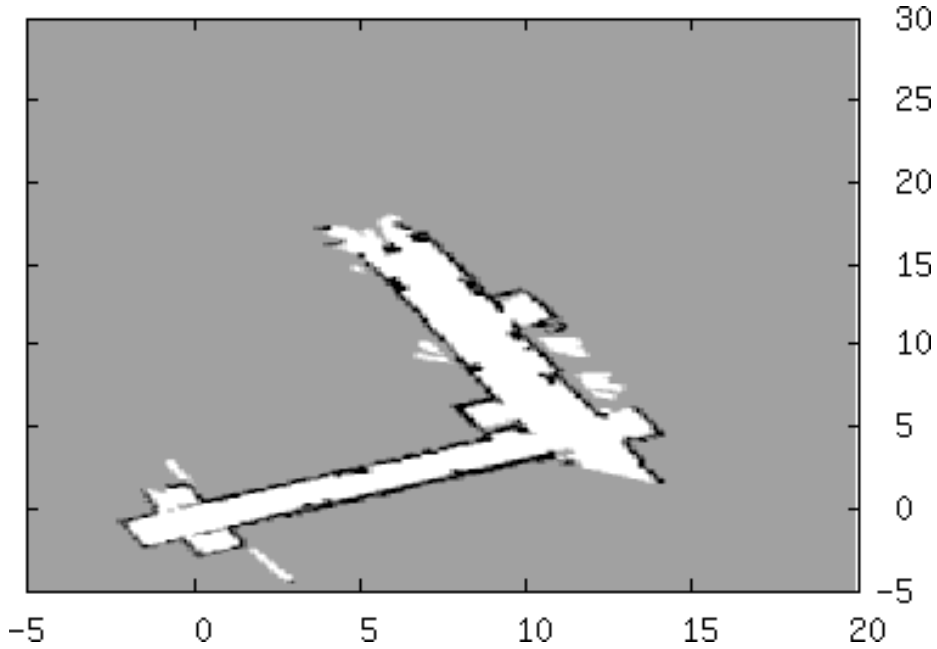


(b) Occupancy grid map

Fig. 9. Experimental result for Run 3



(a) Feature map



(b) Occupancy grid map

Fig. 10. Experimental result for Run 4

to obtain a copy of the log files for the robot runs. The iRobot was made to explore the corridors of the Houston building and such logs were collected for a number of runs of the robot as represented in Fig (7-10).

Hybrid SLAM was used to solve these data sets to generate the estimated map of the environment explored by the robot. The log files were input to the Hybrid SLAM software to solve the SLAM problem at hand. Prominent line features were used by the EKF as map features to localize the robot pose. The line features were extracted from the laser range data as described in appendix C. The odometry path, the estimated path and the feature estimates are plotted in the sub-figure (a) of Figs. (7-10). The path in red shows the raw odometry path and shows that it has substantial error. The estimated path in green shows the estimated mean path generated by the EKF. Qualitatively, by visually inspecting the maps, we see that the EKF was able to correct the error in odometry and successfully localize using the line features. The dense map was built by the frequentist mapping technique using an OG map. Sub-figure (b) in Figs. (7-10) shows that the robot was able to successfully estimate the distributed map and gives us a much more detailed map than is possible by a feature based map.

The robot run lengths varied between 4 meters in Run 1 to about 60 meters in Run 3 and 4. The duration the robot ran to collect the data was between 1 minute for Run 1 and close to 5 minutes for Run 4. The processing time needed by the Hybrid SLAM solver to do the localization and estimate the dense map took considerably less time than the robot exploration times. It took less than 1 minute for Run 1 and about 2 minutes for the rest or the runs. The results are very encouraging and show that the Hybrid SLAM methodology proposed in this dissertation can solve the mapping problem for real life problems in real time. Thus, the proposed Hybrid SLAM methodology is shown to successfully perform under experiments. At the same

time, the Hybrid SLAM software is empirically verified.

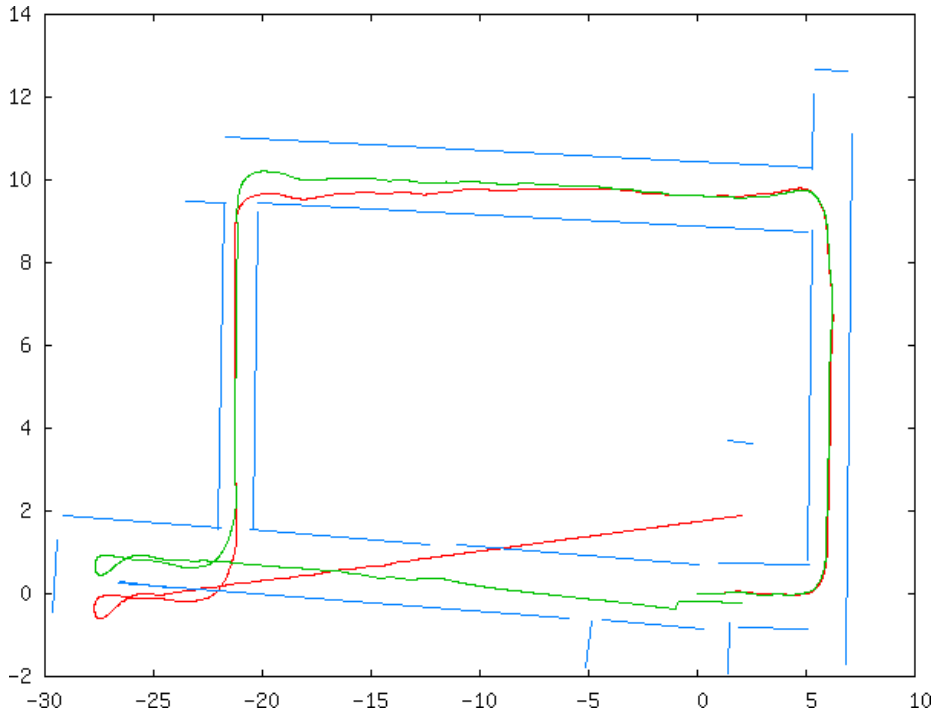
D. Open-source Dataset Results

In this subsection, results are presented from various open-source datasets published through the Radish website [55]. The Radish website contains several datasets collected by various researchers belonging to the SLAM community throughout the world. It provides real experimental data without each researcher having to perform the experiments by himself, saving time and resources needed to perform exploration experiments. It also helps researchers to find common ground to test their algorithms on. Though the website contains just the raw datasets and does not contain any mapping results, still many researchers tend to use these data sets while publishing their work. Special thanks to Andrew Howard and Nicholas Roy for maintaining this repository.

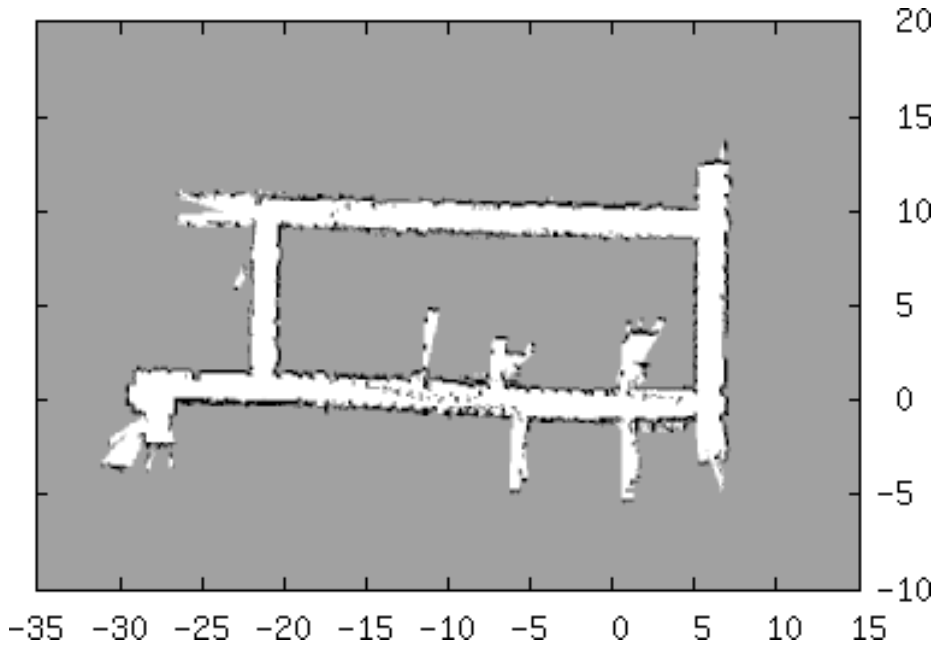
1. USC SAL Building

The Hybrid SLAM solution is tested on the USC SAL data set collected by Dr.Andrew Howard at University of Southern California (USC). It is one of the standard data sets published through Radish [55]. The data set contains two separate tours of the second floor of the USC SAL building by a Pioneer robot fitted with a SICK LMS 200 laser sensor. The laser range sensor is fitted $8cm$ in front of the center of motion of the robot facing forward.

Figs. 11 & 12 show the results from the two tours of the second floor of the USC SAL building. The two data sets are from the same place and a dense map was built for both the data sets to check consistency of the map building process using Hybrid SLAM. Line features were used by the EKF for localizing the robot pose as

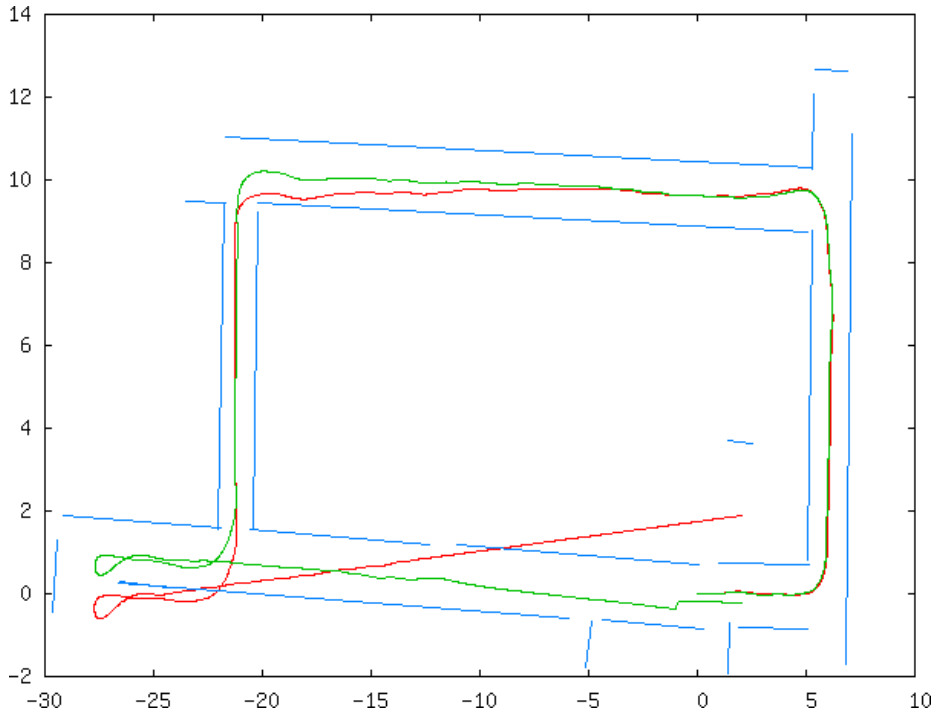


(a) Feature map

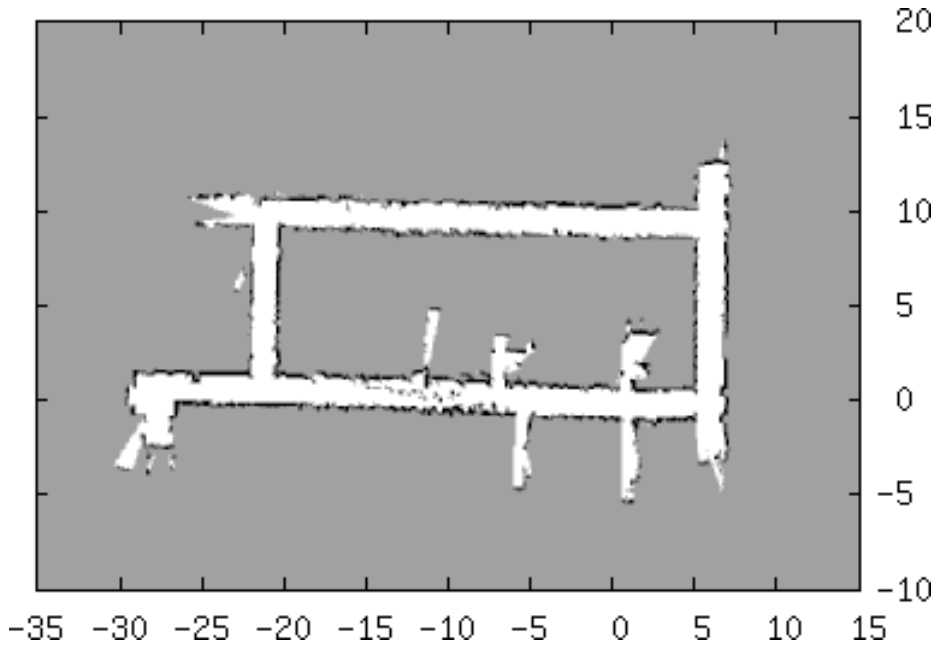


(b) Occupancy grid map

Fig. 11. Results from USC SAL dataset (Run 1)



(a) Feature map



(b) Occupancy grid map

Fig. 12. Results from USC SAL dataset (Run 2)

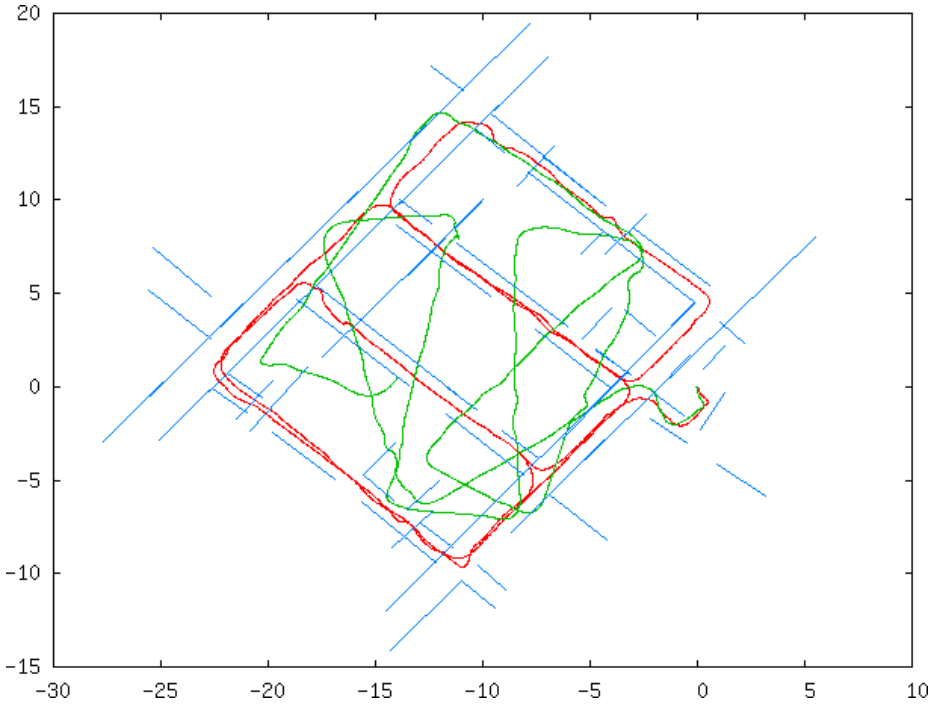
described in appendix C. Sub-figure (a) shows the estimated mean position of the line features, the path given by the raw odometry readings, and, the mean path of the robot as estimated by the EKF. Sub-figure (b) shows the OG map of the distributed environment. The results show that the second floor map of the USC SAL building was successfully built without significant errors.

The mapping algorithm gives us very similar results for both the runs giving us more confidence on the solution provided by the proposed algorithm. The distance traveled by the robot is about 70 meters and the time duration of the runs were a little over 5 minutes each. The total time taken to solve each data set was less than 30 seconds for each of the runs. Thus, the Hybrid SLAM algorithm may be run in real time to get recursive estimates of a map.

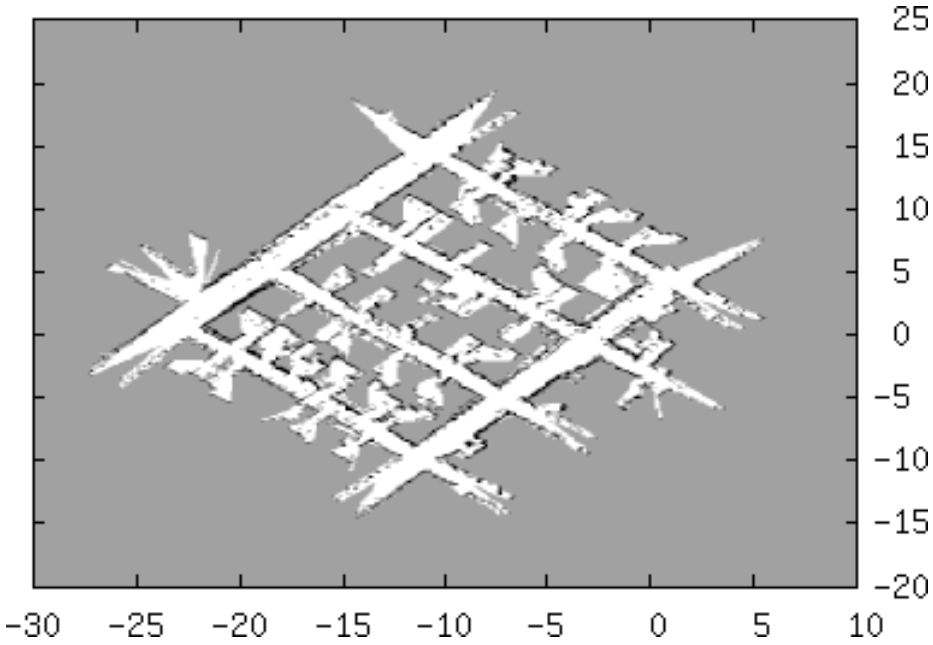
2. Intel Laboratory

The Intel Laboratory dataset is another dataset published on the Radish website which contains the data collected by a Pioneer robot fitted with a SICK LMS 200 laser range sensor. The data was acquired by the robot while making a tour of the part of the Intel laboratory in Hillsboro, Oregon. Special thanks to Maxim Batalin for collecting this data. The laser range finder was mounted 8cm in front of the center of motion of the robot, facing forward. The path of the robot contains multiple loops through the corridors of the building. Though the robot does not enter any of the rooms, the rooms are partially observed because of the angular view of the laser range finder. This data set is collected over a 10 minute run by the robot and covers an area about 30 meters by 30 meters. It is one of the bigger and more difficult data sets considering the area and also the large number of loops.

Hybrid SLAM software was used to solve the data set to get the OG map of the environment as shown in Fig. 13(b). The localization was done using EKF using



(a) Feature map



(b) Occupancy grid map

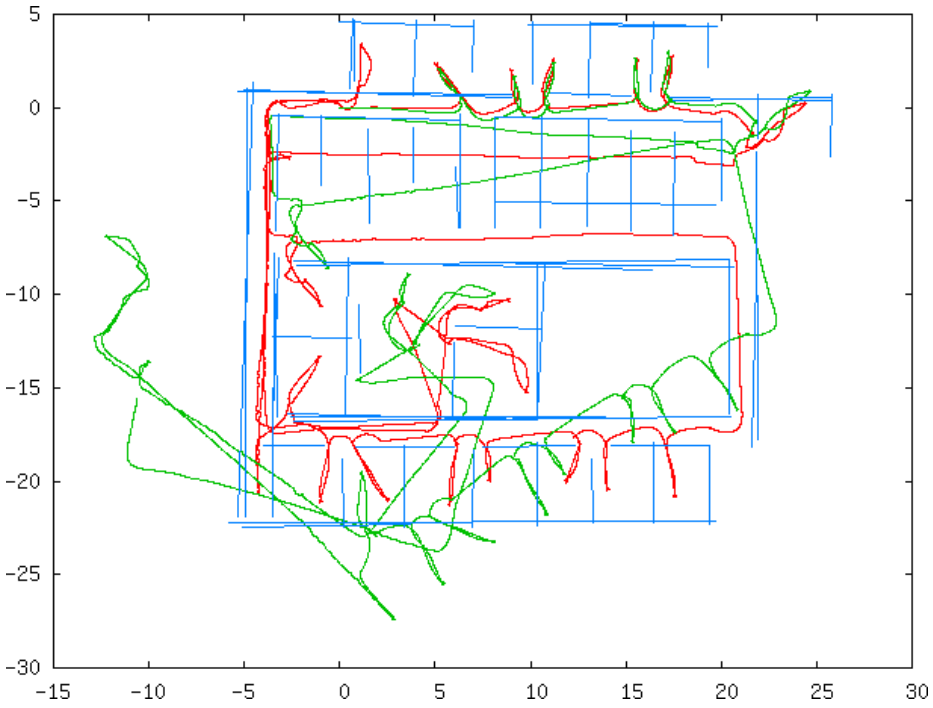
Fig. 13. Results for Intel dataset

line features. The line features were extracted from the raw laser range data using the feature extraction algorithm as described in appendix C. The feature extraction was done such that only long reliable line segments were extracted and other small, unreliable line segments were rejected. This helps the EKF to keep only sparse reliable features. The green line in Fig 13(a) denotes the raw odometry data whereas the red line denotes the estimated robot path produced by the EKF. As can be seen that the raw odometry path has significant error and is in no way suitable for mapping. The EKF on the other hand is able to consistently estimate the robot pose which is used by the frequentist mapping algorithm. The frequentist mapping algorithm generates the dense map of the environment as shown in Fig. 13(b). The dense map is visually studied and is found to satisfactorily estimate the map.

The robot took 10 min to explore the building and collect the data. The Hybrid SLAM software was able to process the data set in 14 seconds which is many fold lesser than the exploration time. So it is seen that Hybrid SLAM can consistently estimate the dense map of a 30 meter by 30 meter region in real time.

3. SDR Building

The SDR data set is one of the most complex datasets on the Radish website among the data sets collected using a laser range sensor. The area to be mapped is an entire floor of the SDR building which contains a large number of rooms. The robot enters into each of the rooms briefly and then keeps continuing to the next room. As is expected in an indoor environment, the corridors are interconnected which leads the robot to come back to the same place multiple times. A Pioneer robot fitted with a SICK LMS 200 laser range finder was used to explore the building. The laser was mounted 8cm in front of the center of the robot facing forward. It took the robot 50 minutes to explore this complex geometry. For all this time, the odometry and laser



(a) Feature map

(b) Occupancy grid map

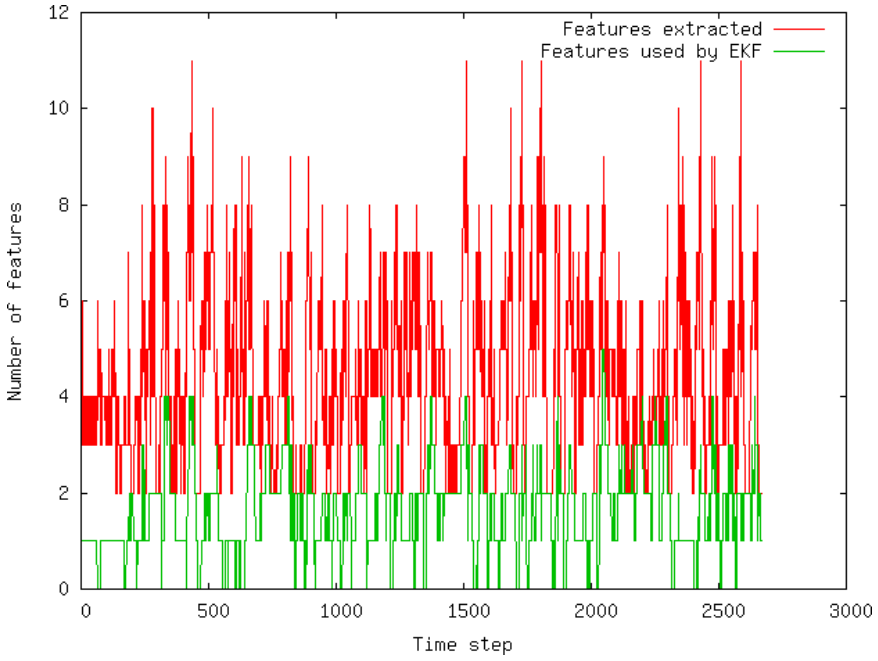
Fig. 14. Results for SDR dataset

observation data was stored in a log file. The robot was tele-operated by a human operator for the entire duration of robot navigation.

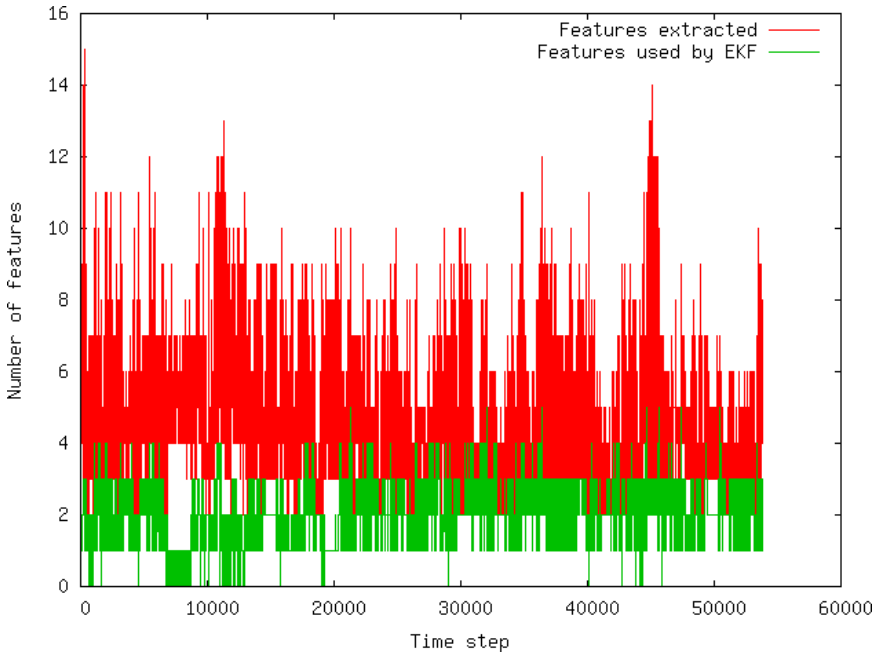
The Hybrid SLAM solver is used to map the distributed environment explored by the robot. The localization is done using CEKF to produce a consistent pose estimate of the robot and also a feature map as shown in Fig. 14(a). The distributed environment is then mapped using the frequentist mapping technique as shown in Fig. 14(b). As can be seen from Fig 14 that the hybrid solver has successfully mapped the entire building consistently. The feature map depicts the position of the longer walls in the building. As expected, these walls are almost perpendicular from each other which is very common for such indoor environments. The feature map does not contain enough information to make out the rooms for which we turn to the dense map produced by the frequentist technique. Here we can clearly see all the rooms that were visited by the robot during exploration. Since we do not know the ground truth we can only inspect the results visually which leads us to say that the hybrid slam did a good job of mapping this complex environment. Also the solver took only 2.5 minutes to produce both the feature map as well as the occupancy grid map. This shows that Hybrid SLAM solves such a complicated data set in about $1/10^{th}$ of the time taken by the robot to record the dataset.

E. Discussion

In Hybrid SLAM, localization is done using a sparse set of features in the feature based EKF. This is attained by using a small set of prominent features at every step in the EKF and not considering all the features extracted from the raw sensor data. Fig. 15 provides a comparison of the number of features extracted from the raw laser data and the number of features actually used by the EKF for the Intel and SDR



(a) Intel Map



(b) SDR dataset

Fig. 15. Comparison between number of features extracted by feature extraction algorithm and number of features used by the EKF

Table III. Mapping times for datasets

Dataset	Exploration Time	Mapping Time	Number of EKF features	Grid size
Intel Lab	10 min	14 sec	57	10 cm
SDR	50 min	154 sec	68	10 cm

dataset. We focus on these two datasets for the purposes of discussion as these are the most complicated and challenging datasets that we have solved in terms of map size, complexity and duration of exploration. The sparse feature assumption helps us tackle the issues faced with EKF based SLAM, namely computational burden, data association and loop closing. From Fig. 15 we see that the total number of features extracted from the raw laser data varies between 2 and 15 whereas the number of features used as observations in EKF varies between 0 and 4. On the average, EKF needs two features to consistently estimate the robot pose. The data association for such low number of features using either nearest neighbor algorithm or JCBB remains reliable without adding much computational burden on the Hybrid SLAM solution. On the average, about $1/3^{rd}$ of the extracted features are used as observations for the Intel dataset and the rest are thrown away. For the SDR dataset about half of the extracted features were thrown away and half were used by the EKF. The fraction of the number of features to be used will depend on the kind of environment the robot is exploring and also the accuracy of the laser range finder. Thus we successfully impose the sparse feature assumption without loosing consistency and keeping the computational burden low at the same time.

The important motivations behind the current work was to have a consistent mapping algorithm which has low computational complexity so that it can map large distributed environments. The frequentist mapping consistently estimates the dis-

tributed environment given that the localization part remains consistent. The consistency of the frequentist part is rigorously proved in appendix A. The consistency for the EKF based SLAM also hold as no approximations have been made to the Bayesian formulation as long as its underlying assumptions of Gaussian noise and linearization holds good. The time complexity for EKF based SLAM in general is cubic in the number of states in the EKF. However, using CEKF the time complexity is reduced to quadratic in local map features with global updates involving the full state space being required only rarely. The computational burden of the EKF based SLAM is further kept under limits by enforcing the sparse feature assumption. The frequentist mapping technique has linear complexity in the number of map grids being estimated. Further, it has been shown in subsection E that under bounded sensor noise it has constant time complexity. The constant time complexity is very important for dense mapping as the number of grids used to define a dense map is very high and anything higher will quickly render the computational burden intractable for real time application.

The details of mapping solution for the Intel and SDR datasets are captured in Table III. The exploration time over which the robot recorded the Intel dataset was about 10 mins and Hybrid SLAM was able to solve it in 14 seconds which is less than $1/40^{th}$ the duration of robot run time. In the case of SDR dataset, the robot needed 50 minutes to run through the corridors and rooms to collect the data. Hybrid SLAM solved it in approximately 2.5 minutes which is about $1/20^{th}$ the duration the robot took to explore the map. This implies that Hybrid SLAM with its current implementation is equipped to solve the distributed mapping problem for such big environments in real time which is a significant step forward for the SLAM problem.

The dense map is created using a grid size of 10 cm which translates to approximately 10^5 grids in a 2-dimensional map covering an area of 30 meters by 30 meters.

Grid size of 10 cm for such large environments creates a good resolution dense map which can be used for various robot activities like navigation. Since the frequentist mapping algorithm is constant time, it does not really get affected by the total number of grids in the map. Also, since only one copy of the dense map is required (as opposed to particle methods) the memory requirements are quite low (about 100 mb) considering the size of the problem. The results were generated on a standard desktop PC from 2007 running on Debian with a Intel Core 2 Duo processor. The system requirements for running the Hybrid SLAM software can be met by many modern robot platforms, especially the ones which have a laptop on-board. The Hybrid SLAM software runs easily on standard laptop computers without any special requirements. A couple of hundred megabytes of RAM is all the Hybrid SLAM software needs to successfully map areas comparable to the results being presented in this dissertation. Writing the software in the C language and following code optimization techniques have contributed to getting such an efficient solver with low system requirements.

Since Hybrid SLAM takes an order of magnitude less time to solve the distributed mapping problem than the time taken to collect the data, it even gives significant amount of breathing space for the robot to do other activities. It assures the scope of adding more features to the algorithm, like planning, without loosing its real time applicability. These results convincingly prove that Hybrid SLAM and the software developed implementing it can successfully solve the SLAM problem for complicated distributed environments. To the best of our knowledge, such claims have not been made by other distributed mapping algorithms which have consistent guarantees. As mentioned earlier, approximations which sacrifice the consistency guarantees of the method are not beneficial for large scale applications as they tend to fail for explorations extending for a long time. With the guaranteed consistency of the Hybrid SLAM algorithm along with the constant time complexity of the frequentist mapping

algorithm, it could solve much larger datasets in real time than has been achieved before. The datasets presented here are collected from real experiments by other researchers and is directly input to the Hybrid SLAM software. This gives us a high level of confidence in the performance capabilities of the frequentist mapping technique and Hybrid SLAM software to successfully estimate large distributed environments in real time.

CHAPTER VII

CONCLUSIONS

A novel frequentist mapping approach has been presented which is provably consistent and converges to the true map probabilities almost surely. The frequentist mapping algorithm has been applied to the case of occupancy grid maps when the sensor being used is a 1D range finder. In general, the complexity of the proposed mapping algorithm is linear in the number of map components/grids that the robot observes. In practice, under the bounded noise assumption, it has been shown that the computational complexity is constant in the number of map components and makes it amenable to real time implementation. Thus a consistent and computationally efficient solution to the mapping problem has been developed.

A Hybrid SLAM solution was developed which uses the principle of "first localize and then map". A feature based EKF is used to localize and compute the pose estimate at each time instant by only including good reliable sparse features in its state vector. Keeping sparse features helps us avoid the ailments that EKF based algorithms suffer from, e.g., data association and computational burden. Thus, the robust EKF gives consistent pose estimate of the robot at each time instant and this pose estimate/belief is used by the frequentist mapping algorithm to map the distributed/dense environment.

The Hybrid SLAM solution has been tested under simulations where it performs well. It also shows an almost exponential convergence in error. However, any rigorous theoretical justification is not available currently and needs investigation. The EKF under the assumption of sparse features remains consistent and is able to close multiple loops which most localization algorithms struggle with.

Applicability of the developed solution to real life problems has been established

through experiments run using an iRobot Create. Further, the efficacy of the method for real time solution was demonstrated by solving publicly available complicated standard datasets.

Highly efficient Hybrid SLAM software has been developed and tested using the C language which is capable of running in real time with very low system requirements. A modular approach was resorted to for the development; simulation environment, feature extraction, feature based EKF and frequentist mapping being the different modules. Thus, on the software side, a complete solution to the SLAM has been developed which could be extended to include more algorithms and could be used either as a test bed or in real experiments.

The method has been demonstrated to perform well under a variety of scenarios and shows great potential. As the computational complexity under implementation is constant, it scales to large environments. Under the sparse feature assumption and using computationally efficient implementations of the localization algorithms, it is envisioned that much bigger maps can be mapped by the Hybrid SLAM method than has been successfully mapped till now. As has been seen from experiments, the Hybrid SLAM software takes an order of magnitude less time to estimate the dense map than the robot takes to record the data, thus showing great real-time potential. Hence it is concluded that the Hybrid SLAM solution proposed has great potential for consistent real time mapping of large distributed environments.

A. Future Research

- The frequentist mapping algorithm is based on averaging and the law of large numbers. It lends itself naturally to be extended to the multiple robot scenario which would include one more averaging step over the multiple robots being

used for mapping.

- More robust and efficient improvements to the feature based EKF could be investigated to improve the performance of the localization step and in turn the Hybrid SLAM solution. Also, the feature extraction and data association remains potential issues for complicated maps.

- Theoretical study of the convergence behavior of the frequentist mapping algorithm would be an important step. From our observations so far, we see that the map probabilities converge exponentially and it could possibly be proved under suitable conditions.

- Mobile robots need to navigate from one point to another (path planning) to perform various tasks. Planning along with SLAM would make the robot completely autonomous, and thus, will remain the holy grail of this line of research.

REFERENCES

[1] S. Chakravorty and J. Junkins, "Motion planning in uncertain environments with vision-like sensors," *Automatica*, vol. 43, no. 12, pp. 2104–2111, 2007.

[2] S. Chakravorty and R. Saha, "Hybrid hierarchical motion planning," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automation Magazine*, pp. 99–108, 2006.

[4] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping: Part II," *IEEE Robotics and Automation Magazine*, pp. 108–117, 2006.

[5] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*. San Francisco, CA: Morgan Kaufmann, 2002.

[6] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 229–241, 2001.

[7] *Concurrent localisation and map building for mobile robots using ultrasonic sensors*, vol. 3, July 1993.

[8] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Auton. Robots*, vol. 5, no. 3-4, pp. 253–271, 1998.

[9] J. Nieto, J. Guivant, and E. Nebot, "Denseslam: Simultaneous localization and dense mapping," *Int. J. Rob. Res.*, vol. 25, pp. 711–744, Aug. 2006.

[10] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *In IEEE Int. Conference on Robotics and Automation*, 1985.

[11] P. SkrzypczyńSki, "Simultaneous localization and mapping: A feature-based probabilistic approach," *Int. J. Appl. Math. Comput. Sci.*, vol. 19, no. 4, pp. 575–588, 2009.

[12] A. Mourikis and S. Roumeliotis, "Analysis of positioning uncertainty in simultaneous localization and mapping (slam)," in *Proc. of the IEEE/RSJ Int. Conference on Robotics and Intelligent Systems (IROS)*, 2004, pp. 13–20.

[13] H. Choset, "Sensor based motion planning: The hierarchical generalized voronoi graph," Ph.D. dissertation, California Institute of Technology, Los Angeles, CA, 1996.

[14] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 125–137, 2001.

[15] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proc. of the Twelfth National Conference on Artificial Intelligence*. The MIT Press, 1994, pp. 979–984.

[16] B. Kuipers and Y. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *J. of Robotics and Autonomous Systems*, vol. 8, pp. 47–63, 1991.

[17] M. Mataric, "A distributed model for mobile robot environment-learning and navigation," Massachusetts Institute of Technology, Cambridge, MA, Tech. Rep.,

1990.

[18] H. Shatkay and L. Kaelbling, "Learning topological maps with weak local odometric information," in *In Proc. of IJCAI-97. IJCAI, Inc*, 1997, pp. 920–929.

[19] J. Blanco, J. Fernndez-Madrigal, and J. Gonzalez, "Towards a unified bayesian approach to hybrid metric-topological slam," *IEEE Trans. on Robotics*, vol. 24, pp. 259–270, 2008.

[20] J. L. Blanco, J. González, and J. A. Fernández-Madrigal, "Subjective local maps for hybrid metric-topological slam," *Robot. Auton. Syst.*, vol. 57, no. 1, pp. 64–74, 2009.

[21] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[22] M. Walter, R. Eustice, and J. Leonard, "Exactly sparse extended information filters for feature-based slam," *Int. J. Rob. Res.*, vol. 26, no. 4, pp. 335–359, 2007.

[23] M. Altermatt, A. Martinelli, R. Siegwart, and N. Tomatis, "Slam with corner features based on a relative map," in *IEEE Int. Conference on Intelligent Robots and Systems*, 2004, pp. 1053–1058.

[24] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE J. of Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.

[25] ——, "Occupancy grids: a probabilistic framework for robot perception and navigation," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 1989.

[26] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, vol. 9, pp. 61–74, 1988.

[27] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Robotics and Automation, 2005. ICRA 2005. Proc. of the 2005 IEEE Int. Conference on*, Apr. 2005, pp. 2432–2437.

[28] G. Grisetti, G. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, "Speeding up rao blackwellized slam," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2006, pp. 442–447.

[29] C. Stachniss, D. Hähnel, W. Burgard, and G. Grisetti, "On actively closing loops in grid-based fastslam," *Advanced Robotics*, vol. 19, no. 10, pp. 1059–1079, 2005.

[30] A. Eliazar and R. Parr, "Dp-slam 2.0," in *Proc. (ICRA '04). IEEE Int. Conference on Robotics and Automation*, vol. 2, May 2004, pp. 1314–1320.

[31] G. Grisetti, G. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, "Fast and accurate slam with rao-blackwellized particle filters," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 30–38, 2007.

[32] K. Wurm, C. Stachniss, and G. Grisetti, "Bridging the gap between feature- and grid-based slam," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 140 – 148, 2010, selected papers from the 2007 European Conference on Mobile Robots (ECMR '07).

[33] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainly," *Int. J. of Robotic Research*, vol. 5, no. 4, pp. 56–68, 1987.

[34] H. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE Trans. on Robotics and Automation*, vol. 4, no. 1, pp. 23–31, 1988.

[35] N. Ayache and O. Faugeras, "Building, registrating, and fusing noisy visual maps," *Int. J. Rob. Res.*, vol. 7, no. 6, pp. 45–65, 1988.

[36] J. Crowley, "World modeling and position estimation for a mobile robot usig ultrasonic ranging," in *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1989, pp. 674–680.

[37] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *In Proc. of the IEEE Conf. on Robotics and Automation*, 1985, pp. 138–143.

[38] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous Robot Vehicles*, pp. 167–193, 1990.

[39] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems (IROS '91)*, vol. 3, Nov. 1991, pp. 1442–1447.

[40] ——, *Directed Sonar Sensing for Mobile Robot Navigation.* Norwell, MA: Kluwer Academic Publishers, 1992.

[41] M. Csorba, J. Uhlmann, and H. Durrant-Whyte, "New approach to simultaneous localization and dynamic map building," in *Navigation and Control Technologies for Unmanned Systems*, S. A. Speigle, Ed., vol. 2738. SPIE, 1996, pp. 26–36.

[42] M. Csorba, "Simultaneous localization and map building," Ph.D. dissertation, Universit of Oxford, Oxford, England, 1997.

[43] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, June 2001.

[44] J. Knight, A. Davison, and I. Reid, "Towards constant time slam using postponement," in *Proc. 2001 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, vol. 1, 2001, pp. 405–413.

[45] S. Williams, "Efficient solutions to autonomous mapping and navigation problems," Ph.D. dissertation, University of Sydney, Sydney, Australia, 2001.

[46] J. Tardos, J. Neira, P. Newman, and J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *Artificial Intelligence*, vol. 21, no. 4, pp. 311–330, 2002.

[47] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular slam," *Robot. Auton. Syst.*, vol. 57, no. 12, pp. 1188–1197, 2009.

[48] J. Neira and J. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 890–897, Dec. 2001.

[49] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Proc. of the AAAI National Conference on Artificial Intelligence*, 2002, pp. 593–598.

[50] A. Doucet, N. Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *UAI '00: Proc. of the 16th Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA: Morgan

Kaufmann Publishers Inc., 2000, pp. 176–183.

[51] K. Murphy, "Bayesian map learning in dynamic environments," in *In Neural Info. Proc. Systems (NIPS*, vol. 12. MIT Press, 1999, pp. 1015–1021.

[52] R. Sim, P. Elinas, and J. Little, "A study of the rao-blackwellised particle filter for efficient and accurate vision-based slam," *Int. J. Comput. Vision*, vol. 74, pp. 303–318, Sept. 2007.

[53] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *The Int. J. of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, Aug. 2004.

[54] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[55] A. Howard and N. Roy, "The robotics data set repository (Radish)," 2003. [Online]. Available: http://radish.sourceforge.net/

[56] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. of the ASMEJ. of Basic Engineering*, vol. 82, no. D, pp. 35–45, 1960.

[57] N. Gordon, D. Salmond, and A. Smith, "A novel approach to non-linear/non-Gaussian Bayesian state estimation," in *IEEE Proc. on Radar and Signal Processing*, vol. 140, no. 2, 1993, pp. 107–113.

[58] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, pp. 174–188, 2001.

[59] M. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[60] G. Welch and G. Bishop, "An introduction to the Kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, Tech. Rep., 1995.

[61] S. Chakravorty and R. Saha, "A frequentist approach to mapping under uncertainty," *Automatica (J. of IFAC)*, vol. 47, no. 3, pp. 477–484, March 2011.

[62] S. Chakravorty and J. Junkins, "Intelligent exploration of unknown environments with vision like sensors," in *Proc. IEEE/ASME Int. Conference on Advanced Intelligent Mechatronics*, July 2005, pp. 1204–1209.

[63] ——, "A methodology for intelligent path planning," in *Proc. of the 2005 IEEE Int. Symposium on Intelligent Control, 2005*, June 2005, pp. 592–597.

[64] J. Davis and S. Chakravorty, "Motion planning in an uncertain environment: Application to an unmanned helicopter," in *45th IEEE Conference on Decision and Control*, Dec. 2006, pp. 6814–6819.

[65] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. New York: Springer-Verlag, 1990.

[66] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer, 2003.

[67] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[68] L. Kneip, F. Tâche, G. Caprari, and R. Siegwart, "Characterization of the compact hokuyo urg-04lx 2d laser range scanner," in *ICRA'09: Proc. of the 2009*

IEEE international conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 2009, pp. 2522–2529.

[69] S. Haykin, *Neural Networks*. New Jersey: Prentice Hall, 1994.

[70] H. K. Khalil, *Nonlinear Systems*. New Jersey: Prentice Hall, 1996.

[71] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, MA: Cambridge University Press, 1993.

[72] I. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle," *Robotics and Automation, IEEE Trans. on*, vol. 7, no. 2, pp. 193–204, Apr. 1991.

[73] P. Jensfelt and H. Christensen, "Pose tracking using laser scanning and minimalistic environmental models," *Robotics and Automation, IEEE Trans. on*, vol. 17, no. 2, pp. 138–147, Apr. 2001.

[74] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, "Orthogonal slam: a step toward lightweight indoor autonomous navigation," in *2006 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2006, pp. 5007–5012.

[75] M. Kurisu, H. Muroi, Y. Yokokohji, and H. Kuwahara, "Development of a laser range finder for 3d map-building in rubble; installation in a rescue robot," in *ICMA 2007. Int. Conference on Mechatronics and Automation*, Aug. 2007, pp. 2054–2059.

[76] D. Klimentjew, M. Arli, and J. Zhang, "3d scene reconstruction based on a moving 2d laser range finder for service-robots," in *2009 IEEE Int. Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, pp. 1129–1134.

[77] K. Arras and R. Siegwart, "Feature extraction and scene interpretation for map-based navigation and map building," in *Proc. of SPIE, Mobile Robotics XII*, 1997, pp. 42–53.

[78] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans," in *Proc. CVPR '94., IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 935–938.

[79] J. Nieto, T. Bailey, and E. Nebot, "Recursive scan-matching slam," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 39–49, 2007, simultaneous Localisation and Map Building.

[80] P. Nunez, R. Vazquez-Martin, J. del Toro, A. Bandera, and F. Sandoval, "Feature extraction from laser scan data based on curvature estimation for mobile robotics," in *Proc. 2006 IEEE Int. Conference on Robotics and Automation, 2006. ICRA 2006.*, May 2006, pp. 1167–1172.

[81] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," *Auton. Robots*, vol. 23, no. 2, pp. 97–111, 2007.

[82] N. Bahari, M. Becker, and H. Firouzi, "Feature based localization in an indoor environment for a mobile robot based on odometry, laser and panoramic vision data," *ABCM Symposium Series in Mathematics*, vol. 3, pp. 266–275, 2008.

[83] K. Arras, "Feature-based Robot Navigation in Known and Unknown Environment," Ph.D. dissertation, Swiss Federal Institure of Technology (EPFL), Luasanne, Switzerland, 2003.

[84] P. Newman, "On the Structure and Solution of the Simultaneous Localization

and Map Building Problem," Ph.D. dissertation, The University of Sydney, Sydney, Australia, 1999.

APPENDIX A

CONSISTENCY OF FREQUENTIST APPROACH

In this section, the consistency of the frequentist mapping algorithm under uncertain robot poses is established. In the current context, consistency implies that the estimated map probabilities converge to the true map probabilities with probability one, or almost surely. The result shall be proved using the powerful ODE method from the stochastic approximation literature [66, 65]. To begin with, a short introduction of the method is presented to clarify the basic idea behind the method.

Consider the mapping algorithm from Eq. 4.31 without the projection operator and using the current estimates of $c_i^*(.)$ and $A_i^*(.)$ from the current map estimates P_t we get,

$$P_{i,t+1} = P_{i,t} + \gamma[c_i(X_t, P_t) - A_i(X_t, P_t)P_{i,t}], \tag{A.1}$$

where $X_t = (b_t, z_t)$, the 2-tuple consisting of the belief state and the observation at any instant. It is easily seen that the state of the algorithm X_t evolves according to a Markov chain, whose transition probabilities, in general, depend on the map probability estimates P_t. If the learning rate parameter γ is small, then the value of the map probabilities does not change quickly, and can be assumed to be essentially equilibrated over N steps, and then

$$P_{i,t+N} \approx P_{i,t} + \gamma N(\frac{1}{N})\sum_{k=1}^{N}[c_i(X_{t+k}, P_t) - A_i(X_{t+k}, P_t)P_{i,t}], \tag{A.2}$$

Then, using the law of large numbers, it follows that if N is large enough:

$$\frac{1}{N}\sum_{k=1}^{N}[c_i(X_{t+k}, P_t) - A_i(X_{t+k}, P_t)P_{i,t}] \approx \bar{h}_i^*(P_t) - \bar{A}_i(P_t)P_{i,t}, \tag{A.3}$$

where $\bar{h}_i^*(.)$ and $\bar{A}_i(.)$ are the averaged values of $c_i(.)$ and $A_i(.)$ respectively. Then, it follows that

$$P_{i,t+N} - P_{i,t} \approx N\gamma[\bar{h}_i^*(P_t) - \bar{h}_i(P_t)], \tag{A.4}$$

where $\bar{h}_i(P_t) = \bar{A}_i(P_t)P_{i,t}$, which happens to be the forward Euler approximation (with step size $N\gamma$) of the differential equation:

$$\dot{P}_i = \bar{h}_i^*(P) - \bar{h}_i(P). \tag{A.5}$$

The idea behind the method is that the asymptotic performance of the estimation/ mapping algorithm can be analyzed by analyzing the behaviour of the "mean/ average" ODE above. This method is very popular in analyzing the behavior of algorithms in many different fields including reinforcement learning [67], neural networks [69], system identification and stochastic adaptive control [66, 65]. In the following, the frequentist mapping algorithm is analyzed using the ODE method.

Consider the mapping algorithm as presented previously:

$$P_{i,t+1} = \Pi_{\mathcal{P}}\{P_{i,t} + \gamma_t[c_i(X_t, P_t) - A_i(X_t, P_t)P_{i,t}]\}, \tag{A.6}$$

where $X_t = (b_t, z_t)$ is the 2-tuple consisting of the belief state and the observation at any instant. The mean true observation probabilities of the i^{th} map component and the mean "current" predicted value are defined as:

$$h_i^*(b, p) \equiv E_z^*[c_i(b, z, P)] = \sum_z p^*(z/b)c_i(b, z, P), \tag{A.7}$$

$$h_i(b, P) \equiv E_z[c_i(b, z, P)] = A_i(b, z, P)P_i = \sum_z p(z/b)c_i(b, z, P), \tag{A.8}$$

where $p^*(z/b)$ is the probability of an observation z given the true map probabilities P^*, and $p(z/b)$ are the probabilities given the estimate of the map probabilities P,

and given that the belief state is b. Recall that $c_i(b, z, P)$ is the vector containing the observation probabilities of the i^{th} map component in its various different states, given that the belief on the robot pose is b, the reading from the sensor is z and the estimate of the map probabilities is P. Note that $c_i(b, z, P)$ is the approximation of the true observation probability vector $c_i^*(b, z, P)$ where the vector of true map probabilities, P^*, is replaced by the approximate map probabilities P. If the map probabilities were truly P, then the average of $C_i(b, z, P)$ over all observations z would result in $h_i(b, P)$. However, since the observation z is generated by the true map probabilities P^*, not P, the quantity $h_i^*(b, P)$ is in general different from $h_i(b, P)$. In fact, only at $P = P^*$ are the two quantities equal and the algorithm uses this fact to guide the map estimates towards the true map probabilities. It can be seen that

$$p^*(z/b) = \sum_{s,q_1,q_2,\cdots,q_M} p(z/s, q_1, \cdots q_M)p^*(q_1)\cdots p^*(q_M)b(s), \tag{A.9}$$

$$p(z/b) = \sum_{s,q_1,q_2,\cdots,q_M} p(z/s, q_1, \cdots q_M)p(q_1)\cdots p(q_M)b(s). \tag{A.10}$$

For the sake of simplicity, the rest of the development is done for the two dimensional case, i.e., each map component q_i can take two values. The occupancy grid is a binary variable where each grid could either be occupied or empty. The extension to higher dimension is fairly straightforward. Let $q_i \in \{O, E\}$, O for the grid being occupied and E for the grid being empty.

First, the following assumption is made.

Assumption A.1. *Corresponding to every map probability P, let the belief process b_t have a stationary distribution $\pi_\infty(b, P)$. Moreover, let the belief process be geometrically ergodic. Let*

$$\bar{h}_i^*(P) = \int_{b\in B_i} h_i^*(b, P)\frac{d\pi_\infty(b, P)}{\pi_\infty(B_i, P)}, \bar{h}_i(P) = \int_{b\in B_i} h_i(b, P)\frac{d\pi_\infty(b, P)}{\pi_\infty(B_i, P)}, \tag{A.11}$$

where B_i are all the belief states that map component i is observed from. In particular, the above assumption implies that there exist $K < \infty$, $\rho < 1$ such that

$$E[C_i(b_t, z_t, P) - \bar{h}_i^*(P)] \leq K\rho^t, E[A_i(b_t, P)P_i - \bar{h}_i(P)] \leq K\rho^t, \tag{A.12}$$

i.e., the quantities $C_i(.)$ and $A_i(.)$ converge to their average values exponentially fast.

Define $\bar{H}^*(P) = [\bar{h}_1^*(P), \cdots \bar{h}_M^*(P)]^t$, and $\bar{H}(P) = [\bar{h}_1(P), \cdots \bar{h}_M(P)]^t$. Then, under assumption A.1, it can be shown that the asymptotic behavior of the mapping algorithm is characterized by the solution of the following ODE ([66], pg. 187, Ch. 6, Theorem 6.1):

$$\dot{P} = \bar{H}^*(P) - \bar{H}(P). \tag{A.13}$$

In particular, the following result holds.

Proposition A.2. *Let the point $P = P^*$ be an asymptotically stable equilibrium of the ODE (A.13) with domain of attraction D^*. Let $C \subseteq D^*$ be some compact subset of D^*. Let the learning rate parameters $\{\gamma_t\}$ be such that $\sum_t \gamma_t = \infty$, and $\gamma_t \to 0$ as $t \to \infty$. If the trajectories of the mapping algorithm (A.1) enter the subset C infinitely often, the estimates $P_t \to P^*$ almost surely.*

Hence, it is left to be shown that the set of true map probabilities $P^* = [P_1^*, ..., P_M^*]$ is an asymptotically stable equilibrium of ODE (A.13). In order to show this, it would be shown that the linearization of the mean ODE (A.13) about P^* is asymptotically stable and hence, so is the nonlinear ODE ([70], Chapter 3, Theorem 3.7). Current treatment is limited to the case when $D = 2$, i.e., the map components can take one of two values. The gradient of the vector field $\bar{H}^*(P) - \bar{H}(P)$ is defined by the matrix:

$$\nabla(\bar{H}^*(P) - \bar{H}(P)) = [\partial_i(\bar{h}_j^*(P) - \bar{h}_j(P))]. \tag{A.14}$$

The following assumption is made.

Assumption A.3. *It is assumed that*

$$\partial_i(\bar{h}_i^*(P) - \bar{h}_i(P)) < -\epsilon, \forall\, i, \tag{A.15}$$

$$|\sum_{j\neq i}[\partial_j(\bar{h}_i^*(P) - \bar{h}_i(P))]| \leq |\partial_i(\bar{h}_i^*(P) - \bar{h}_i(P))|, \tag{A.16}$$

where all the partial derivatives above are evaluated at $P = P^*$, and $\epsilon > 0$ is a positive constant.

The justification of the above assumptions are provided after the following proposition.

Proposition A.4. *Let assumption A.3 be satisfied. Then the true map probability vector P^* is an asymptotically stable equilibrium of ODE A.13 with a non-empty region of attraction D^*. Thus, if the mapping algorithm estimates P_t visit a compact subset $C \subseteq D^*$ infinitely often, then $P_t \to P^*$ almost surely, due to Proposition A.2.*

Proof. Under assumption A.3, the linearization of the mean ODE (A.13) about P^* is row-dominant, and hence, all its eigen values lie in the open left half plane and their real parts are bounded at least ϵ away from the imaginary axis [71]. Therefore, it follows that P^* is an asymptotically stable equilibrium point of the mean ODE and hence, all the other results follow. □

It is left to furnish the justification for assumption A.3.

In order to do this, note that

$$\partial_j(\bar{h}_i^*(P) - \bar{h}_i(P)) = \int_{b\in B_i} \partial_j(h_i^*(b,P) - h_i(b,P))\frac{d\pi_\infty(b,P)}{\pi_\infty(B_i,P)}. \tag{A.17}$$

Consider the term $\partial_j(h_i^*(b,P) - h_i(b,P))$ for some belief state b. In the following to simplify notation, all the partial derivatives are assumed to be evaluated at $P = P^*$.

It may be shown that:

$$\partial_j(h_i^*(b,P) - h_i(b,P)) = -\sum_z \partial_j p(z/b) c_i(b,z,P), \tag{A.18}$$

where the partial derivative above is evaluated at $P = P^*$, and that:

$$\partial_j p(z/b) = p^*(z/b, q_j = O) - p^*(z/b, q_j = E), \tag{A.19}$$

i.e., the difference in the probabilities of observing z given belief state b, and whether q_j is in state E or state O. Then, it follows that

$$\partial_j(h_i^*(b,P) - h_i(b,P)) = -\sum_z (p^*(z/b, q_j = O) - p^*(z/b, q_j = 0)) p^*(\hat{q}_i = O/z, b), \tag{A.20}$$

where recall that the variable $p^*(\hat{q}_i = O/z, b) = c_i^*(b,z)$ is the "true" probability that the observation of the i^{th} map component is O given the belief state b and the observation z (see Section 2.2). Hence, it follows that

$$\begin{aligned}\partial_i(h_i^*(b,P) - h_i(b,P)) &= -\sum_z [p^*(z/b, q_i = O) - p^*(z/b, q_i = E)] p^*(\hat{q}_i = O/z, b), \\ &= -[p^*(\hat{q}_i = O/q_i = O, b) - p^*(\hat{q}_i = O/q_i = E, b)].\end{aligned} \tag{A.21}$$

Hence, $\partial_i(h_i^*(b,P) - h_i(b,P)) < -\epsilon$ if $p^*(\hat{q}_i = O/q_i = O, b) - p^*(\hat{q}_i = O/q_i = E, b) > \epsilon$. The above equation thus implies that the probability of observing the map component occupied when it is actually occupied should be more than the probability of seeing it occupied when it is actually unoccupied (i.e., a spurious observation due to some other map component). This in turn is a "good sensor" assumption, i.e, the right observation is made more number of times than the wrong one. This corresponds to the heuristic that observations that are too far from the current belief state are discarded. Thus, the set B_i in Eq. A.17 above should consist of only those belief states from which the observation of map component i can be reliable. Ensuring that

the set B_i is chosen in the above fashion implies that $\partial_i(h_i^*(b,P) - h_i(b,P)) < -\epsilon$ for all $b \in B_i$, and hence, it follows that Eq. (A.15) is automatically satisfied.

Recall the definitions of $h_i(b,P)$ and $h_i^*(b,P)$ (cf. Eqs. (A.7)-(A.8)). The difference between the two signifies the average observation prediction error of the i^{th} map component given that the map probability estimates are P. Recall that it is zero for $P = P^*$. Thus, $\partial_j(h_i^*(b,P) - h_i(b,P))$ represents the sensitivity of this error to the j^{th} component of the map probabilities. Now, under the assumption that

$$|\sum_{j \neq i}[\partial_j(h_i^*(b,P) - h_i(b,P))]| \leq |\partial_i(h_i^*(b,P) - h_i(b,P))|, \tag{A.22}$$

for all $b \in B_i$, then Eq. A.16 in assumption A.3 is automatically satisfied. The equation above implies that the sensitivity of the observation error of the i^{th} map component to its own map probabilities should dominate the cumulative sensitivity of the error to all other map components. This is a structural assumption that is required regardless of whether the robot pose is uncertain or perfectly known. In fact, it may be reasonably expected that it is satisfied if the map components are updated only from "good" belief states, i.e., belief states from which the sensors can be assumed to be reliable. Experimental evidence seems to suggest the same as well. The development above can then be summarized in the following proposition.

Proposition A.5. *Given any map component q_i let there exist a set of belief states G_i s.t. for all beliefs $b \in G_i$, the following hold:*

$$p^*(\hat{q}_i = O / q_i = O, b) - p^*(\hat{q}_i = O / q_i = E, b) > \epsilon,$$
$$|\partial_i(h_i^*(b,P) - h_i(b,P))| \geq |\sum_{j \neq i} \partial_j(h_i^*(b,P) - h_i(b,P))|,$$

where all the partial derivatives above are evaluated at $P = P^*$. If the sets B_i are chosen such that $B_i \subseteq G_i$, then assumption A.3 is automatically satisfied and hence,

Proposition holds.

This completes the proof of the consistency of the mapping algorithm given an uncertain time-varying belief state. In practice, the sets B_i have to be chosen heuristically. In most cases it should be easy to choose the set B_i given the sensor characteristics. Throughout the implementations in the current work, all range readings were discarded which were close to or more than the maximum range specified in the sensor characteristics.

APPENDIX B

OBSERVATION MODEL FOR LASER RANGE FINDER

In this section, specific observation models are developed for the case when a 2D OG map is constructed from the observations made by a mobile laser range sensor whose position is defined by the belief $b(s)$. Among different types of sensors, 2D laser range finders have become increasingly popular in mobile robotics. For instance, laser scanners have been used in localization [72, 73], feature based SLAM [74, 23] and building 3D maps [75, 76]. The laser sensor is a one dimensional range sensor which gives range readings at specified bearing angles. All readings from a laser range finder are assumed to be in the horizontal plane. It is assumed that there is no error in the bearing angle. This is a standard assumption for the laser range sensor as the bearing angle has very low random error and makes the formulation of the observation models much simpler [77]. A range reading is interpreted as the nearest point in the direction of the laser beam that is occupied. The range reading is assumed to be corrupted by an additive zero mean Gaussian noise with variance σ. Thus the range reading, z, is the sum of the actual range z^* and a zero mean Gaussian random noise whose variance is σ and can be represented as,

$$z \sim N(z^*, \sigma).$$

Suppose that the sensor is at point s. Let the true reading that the laser should make, z^*, correspond to the i^{th} grid i.e., $(q_i = O)$. Let the actual reading z correspond to some grid, q_k, lying in the path of the laser beam. Then the sensor model in the discretized sense is given as the probability that the observation $\hat{q}_k = O$ is made when grid q_i is occupied given that the sensor is in pose s.

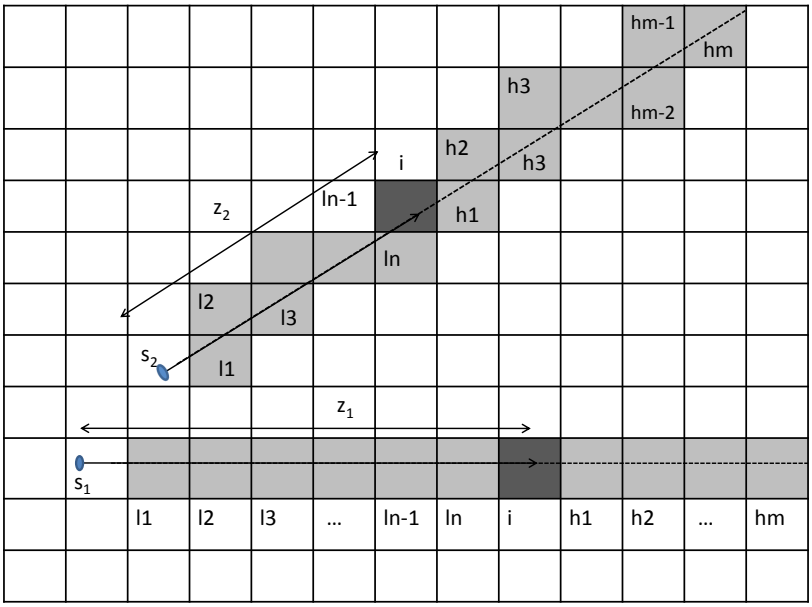


Fig. 16. Observation model for laser range finder

$$p(\hat{q}_k = O/s, q_i = O) = \int_{q_k} p(z/s, q_i = O)dz = \int_{q_k} \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(z - z^*)^2}{2\sigma^2})dz, \quad \text{(B.1)}$$

where $p(z)$ is the probability density function for normal distribution function with mean z^* and variance σ. The integration is done over the interval for which the range reading z corresponds to the grid q_k. It should be noted that the sensor can get a reading from q_i only when all the grids between s and q_i are empty and this assumption is made implicitly when representing the sensor model in the above form. While making observations, the implicit assumption that all the grids between the vantage point and the observed grid is empty is not true. We need $p(\hat{q}_k = O/s, q_k = O)$ and $p(\hat{q}_k = O/s, q_k = E)$ independent of this assumption to construct $A_k(s)$ by Eq. 4.8. We can drop the implicit assumption and calculate the required probabilities as follows.

Suppose that the sensor is in pose s and makes a range observation z at some

angle θ relative to the local robot frame. Suppose the reading z corresponds to the i^{th} grid in the map, i.e, $\hat{q}_i = O$. Also a ray, representing the laser beam, is made originating from s in the direction θ and intersecting grids are labeled as follows. The grids between s and q_i, starting from s are called $\{q_{l1}, \ldots, q_{ln}\}$ and the grids beyond q_k are labeled $\{q_{h1}, \ldots, q_{hm}\}$, as shown in Fig. B. Using Eq. 4.5

$$p(\hat{q}_i/s, q_i) = \sum_{q_1 \cdots q_{i-1} q_{i+1} \cdots q_M} p(\hat{q}_i/s, q_1 \cdots q_M) p^*(q_1) \cdots p^*(q_{i-1}) p^*(q_{i+1}) \cdots p^*(q_M). \tag{B.2}$$

For the occupancy grid case and the sensor being used is a laser range finder, the above reduces to the following form.

$$\begin{aligned} p(\hat{q}_i = O/s, q_i = O) &= p(\hat{q}_i = O/s, q_{l1} = O) p(q_{l1} = O) \\ &+ p(\hat{q}_i = O/s, q_{l1} = E, q_{l2} = O) p(q_{l2} = O) p(q_{l1} = E) + \ldots \\ &+ p(\hat{q}_i = O/s, q_{l1} = E, \ldots, q_{l(n-1)} = E, q_{ln} = O) p(q_{ln} = O) \prod_{i=1}^{n-1} p(q_{li} = E) \\ &+ p(\hat{q}_i = O/s, q_{l1} = E, \ldots, q_{ln} = E, q_i = O) \prod_{i=1}^{n} p(q_{li} = E), \end{aligned} \tag{B.3}$$

$$
\begin{aligned}
p(\hat{q}_i = O/s, q_i = E) &= p(\hat{q}_i = O/s, q_{l1} = O)p(q_{l1} = O) \\
&+ p(\hat{q}_i = O/s, q_{l1} = E, q_{l2} = O)p(q_{l2} = O)p(q_{l1} = E) + \ldots \\
&+ p(\hat{q}_i = O/s, q_{l1} = E, \ldots, q_{l(n-1)} = E, q_{ln} = O)p(q_{ln} = O)\prod_{i=1}^{n-1} p(q_{li} = E) \\
&+ p(\hat{q}_i = O/s, q_{l1} = E, \ldots, q_{ln} = E, q_{h1} = O)p(q_{h1} = O)\prod_{i=1}^{n} p(q_{li} = E) \\
&+ p(\hat{q}_i = O/s, q_{l1} = E, \ldots, q_{ln} = E, \hat{q}_i = E, q_{h1} = E, q_{h2} = O) \\
&\quad p(q_{h2} = O)p(q_{h1} = E)\prod_{i=1}^{n} p(q_{li} = E) \;+\; \ldots \\
&+ p(\hat{q}_i = O/s, q_{l1} = E, \ldots, q_{ln} = E, \hat{q}_i = E, q_{h1} = E, \ldots, q_{hm} = O) \\
&\quad p(q_{hm} = O)\prod_{i=1}^{m-1} p(q_{hj} = E)\prod_{i=1}^{n} p(q_{li} = E). \qquad \text{(B.4)}
\end{aligned}
$$

Now $A_i(s)$ can be constructed as:

$$
A_i(s) = \begin{bmatrix} p(\hat{q}_i = O/s, q_i = O) & p(\hat{q}_i = O/s, q_i = E) \\ p(\hat{q}_i = E/s, q_i = O) & p(\hat{q}_i = E/s, q_i = E) \end{bmatrix} \qquad \text{(B.5)}
$$

$$
= \begin{bmatrix} p(\hat{q}_i = O/s, q_i = O) & p(\hat{q}_i = O/s, q_i = E) \\ 1 - p(\hat{q}_i = O/s, q_i = O) & 1 - p(\hat{q}_i = O/s, q_i = E). \end{bmatrix} \qquad \text{(B.6)}
$$

Note that

$$
p(\hat{q}_i = E/s, q_i = O) = 1 - p(\hat{q}_i = O/s, q_i = O),
$$

and

$$
p(\hat{q}_i = E/s, q_i = E) = 1 - p(\hat{q}_i = O/s, q_i = E)
$$

Using Eq. 4.4 $p(z/s)$ can be constructed as

$$
\begin{aligned}
p(z/s) &= \sum_{q_i} p(\hat{q}_i/s, q_i)p(q_i) \\
&= p(\hat{q}_i = O/s, q_i = O)p(q_i = O) + p(\hat{q}_i = O/s, q_i = E)p(q_i = E). \qquad \text{(B.7)}
\end{aligned}
$$

Recall Eq. 4.31 which is used to estimate the map probabilities in the case of pose uncertainty. Let the belief of the current pose, $b(s)$, and current observation z be given. Then $c_i(b, z)$ can be computed using Eq. 4.27

$$\begin{aligned} c_i(b, z) &= \sum_s \frac{1(\hat{q}_i/s, z)p(z/s)b(s)}{p(z/b)} \\ &= \eta \sum_s 1(\hat{q}_i/s, z)p(z/s)b(s) \end{aligned} \tag{B.8}$$

where η is a normalizing factor to ensure that $c_i(b, z)$ is a probability vector. And, $A_i(b)$ can be computed as in Eq. 4.30

$$A_i(b) = \sum_s A_i(s)b(s).$$

Thus, the application of the frequentist mapping under pose uncertainty in a 2D occupancy grid map observed by a laser range finder is provided. This is exactly how the algorithm is implemented as a part of the Hybrid SLAM solution.

APPENDIX C

FEATURE EXTRACTION

A primary aspect of any localization method is to accurately match current observations to the prior map or information to estimate the pose of the robot. One way of doing this is using scan matching based approaches, where the alignment of two consecutive scans is approximated by minimizing some metric based on the entire set of scanned points [72, 78, 79]. This kind of method is not suitable for use with EKF based localization as the state space would be too large, and also, the data association would be very complicated. Instead of working directly with raw measurements, EKF based localization methods first extract geometric features which are added to the EKF state vector and estimated along with the pose of the robot. Feature extraction has been studied widely [77, 80, 81] and applied to localization [82] and SLAM [22, 83, 84]. Feature representation is much more compact than the raw measurement data and thus uses much less memory while still providing rich and accurate information. Also, as agglomeration of points is used to approximate a feature, the random noise tends to average out more effectively making the data association relatively simpler. For 2D laser range finders a variety of feature extraction methods have been proposed and a good comparative study can be found in [81].

The feature extraction method used in this proposal is based on the work of Arras et al. [77, 83]. The feature extraction process followed here is a two step process. First, the segmentation is done by defining a metric on the model fidelity of adjacent groups of measurements from the laser range finder. A segment matching step follows where segments belonging to the same geometrical structure in the environment are joined together based on the Mahalanobis distance matrix. Straight line segments are

extracted by this method in the generalized least squares sense using polar coordinates along with the covariance estimate of the line parameters as follows.

$$tan2\alpha = \frac{2\sum\sum_{i<j} w_i w_j \rho_i \rho_j sin(\theta_i + \theta_j) + \sum(w_i - \sum w_j) w_i \rho^2 sin2\theta_i}{2\sum\sum_{i<j} w_i w_j \rho_i \rho_j cos(\theta_i + \theta_j) + \sum(w_i - \sum w_j) w_i \rho^2 cos2\theta_i}, \tag{C.1}$$

$$r = \frac{w_i \rho_i cos(\theta_i - \alpha)}{\sum w_i}, \tag{C.2}$$

where (ρ_i, θ_i) are the i^{th} measurement and w_i is the associated weight for the reading, and (α, r) are the parameters for the line model

$$\rho cos(\theta - \alpha) - r = 0. \tag{C.3}$$

The covariance matrix C can be computed as

$$\begin{aligned} \sigma_{\alpha\alpha} &= \sum \Big(\frac{\partial \alpha}{\partial \rho}\Big)^2 \sigma_{\rho_i}^2 \\ \sigma_{rr} &= \sum \Big(\frac{\partial r}{\partial \rho}\Big)^2 \sigma_{\rho_i}^2 \\ \sigma_{\alpha r} &= \sum \frac{\partial \alpha}{\partial \rho} \frac{\partial r}{\partial \rho} \sigma_{\rho_i}^2 \end{aligned} \tag{C.4}$$

where,

$$\begin{aligned} \frac{\partial \alpha}{\partial \rho} &= \frac{1}{D^2 + N^2} \sum w_i^2 [N(\bar{x} cos(\theta_i) - \bar{y} sin\theta_i - \rho_i cos2\theta_i) - D(sin(\theta_i) + \bar{y} cos\theta_i - \rho_i sin2\theta_i)], \\ \frac{\partial r}{\partial \rho} &= \sum \Big[\frac{w_i}{\sum w_i} cos(\theta_i - \alpha) + \frac{\partial \alpha}{\partial \rho}(\bar{y} cos\alpha - \bar{x} sin\alpha)\Big], \end{aligned} \tag{C.5}$$

$\bar{x} = 1/\sum w_i \sum w_i \rho_i cos\theta_i$, $\bar{y} = 1/\sum w_i \sum w_i \rho_i sin\theta_i$, N and D are the numerator and denominator of the right hand side of Eq. C.1. Note that angular uncertainty is neglected for laser range finders.

Segmentation

The regression line model is fitted in each n_f neighboring points and the covariance matrix is calculated obtaining the same number of points in the model space. When

adjacent groups of range readings lie on the same landmark, the associated groups constitute a cluster in the model space corresponding to the landmark. Feature extraction is now the task of finding these clusters. The fact that points on the same landmark are almost always consecutive points is exploited by defining a distance measure, d_i, in the model space for n_m adjacent points.

$$d_i = \sum_j (x_j - x_w)'(C_j + C_w)^{-1}(x_j - x_w) \tag{C.6}$$

$$x_w = C_w \sum C_j^{-1} x_j, \tag{C.7}$$

$$C_w^{-1} = \sum C_j^{-1}. \tag{C.8}$$

where $j = i - (n_m - 1)/2, \ldots, i + (n_m)/2$ and x_w is the weighted mean. Low distances indicates that the points involved have high model fidelity. A threshold d_m is applied cutting off the regions of low distance. A segment is now defined as the set of measurement points whose representations in the model space satisfy the condition

$$d_i = d_m \tag{C.9}$$

Segment merging

The segment matching problem is to find and merge all segments that belong to the same geometrical structure while avoiding false associations. The Mahalanobis distance is widely used for this purpose. The Mahalanobis distance between each pair of clusters is calculated as

$$d_{ij}^2 = (x_i - x_j)'(C_i + C_j)^{-1}(x_i - x_j) \tag{C.10}$$

If x_i and x_j belong to the same landmark then d_{ij}^2 has a chi-squared distribution and an appropriate threshold $\mathcal{X}_{\alpha,2}^2$ is chosen below which the pair is merged to become one segment.

Thus line segments are extracted from laser range data whose parameters are included in the EKF state vector and are recursively estimated. Point features are extracted by finding the intersecting point of nearby line segments that represent a geometrical corner in the environment.

VITA

Roshmik Saha was born in Ranchi, India. After graduating from Delhi Public School in Ranchi, Roshmik went on to attended Inidan Institute of Technology Kharagpur, where he earned a Bachelor of Technology in mining engineering. He joined the aerospace engineering department of Texas A&M University in summer 2006 to pursue his graduate studies under the supervision of Dr. Suman Chakravorty. He received his Ph.D. in aerospace engineering in August 2011. He has broad interest in robotics, stochastic methods and embedded software development. He can be reached via email at roshmik@gmail.com or by contacting Dr. Suman Chakravorty at Department of Aerospace Engineering, 3141 TAMU, College Station, TX 77843.

The typist for this thesis was Roshmik Saha.