Sparse Bayesian Information Filters for Localization and Mapping

by

Matthew R. Walter

B.S., University of Illinois at Urbana-Champaign, 2000

Submitted to the Joint Program in Applied Ocean Science & Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

February 2008

Author . . . .
Joint Program in Applied Ocean Science & Engineering
Massachusetts Institute of Technology
and Woods Hole Oceanographic Institution
February 6, 2008

Certified by . . . .
John J. Leonard
Professor of Mechanical and Ocean Engineering
Massachusetts Institute of Technology
Thesis Supervisor

Accepted by . . . .
Henrik Schmidt
Chairman, Joint Committee for Applied Ocean Science & Engineering
Massachusetts Institute of Technology/
Woods Hole Oceanographic Institution

Sparse Bayesian Information Filters for Localization and Mapping

by

Matthew R. Walter

Submitted to the Joint Program in Applied Ocean Science & Engineering
on February 6, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis formulates an estimation framework for Simultaneous Localization and Mapping (SLAM) that addresses the problem of scalability in large environments. We describe an estimation-theoretic algorithm that achieves significant gains in computational efficiency while maintaining consistent estimates for the vehicle pose and the map of the environment.

We specifically address the feature-based SLAM problem in which the robot represents the environment as a collection of landmarks. The thesis takes a Bayesian approach whereby we maintain a joint posterior over the vehicle pose and feature states, conditioned upon measurement data. We model the distribution as Gaussian and parametrize the posterior in the canonical form, in terms of the information (inverse covariance) matrix. When sparse, this representation is amenable to computationally efficient Bayesian SLAM filtering. However, while a large majority of the elements within the normalized information matrix are very small in magnitude, it is fully populated nonetheless. Recent feature-based SLAM filters achieve the scalability benefits of a sparse parametrization by explicitly pruning these weak links in an effort to enforce sparsity. We analyze one such algorithm, the Sparse Extended Information Filter (SEIF), which has laid much of the groundwork concerning the computational benefits of the sparse canonical form. The thesis performs a detailed analysis of the process by which the SEIF approximates the sparsity of the information matrix and reveals key insights into the consequences of different sparsification strategies. We demonstrate that the SEIF yields a sparse approximation to the posterior that is inconsistent, suffering from exaggerated confidence estimates. This overconfidence has detrimental effects on important aspects of the SLAM process and affects the higher level goal of producing accurate maps for subsequent localization and path planning.

This thesis proposes an alternative scalable filter that maintains sparsity while preserving the consistency of the distribution. We leverage insights into the natural structure of the feature-based canonical parametrization and derive a method that actively maintains an exactly sparse posterior. Our algorithm exploits the structure of the parametrization to achieve gains in efficiency, with a computational cost that scales linearly with the size of the map. Unlike similar techniques that sacrifice consistency for improved scalability, our algorithm performs inference over a posterior that is conservative relative to the nominal Gaussian distribution. Consequently, we preserve the consistency of the pose and map estimates and avoid the effects of an overconfident posterior.

We demonstrate our filter alongside the SEIF and the standard EKF both in simulation as well as on two real-world datasets. While we maintain the computational advantages of an exactly sparse representation, the results show convincingly that our method yields conservative estimates for the robot pose and map that are nearly identical to those of the original Gaussian distribution as produced by the EKF, but at much less computational expense.

The thesis concludes with an extension of our SLAM filter to a complex underwater environment. We describe a systems-level framework for localization and mapping relative to a ship hull with an Autonomous Underwater Vehicle (AUV) equipped with a forward-looking sonar. The approach utilizes our filter to fuse measurements of vehicle attitude and motion from onboard sensors with data from sonar images of the hull. We employ the system to perform three-dimensional, 6-DOF SLAM on a ship hull.

Thesis Supervisor: John J. Leonard
Title: Professor of Mechanical and Ocean Engineering
Massachusetts Institute of Technology

Acknowledgments

There are many people that I would like to thank for their time, patience, and support over the last several years. I've had a wonderful time at MIT, and I credit this fortune to my family, friends, and colleagues.

I would first like to thank my advisor, John Leonard, who has been tremendously supportive of me during my time at MIT. Our discussions have played an integral part in developing my thesis work and, more generally, in discovering my research interests. John has provided me with many unique opportunities, both academic and otherwise. For example, I can't imagine that I'd ever have had the chance to go on a research cruise based out of a prison island off of Italy. From day one, he has made it a priority to ensure that I was enjoying my work at MIT, and, for that, I am grateful.

In addition to John, there are several other faculty and staff, both at MIT and WHOI, to whom I am thankful. In particular, Nick Roy has been generous enough to let me participate in his group meetings for the past few years. I also had the opportunity of working with Nick as a teaching assistant for the Principles of Autonomy and Decision Making course during the Fall of 2006. It was a wonderful experience, largely due to the students in the class, who made all the time and effort enjoyable. Additionally, Mark Grosenbaugh has gone out of his way to be supportive of me over the last couple years. I would also like to thank my committee members, Seth and Hanu, for their useful discussions in writing the thesis. Bryt Bradley, Julia Westwater, and Marsha Gomes, also deserve thanks for their support.

I am also grateful to my fellow lab-mates, many of whom have become close friends. I've really enjoyed hanging out with them both in and outside of the lab (ok, more outside of the lab). Among them is Alex, who has been my closest friend at MIT since he came here several years ago. Between working on the Unicorn and Caribou AUVs, going to sea on two research cruises, and going to 1369 for coffee in the morning, I've been very fortunate to get to know Alex and consider him one of my best friends. Additionally, I'd like to thank Albert, Alec, Iuliu, Olivier, and Tom, who have helped to make my time in Cambridge much more enjoyable. They are great colleagues and, more importantly, good friends. More recently, I have had the chance to become friends with Sam, RJ, and Valerie, and I have enjoyed spending time with them in lab and out. Additionally, I am thankful to Ed, Emma, Christian, Jacob, and Kristian for the many interesting discussions that we have had. It was great to work with David in preparing and operating the AUVs for the two cruises off of Italy. I learned a lot from him during the months that we spent preparing the vehicles in our ever-relocating lab and at the sailing pavilion. Additionally, Mike Kokko deserves credit for all his help with the HAUV experiments. Last, but not least, I was fortunate to work closely with Ryan, who was a few years ahead of me in the Joint Program. Much of the work in the thesis would not have been possible without our extensive discussions and collaboration.

Most importantly, I owe an unlimited amount of gratitude to my family, who has given me a tremendous amount of support over all these years. I am truly lucky to have had such a close relationship with my grandparents, aunts, uncles, and cousins all of my life. I attribute this largely to my grandparents, who have been role models

for all of us, teaching us how important our family really is. During my many years in Cambridge, my aunt and uncle, Marisa and Bob, and their family have been extremely generous, between letting me live with them down in Sandwich for many months, and having me stay any weekend that I wanted to come down. I don't know what I would have down without them. I am also grateful to my Nonna and Grandpa and Aunt Loretta, who have shown nothing but love and support in an uncountable number of ways. Finally, there are my parents, brother, and sisters. My dad has always stressed the importance of education and of working hard and I have always looked up to him for that. My mom's values are unwavering and she is a role model for the type of person that I strive to be. I will forever cherish growing up with my sisters, Mia and Sarah, whether building forts in our woods, putting on plays in our foyer, or going out for a beer now that Sarah is of-age. Finally, there is my brother, John, who makes me laugh more than anyone else. John is my best friend, and I admire him in many ways.

To my grandfathers

Contents

Appendices

List of Figures

List of Tables

List of Acronyms

AUV	autonomous underwater vehicle.
COM	center of mass.
DIDSON	Dual Frequency Identification Sonar.
DOF	degrees of freedom.
DVL	Doppler Velocity Log.
EIF	Extended Information Filter.
EKF	Extended Kalman Filter.
ESDSF	Exactly Sparse Delayed-State Filter.
ESEIF	Exactly Sparse Extended Information Filter.
FOV	field-of-view.
GMRF	Gaussian Markov random field.
GPS	global positioning system.
HAUV	Hovering Autonomous Underwater Vehicle.
IMU	inertial measurement unit.
KF	Kalman Filter.
LBL	long baseline.
LG	linear Gaussian.
MAP	maximum a posteriori.
MIT	Massachusetts Institute of Technology.
ML	maximum likelihood.
MRF	Markov random field.

NEES	normalized estimation error squared.
RANSAC	random sample consensus.
SEIF	Sparse Extended Information Filter.
SFM	structure from motion.
SIR	sampling importance resampling.
SLAM	Simultaneous Localization and Mapping.
TJTF	Thin Junction Tree Filter.
WHOI	Woods Hole Oceanographic Institution.

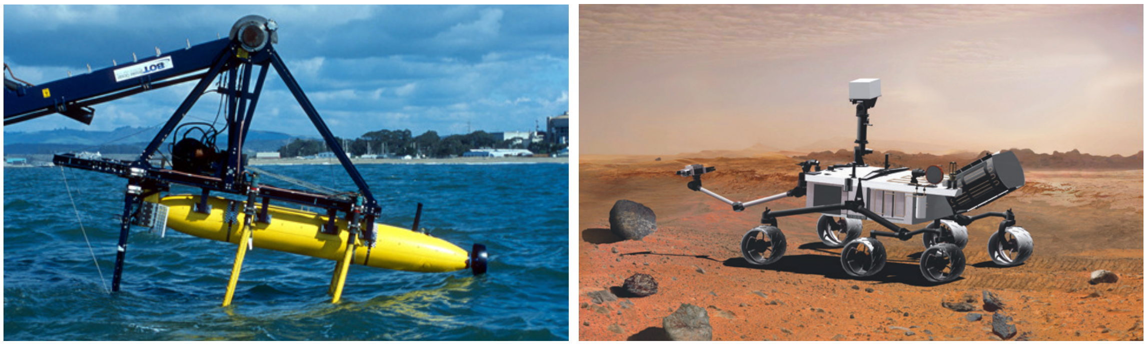
Chapter 1

Introduction

Scientific advancements are both benefactors of improvements in robotic technology as well as driving forces toward the development of more capable autonomous platforms. This is particularly evident in the fields of interplanetary science and subsea exploration. Recent discoveries in areas such as climate change, hydrothermal vent biology, and deep space exploration are driving scientists to study environments that are less and less accessible to humans, be they 5,000 meters below the ocean surface or on the surface of Mars. Traditionally, remotely operated vehicles have proven to be a viable substitute, allowing people to interact with the environment from a surface ship or a lab on land. Many open problems, such as those in ocean circulation modeling [20, 1] and planetary exploration [100], though, require a long-term presence that makes it difficult, if not impossible to have a human constantly in the loop.

An integral component to oceanographic modeling and prediction is the persistent observation of coastal and near-surface ocean processes. As part of the Autonomous Ocean Sampling Network (AOSN) [20], for example, a team of scientists are developing models that can predict ocean upwelling and mixing as well as the distribution of algae and other biological organisms within Monterey Bay. In developing these models, researchers rely upon prolonged, continuous observations of various physical oceanographic properties. While traditional sampling tools such as floats, satellites, and moorings provide useful data, they are not capable of the persistent, adaptive monitoring that is necessary to fully observe the different processes. Consequently, the AOSN initiative is working towards the long-term presence of an ocean observation system that will combine traditional monitoring tools with autonomous vehicles. These vehicles, such as the Dorado-class autonomous underwater vehicle (AUV) shown in Figure 1-1(a), must be capable of sustained, intelligent sampling with little or no human intervention.

Similarly, the Mars Exploration Rover (MER) mission is faced with operating a pair of vehicles in an uncertain environment with communication delays on the order of tens of minutes. The MER mission relies on a large team of ground-based personnel to carefully plan the rovers' day-to-day operation, including their scientific activities and low-level motion. This risk-averse strategy has arguably contributed to the rovers' surprising longevity but imposes significant operational costs and reduces the scientific throughput. Recognizing this, the follow-up Mars Science Laboratory



(a) (b)

Figure 1-1: The image in (a) shows the launch of a Dorado-class AUV as part of the Autonomous Ocean Sampling Network project [20]. The artistic rendering in (b) depicts the Mars Science Laboratory vehicle, which is scheduled to launch in 2009. Roughly two meters in length and weighing 800 kilograms, the rover is equipped with an extensive scientific payload that includes a manipulator, several cameras, an X-ray diffraction unit, and a mass spectrometer. The rover will operate with greater autonomy than the MER vehicles, and is expected to traverse upward of 20 kilometers during the first Martian year. The AUV photograph is courtesy of Todd Walsh ©2003 MBARI. The MSL rendering is courtesy NASA/JPL-Caltech.

(MSL) Project [134] will rely less on ground control and more on vehicle autonomy to plan longer paths and collect and analyze geological samples from the Martian surface. An increased level of scientific and mobile autonomy will allow the vehicle, shown in Figure 1-1(b), to cover as much as 20 kilometers over the course of a single Martian year. As mentioned in a recent issue of the journal *Science* devoted to the role of robotics in science, it is this need for robust, increasingly complex autonomous capabilities that is driving the state of the art in robotics [10].

1.1 Key Capabilities for Autonomous Robotics

The Mars Science Laboratory Project and others like it require highly advanced vehicles that are capable of sustained, long-term autonomous operation. This demand is driving the frontier of robotics toward the development of efficient and consistent algorithms that are suited to persistent autonomy in unknown, unstructured environments. In the case of mobile robotics, three fundamental capabilities serve as the critical building blocks for autonomous behavior [85]: mapping, localization, and path planning. Put broadly, *mapping* refers to the robot's ability to model its environment. Depending on the application, this may be a coarse model that consists of a set of key locations within the world, or it may be a highly detailed representation of the environment. *Path planning* encompasses both the problem of choosing the best route to take as well as the search for immediate, low-level control actions. In order to perform either of these two tasks, the robot must know where it is in the world, to *localize* itself based upon a combination of position and motion observations.

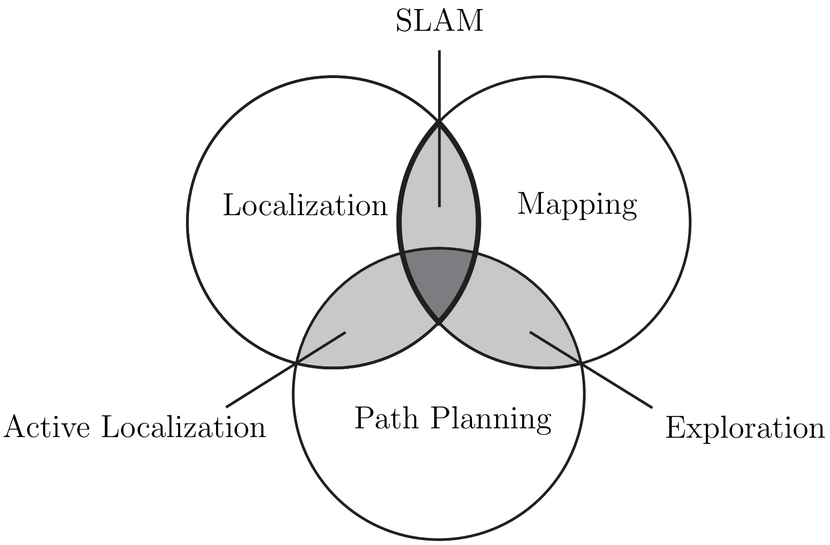


Figure 1-2: Localization, mapping, and path planning are fundamental components of robot autonomy. The three are closely connected and the interdependencies define key problems in artificial intelligence, including that of Simultaneous Localization and Mapping (SLAM), the focus of this thesis. Adopted from [85, 123].

It is difficult to consider mapping, localization, or planning as separate problems to be addressed independently. As the simple diagram in Figure 1-2 demonstrates, there are important areas of overlap between the three. Operating in an unknown environment, it is difficult to decouple motion planning from mapping. Both short and long-term planning require a model of the robot's surroundings. At the same time, the vehicle must plan suitable paths in order to map undiscovered parts of the environment and improve existing maps. This coupling defines the *exploration* problem. Similarly, in the context of mapping, there is rarely a "black box" solution to the localization problem. There is an inherent coupling between map building and localization. The quality of the map depends on an accurate estimate of the robot's pose, yet localization strategies typically estimate the robot's position based upon a map of the environment. The most successful algorithms for autonomous robotics are developed with the implicit recognition that these three problems are highly dependent.

This thesis is primarily concerned with robotic mapping and localization capabilities and, more specifically, algorithms that tackle these two problems concurrently. Let us then take a closer look at these two problems in particular.

1.1.1 Mapping

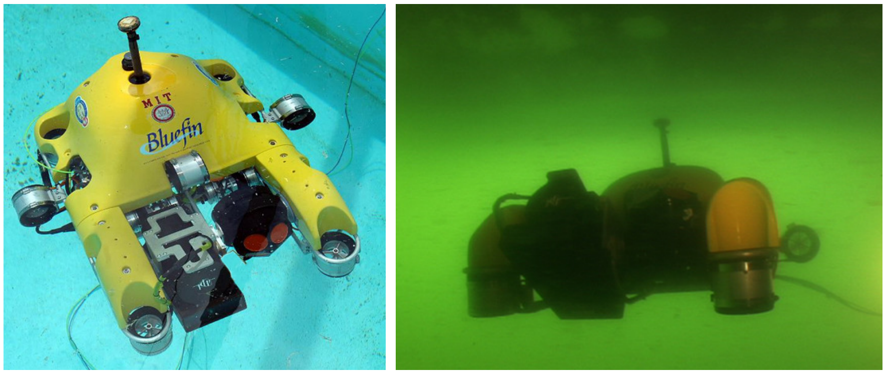
A map of the robot's environment is essential for a number of tasks, not the least of which are path planning and localization. Standard motion planning algorithms rely on representations of the environment in the search for motion plans that bring the robot to a desired goal state [73]. The map provides the coarse structure of the world that serves as the input to the global planner, which solves for an optimal route. In order to follow this global path, the local planner uses the map's finer-scale information, largely to avoid obstacles as the vehicle navigates the global plan.

Robots often operate in environments whose structure is unknown *a priori.* When an initial map does exist, it is often incomplete. In general, the robot is required to build either a coarse (topological) or fine (metric) map by fusing observations of its surroundings with the help of location estimates. Consider, for example, AUVs that are used by the scientific community to detect and locate hydrothermal vents on the ocean floor. Prior to vehicle deployment, the team conducts a survey of the site with a ship-based multibeam sonar, which yields a bathymetric (depth) map of the local seabed. With water depths of several thousand meters, though, the spatial resolution of the maps is, at best, on the order of tens of meters. The maps reveal large-scale geological structure but are too coarse to identify vent locations and are insufficient for planning the fine-scale motion of the AUV [142]. Instead, Yoerger *et al.* [142] describe an adaptive mapping strategy whereby the AUV autonomously plans its survey based upon hydrographic measurement data. More specifically, the AUV first executes a series of coarse, preplanned tracklines that cover the site, using a network of underwater beacons for localization. As the vehicle conducts the initial survey, it generates a map that describes the likelihood of vent sites based upon hydrographic measurement data. The algorithm then identifies promising features that it surveys at a fine scale with a photographic camera and multibeam sonar. The authors have applied this nested exploration strategy to locate and map hydrothermal vents at different sites with the Autonomous Benthic Explorer (ABE) AUV.

1.1.2 Localization

The ability to operate in *a priori* unknown environments plays an integral role in achieving robot autonomy. In the case of the hydrothermal vent surveys, the AUV relies upon an estimate of its pose in order to reference measurement data and build a map of the site. The quality of this map depends directly on the accuracy of the vehicle's estimate of its position relative to the vent field. In addition to mapping, localization plays a critical role in other core aspects of autonomy, including path planning and exploration.

Standard localization methods utilize on-board sensors that observe vehicle velocities and accelerations in conjunction with attitude and heading measurements to integrate position over time [11]. Well-equipped AUVs such as ABE, for example, have access to three-axis linear velocity data along with angular rates and accelerations. Errors in the sensor data, however, give rise to dead-reckoned position estimate errors that grow unbounded with time, typically on the order of one percent of distance traveled [11]. Localization algorithms bound this drift by taking advantage of external infrastructure, such as a global positioning system (GPS), that provides periodic absolute position fixes. Underwater, though, GPS is not available due to the rapid attenuation of electromagnetic signals. ABE and other underwater vehicles estimate their position based upon acoustic time-of-flight measurements to a set of beacons. Known as long baseline (LBL), this navigation strategy mimics the functionality of GPS and yields accurate 3D position data at the cost of setting up the



(a) (b)

Figure 1-3: (a) The Hovering Autonomous Underwater Vehicle (HAUV) [21] is designed to be a highly stable, highly maneuverable platform for the inspection of underwater structures, including ship hulls. The vehicle is equipped with a high resolution imaging sonar that serves as the primary sensor for surveys. (b) The vehicle searching a ship for mine-like targets mounted to the hull.

as-needed beacon infrastructure.

GPS is available in most outdoor settings and requires relatively little infrastructure (i.e. a receiver) on the user end. It suffers from relatively few faults, namely those that result from multipath interference or signal loss due to tree or building coverage. LBL navigation and similar variants, on the other hand, require the deployment and calibration of the acoustic beacon network before they can be used for localization. Secondly, vehicles that are reliant on LBL are confined to operate near the fixed network. These constraints are tolerable with longer-term operations that focus on a specific site, such as the hydrothermal vent survey, but they make rapid, dynamic deployments difficult. Consider the problem of inspecting ship hulls for anomalies as part of regular maintenance or for explosive ordinances in the case of military security. Manual, in-service surveys are often conducted by hand (literally) due to poor visibility and can be time-consuming with ships as large as 70 m in length, as well as hazardous to the divers. As a result, there is an increasing demand for on-site, autonomous inspection with AUVs equipped with a camera or sonar that can quickly map features of interest on the hull. Figure 1-3(b) shows an image of one such vehicle, the Hovering Autonomous Underwater Vehicle (HAUV)[1], during the survey of a barge for mine-like objects. The main goal of autonomous inspection is to conduct quick, thorough surveys of the hull that yield accurate feature maps. While LBL navigation can contribute to the coverage and accuracy goals, the infrastructure constraints prevent rapid, on-site surveys. Furthermore, compared with deep water, near-bottom deployments, LBL performance in shallow water harbors degrades significantly due to interference and multipath induced by surface effects and the ship's hull.

[1] The images depict the successor to the original HAUV prototype, which we refer to as the HAUV1B.

1.1.3 Simultaneous Localization and Mapping

One alternative is to forgo the beacon array and instead rely upon the static structure of the hull or seafloor, in the case of bathymetric surveys, to provide absolute points of reference. Given an initial map of the environment that captures this structure, the robot can take advantage of observations provided by exteroceptive sensors, such as cameras, sonar, or laser range finders, in order to localize itself within the map. Map-based localization is relatively straightforward and is accurate, subject to the quality of the map and the sensors. This approach offers a viable option in some indoor or ground-based environments where architectural plans exist or initial surveys can be performed to generate a suitable map. Oftentimes, though, there is no *a priori* metric map of the environment that is both complete and accurate. Most architectural drawings of offices, for example, describe only the two-dimensional building structure and do not capture "dynamic" features, such as desks and chairs. In comparison, detailed bathymetric maps exist only for a small fraction of the seafloor. Of those that are available, most result from ship-mounted multibeam sonar surveys, which provide the greatest spatial coverage at the expense of resolution. As with the hydrothermal vent deployments, this resolution is far too coarse for map-based localization at most ocean depths.

Simultaneous Localization and Mapping (SLAM) offers a solution to the problem of unencumbered navigation in *a priori* unknown environments. The SLAM formulation to the problem is based upon two coupled ideas: (1) given a model of the world, the robot can accurately estimate its pose by registering sensor data relative to the map, and (2) the robot can build a map of the environment if it has an estimate for its location. Simultaneous Localization and Mapping (SLAM)[2] builds a map of the environment online while concurrently estimating its location based upon the map. More specifically, by building a map online while using inertial and velocity measurement data to predict vehicle motion, the robot utilizes observations of the environment to localize itself within the map and thereby bound error growth. SLAM is a classic chicken-and-egg problem in which an accurate estimate for the robot's pose is necessary to build a map based upon sensor data, while accurate localization depends on the same map to estimate pose. The coupling between mapping and localization is further complicated by uncertainty in the vehicle motion and measurement models, and by noise that corrupts sensor data. Many successful SLAM algorithms address these issues by formulating the problem in a probabilistic manner, tracking the joint distribution over the vehicle pose and map. This approach to the problem offers a principled means of explicitly accounting for the coupling between map building and navigation.

SLAM is an approach to localization and mapping that dates back to the seminal work of Chatila and Laumond [17], along with that of Smith, Self, and Cheeseman [120], among others. Over the last two decades, SLAM has emerged as a central problem in robotics. The community has paid a great deal of attention to the problem, advancing the state of the art in various aspects of SLAM. These include the

[2] Simultaneous Localization and Mapping (SLAM) is also referred to as Concurrent Mapping and Localization (CML) within the literature.

core formulation of the SLAM problem, both with regards to the original Gaussian model of the distribution [76, 14, 107, 28, 52, 129, 113, 45] as well as Monte Carlo representations [98, 29, 93, 54, 50] of the posterior. There have also been significant contributions in the areas of both 2D [125, 16, 82] and 3D [25, 34, 104] world modeling, which extend SLAM from simple, man-made office-like environments to complex, unstructured domains. Exemplifying this progress is a long list of successful applications of SLAM within indoor [126, 12], outdoor [2, 52, 68, 104], and underwater [140, 106, 34] environments of increasing size.

The robotics community has made a great deal of progress towards a better understanding of the SLAM problem. Despite these contributions, though, a number of key outstanding issues remain. Foremost among them is the problem of scalability to increasingly larger maps. As SLAM is applied in larger domains, there is the need for a map representation of the environment together with a model for the coupling between the map and vehicle pose that combine to yield robust map and pose estimates in a computationally efficient manner. In addition to the relatively new problem of scalability are unresolved issues that date back to some of the first work in SLAM. These include robust data association and the problem of dynamic environments.

A well-known open problem in localization and mapping is that of dealing with dynamic environments. The majority of SLAM algorithms explicitly assume that the environment is static in order to simplify map estimation. In reality, though, many environments change over time as people move about, the position of furniture changes, and doors open and close. With a static map model, these changes give rise to seeming discrepancies in the robot's observations of the world. Algorithms often treat this variation as measurement noise and reject the observation as erroneous when the difference is significant. This strategy is robust to a limited degree of change in the environment, but significant variation will cause the map and pose estimates to diverge. In the context of mobile robot localization, Fox *et al.* [43] utilize a pair of filters to detect measurements that correspond to dynamic objects based upon an *a priori* known map of the environment. The authors then perform Markov localization based upon observations of static elements. Wang and Thorpe [135] describe a SLAM framework that employs a separate dynamic object tracking algorithm to identify observations of non-stationary objects. These measurements do not contribute to the SLAM process but are used by the object tracker, which also leverages the SLAM vehicle pose estimate. An alternative approach is to drop the static map assumption and explicitly track dynamic elements within the environment. This approach requires a motion model for the dynamic map elements in order to predict their pose as it changes over time. Secondly, they must account for a varying number of dynamic features as objects appear and disappear from the environment (e.g. as people enter and leave a room). Montemerlo *et al.* [95] consider the problem of tracking people in an office-like environment. They adopt a Brownian motion model for each person and update the number of people within their map based upon the Minimum Description Length metric. The algorithm employs a Monte Carlo filter to track the robot pose and a second Monte Carlo filter for the people.

Data association plays an integral role in most SLAM algorithms, which for correctness must arrange that exteroceptive sensor measurements are correctly matched to their corresponding map elements. Algorithms typically treat this correspondence as a hard constraint and are not robust to errors since they cannot undo the effects of incorrect data association. One exception is the FastSLAM algorithm [93], which effectively tracks multiple hypothesis over data association and, as a result, is robust to errors [53]. Traditionally, data association considers measurements on an individual basis and chooses the maximum likelihood map correspondence, ignoring the mutual consistency among these pairings. This approach is prone to errors and can cause the robot location and map estimates to diverge, particularly when revisiting regions of the map that have not been observed for some time (loop closing). Neira and Tardós [103] propose an improved technique that considers the joint likelihood over the set of correspondences. This approach yields fewer errors but must search over a space of correspondences that is exponential in the number of measurements. Alternatively, Olson, Walter, Teller, and Leonard [112] propose an algorithm that formulates the set of possible data association hypotheses as an adjacency graph whereby they treat the problem as one of graph partitioning. They analyze the spectral properties of the corresponding adjacency matrix to identify the largest set of compatible correspondences in polynomial time. Meanwhile, Cox and Leonard [18] describe a multi-hypothesis tracking algorithm that concurrently maintains several correspondence estimates in the form of a hypothesis tree. The method employs Bayesian techniques to evaluate the likelihood associated with each data association hypothesis. This likelihood effectively serves as a soft correspondence constraint and allows the SLAM filter to assess the accuracy of each assignment over multiple time steps. This approach also has limitations, namely the need to maintain multiple hypotheses that, depending on the underlying filter, can be computationally demanding. While not explicitly a multi-hypothesis implementation, Feder *et al.* [38] and Leonard *et al.* [78] track a delayed-state estimate over vehicle pose that similarly allows them to postpone data association decisions. More recently, Newman and Ho [105] describe a framework for data association that combines a delayed-state representation with salient optical image features that are independent of the vehicle pose and map estimates. The combined benefits of the delayed-state filter and the image filters provide robust loop closure that the authors demonstrate in a 3D outdoor environment [104].

The robotics community has long recognized data association and the modeling of dynamic environments as key problems in localization and mapping. More recently, as SLAM is applied to a greater number of domains and larger environments, other issues arise. Key among them is the problem of scalability, as many SLAM filters impose computational and memory costs that make them intractable for large environments. This limitation has given rise to a number of algorithms that reduce the complexity of localization and mapping in an attempt to scale to larger environments. Among the different strategies are algorithms that break the environment into a set of smaller, more manageable maps [76, 51, 139, 77, 12]. These appropriately-named submap algorithms greatly reduce the effects of map size on the computational cost but are thought to suffer from slower convergence speed [77].

Recently, novel strategies have emerged that offer the promise of scalability through an alternative model for the SLAM distribution. Specifically, Frese and Hirzinger [46] and Thrun *et al.* [130] make key insights into the canonical (information) parametrization of the Gaussian distribution. Their analysis reveals that the graphical model that describes the feature-based SLAM posterior is *almost* sparse.[3] Thrun *et al.* show that, in the case that the parametrization is truly sparse, SLAM filtering is nearly constant time, irrespective of the size of the map. They then present the seminal Sparse Extended Information Filter (SEIF), which explicitly enforces a sparse parametrization to perform efficient, scalable SLAM. However, we show later that the method by which they achieve an exactly sparse representation induces an inconsistent posterior distribution. In the same vein, Paskin [113] provides similar insights into the structure of the corresponding probabilistic graphical model. He describes the Thin Junction Tree Filter (TJTF), which achieves linear complexity by maintaining a sparse graph structure. Leveraging his earlier work in the canonical formulation to SLAM [46], Frese [45] similarly approximates the distribution with a sparse graphical model. His Treemap filter enforces a tree structure for the graphical model and is capable of localization and mapping that is logarithmic in the size of the map. Alternatively, Eustice, Singh, and Leonard [34] derive the Exactly Sparse Delayed-State Filter (ESDSF) that maintains a canonical model for the distribution over the vehicle's pose history. The authors show that this delayed-state representation exhibits a parametrization that is naturally sparse. The ESDSF exploits this structure to perform SLAM in near constant time without having to rely upon additional approximations for the distribution. In order to achieve an exactly sparse representation, though, the ESDSF must maintain a distribution over the entire pose history. Consequently, the state grows linearly with time irrespective of the size of the environment. In the case of sustained, long-term operation within a fixed area, this is an undesirable effect.

With the exception of the ESDSF, the problem with these canonical filters is that the feature-based SLAM parametrization is not sparse. As we discuss later in the thesis, the canonical representation for feature-based SLAM is naturally dense. Consequently, the SEIF, TJTF, and Treemap must approximate the posterior with a distribution that is sparse by pruning the graphical model or modifying the canonical form. The key issue is *how* to approximate the distribution with a sparse canonical parametrization.

1.2 Contributions of this Thesis

SLAM has grown to be a fundamental problem of interest in the robotics community. This interest has helped to answer a number of key questions related to SLAM, but, in the process, raised new questions that remain unresolved. This thesis seeks to address one of these problems, namely that of scaling Simultaneous Localization and Mapping (SLAM) to large environments. Leveraging the insights that Frese *et al.* [46], Paskin [113] and Thrun *et al.* [130] have made regarding the information parametrization, we consider the canonical representation of the Gaussian SLAM distribution as a consistent means to achieve computational and memory efficiency. The main contributions of the thesis are twofold. We first present a thorough analysis of

[3] We refer to a representation as *almost* sparse if it is well-approximated by an exactly sparse form.

the canonical formulation to feature-based SLAM and shed light on the consequences of approximating the distribution as sparse. Based on this analysis, we present a new estimation-theoretic algorithm that maintains exact sparsity in a principled manner. We then describe the application of the algorithm for a challenging underwater research problem – mapping the underside of ship hulls at close range with a narrow field-of-view acoustic camera.

1.2.1 Analysis

The first contribution of the thesis is an in-depth investigation of the canonical parametrization of the Gaussian SLAM distribution. We describe the core aspects of Bayesian SLAM filtering with this model in the context of the fundamental steps of conditioning and marginalization. The derivation reveals important characteristics of the canonical SLAM posterior, notably the means by which the parametrization naturally becomes dense. This discussion lays the groundwork for our sparsification analysis and the subsequent description of our filtering strategy.

The thesis offers a thorough analysis of sparse approximations to the canonical SLAM distribution. We focus, in particular, on the Sparse Extended Information Filter (SEIF) and shed insight into the SEIF sparsification strategy. We explore, in-depth, the approximation employed by the SEIF and show that it relies on a specific premise that results in a posterior that is inconsistent. In particular, we reveal that the SEIF induces global estimates for the robot pose and map that are overconfident. However, empirical evidence suggests that the SEIF sparsification strategy preserves the local consistency of the map. We demonstrate the consequences of SEIF sparsification in detail through both a controlled linear Gaussian (LG) simulation as well as a real-world nonlinear dataset.

1.2.2 Algorithm

The analysis of the SEIF sparsification strategy motivates the need for a sparse formulation to the posterior that retains consistency. The main contribution of the thesis is the Exactly Sparse Extended Information Filter (ESEIF). The ESEIF is a scalable SLAM filter based in the information form that maintains sparsity while preserving the consistency of the pose and map estimates. The thesis describes an integral component of the ESEIF: a method for controlling the density of the canonical parametrization whereby we track a modified version of the SLAM posterior, essentially by ignoring a small fraction of temporal measurements. In this manner, our Exactly Sparse Extended Information Filter (ESEIF) performs inference over a model that is conservative relative to the standard Gaussian distribution. We compare the performance of our algorithm to the standard Extended Kalman Filter (EKF) with a controlled linear Gaussian (LG) simulation and confirm that the ESEIF preserves the consistency of the map and pose estimates. We also demonstrate the SEIF alongside the gold-standard EKF on a pair of benchmark nonlinear datasets. The results convincingly show that our method yields conservative estimates for the robot pose and map that are nearly identical to those produced by the EKF, yet at much less computational expense.

1.2.3 Underwater Application

The final contribution of the thesis is the application of the ESEIF to underwater localization and mapping. We consider the problem of ship hull mapping with an autonomous underwater vehicle (AUV) equipped with a forward-looking sonar. We treat the sonar as an acoustic camera and describe the imaging geometry that underlies the corresponding camera model. The acoustic image data is fused with measurements of the relative position of the ship hull to generate an observation of features on the hull surface. We incorporate this measurement data within the ESEIF and use the vehicle's suite of onboard velocity and attitude sensors to track the six degrees of freedom (DOF) vehicle pose and three-dimensional map of a ship hull.

1.3 Thesis Outline

The remainder of the thesis is organized as follows:

Chapter 2: SLAM: A State Estimation Problem
This chapter serves as an introduction to the probabilistic interpretation of the SLAM problem and describes the application of Bayes filters that form the basis of estimation-theoretic solutions. We introduce the common formulations to localization and mapping and discuss the current state of the art in SLAM. The thesis subsequently focuses on the canonical parametrization of the Gaussian and describes the information form of the Bayesian filter. We conclude the chapter with a discussion on the unique characteristics of the canonical representation as they pertain to scalability.

Chapter 3: Sparsification via Enforced Conditional Independence
The canonical parametrization of feature-based SLAM is naturally dense. Algorithms that leverage a sparse representation must approximate the posterior with a sparse distribution. This chapter explores the implication of such approximations on the probabilistic relationships among the robot and map. We present an in-depth analysis of the sparsification strategy employed by the SEIF and demonstrate that the approximation yields an inconsistent posterior.

Chapter 4: Exactly Sparse Extended Information Filters
In light of our discussion on the consequences of sparsification, we offer an alternative strategy for controlling the structure of the canonical form. This chapter presents the Exactly Sparse Extended Information Filter (ESEIF), an efficient extension of the Bayesian information filter that maintains exact sparsity while preserving consistency. We describe algorithm in detail, including the sparsification rule, as well as efficient inference strategies as they relate to mean estimation and data association. The chapter concludes with an analysis of the ESEIF in a LG simulation, as well as on a pair of nonlinear datasets.

Chapter 5: ESEIF for an Underwater Vehicle with an Imaging Sonar In this chapter, we consider the problem of underwater localization and mapping with an AUV equipped with an acoustic imaging sonar. We apply the ESEIF to perform SLAM on a ship hull, based upon features detected within the acoustic imagery.

Chapter 6: Conclusion
The thesis concludes by establishing our contributions in the context of fundamental problems in the area of SLAM. We discuss the key assumptions that we have made in deriving the ESEIF, as well as the algorithm's possible failure modes. The chapter then presents directions for future research with regards to both improvements to the ESEIF algorithm, as well as its integration with other aspects of robotics.

Appendix A: Implementation Details
The first addendum describes the details of the filter implementations in simulation, as well as real-world datasets, which we refer to throughout the thesis.

Appendix B: Acoustic Imaging Sonar
This second addendum describes the imaging geometry that underlies the HAUV sonar. We present a measurement model that approximates the sonar as an affine imaging acoustic camera.

Chapter 2

SLAM: A State Estimation Problem

This chapter addresses the estimation theoretic view of the Simultaneous Localization and Mapping (SLAM) problem. We first present the general probabilistic interpretation of the state estimator that is common to most SLAM algorithms. Subsequent sections are devoted to distinguishing between the different SLAM implementations. We present a coarse-to-fine analysis, as we first discuss a few alternative representations for the map. We then look at the various characterizations of the probability distributions and how they go about maintaining this estimator. For each, the discussion elaborates on several particular algorithms, so as to highlight the state of the art in SLAM.

2.1 SLAM Problem Formulation

As we discussed in the previous chapter, the SLAM problem is fundamentally rather simple. The robot moves about in an unknown environment, making observations of the world around it. Odometry and velocity measurements provide an estimate of the vehicle's motion. Due to the noise that corrupts this data, the error in the estimated pose drifts with time. SLAM algorithms bound this error by concurrently building a local map against which they reference observations of the environment to localize the vehicle. This leads to the well-known chicken-and-egg problem that characterizes SLAM. The accuracy of the map depends upon how well the robot's position is known while the pose, itself, is estimated based upon this map.

Simultaneous Localization and Mapping (SLAM) can be viewed as a state estimation problem. Simply put, the goal is to improve an estimate for the robot pose and map (the state) over time based upon noisy sensor readings. Most successful SLAM algorithms have been developed with this formulation in mind and take an estimation theoretic approach to address the SLAM problem. What distinguishes one approach from another is how they implement the state estimator. For example, what is the best way to parametrize the map? How do we sufficiently capture the uncertainty in the state that arises as a result of noisy data? What is the best way to track the robot pose and map estimates over time? We present an overview of SLAM and offer answers to some of these questions. For a more detailed discussion on SLAM in the context of probabilistic robotics, the reader is referred to the text by Thrun, Burgard, and Fox [128]. Additionally, Durrant-Whyte and Bailey [32, 3] provide an informative review of the history of the SLAM problem, along with a discussion on the state of the art.

2.1.1 State Representation

Simultaneous Localization and Mapping (SLAM) maintains an estimate for the robot's pose and the map as the robot navigates in the world. As such, the state space includes the robot pose, $\mathbf{x}$, along with a representation for the map, $\mathbf{M}$.

Let us denote by $\mathbf{x}(t) \in \mathbb{R}^p$ the continuous-time state of the robot at time, t. The state vector describes the vehicle pose, i.e. its position and orientation, and may also include linear and angular velocities. The pose space for vehicles that operate in a planar world such as an office environment is $\mathbf{x}(t) \in \mathbb{R}^3$, which includes its (x, y) Cartesian position along with the orientation, θ. Others such as autonomous underwater vehicles (AUVs) and aerial vehicles operate in three-space and require a six element pose vector, $\mathbf{x}(t) \in \mathbb{R}^6$ to describe their six degrees of freedom (DOF). Typically, the state includes the (x, y, z) position of the center of mass (COM), along with an Euler angle representation for orientation, (ϕ, θ, ψ).

The variable, $\mathbf{M}$, denotes the parametrization of the map that describes the salient statistics of the environment. The two types of maps common to SLAM are *topological* and *metric*. Topological maps [72] consider the environment to be a collection of "distinct places" that may include intersections, doorways, and offices in an indoor environment. A topological framework represents the map as a graph in which the distinct places form the nodes and the edges denote connections, such as hallways, between these places. Metric maps, for which this thesis is concerned, explicitly represent the geometrical properties of the environment, typically in the context of Euclidean space.

The two standard metric map representations are occupancy grids and feature-based maps. Occupancy grid maps [96] discretize the world into a set of grid cells. Each cell is associated with a position and a binary label where 1 corresponds to the cell being occupied and 0 signifies that it is empty. The map is then a collection of cells, $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_n\}$, where each $\mathbf{m}_i$ represents the position of the i^{th} cell and its binary label. The space of possible maps is then exponential in the total number of grid cells. Feature-based maps [74, 75], on the other hand, describe the world as a collection of geometric primitives, such as lines and points. The parameters that model each landmark form the continuous-valued state, $\mathbf{m}_i$, and together define the map as the set $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_n\}$.

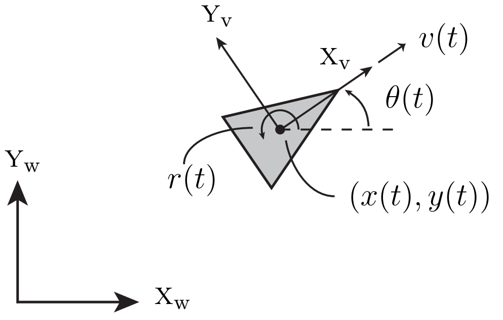
2.1.2 Robot Motion Model

We model the vehicle dynamics with rigid body equations of motion. The time invariant, continuous-time state space model is, in its general form, a nonlinear function of the vehicle state and control input, $\mathbf{u}(t)$,

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right). \tag{2.1}$$

The simplest form of (2.1) is a kinematic, first-order motion model for which the control inputs are either odometry or velocity. The constant-velocity model assumes zero acceleration and considers the vehicle state to be the pose and body-frame velocities. For example, the constant velocity model of a planar wheeled robot is given in Example 2.1.

Example 2.1 (Continuous-time Motion)
Consider a wheeled robot that operates in a planar environment. We define a body-fixed reference frame with the x axis pointing forward and the y axis to the left as shown in the figure. The state vector, $\mathbf{x}(t) = [x(t)\ y(t)\ \theta(t)]^\top$, denotes the position and orientation in the world frame. The kinematics constrain the vehicle to move forward and to turn, but, assuming no slippage, not move laterally. The control inputs, $\mathbf{u}(t) = [v(t)\ r(t)]^\top$, are the body-frame forward velocity, $v(t)$, and rotation rate, $r(t)$. The continuous-time, constant-velocity dynamical model is



$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$$

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\left(\theta(t)\right) \\ v(t)\sin\left(\theta(t)\right) \\ r(t) \end{bmatrix} \tag{2.2}$$

We can model the robot dynamics only to a limited degree of accuracy. There will be some degree of error in the estimated system parameters; the velocity measurements are not exact; the wheels will slip depending on the ground surface; etc. These factors give rise to a model such as that in (2.1) that does not completely describe reality. A modified dynamics model explicitly accounts for the structured and unstructured uncertainties with the addition of a noise term, $\mathbf{w}(t)$. This term may denote external perturbations to the system, as well as modeling (parameter) errors, and is not observable. With its inclusion in the dynamics, the actual system is said to behave according to the model,

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)\right). \tag{2.3}$$

While the vehicle state evolves continuously in time, sensing events, including environment observations and measurements of vehicle velocity, occur at discrete time steps. For the sake of implementation, we consider a discrete-time representation for the vehicle state. Assuming a uniform sampling period of ΔT, $t_k = k \cdot \Delta T$ denotes the k^{th} time step and $\mathbf{x}_k = \mathbf{x}(t_k)$ the corresponding robot state. We convert the continuous-time equations of motion (2.3) to a discrete-time representation (2.4) that describes the evolution of the discrete-time state,[1]

$$\mathbf{x}_{t+1} = \mathbf{f}\left(\mathbf{x}_t, \mathbf{u}_{t+1}, \mathbf{w}_t\right). \tag{2.4}$$

Note that vector function $\mathbf{f}(\cdot)$ in the difference equation is not the same as that in the continuous-time model in 2.3.

Appendix A provides detailed examples of both the continuous-time and discrete-time motion models for different robotic platforms considered in the thesis. We present the motion models for three-DOF wheeled vehicles as well as a full three-dimensional, six-DOF underwater robot. A thorough discussion of the dynamics that govern land-based vehicles can be found in Borenstein *et al.* [11] and Siegwart *et al.* [119]. Fossen [42], meanwhile, is an excellent reference for underwater vehicle dynamical models.

2.1.3 Measurement Model

As the vehicle explores the environment, onboard sensors provide observations of the environment, as well as proprioceptive data. Land robots, for example, commonly use laser range finders to measure the relative position of objects in the world. Underwater vehicles, meanwhile, rely on acoustic sonar to sense the environment and Doppler sensors to measure velocity.

As previously mentioned, sensors provide measurement data at discrete time steps. At any time step, t, there are likely to be multiple different measurements, $\mathbf{z}_t = \left\{{}^1\mathbf{z}_t, {}^2\mathbf{z}_t, \ldots, {}^J\mathbf{z}_t\right\}$, the j^{th} of which we denote by the vector, ${}^j\mathbf{z}_t = {}^j\mathbf{z}(t) \in \mathbb{R}^l$. Each corresponds to an observation of a different map element, m_i, within the sensor's field-of-view (FOV) or of vehicle pose. The general form of a vehicle-relative observation of the object within the environment, ${}^j\mathbf{z}_t$, is a nonlinear function of the vehicle pose and the corresponding map element state, $\mathbf{m}_i$,

$${}^j\mathbf{z}_t = \mathbf{g}\left(\mathbf{x}_t, \mathbf{m}_i\right). \tag{2.5}$$

This model describes the measurement acquisition process and is specific to the operation of different sensors. As detailed as this model may be, though, it is not exact as it is prone to both structured and unstructured uncertainty. For example, the accuracy of parameter estimates is limited, the sensor may not be calibrated, data is corrupted by noise, etc. In similar fashion to the robot dynamic model, we account for the uncertainty with the addition of an unobserved noise term, $\mathbf{v}_t$. The true model is then

$${}^j\mathbf{z}_t = \mathbf{g}\left(\mathbf{x}_t, \mathbf{m}_i, \mathbf{v}_t\right). \tag{2.6}$$

Again, we provide specific examples of measurement models in Appendix A, where

[1] A slight abuse of our earlier notation, we will use t within subscripts to denote an incremental time step.

we describe the laser range and bearing sensors as well as the acoustic camera models that we employ later in the thesis.

2.2 Probabilistic State Representation

The fundamental objective in *online* SLAM implementations is to maintain, over time, an estimate for the current pose and map, $\mathbf{x}_t$ and $\mathbf{M}$, which comprise the latent state, based upon the available observables, $\mathbf{z}^t$ and $\mathbf{u}^t$. The stochastic nature of the vehicle motion and sensor data complicates the estimation and, in particular, the coupling between navigation and mapping that is inherent to SLAM. Many successful SLAM algorithms address these issues by formulating the problem in a probabilistic manner, using generative methods to track the joint distribution over the robot pose and map.

Probabilistic interpretations of SLAM represent the problem in terms of the tuple, $\mathcal{S}_t = \langle \mathsf{x}_t, \mathsf{M}, \mathsf{z}^t, \mathsf{u}^t, T, O \rangle$. The formulations model the robot pose as a stochastic process and the map as a random vector. The random vectors x_t and M denote the latent pose and map states.[2] Similarly, the vehicle motion (odometry, velocity) data and map observations are particular realizations of the random variables, z_t and u_t that constitutes the evidence. Probabilistic SLAM algorithms consider the entire time history of measurements, $\mathsf{z}^t = \{\mathsf{z}_1, \mathsf{z}_2, \ldots, \mathsf{z}_t\}$, and $\mathsf{u}^t = \{\mathsf{u}_1, \mathsf{u}_2, \ldots, \mathsf{u}_t\}$.

The probabilistic state space model specifies a transition function, measurement likelihood model, and a prior over the state and observations. The state transition function, $T : \mathsf{x}_t \times \mathsf{u}_{t+1} \times \mathsf{x}_{t+1} \rightarrow [0, 1]$, is a stochastic model for the dynamic behavior of the robot pose in response to control inputs. Typical SLAM algorithms assume that the map is static, i.e. that $p(\mathbf{M}_{t+1}) = p(\mathbf{M}_t) = p(\mathbf{M})$, and that only the robot pose is dynamic [127]. Note that, while we adopt this assumption in the thesis, it is not always valid, since environments often contain dynamic elements (e.g. chairs within an office, people). Probabilistic algorithms that assume a static world are robust to some change [128], yet, as we discussed in Section 1.1.3, the ability to deal with dynamic environments remains an open problem in SLAM.

We model the vehicle dynamics according to a first-order Markov process of the form, $T : p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_0, \mathbf{u}_{t+1}) = p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_{t+1})$, where $\mathbf{x}_t$ is the current state and $\mathbf{u}_t$ the control[3] [128]. Given the previous pose and current control input, the current pose is conditionally independent of the map and the historical data, i.e. $p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{M}, \mathbf{z}^t, \mathbf{u}^{t+1}) = p(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_{t+1})$. The distribution captures the uncertainty in the vehicle dynamics model along with the noise that corrupts the odometry data, which we describe in (2.4).

The perceptual model, $O : \mathsf{x}_t \times \mathsf{M} \times \mathsf{z}_t \rightarrow [0, 1]$, is a sensor-specific stochastic representation for the physical properties underlying the formation of measurements (2.6).

[2] We use fonts without serifs to denote random variables (e.g. x) and fonts with serifs to represent particular instantiations of the random variable (e.g. x). The same applies to bold symbols that represent vectors.

[3] This assumption is not restrictive as higher-order Markov models can easily be represented as first-order Markov by defining a new state.

A generative model, $p\left(\mathbf{z}_t \mid \mathbf{x}_t, \mathbf{M}\right)$ specifies the dependency of sensor measurements on the robot pose and the observed map elements. The distribution explicitly models noise in the data along with inaccuracies in the physical models.

Finally, the prior specifies the initial probability density over the state. In the context of SLAM, which starts with an empty map, the prior is over the initial robot pose, $p\left(\mathbf{x}_0\right)$.

2.2.1 Bayesian Filtering

This thesis treats SLAM as a filtering problem whereby we track the joint distribution over the state based upon current and historical observation data, i.e. $\mathbf{z}^t = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\}$ and $\mathbf{u}^t = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_t\}$. The goal is then to maintain the conditional density

$$p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right) \tag{2.7}$$

as it evolves over time. With each subsequent time step, data arrives in the form of motion control inputs and relative measurements of the environment. The filter recursively updates the distribution to reflect these new observations, $\{\mathbf{u}_{t+1}, \mathbf{z}_{t+1}\}$. This update follows from the application of Bayes' rule,

$$p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right) \xrightarrow{\mathbf{u}_{t+1}, \mathbf{z}_{t+1}} p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t+1}, \mathbf{u}^{t+1}\right)$$

$$\begin{aligned} p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t+1}, \mathbf{u}^{t+1}\right) = \eta\, & p\left(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1}, \mathbf{M}\right) \\ & \cdot \int p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_{t+1}\right) \cdot p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right) d\mathbf{x}_t. \end{aligned} \tag{2.8}$$

The term, η, is a normalizing constant that ensures the new distribution is a valid probability function. The recursive update (2.8) assumes a first-order Markov model for the motion model and represents the measurements as temporally independent given the state, i.e. $p\left(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1}, \mathbf{M}, \mathbf{z}^t, \mathbf{u}^{t+1}\right) = p\left(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1}, \mathbf{M}\right)$.

The Bayesian filter (2.8) accomplishes two core updates to the distribution: *time prediction* and *measurement update*. The time prediction step updates the distribution to reflect the vehicle's motion from time t to time $t+1$,

$$p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right) \xrightarrow{\mathbf{u}_{t+1}} p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) \tag{2.9}$$

per the state transition function, T. It is useful to break the time prediction component into two sub-processes: state augmentation and roll-up. *State augmentation* first adds the new pose to the state according to the vehicle motion model based upon odometry data and control inputs.

$$p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right) \longrightarrow p\left(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) \tag{2.10}$$

A *roll-up* process follows augmentation and marginalizes away the previous pose, effectively transferring information from the prior to the new distribution,[4]

$$\begin{aligned} p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t+1}\right) &= \int p\left(\mathbf{x}_{t+1}, \mathbf{x}_{t}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t+1}\right) d\mathbf{x}_{t} \\ &= \int p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_{t}, \mathbf{u}_{t+1}\right) \cdot p\left(\mathbf{x}_{t}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t}\right) d\mathbf{x}_{t}. \end{aligned} \tag{2.11}$$

We summarize two two-step interpretation of time prediction in (2.12).

After projecting the dynamics forward in time, the filter incorporates new measurement data into the distribution. The measurement update step is a Bayesian update to the distribution in (2.12b), based upon the generative model of the observation likelihood. The result is the posterior distribution at time $t+1$, which results from conditioning the state upon the measurement data, $\mathbf{z}_{t+1}$, revealed in (2.13).

Algorithm 2.1 (Bayesian SLAM Filter)

$$p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t+1}, \mathbf{u}^{t+1}\right) \propto p\left(\mathbf{z}_{t+1} \mid \mathbf{x}_{t+1}, \mathbf{M}\right) \int p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_{t}, \mathbf{u}_{t+1}\right) p\left(\mathbf{x}_{t}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t}\right) d\mathbf{x}_{t}$$

1) <u>Time Prediction</u>:

$$p\left(\mathbf{x}_{t}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t}\right) \xrightarrow{\mathbf{u}_{t+1}} p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t+1}\right)$$

Augmentation:

$$p\left(\mathbf{x}_{t}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t}\right) \xrightarrow{\mathbf{u}_{t+1}} p\left(\mathbf{x}_{t+1}, \mathbf{x}_{t}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t+1}\right) \tag{2.12a}$$

Roll-up:

$$p\left(\mathbf{x}_{t+1}, \mathbf{x}_{t}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t+1}\right) \longrightarrow p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t+1}\right) \tag{2.12b}$$

2) <u>Update</u>:

$$p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t}, \mathbf{u}^{t+1}\right) \xrightarrow{\mathbf{z}_{t+1}} p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t+1}, \mathbf{u}^{t+1}\right) \tag{2.13}$$

Algorithm 2.1 summarizes the principle Bayesian filter steps. These components form the basis of the many different algorithms that address the SLAM estimation problem [128].

[4] Note that smoothing SLAM filters, also known as delayed-state filters, explicitly maintain the robot pose history and do not marginalize over poses.

2.3 State of the Art in Localization and Mapping

The Bayesian formulation to SLAM is seemingly straightforward, involving fundamental marginalization and conditioning operations that take the form of the time projection and measurement update steps. In reality, though, an exact, general implementation is intractable. For one, the space of maps is continuous and prohibits a generic representation of the environment. The space of robot poses is also continuous, which complicates the marginalization component to time prediction. These difficulties have given rise to a number of SLAM estimation algorithms that rely upon different assumptions regarding the problem in order to alleviate these complications. The approaches differ in their model of the environment, their assertions regarding the form of the motion and measurement models, and their representation of the SLAM posterior. We continue the chapter with a description of the state-of-the-art in SLAM filtering. We take a hierarchical approach, whereby we frame specific algorithms in the context of a few primary schools of thought in the SLAM community.

2.3.1 Feature-based Map Representations

SLAM algorithms avoid the infinite dimensional distribution over the map by reducing the environmental model to a tractable form. Section 2.1.1 described the two metric representations of the environment: occupancy grids and feature-based parametrizations. Occupancy grids discretize the continuous environment into a collection of n grid cells, $c_i = \{(x_i, y_i), m_i\} \in \mathcal{C}$. Assigned to each cell is a binary label, $m_i \in \{0, 1\}$, that classifies it as either occupied or free space. Grid-based probabilistic models track the joint distribution over the cell labels, $p(\mathbf{M}) = p(m_1, m_2, m_3, \ldots, m_n)$. In order to avoid the 2^n dimensionality of the space, the models assume that the occupation values stored within each cell are independent variables,

$$\begin{aligned} p(\mathbf{M}) &= p(m_1, m_2, m_3, \ldots, m_n) \\ &\approx \prod_{c_i \in \mathcal{C}} p(m_i) \,. \end{aligned} \tag{2.14}$$

Rather than an occupancy grid parametrization, this thesis focuses on a feature-based model of the environment. Feature-based approaches account for the intractable dimensionality of the environment by representing the map in terms of a set of geometric primitives. This model reduces the map to a collection of lines, points, planes, etc. Each map element, $\mathbf{m}_i \in \mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \ldots, \mathbf{m}_n\}$, consists of the parameters that model the corresponding landmark primitive, e.g. Cartesian coordinates, length, and orientation. The probability distribution over the map is then a joint distribution over the set of feature parameters. Unlike occupancy grid representations, the landmarks are not assumed to be independent.

Feature-based models of the environment have the benefit that they often provide a decomposition of the map that is more succinct. Unlike an occupancy grid representation, they do not suffer from exponential spatial growth. Instead, the size of feature-based maps is a direct function of the structure and not the size of the environment. Of course, these representations rely upon the presumption that the world is amenable to parametrization in terms of geometric primitives. This assumption is often valid in indoor, office-like environments, where straight walls and clean corners offer quality line and point features. In less structured domains, such as underwater or outdoor environments, the features tend to be simple point landmarks, as we show later in the thesis.

2.3.2 Gaussian Filter Models

A concise representation for the environment alleviates much of the complexity that comes with modeling a distribution over the continuous space of maps. Nonetheless, tracking the SLAM posterior (2.7) with a Bayesian filter remains difficult. Particularly challenging is an appropriate model for the distribution over the robot pose and map. In general, models that are more detailed offer a more accurate representation for the posterior, but at the cost of filtering complexity. Just as the representation of the environment helps to differentiate SLAM algorithms, so does the model for the posterior distribution.

Perhaps the most common representation for the distribution is the multivariate Gaussian (2.15). One benefit of the Gaussian model is that it is compact: it completely describes the distribution by a mean vector, $\boldsymbol{\mu}_t$, and covariance matrix, Σ_t. Additionally, the existence of closed-form realizations of the conditioning and marginalization processes simplify Bayesian filtering. The general Gaussian model for the SLAM posterior is of the form,

$$p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right) = \mathcal{N}\big(\boldsymbol{\mu}_t, \Sigma_t\big) \propto \exp\left\{-\frac{1}{2}\left(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\right)^\top \Sigma_t^{-1}\left(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\right)\right\} \tag{2.15}$$

In their seminal paper [121], Smith *et al.* first present an approach to the SLAM problem based upon a Gaussian model for the distribution. The authors describe a novel solution that leverages an Extended Kalman Filter (EKF) [67, 86] to greatly simplify the Bayesian filtering process. In part as a result of this relative simplicity, this model has become the standard tool of choice for a majority of SLAM algorithms [97, 75, 14, 28]. The EKF explicitly tracks the correlation between the robot state and map elements, which accounts for the coupling that is inherent to SLAM. Consequently, the EKF actively updates the filter estimates for the entire pose and map states, based upon observations of only a subset of the map by exploiting the correlation between the robot state and map elements. An account of the correlations also improves the consistency of the resulting SLAM posterior. Maintaining these correlations, though, imposes an $\mathcal{O}(n^2)$ memory requirement, where n is proportional to the size of the map [107]. Furthermore, while the EKF efficiently predicts the vehicle motion, the standard EKF measurement updates are quadratic in the number of states (map elements). As a consequence, it is well known that the standard EKF SLAM algorithm is limited to relatively small environments (i.e. on the order of a few hundred features) [128].

Submap Decompositions

As robots are deployed in larger environments, extensive research has focused on the scalability restrictions of EKF SLAM. An intuitive way of dealing with this limitation is to divide the world into numerous sub-environments, each comprised of a more manageable number of features. Such is the approach of the appropriately named submap algorithms [76, 51, 139, 77, 12] that shed some of the computational burden of the full EKF by performing filtering over only the robot's current submap. This framework describes the environment hierarchically as a graph in which each node corresponds to a feature-based submap, each represented relative to a local coordinate frame. Edges within the graph describe the transformation between submap reference frames that arise from shared features. The filtering process operates at the local (node) level, with measurement updates performed over individual submaps rather than the entire map. By actively controlling the size of each map partition to contain no more than $l \ll n$ features, the algorithms bound the cost of updates and, in turn, filtering at $\mathcal{O}(l^2)$ rather than the standard $\mathcal{O}(n^2)$.

The computational efficiency of the submap framework comes at the cost of slower estimation convergence and the absence of a consistent global map of the environment. In essence, these approaches ignore what are generally weak correlations among distant landmarks in order to reduce the burden on filtering to track only local features. Consequently, unlike the standard single-map estimator, filter updates improve only local map estimates. It is not surprising, then, that submap algorithms suffer from slower convergence speed [77]. A second drawback is the inability to transform the network of submaps into a single, global map of the environment that respects the transformations between submaps that are encoded within the edges of the graph. These algorithms stitch individual submaps together to form a single map by concatenating frame-to-frame transformations. Typically, though, it is difficult to ensure that the series of transformations are consistent over different paths in the graph (i.e. cycles in the graph)[5], which only coarsely account for the constraints between overlapping submaps. The Atlas algorithm [12], for example, which lacks a notion of the world origin, reduces the graph to a "global" map by registering each submap to an arbitrarily chosen reference node. The registration follows from a shortest path search over the transformations between submaps and is generally not consistent across cycles.

Sparse Information Filtering

Recently, strategies have emerged that offer the promise of scalability through the canonical parametrization for the Gaussian SLAM distribution. Rather than a dense covariance matrix and mean vector, the canonical form completely describes the Gaussian by the information (inverse covariance) matrix and information vector. Analogous to the EKF, the evolution of the posterior is tracked over time via a two step process comprising the Extended Information Filter (EIF) [86, 99]. The canonical

[5] Ideally, the transformation between non-adjacent submaps would be identical irrespective of the path within the graph.

parametrization is the dual of the covariance form, and, in turn, the EIF update step is efficient as it is quadratic in the number of measurements and not the size of the map.[6] On the other hand, the time projection step is, in general, quadratic in the number of landmarks. Also, recovering the mean from the information vector and matrix requires a costly $\mathcal{O}(n^3)$ matrix inversion. Together, these characteristics would seem to rule out the information parametrization as a viable remedy to the scalability problem of the standard EKF and are largely the reason for its limited application to SLAM.

Pivotal insights by Thrun *et al.* [130] and Frese *et al.* [46] reveal that the canonical form is, in fact, particularly beneficial in the context of feature-based SLAM, as a majority of the off-diagonal elements in the normalized information matrix are inherently very small. By approximating some of these small entries as zero, Thrun *et al.* take advantage of what is then a sparse information matrix, presenting the Sparse Extended Information Filter (SEIF), an adaptation of the EIF. In addition to the efficient measurement updates, the SEIF performs the time projection step at a significant savings in cost over the nominal EIF, offering a near constant-time solution to the SLAM problem. The caveat is that a subset of the mean is necessary to linearize the motion and measurement models, as well as to enforce the sparsity of the information matrix. To that end, the authors estimate the mean of the robot pose and a limited number of features as the solution to a sparse set of linear equations that is approximated using relaxation. We discuss the SEIF algorithm in much more detail throughout the remainder of the thesis.

Similar benefits extend from interpreting the canonical parametrization as a Gaussian Markov random field (GMRF) [122] where small entries in the normalized information matrix correspond to weak links in the graphical model. By essentially breaking these weak links, Paskin [113] and Frese [45] approximate the graphical model with a sparse tree structure. Paskin's Thin Junction Tree Filter (TJTF) and Frese's Treemap filter exploit this representation to operate on the graph in $\mathcal{O}(n)$ and $\mathcal{O}(\log n)$ time, respectively.

2.3.3 Rao-Blackwellized Particle Filters

Sequential Monte Carlo methods offer an alternative to the Gaussian model for the SLAM distribution. Rao-Blackwellized particle filters were first applied to the SLAM problem by by Murphy [98] and Doucet *et al.* [29]. Unlike Gaussian filters, which are concerned only with the current robot pose, particle filter techniques consider a form of the posterior that includes the entire trajectory, $\mathbf{x}^t = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$. Particle filter algorithms rely upon a non-parametric model that represents the posterior as a collection of samples (particles) over the state space. The process of Rao-Blackwellization factors the posterior into two distributions, one over the map and the other over the robot pose,

$$p\left(\mathbf{x}^t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right) = p\left(\mathbf{M} \mid \mathbf{x}^t, \mathbf{z}^t\right) \cdot p\left(\mathbf{x}^t \mid \mathbf{z}^t, \mathbf{u}^t\right) \tag{2.16}$$

[6] This assumes knowledge of the mean, which is necessary for observations that are nonlinear in the state.

where the map is assumed to be conditionally independent of $\mathbf{u}^t$ given the pose history. Rao-Blackwellized particle filters maintain a representation for the distribution over the vehicle's trajectory, $p\left(\mathbf{x}^t \mid \mathbf{z}^t, \mathbf{u}^t\right)$, as a set of samples. Associated with each particle is its own model for the map distribution, $p\left(\mathbf{M} \mid \mathbf{x}^t, \mathbf{z}^t\right)$. The most common method for tracking this model for the posterior is via sampling importance resampling (SIR), whereby the filter (i) draws samples from a proposal distribution to reflect the next robot pose, then (ii) weights the samples according to the most recent measurement information in order to better fit the target distribution, and (iii) resamples according to the weights. The filter then updates the map distribution associated with each particle.

The FastSLAM algorithm by Montemerlo *et al.* [93] represents the map distribution for each particle as a Gaussian. FastSLAM treats the map elements as conditionally independent (i.e. uncorrelated) given knowledge of the robot pose. The algorithm then efficiently tracks the map for each of the m particles with a collection of n independent EKFs, one for each feature. The $\mathcal{O}(m \log n)$ computational cost is dependent on the number of particles and offers improved scalability in situations where relatively few particles are necessary to accurately represent the robot pose.

A problem with FastSLAM and Rao-Blackwellized particle filters, in general, is the difficulty in fitting the samples to the target distribution, $p\left(\mathbf{x}^t \mid \mathbf{z}^t, \mathbf{u}^t\right)$. Without being able to sample directly from this posterior, SIR often leads to the phenomenon of "particle depletion" [133], in which resampling eliminates the good particles that closely approximate the actual state. The limitation is due largely to a poor choice for the proposal distribution. Standard SIR algorithms rely on the vehicle's motion model as the proposal distribution. Typically, this distribution exhibits high variance in comparison to the measurement model, which is highly peaked. Samples drawn from the motion model will be distributed over the state space with the greatest density near the mean. In the case where the measurement distribution is aligned with the tail of the proposal, only a small number of samples will receive significant weight. The likelihood of these "good" particles surviving subsequent resampling is diminished, giving rise to particle depletion. The filter can improve the chances that correct samples survive by increasing the number of particles that are tracked. This approach soon becomes intractable due to the curse of dimensionality [49] that arises as a result of having to maintain a distribution over a sequence of poses that grows with time. As the dimensionality of this space increases, a larger number of particles are necessary to describe the likelihood, particularly in the case of greater uncertainty. Particle depletion only exacerbates this problem. Consequently, the efficiency benefits are not as obvious, as the cost of Rao-Blackwellized particle filters is proportional to the number of particles.

One way to reduce particle depletion is to improve the accuracy of the proposal distribution to better approximate the target distribution. As an update to the FastSLAM algorithm, Montemerlo *et al.* [94] incorporate an improved pose sampler that mimics work by de Freitas [26] on alternative proposal distributions. The authors use an EKF to update a Gaussian approximation to a proposal distribution that incorporates the most recent map observations. They show that sampling from this updated proposal greatly reduces the number of particles necessary to model the posterior distribution. Similarly, Grisetti and colleagues [50] describe a two-fold improvement on SIR. They first sample from a Gaussian approximation to a proposal distribution that incorporates the current measurement data so as to minimize the variance in the subsequent sample weights. Rather than resampling with every step as in standard SIR, they resample only when the variance of the weights is deemed sufficiently high. Together, these steps reduce the occurrence of particle depletion and allow them to perform SLAM using an order of magnitude fewer particles than with the standard method.

2.3.4 Nonlinear Graph Optimization

An alternative formulation to SLAM treats the problem as a constrained optimization over the robot pose and map states [30, 41, 47, 27, 128, 110]. This approach solves for the maximum likelihood (ML) estimate of the vehicle pose history by optimizing a full, nonlinear log-likelihood that incorporates the history of robot motion and measurement data. The optimization is performed over a series of iterations, which eases the computational burden and provides robustness to linearization and data association errors.

The initial research in this area dates back to the work of Lu and Milios [82], who formulate the problem as a pose graph in which each node corresponds to a vehicle reference frame and the edges denote spatial constraints between these frames. The constraints encode observations of the relative transformation between two vehicle pose reference frames. Lu and Milios utilize odometry information to establish constraints between consecutive poses and laser scan matching [83] to extract a measurement of the relative transformation between poses based upon overlapping laser scans. Given a network of poses and constraints, they solve for the ML estimate of pose history by optimizing the full, nonlinear log-likelihood. Assuming that the measurement noise is Gaussian the joint likelihood is itself Gaussian and the optimization is equivalent to the minimization of a Mahalanobis distance. The algorithm linearizes the odometry and measurement constraints about the current estimate for the robot trajectory to achieve what is then a linear least-squares formulation to the minimization. The result is a set of linear equations of the form $\mathrm{A}\mathbf{x} = \mathbf{b}$, where the matrix of coefficients, A, is the information matrix for the Gaussian distribution and $\mathbf{x}$ is the unknown pose history. Lu and Milios then solve for the complete robot trajectory, $\mathbf{x}$, by inverting the information matrix at a cost that is cubic in the number of poses. As the length of the robot trajectory grows large, though, this inversion quickly becomes intractable, limiting the algorithm's scalability.

Alternatively, Duckett, Marsland, and Shapiro [30] propose an algorithm that utilizes Gauss-Siedel relaxation to iteratively optimize the joint likelihood. With each iteration, they optimize over only a single node (robot pose) in the graph according to its local constraints, while holding the other nodes fixed. Rather than invert the information matrix to fully solve the set of linear equations, relaxation is equivalent to estimating a particular robot pose by solving the limited set of linear equations that correspond to its local constraints. Repeating this process for the sequence of robot poses leads to convergence to a minimum of the cost function, though not necessarily the global minimum.[7] The computational cost associated with each relaxation iteration is a function of the number of constraints associated with each node, which is often fixed due to the limited FOV of the robot's sensors. In turn, each iteration tends to be $\mathcal{O}(n)$, where n is the number of nodes in the graph. While Duckett *et al.* have found that most online SLAM steps require only a single iteration, large loop closures may require multiple iterations at a total cost of $\mathcal{O}(n^2)$ [31].

Frese, Larsson, and Duckett [47] improve upon this performance with the Multilevel Relaxation algorithm, which adapts multigrid linear equation solvers [13] to perform the estimation. Multilevel relaxation describes the pose graph at various levels of resolution, and performs optimization on this hierarchy. Underlying this graph decomposition, is a multigrid representation of the set of linear equations, $\mathrm{A}\mathbf{x} = \mathbf{b}$, that describe the least-squares solution. Given a new constraint, the algorithm progresses from the finest to coarsest resolutions of the hierarchy, performing relaxation on the set of linear equations corresponding to each level. At the lowest resolution, the algorithm directly solves the corresponding coarse set of linear equations and subsequently propagates the coarse state estimates to the finer levels of the hierarchy. In essence, relaxation at high resolution refines the error in the pose estimates while solutions at the coarse level reduce the error that tends to be spatially smooth by coarsely adapting the estimates. Multilevel relaxation then requires $\mathcal{O}(n)$ operations for each iteration update, including those that involve large loop closures.

Dellaert [27] computes the ML estimate of the robot's pose history along with a feature-based map by directly solving the linear least-squares problem. The Square Root SAM algorithm takes advantage of what is a naturally sparse information matrix [113, 34] to solve the for the mean over the state as the solution to the set of linear equations. The algorithm decomposes the information matrix into either its QR or Cholesky (LDL) factorization [48], paying close attention to variable ordering.[8] In turn, the estimator jointly solves for the mean over the pose and map via back-substitution. As the author insightfully shows, the approach closely parallels aspects of the aforementioned graphical model methods. The results demonstrate promising advantages over the EKF with regards to performance, though the algorithm currently does not address some important aspects of the SLAM problem such as data association.

In its general form, the algorithm operates as a batch estimator, solving the full least-squares problem anew as new constraints arise. Each time, the algorithm builds the information matrix, performs the factorization and subsequently solves for the entire trajectory and map. In order to achieve robustness to linearization errors, this approach requires that the constraints be occasionally re-linearized when the state estimate changes significantly. This batch implementation is sub-optimal for online SLAM as the computational complexity grows with time. In the case that the current linearization point is sufficiently accurate, one can avoid recomputing factorizing the

[7] Duckett *et al.* further rely upon knowledge of the robot's heading to express the constraints as linear in the robot's position history. Consequently, the log-likelihood is quadratic in the state, and the optimization is guaranteed to converge to the global minimum.

[8] Dellaert notes that COLAMD [22] yields a good variable ordering and that the Cholesky (LDL) decomposition is particularly efficient at factorization.

information matrix and take advantage of techniques that incrementally update an existing matrix decomposition [23]. Kaess *et al.* [66] describe a variation on Square Root SAM that incrementally updates the QR factorization of the information matrix based upon a particular variable ordering. This algorithm is efficient when the robot explores new regions of the environment, allowing a full state solution in time that is linear in the number of poses and landmarks. Loop closures, on the other hand, alter the structure of the information matrix and, with the same variable ordering, lead to the fill-in of the QR matrix decomposition. Kaess *et al.* avoid this fill-in with a process that periodically identifies a new variable ordering and subsequently performs a full matrix factorization. So as to avoid linearization errors, the algorithm occasionally updates the information matrix with a new linearization prior to full factorization. The limitation of both the incremental and batch versions of the algorithm is the cost of the matrix decomposition, which grows unbounded with time as a result of jointly estimating the entire robot trajectory history. Nonetheless, empirical evidence suggests that the factorization can be made tractable for large state dimensions with a good variable ordering [27].

Olson, Leonard, and Teller [110] similarly treat SLAM as a pose graph and optimize the joint likelihood associated with the inter-pose measurement and odometry constraints. Rather than solve for the full least-squares solution for the state all at once, though, they optimize the cost function iteratively. The algorithm performs the optimization via stochastic gradient descent (SGD) [116], whereby it considers only a single constraint (an edge in the graph) and its corresponding gradient at each iteration. With each update, the algorithm searches for the new state estimate along the direction of this gradient by an amount determined by a variable learning rate. The batch version of the algorithm iterates over the set of constraints, gradually decreasing the learning rate to reduce the likelihood that the optimization converges to a local minimum. A fast approximation to optimization, though, SGD generally does not converge to the exact global minimum. Olson *et al.* complement SGD with a state representation in terms of the incremental pose, i.e. the difference between subsequent poses. As a result of this formulation, the SGD optimization of a constraint between two poses in the graph has the effect of updating the estimates for the entire temporal sequence of poses in between. The computational cost of each constraint iteration is $\mathcal{O}(\log n)$, where n is the size of the state (number of poses). While the number of iterations necessary for convergence is not clear, the algorithm has been shown to yield accurate state estimates in far less time than other iterative optimization techniques [110].

Olson *et al.* [111] propose a variation of their batch algorithm suitable for online application that incorporates constraints in an incremental fashion. While it may seem straightforward to develop an online extension of the SGD-based optimization that incorporates measurement constraints as they arrive, one challenge is the effect of the variable learning rate. The batch algorithm reduces the magnitude of the learning rate as the number of constraints increases, which, in turn, would diminish the contribution that newly introduced constraints have on the optimization. Instead, the online algorithm introduces a pose-specific learning rate that varies spatially over the graph and does not explicitly depend upon the number of iterations. New constraints and their subsequent poses are assigned a learning rate that balances the information available in the corresponding pose measurements with the predicted confidence as currently maintained by the graph. Meanwhile, the algorithm updates the learning rate associated with old constraints as the learning rate assigned to their intermediate poses changes. This learning rate allocation allows the algorithm to integrate new measurement data in such a way that the optimization modifies the local pose estimates without significantly disturbing the configuration of distant nodes. The actual optimization is performed on constraints whose corresponding subgraphs have converged the least, as indicated by learning rates that exceed a desired threshold. Results demonstrate that these updates involve only a small fraction of the total number of constraints. Nonetheless, the optimizations significantly improve the overall likelihood cost, which is not surprising as the corresponding nodes are farthest from convergence.

2.4 Information Formulation to SLAM

The Gaussian distribution remains the most widely used parametric model for the SLAM posterior. In SLAM as well as other probabilistic inference problems, one typically represents the Gaussian distribution in what is referred to as the **standard form**, in terms of its mean vector, $\boldsymbol{\mu}$, and covariance matrix, Σ. The popularity of this parametrization is largely due to the ability to track the distribution over time with the EKF. As we have discussed, however, the need to maintain correlations among the state, an implicit characteristic of the standard form, restricts the size of the map for feature-based SLAM. In this section, we describe the details of the canonical parametrization for the Gaussian distribution, which we mentioned earlier in §2.3.2. We show that this representation offers advantages over the standard form and, in subsequent chapters, we exploit these characteristics to achieve scalable, feature-based SLAM.

2.4.1 Canonical Gaussian Representation

We first present an alternative parametrization to a general Gaussian distribution and contrast this representation with the standard covariance form. We focus, in particular, on the duality between the two parametrizations in the context of the fundamental aspects of SLAM filtering.

Let $\boldsymbol{\xi}_t$ be a random vector governed by a multivariate Gaussian probability distribution, $\boldsymbol{\xi}_t \sim \mathcal{N}\big(\boldsymbol{\mu}_t, \Sigma_t\big)$, traditionally parametrized in full by the mean vector, $\boldsymbol{\mu}_t$, and covariance matrix, Σ_t. Expanding the quadratic term within the Gaussian exponential, we arrive at an equivalent representation for the multivariate distribution,

$\mathcal{N}^{-1}\big(\boldsymbol{\eta}_t, \Lambda_t\big)$.

$$\begin{aligned} p\left(\boldsymbol{\xi}_t\right) &= \mathcal{N}\big(\boldsymbol{\xi}_t; \boldsymbol{\mu}_t, \Sigma_t\big) \\ &\propto \exp\big\{-\tfrac{1}{2}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t)^\top \Sigma_t^{-1}(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t)\big\} \\ &= \exp\big\{-\tfrac{1}{2}\left(\boldsymbol{\xi}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t - 2\boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t + \boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\mu}_t\right)\big\} \\ &\propto \exp\big\{-\tfrac{1}{2}\boldsymbol{\xi}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t + \boldsymbol{\mu}_t^\top \Sigma_t^{-1}\boldsymbol{\xi}_t\big\} \\ &= \exp\big\{-\tfrac{1}{2}\boldsymbol{\xi}_t^\top \Lambda_t \boldsymbol{\xi}_t + \boldsymbol{\eta}_t^\top \boldsymbol{\xi}_t\big\} \\ &\propto \mathcal{N}^{-1}\big(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t\big) \end{aligned} \tag{2.17}$$

The **canonical form** of the Gaussian (2.17) is completely parametrized by the information matrix, Λ_t, and information vector, $\boldsymbol{\eta}_t$, which are related to the mean vector and covariance matrix by (2.18).

$$\Lambda_t = \Sigma_t^{-1} \tag{2.18a}$$

$$\boldsymbol{\eta}_t = \Sigma_t^{-1}\boldsymbol{\mu}_t \tag{2.18b}$$

Duality between Standard and Canonical Forms

The canonical parametrization for the multivariate Gaussian is the dual of the standard form in regard to the marginalization and conditioning operations [113], as demonstrated in Table 2.1. Marginalizing over variables with the standard form is *easy* since we simply remove the corresponding elements from the mean vector and covariance matrix. However, the same operation in the canonical form involves calculating a Schur complement and is computationally *hard*. The opposite is true when computing the conditional from the joint distribution; it is *hard* with the standard form yet *easy* with the canonical parametrization.

The duality between the two parametrizations has important consequences for SLAM implementations as marginalization and conditioning are integral to the filtering process. The marginalization operation is fundamental to the time prediction step as part of the roll-up process (2.11). Measurement updates (2.13), meanwhile, implement a conditioning operation in order to incorporate new observation data in the distribution over the state. The duality between the two Gaussian parametrizations then helps to explain why time prediction is computationally easy/hard with a standard/canonical parametrizations while measurement updates are hard/easy. The quadratic complexity of measurement updates is implicit to the standard form and contributes of the EKF's scalability problem. However, the subsequent chapters demonstrate the ability to exploit the structure of the information matrix in order to make what is otherwise a hard marginalization operation easy in the canonical form. Consequently, both the measurement update and time prediction components of SLAM information filters can be made to better scale with the size of the environment.

Table 2.1: Summary of the marginalization and conditioning operations on a Gaussian distribution expressed in the covariance and information forms.

$$p\left(\boldsymbol{\alpha},\boldsymbol{\beta}\right) = \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta \end{bmatrix}, \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta} \end{bmatrix}\Big) = \mathcal{N}^{-1}\Big(\begin{bmatrix} \boldsymbol{\eta}_\alpha \\ \boldsymbol{\eta}_\beta \end{bmatrix}, \begin{bmatrix} \Lambda_{\alpha\alpha} & \Lambda_{\alpha\beta} \\ \Lambda_{\beta\alpha} & \Lambda_{\beta\beta} \end{bmatrix}\Big)$$

	MARGINALIZATION	CONDITIONING
	$p\left(\boldsymbol{\alpha}\right) = \int p\left(\boldsymbol{\alpha},\boldsymbol{\beta}\right) d\boldsymbol{\beta}$	$p\left(\boldsymbol{\alpha} \mid \boldsymbol{\beta}\right) = p\left(\boldsymbol{\alpha},\boldsymbol{\beta}\right) / p\left(\boldsymbol{\beta}\right)$
COVARIANCE FORM	$\boldsymbol{\mu} = \boldsymbol{\mu}_\alpha$ $\Sigma = \Sigma_{\alpha\alpha}$	$\boldsymbol{\mu}' = \boldsymbol{\mu}_\alpha + \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta)$ $\Sigma' = \Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha}$
INFORMATION FORM	$\boldsymbol{\eta} = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\boldsymbol{\eta}_\beta$ $\Lambda = \Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}$	$\boldsymbol{\eta}' = \boldsymbol{\eta}_\alpha - \Lambda_{\alpha\beta}\boldsymbol{\beta}$ $\Lambda' = \Lambda_{\alpha\alpha}$

Encoding the Markov Random Field

Throughout the thesis, we take advantage of the graphical model [63] representation of the SLAM distribution to better understand the estimation process. This is particularly true in the case of the information form of the Gaussian, as we use the graphical model to motivate novel filtering algorithms. An advantageous property of the canonical parametrization is that the information matrix provides an explicit representation for the structure of the corresponding undirected graph or, equivalently, the GMRF [122, 113]. This property follows from the factorization of a general Gaussian density

$$\begin{aligned} p\left(\boldsymbol{\xi}\right) &\propto \exp\bigl\{-\tfrac{1}{2}\boldsymbol{\xi}^\top\Lambda\boldsymbol{\xi} + \boldsymbol{\eta}^\top\boldsymbol{\xi}\bigr\} \\ &= \prod_i \exp\bigl\{-\tfrac{1}{2}\left(\lambda_{ii}\xi_i^2 - \eta_i\xi_i\right)\bigr\} \cdot \prod_{\substack{i,j \\ i\neq j}} \exp\bigl\{-\tfrac{1}{2}\xi_i\lambda_{ij}\xi_j\bigr\} \\ &= \prod_i \Psi_i(\xi_i) \cdot \prod_{\substack{i,j \\ i\neq j}} \Psi_{ij}(\xi_i,\xi_j) \end{aligned}$$

where

$$\begin{aligned} \Psi_i(\xi_i) &= \exp\bigl\{-\tfrac{1}{2}\left(\lambda_{ii}\xi_i^2 - \eta_i\xi_i\right)\bigr\} \\ \Psi_{ij}(\xi_i,\xi_j) &= \exp\bigl\{-\tfrac{1}{2}\xi_i\lambda_{ij}\xi_j\bigr\} \end{aligned}$$

are the node and edge potentials, respectively, for the corresponding undirected graph. Random variable pairs with zero off-diagonal elements in the information matrix (i.e. $\lambda_{ij} = 0$) have an edge potential $\Psi_{ij}\left(\xi_i,\xi_j\right) = 1$, signifying the absence of a link between the nodes representing the variables. Conversely, non-zero shared information indicates that there is an edge joining the corresponding nodes with the strength of

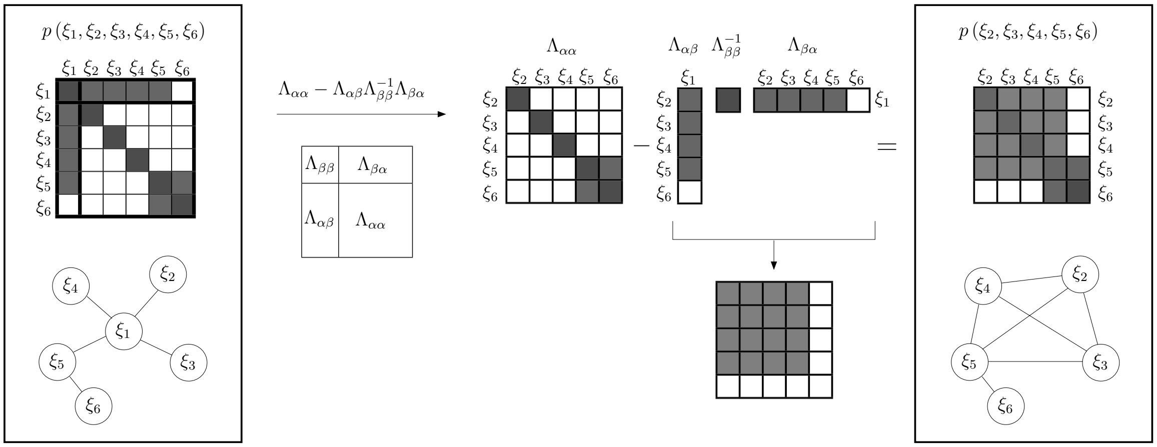

$p(\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6)$
ξ_1 ξ_2 ξ_3 ξ_4 ξ_5 ξ_6
$\Lambda_{\alpha\alpha} - \Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}$
$\Lambda_{\beta\beta}$
$\Lambda_{\beta\alpha}$
$\Lambda_{\alpha\beta}$
$\Lambda_{\alpha\alpha}$
$\Lambda_{\alpha\alpha}$
$\Lambda_{\alpha\beta}$ $\Lambda_{\beta\beta}^{-1}$
$\Lambda_{\beta\alpha}$
$-$
$=$
$p(\xi_2, \xi_3, \xi_4, \xi_5, \xi_6)$

Figure 2-1: An example of the effect of marginalization on the Gaussian information matrix. We start out with a joint posterior over $\xi_{1:6}$ represented by the information matrix and corresponding Markov network pictorialized on the left. The information matrix for the marginalized density, $p(\xi_{2:6}) = \int p(\xi_{1:6})\, d\xi_1$, corresponds to the Schur complement of $\Lambda_{\beta\beta} = \Lambda_{\xi_1\xi_1}$ in $\Lambda_{\xi_{1:6}\xi_{1:6}}$. This calculation essentially passes information constraints from the variable being removed, ξ_1, onto its adjacent nodes, adding shared information between these variables. We see, then, that a consequence of marginalization is the population of the information matrix.

the edge proportional to λ_{ij}. In turn, as the link topology for an undirected graph explicitly captures the conditional dependencies among variables, so does the structure of the information matrix. The presence of off-diagonal elements that are equal to zero then implies that the corresponding variables are conditionally independent given the remaining states.

It is interesting to note that one comes to the same conclusion from a simple analysis of the conditioning operation for the information form. Per Table 2.1, conditioning a pair of random variables, $\boldsymbol{\alpha} = [\boldsymbol{\xi}_i^\top \ \boldsymbol{\xi}_j^\top]^\top$, on the remaining states, $\boldsymbol{\beta}$, involves extracting the $\Lambda_{\alpha\alpha}$ sub-block from the information matrix. When there is no shared information between $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$, $\Lambda_{\alpha\alpha}$ is block-diagonal, as is its inverse (i.e. the covariance matrix). Conditioned upon $\boldsymbol{\beta}$, the two variables are uncorrelated, and we can conclude that they are conditionally independent:[9] $p\big(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j \mid \boldsymbol{\beta}\big) = p(\boldsymbol{\xi}_i \mid \boldsymbol{\beta}) \cdot p\big(\boldsymbol{\xi}_j \mid \boldsymbol{\beta}\big)$. The fact that the information matrix characterizes the conditional independence relationships emphasizes the significance of its structure.

With particular regard to the structure of the information matrix, it is important to make a distinction between elements that are truly zero and those that are just small in comparison to others. On that note, we return to the process of marginalization, which modifies zeros in the information matrix, thereby destroying some conditional independencies [113]. Consider a six-state Gaussian random vector, $\boldsymbol{\xi} \sim \mathcal{N}^{-1}\big(\boldsymbol{\eta}, \Lambda\big)$, characterized by the information matrix and GMRF depicted in the left-hand side of Figure 2-1. The canonical form of the marginal density $p(\xi_{2:6}) = \int p(\xi_{1:6})\, d\xi_1 = \mathcal{N}^{-1}\big(\boldsymbol{\eta}', \Lambda'\big)$ follows from Table 2.1 with $\boldsymbol{\alpha} = [\xi_2 \ \xi_3 \ \xi_4 \ \xi_5 \ \xi_6]^\top$

[9] This equality holds for Gaussian distributions but is, otherwise, not generally valid.

and $\beta = \xi_1$. The correction term in the Schur complement, $\Lambda_{\alpha\beta}\Lambda_{\beta\beta}^{-1}\Lambda_{\beta\alpha}$, is non-zero only at locations associated with variables directly linked with ξ_1. This set, denoted as $m^+ = \{\xi_2, \xi_3, \xi_4, \xi_5\}$, comprises the Markov blanket [114] for ξ_1. Subtracting the correction matrix modifies a number of entries in the $\Lambda_{\alpha\alpha}$ information submatrix, including some that were originally zero. Specifically, while no links exist between $\xi_{2:5}$ in the original distribution, the variables in m^+ become fully connected due to marginalizing over ξ_1. Marginalization results in the population of the information matrix, a characteristic that has important consequences when it comes to applying the information form to feature-based SLAM.

2.5 Feature-based SLAM Information Filters

Now that we have discussed the fundamental aspects of the canonical Gaussian representation, we describe its application to the SLAM problem. Much like the Extended Kalman Filter (EKF) tracks the mean and covariance of the Gaussian distribution, the Extended Information Filter (EIF) tracks the information vector and information matrix that comprise the canonical form. Stemming from the duality between the standard and canonical parametrizations, there are fundamental differences in the way in which the EIF formulates the time projection and measurement update steps. The remainder of this section is devoted to a detailed description of the canonical formulation to these processes. We show that the canonical parametrization of feature-based SLAM naturally evolves into a unique form. Subsequent sections describe filtering algorithms that exploit this structure to address the scalability problem.

2.5.1 Active versus Passive Features

Throughout this section and the remainder of the thesis, we differentiate between *active* features and *passive* features as first proposed by Thrun *et al.* [130]. We borrow their notation and refer to *active* features as those that share information with the robot state. In terms of the graphical model, active landmarks are defined by the existence of an edge that pairs them with the robot node. We denote the set of active landmarks as m^+, which is consistent with our earlier notation for the Markov blanket for the robot pose. In turn, *passive* landmarks are features with zero entries in the information matrix corresponding to the robot pose. They are not directly linked to the robot node in the GMRF and are, equivalently, conditionally independent of the vehicle state. We denote this set of landmarks as m^-.

2.5.2 Measurement Updates in the EIF

Measurement updates are a bottleneck of the EKF, since they involve modifying the entire covariance matrix at quadratic cost. In contrast, the EIF updates only a small, bounded number of terms within the information vector and matrix. The difference is a consequence of the duality between the standard and canonical parametrizations and, in particular, the differences in the fundamental conditioning operation.

Consider the state of the system at time t after having projected the SLAM posterior forward in time, $p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^{t-1}, \mathbf{u}^t\right) = \mathcal{N}^{-1}\big(\bar{\boldsymbol{\eta}}_t, \bar{\Lambda}_t\big)$. Given a set of observations of neighboring landmarks, $\mathbf{z}_t$, the update process follows from Bayes' rule along with independence assumptions regarding the measurements as detailed in (2.8),

$$p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right) \propto p\left(\mathbf{z}_t \mid \boldsymbol{\xi}_t\right) \cdot p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^{t-1}, \mathbf{u}^t\right). \tag{2.19}$$

The general measurement model (2.20a) is a nonlinear function of the state corrupted by white Gaussian noise, $\mathbf{v}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}\big)$. As part of our Gaussian approximation to the probabilistic measurement model, $p\left(\mathbf{z}_t \mid \boldsymbol{\xi}_t\right)$, we linearize (2.20a) with respect to the state. Equation (2.20b) is the first-order Taylor series linearization about the mean of the robot pose and observed features with the Jacobian, H, evaluated at this mean.

$$\begin{aligned} \mathbf{z}_t &= \mathbf{h}\big(\boldsymbol{\xi}_t\big) + \mathbf{v}_t && (2.20\text{a}) \\ &\approx \mathbf{h}\big(\bar{\boldsymbol{\mu}}_t\big) + \mathrm{H}\big(\boldsymbol{\xi}_t - \bar{\boldsymbol{\mu}}_t\big) + \mathbf{v}_t && (2.20\text{b}) \end{aligned}$$

The EIF estimates the canonical form of the new posterior via the update step:

$$p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right) = \mathcal{N}^{-1}\big(\boldsymbol{\eta}_t, \Lambda_t\big)$$

$$\begin{aligned} \Lambda_t &= \bar{\Lambda}_t + \mathrm{H}^\top \mathrm{R}^{-1} \mathrm{H} && (2.21\text{a}) \\ \boldsymbol{\eta}_t &= \bar{\boldsymbol{\eta}}_t + \mathrm{H}^\top \mathrm{R}^{-1}\big(\mathbf{z}_t - \mathbf{h}\left(\bar{\boldsymbol{\mu}}_t\right) + \mathrm{H}\bar{\boldsymbol{\mu}}_t\big) && (2.21\text{b}) \end{aligned}$$

A detailed derivation may be found elsewhere [130].

At any time step, the robot typically makes a limited number, m, of relative observations to individual landmarks. The measurement model is then a function only of the vehicle pose and this small subset of map elements, $\mathbf{m}_i$ and $\mathbf{m}_j$ and, in turn, a majority of terms in the Jacobian (2.22) are zero.

$$\mathrm{H} = \begin{bmatrix} \frac{\partial h_1}{\partial \mathbf{x}_t} & \cdots & \mathbf{0} & \cdots & \frac{\partial h_1}{\partial \mathbf{m}_i} & \cdots & \mathbf{0} \\ \vdots & & & \ddots & & & \vdots \\ \frac{\partial h_m}{\partial \mathbf{x}_t} & \cdots & \frac{\partial h_m}{\partial \mathbf{m}_j} & \cdots & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \tag{2.22}$$

The matrix outer-product in (2.21a), $\mathrm{H}^\top \mathrm{R}^{-1} \mathrm{H}$, is zero everywhere except at positions associated with the vehicle pose and observed features. More specifically, the matrix is populated at the $\mathbf{x}_t$, $\mathbf{m}_i$, and $\mathbf{m}_j$ positions along the diagonal, as well as at the off-diagonal positions for the $(\mathbf{x}_t, \mathbf{m}_i)$ and $(\mathbf{x}_t, \mathbf{m}_j)$ pairs. The addition of this matrix to the original information matrix modifies only the terms exclusively related to the robot and the observed landmarks. The update then acts to either strengthen existing constraints between the vehicle and these features or to establish new ones (i.e. make them active).

Due to the sparseness of H, computing $\mathrm{H}^\top \mathrm{R}^{-1} \mathrm{H}$ involves $\mathcal{O}(m^2)$ multiplications. Assuming knowledge of the mean for the robot pose and observed features for the linearization, this matrix product is the most expensive component of (2.21). Since the number of observations, m, is limited by the robot's FOV, the EIF update time is bounded and does not grow with the size of the map. In general, though, we do not have an estimate for the current mean, and computing it via (2.18b) requires an $\mathcal{O}(n^3)$ matrix inversion. The exception is when the measurement model is linear, in which case the mean is not necessary and the update step is indeed constant-time.

2.5.3 Time Projection Step

While the measurement step is computationally easy for the EIF but hard for the EKF, the opposite is true of the time projection component. In Section 2.2.1, we described time projection as a two step process whereby we first augment the state with the new robot pose and subsequently marginalize out the previous pose. We present the EIF projection step in the same way in order to make clear some fundamental characteristics of the information formulation to feature-based SLAM. We show that the complexity of the EIF projection step is a direct consequence of the duality of the fundamental marginalization process.

State Augmentation

A Markov model governs the motion of the robot and is, in general, a nonlinear function (2.23a) of the previous pose and the control input. The additive term, $\mathbf{w}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{Q}\big)$, represents a Gaussian approximation to the uncertainty in the model. The first-order linearization about the mean robot pose, $\boldsymbol{\mu}_{x_t}$, follows in (2.23b) where F is the Jacobian matrix.

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_{t+1}\big) + \mathbf{w}_t && \text{(2.23a)} \\ &\approx \mathbf{f}\big(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\big) + \mathrm{F}\big(\mathbf{x}_t - \boldsymbol{\mu}_{x_t}\big) + \mathbf{w}_t && \text{(2.23b)} \end{aligned}$$

In the first step, we grow the state vector to also include the new robot pose, $\hat{\boldsymbol{\xi}}_{t+1} = \big[\mathbf{x}_t^\top \; \mathbf{x}_{t+1}^\top \; \mathbf{M}^\top\big]^\top$. The distribution over $\hat{\boldsymbol{\xi}}_{t+1}$ follows from the current posterior, $p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right) = \mathcal{N}^{-1}\big(\boldsymbol{\eta}_t, \Lambda_t\big)$, through the factorization

$$p\left(\hat{\boldsymbol{\xi}}_{t+1} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = p\left(\mathbf{x}_{t+1}, \boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = p\left(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_{t+1}\right) \cdot p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right)$$

where we have exploited the Markov property. Accordingly, the augmentation to the information matrix and vector is shown by Eustice *et al.* [34] to have the form given in (2.24). Notice that the new robot pose shares information with the previous pose but not the map. This is exemplified in the middle schematic within Figure 2-2 by the fact that the only effect on the structure of the graphical model is the addition of the $\mathbf{x}_{t+1}$ node linked to that of $\mathbf{x}_t$. Given $\mathbf{x}_t$, the $\mathbf{x}_{t+1}$ pose is conditionally independent of the map as a consequence of the Markov property.

$$p\left(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\hat{\boldsymbol{\eta}}_{t+1}, \hat{\Lambda}_{t+1}\big)$$

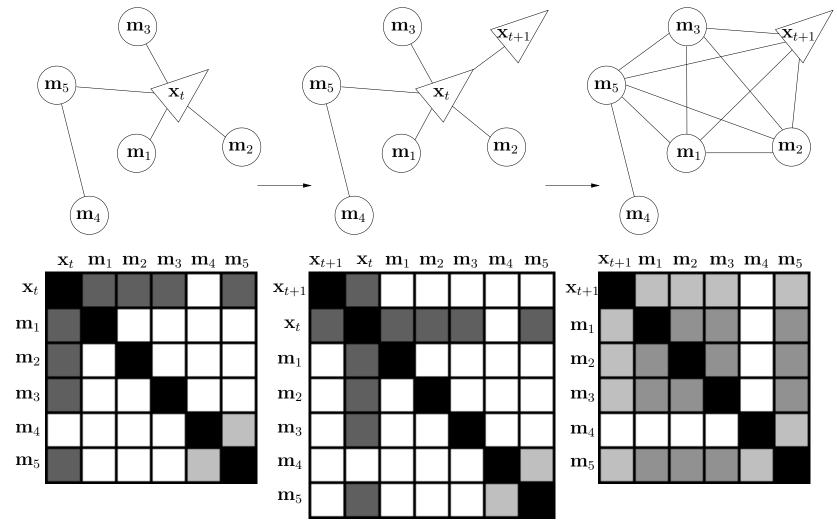


Figure 2-2: A graphical explanation for the inherent density of the information matrix due to the motion update step. Darker shades in the matrix imply larger magnitude. On the left are the Markov network and sparse information matrix prior to time projection in which the robot shares information with the active features, $\mathbf{m}^+ = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_5\}$. We augment the state with the new robot pose, which is linked only to the previous pose due to the Markovian motion model. Subsequently, we marginalize over $\mathbf{x}_t$, resulting in the representation shown on the right. The removal of $\mathbf{x}_t$ creates constraints between the robot and each map element in $\mathbf{m}^+$, which are now fully connected. Along with filling in the information matrix, we see from the shading that the time projection step weakens many constraints, explaining the approximate sparsity of the normalized information matrix.

$$\hat{\Lambda}_{t+1} = \left[\begin{array}{c|cc} \left(\Lambda_{x_tx_t} + \mathrm{F}^\top \mathrm{Q}^{-1}\mathrm{F}\right) & -\mathrm{F}^\top \mathrm{Q}^{-1} & \Lambda_{x_tM} \\ \hline -\mathrm{Q}^{-1}\mathrm{F} & \mathrm{Q}^{-1} & 0 \\ \Lambda_{Mx_t} & 0 & \Lambda_{MM} \end{array}\right] = \left[\begin{array}{c|c} \hat{\Lambda}^{11}_{t+1} & \hat{\Lambda}^{12}_{t+1} \\ \hline \hat{\Lambda}^{21}_{t+1} & \hat{\Lambda}^{22}_{t+1} \end{array}\right] \tag{2.24a}$$

$$\hat{\boldsymbol{\eta}}_{t+1} = \left[\begin{array}{c} \boldsymbol{\eta}_{x_t} - \mathrm{F}^\top \mathrm{Q}^{-1}\big(\mathbf{f}\left(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\right) - \mathrm{F}\boldsymbol{\mu}_{x_t}\big) \\ \hline \mathrm{Q}^{-1}\big(\mathbf{f}\left(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\right) - \mathrm{F}\boldsymbol{\mu}_{x_t}\big) \\ \boldsymbol{\eta}_M \end{array}\right] = \left[\begin{array}{c} \hat{\boldsymbol{\eta}}^1_{t+1} \\ \hline \hat{\boldsymbol{\eta}}^2_{t+1} \end{array}\right] \tag{2.24b}$$

Marginalization

We complete the time projection step by marginalizing $\mathbf{x}_t$ from the posterior to achieve the desired distribution over $\boldsymbol{\xi}_{t+1} = \left[\mathbf{x}_{t+1}^\top\ \mathbf{M}^\top\right]^\top$,

$$p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \int_{\mathbf{x}_t} p\left(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) d\mathbf{x}_t.$$

This brings us back to the expression for marginalization in the canonical representation from Table 2.1. Let $\boldsymbol{\alpha} = \left[\mathbf{x}_{t+1}^\top\ \mathbf{M}^\top\right]^\top$ and $\boldsymbol{\beta} = \mathbf{x}_t$. Using the decomposition of the information matrix and information vector in (2.24), we apply the canonical form of the marginalization here:

$$p\left(\boldsymbol{\xi}_{t+1} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\left(\bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1}\right)$$

$$\bar{\Lambda}_{t+1} = \hat{\Lambda}^{22}_{t+1} - \hat{\Lambda}^{21}_{t+1}\left(\hat{\Lambda}^{11}_{t+1}\right)^{-1}\hat{\Lambda}^{12}_{t+1} \tag{2.25a}$$

$$\bar{\boldsymbol{\eta}}_{t+1} = \hat{\boldsymbol{\eta}}^{2}_{t+1} - \hat{\Lambda}^{21}_{t+1}\left(\hat{\Lambda}^{11}_{t+1}\right)^{-1}\hat{\boldsymbol{\eta}}^{1}_{t+1} \tag{2.25b}$$

The resulting information matrix (2.25a) is the Schur complement of the on-diagonal matrix sub-block that corresponds to the previous pose, $\hat{\Lambda}^{11}_{t+1} = \left(\Lambda_{x_t x_t} + \mathrm{F}^\top \mathrm{Q}^{-1}\mathrm{F}\right)$.

Computational Complexity

The computational cost of the augmentation component is nearly constant-time. In order to modify the canonical distribution to include the new pose, we add a row and column to the information matrix that correspond to the new pose. This row and column pair is zero everywhere except for the fixed-size sub-block that denotes shared information with the previous pose. We also modify the diagonal matrix sub-block for the previous pose. Both operations involve the product of the Jacobian and noise matrices at a cost that depends only on the number of elements that comprise the previous pose state. The complexity of pose augmentation is independent of the size of the map with one caveat. In the common case that the motion model is nonlinear, linearization using (2.23b) requires the mean of the previous pose. Recovering the entire mean vector directly from the information matrix and vector via (2.18) requires that we invert the information matrix. The cost of this inversion is cubic in the size of the state, $\boldsymbol{\xi}_t$, and, in turn, the map size. Fortunately, as we show in later chapters, there are a number of alternatives that yield an estimate for a subset of the mean vector in constant time.

Assuming a linear motion model or efficient access to the robot mean, the marginalization component dictates the computational cost of the time projection step. The correction term of the Schur complement (2.25a) is calculated as the outer product with the off-diagonal submatrix for the old pose, $\hat{\Lambda}^{21}_{t+1}(\hat{\Lambda}^{11}_{t+1})^{-1}\hat{\Lambda}^{12}_{t+1}$. The complexity of this outer product is quadratic in the number of nonzero elements within the matrix, $\hat{\Lambda}^{21}_{t+1} = \hat{\Lambda}^{12\top}_{t+1}$, which encodes the shared information between the old pose and map. Consequently, the Schur complement is quadratic in the number of map elements linked to the old pose (i.e. the size of $\mathbf{m}^+$). The number of nonzero elements within the resulting matrix outer product is quadratic in the size of the active map. As the example in Figure 2-2 demonstrates, the Schur complement modifies $\mathcal{O}(|\mathbf{m}^+|^2)$ components of the information matrix, including elements that correspond to information shared among the map and between the map and new robot pose.

To summarize, the time projection step imposes an $\mathcal{O}(|\mathbf{m}^+|^2)$ computational cost on the EIF and results in $\mathcal{O}(|\mathbf{m}^+|^2)$ modifications to the information matrix.

2.5.4 Structure of the SLAM Information Matrix

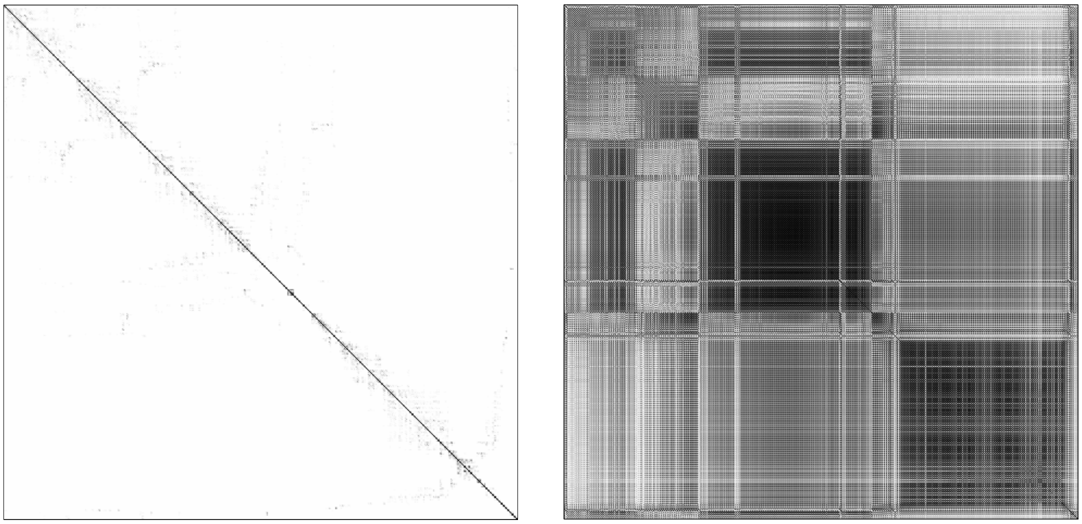
Beyond the immediate computational issues, the time projection step has important consequences regarding the structure of the information matrix. The marginalization component, in particular, gives rise to a fully populated matrix over the course of the SLAM filtering process. We describe the natural evolution of the feature-based SLAM canonical form and show that the information matrix takes on a *relatively* sparse form that is, nonetheless, fully populated.[10]

To better understand the consequences of this marginalization, we refer back to the discussion at the end of Section 2.4.1. Prior to marginalization, the old robot pose is linked to the active features, $\mathbf{m}^+$, while the new pose shares information only with $\mathbf{x}_t$. When we remove the old pose, though, a link is formed between the new pose and each feature in $\mathbf{m}^+$, and this set itself becomes fully connected. The information matrix that was originally sparse is now populated as a consequence of (2.25a). In the scenario depicted in the right-hand side of Figure 2-2, the only remaining zero entries correspond to the lone feature, $\mathbf{m}_4$, which will become active upon the next observation. The subsequent time projection step will instantiate shared information between this landmark and the other features within the active map. Over time, more features will become active and links between the robot pose and map will persist, leading to an ever-growing active map. As Paskin [113] previously showed, the time projection step naturally leads to a full information matrix in online, feature-based SLAM, with a single maximum clique over the GMRF.

The population of the information matrix, particularly the number of off-diagonal elements that associate the robot pose with the map, affects the computational cost of the filtering process. While the EIF incorporates new measurements in near constant-time, the trade-off is a time projection step that is quadratic in the number of active landmarks. As we have just stated, though, the active map inherently grows to include the entire map. In its natural form, then, the EIF is also $\mathcal{O}(n^2)$ per iteration and does not offer an alternative to the EKF in addressing scalability issues. A closer look at the structure of the information matrix, though, reveals that an adapted form of the EIF may provide a solution. Figure 2-3(a) depicts the final normalized information matrix for a nonlinear dataset, where darker shades denote greater magnitudes. While every element within the matrix is nonzero, the large majority of off-diagonal values are negligible in comparison to a few entries that have large magnitude. The structure of the matrix implies a GMRF in which a small number of edges are strong while most are relatively weak. Figure 2-3(b) compares the information matrix with the corresponding normalized covariance (correlation) matrix. The covariance matrix is also fully-populated, but the elements are fairly uniform in magnitude over the entire matrix. This disparity is typical of feature-based SLAM for which landmarks become fully-correlated in the limit [107], yet the information matrix is naturally relatively sparse.

Ironically, while the time projection step induces the fill-in of the information matrix, it also results in the matrix's relatively sparse structure. Returning to the

[10] We describe *relative* sparsity in the context of the normalized information matrix for which the majority of elements are very small relative to a few dominant elements.



(a) Information Matrix

(b) Covariance Matrix

Figure 2-3: The (a) normalized information and (b) covariance (correlation) matrices for a feature-based SLAM distribution. Darker shades denote elements with larger magnitude, and white elements represent zero values. Typical of feature-based SLAM, the information matrix, while fully populated, is relatively sparse, dominated by a small number of large elements. On the other hand, the elements within the normalized covariance matrix are uniformly large as a result of the high correlation among the map.

example pictorialized in Figure 2-2, note that, aside from populating the information matrix, the time projection step weakens the off-diagonal links. The correction term within the Schur complement component of marginalization (2.25a) creates shared information among any active landmarks that were previously conditionally independent. At the same time, the correction term has the effect of weakening the information shared among features already linked in the graph. Marginalization also degrades the information that pairs the robot pose with the active map. When the vehicle re-observes landmarks, the update to the canonical distribution strengthens the edge between the robot pose and these features. In turn, by marginalizing over this pose, the subsequent time projection step distributes the information associated with these links jointly among the corresponding features. This has the effect of strengthening the information shared among these features, which, prior to the observation, had decayed due to marginalization. Over the course of the SLAM filtering process, this has been shown to result in a normalized information matrix that is nearly sparse [44].

2.6 Efficient Filtering via Sparsity Approximation

In the limit of the EIF filtering process, the GMRF naturally forms into a single maximum clique over the robot pose and map [113]. The information matrix is *rela-*

tively sparse but nonetheless fully-populated, as even the small elements are nonzero. The computational complexity of the time projection step, in turn, is quadratic in the size of the map. In order to store the full matrix, the canonical parametrization requires an amount of memory that is also quadratic in the number of landmarks. Consequently, the filter suffers from the same problem of scalability as the EKF.

A close analysis of the canonical parametrization reveals that, by approximating the matrix as being exactly sparse, it is possible to achieve significant gains when it comes to both storage and time requirements [130, 46, 113]. Specifically, a bound on the number of links between the robot and the map allows for near constant-time implementation of the time projection step and controls the fill-in of the information matrix resulting from marginalization. The delicate issue is *how* to approximate the posterior, $p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right)$, with a sparse canonical form. Paskin's Thin Junction Tree Filter (TJTF) [113] essentially breaks weak links in the GMRF in order to approximate the graphical model with a sparse tree structure. Frese [45] adopts a similar strategy with the Treemap filter that represents the environment as a hierarchical tree structure. Analogous to message passing in the TJTF, the Treemap algorithm efficiently maintains an information parametrization of the posterior by exploiting the sparse tree structure. Meanwhile, Thrun and colleagues [130] describe the Sparse Extended Information Filter (SEIF), which forces weaker, nonzero information that is shared between the robot and map to be zero in order to maintain a sparse information matrix.

2.7 Discussion

Simultaneous Localization and Mapping (SLAM) has taken its place as a fundamental problem within robotics. Much attention has been paid to the problem, giving rise to several estimation-theoretic algorithms that maintain a joint distribution over the vehicle pose and map (2.7). We have reviewed the fundamental characteristics that define these approaches, notably the representation for the posterior distribution and the model of the environment. Each have their respective advantages and disadvantages but are faced with the common challenge of robustly scaling to larger environments. This thesis addresses that challenge in the context of the Gaussian representation of the feature-based SLAM posterior (2.15).

We have reviewed much of the state of the art in scalable SLAM filtering. Of particular promise for overcoming the quadratic complexity of the EKF are the insights into the structure of the canonical parametrization of the posterior (2.17). The key observation that the feature-based canonical form is nearly sparse has given rise to filtering strategies that exploit exactly sparse approximations to achieve scalability. While a majority of the elements within the information matrix are relatively small, the feature-based parametrization is nonetheless fully populated. Consequently, each of these algorithms must approximate the SLAM posterior by a distribution that exhibits exact sparsity. The delicate issue in this case is *how* to perform this approximation. The specific sparsification strategy has important consequences regarding the accuracy of the resulting posterior distribution.

In the next chapter, we present a detailed analysis of the Sparse Extended Information Filter (SEIF) approach to sparsification from the perspective of our discussion in Section 2.4.1 on the conditional independence encoded within the information matrix. We show that the SEIF approximates the conditional independence between the robot and a subset of the map in order to enforce sparsity. Our analysis reveals that a consequence of SEIF sparsification strategy is an inconsistent posterior over the robot pose and map. We subsequently propose an alternative sparsification strategy that preserves consistency while simultaneously providing the benefits of a sparse information parametrization.

Chapter 3

Sparsification via Enforced Conditional Independence

Chapter 2 discussed the structure inherent to the canonical parametrization of the SLAM posterior. We have shown that the information matrix exhibits the appealing property that the majority of terms are very small in comparison to a few large elements. In the case that the parametrization is actually sparse, filtering can be performed in near constant time. Unfortunately, while the matrix is *relatively* sparse, it remains fully populated as every element is generally nonzero.

The fill-in of the information matrix induced by the motion update step, together with its computational complexity, are proportional to the number of links between the robot and the map. Unfortunately, as discussed in Section 2.5.4, these links may weaken as a result of the time projection step, but they never disappear. The size of the active map will only increase over time and quickly leads to a fully populated information matrix. As a result, in order to bound the size of the active map and, in turn, preserve the sparsity of the canonical parametrization, the Sparse Extended Information Filter (SEIF) actively breaks links between the robot and the map. In this chapter, we explore, in depth, the approximation that the SEIF employs in order to break these links.[1] We show that a particular assumption of the SEIF sparsification process leads to an inconsistent global estimate for the map. Meanwhile, empirical testing suggests that the relative map relations are better preserved. Nonetheless, the general sparsification strategy that forces the robot to be conditionally independent of a set of landmarks inherently yields an inconsistent distribution over the map.

3.1 SEIF Sparsification Rule

The SEIF describes the structure of the information matrix in terms of Γ_a, the desired size of the active map, and Γ_p, the number of links among the entire map. The SEIF enforces a bound on Γ_a in order to maintain a desired level of sparsity reflected by Γ_p. Recalling the conditional dependency relationships implicit in the GMRF, the

[1] The analysis of the SEIF sparsification strategy that we discuss in this chapter was performed in collaboration with Ryan Eustice [37].

SEIF breaks links between the robot and a set of landmarks by replacing the SLAM posterior with a distribution that approximates the conditional independence between the pose and these features. In describing the sparsification rule, we decompose the map into three disjoint sets, $\mathbf{M} = \{\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\}$. The set $\mathbf{m}^-$ consists of the passive features that will *remain* passive. In a slight abuse of notation, $\mathbf{m}^+$ denotes the active features that are to *remain* active. We represent the active landmarks that will be *made* passive as $\mathbf{m}^0$. The SEIF sparsification routine proceeds from a decomposition of the SLAM posterior

$$\begin{aligned} p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right) &= p\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \\ &= p\left(\mathbf{x}_t \mid \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \qquad (3.1a) \\ &= p\left(\mathbf{x}_t \mid \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^- = \boldsymbol{\varphi}\right) \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right), \qquad (3.1b) \end{aligned}$$

where we have omitted the dependence upon $\mathbf{z}^t$ and $\mathbf{u}^t$ for notational convenience. In (3.1b), we are free to condition on an arbitrary instantiation of the passive features, $\mathbf{m}^- = \boldsymbol{\varphi}$, due to the conditional independence between the robot and these landmarks.

The SEIF deactivates landmarks by replacing (3.1b) with an approximation to the posterior that drops the dependence of the robot pose on $\mathbf{m}^0$:

$$\begin{aligned} \tilde{p}_{\text{SEIF}}\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right) &= \tilde{p}_{\text{SEIF}}\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \\ &= p\left(\mathbf{x}_t \mid \mathbf{m}^+, \mathbf{m}^- = \boldsymbol{\varphi}\right) \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right). \qquad (3.2) \end{aligned}$$

While the expression in (3.1b) is theoretically exact, it is no longer valid to condition upon a particular value for the passive map elements while ignoring the dependence upon $\mathbf{m}^0$ as we have done in (3.2). Given only a subset of the active map, the robot pose and passive features are *dependent*, suggesting that the particular choice for $\boldsymbol{\varphi}$ affects the approximation. Indeed, the dependence on the specific setting of the passive map is apparent from the covariance formulation to sparsification. Applying Bayes' rule to (3.2), we factor the sparsified SEIF distribution as

$$\begin{aligned} \tilde{p}_{\text{SEIF}}\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right) &= p\left(\mathbf{x}_t \mid \mathbf{m}^+, \mathbf{m}^- = \boldsymbol{\varphi}\right) \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \\ &= \frac{p_B\left(\mathbf{x}_t, \mathbf{m}^+ \mid \mathbf{m}^- = \boldsymbol{\varphi}\right)}{p_C\left(\mathbf{m}^+ \mid \mathbf{m}^- = \boldsymbol{\varphi}\right)} \cdot p_D\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \qquad (3.3) \end{aligned}$$

The standard (3.5) and canonical (3.7) parametrizations for the p_B, p_C and p_D distributions follow directly from $p\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) = \mathcal{N}\left(\boldsymbol{\mu}_t, \Sigma_t\right) = \mathcal{N}^{-1}\left(\boldsymbol{\eta}_t, \Lambda_t\right)$ according to the basic marginalization and conditioning operations detailed in Table 2.1. For the sake of consistency, we adopt the notation employed by Thrun *et al.* [130], in which S denotes a projection matrix over the state $\boldsymbol{\xi}_t$ (e.g., $\mathbf{x}_t = \mathrm{S}_{x_t}^\top \boldsymbol{\xi}_t$ extracts the robot pose). We reveal the parametrization for the SEIF sparsified posterior (3.3) in both the standard form (3.4), as well as the canonical form (3.6).

Covariance Form

$$\tilde{\Sigma}_t = \big(\mathrm{S}_{x_t,m^+}\Sigma_B^{-1}\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Sigma_C^{-1}\mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Sigma_D^{-1}\mathrm{S}_{m^0,m^+,m^-}^\top\big)^{-1} \tag{3.4a}$$

$$\tilde{\boldsymbol{\mu}}_t = \boldsymbol{\mu}_t + \tilde{\Sigma}_t\big(\mathrm{S}_{x_t,m^+}\Sigma_B^{-1}\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Sigma_C^{-1}\mathrm{S}_{m^+}^\top\big)\Sigma_t\mathrm{S}_{m^-}\big(\Sigma_{m^-m^-}\big)^{-1}\big(\boldsymbol{\varphi} - \boldsymbol{\mu}_{m^-}\big) \tag{3.4b}$$

where

$$\begin{aligned}
\Sigma_B &= \mathrm{S}_{x_t,m^+}^\top\Sigma_t\mathrm{S}_{x_t,m^+} - \mathrm{S}_{x_t,m^+}^\top\Sigma_t\mathrm{S}_{m^-}\big(\mathrm{S}_{m^-}^\top\Sigma_t\mathrm{S}_{m^-}\big)^{-1}\mathrm{S}_{m^-}^\top\Sigma_t\mathrm{S}_{x_t,m^+}\\
\Sigma_C &= \mathrm{S}_{m^+}^\top\Sigma_t\mathrm{S}_{m^+} - \mathrm{S}_{m^+}^\top\Sigma_t\mathrm{S}_{m^-}\big(\mathrm{S}_{m^-}^\top\Sigma_t\mathrm{S}_{m^-}\big)^{-1}\mathrm{S}_{m^-}^\top\Sigma_t\mathrm{S}_{m^+}\\
\Sigma_D &= \mathrm{S}_{m^0,m^+,m^-}^\top\Sigma_t\mathrm{S}_{m^0,m^+,m^-}
\end{aligned} \tag{3.5}$$

Information Form

$$\tilde{\Lambda}_t = \mathrm{S}_{x_t,m^+}\Lambda_B\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Lambda_C\mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Lambda_D\mathrm{S}_{m^0,m^+,m^-}^\top \tag{3.6a}$$

$$\tilde{\boldsymbol{\eta}}_t = \mathrm{S}_{x_t,m^+}\boldsymbol{\eta}_B - \mathrm{S}_{m^+}\boldsymbol{\eta}_C + \mathrm{S}_{m^0,m^+,m^-}\boldsymbol{\eta}_D \tag{3.6b}$$

where

$$\begin{aligned}
\boldsymbol{\eta}_{\boldsymbol{\varphi}} &= \Lambda_t\mathrm{S}_{m^-}\boldsymbol{\varphi}\\
\Lambda_B &= \mathrm{S}_{x_t,m^+}^\top\Lambda_t\mathrm{S}_{x_t,m^+} - \mathrm{S}_{x_t,m^+}^\top\Lambda_t\mathrm{S}_{m^0}\left(\mathrm{S}_{m^0}^\top\Lambda_t\mathrm{S}_{m^0}\right)^{-1}\mathrm{S}_{m^0}^\top\Lambda_t\mathrm{S}_{x_t,m^+}\\
\boldsymbol{\eta}_B &= \mathrm{S}_{x_t,m^+}^\top\left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\varphi\right) - \mathrm{S}_{x_t,m^+}^\top\Lambda_t\mathrm{S}_{m^0}\left(\mathrm{S}_{m^0}^\top\Lambda_t\mathrm{S}_{m^0}\right)^{-1}\mathrm{S}_{m^0}^\top\left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\varphi\right)\\
\boldsymbol{\Lambda}_C &= \mathrm{S}_{m^+}^\top\Lambda_t\mathrm{S}_{m^+} - \mathrm{S}_{m^+}^\top\Lambda_t\mathrm{S}_{x_t,m^0}\left(\mathrm{S}_{x_t,m^0}^\top\Lambda_t\mathrm{S}_{x_t,m^0}\right)^{-1}\mathrm{S}_{x_t,m^0}^\top\Lambda_t\mathrm{S}_{m^+}\\
\boldsymbol{\eta}_C &= \mathrm{S}_{m^+}^\top\left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\varphi\right) - \mathrm{S}_{m^+}^\top\Lambda_t\mathrm{S}_{x_t,m^0}\left(\mathrm{S}_{x_t,m^0}^\top\Lambda_t\mathrm{S}_{x_t,m^0}\right)^{-1}\mathrm{S}_{x_t,m^0}^\top\left(\boldsymbol{\eta}_t - \boldsymbol{\eta}_\varphi\right)\\
\boldsymbol{\Lambda}_D &= \mathrm{S}_{m^0,m^+,m^-}^\top\Lambda_t\mathrm{S}_{m^0,m^+,m^-} - \mathrm{S}_{m^0,m^+,m^-}^\top\Lambda_t\mathrm{S}_{x_t}\left(\mathrm{S}_{x_t}^\top\Lambda_t\mathrm{S}_{x_t}\right)^{-1}\mathrm{S}_{x_t}^\top\Lambda_t\mathrm{S}_{m^0,m^+,m^-}\\
\boldsymbol{\eta}_D &= \mathrm{S}_{m^0,m^+,m^-}^\top\boldsymbol{\eta}_t - \mathrm{S}_{m^0,m^+,m^-}^\top\Lambda_t\mathrm{S}_{x_t}\left(\mathrm{S}_{x_t}^\top\Lambda_t\mathrm{S}_{x_t}\right)^{-1}\mathrm{S}_{x_t}^\top\boldsymbol{\eta}_t
\end{aligned} \tag{3.7}$$

Notice in (3.4b) that the mean for the sparsified distribution depends upon the choice for $\boldsymbol{\varphi}$. Conditioning on a value for the passive features other than their mean (i.e. $\boldsymbol{\varphi} \neq \boldsymbol{\mu}_{m^-}$) yields a mean of $\tilde{p}_{\text{SEIF}}\big(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\big)$ that differs from that of the original posterior,[2] $p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right)$. Furthermore, we will demonstrate that by ignoring the dependence relationships in (3.2), the SEIF sparsification algorithm leads to inconsistent covariance estimates.

3.2 Modified Sparsification Rule

The SEIF sparsification strategy breaks links between the robot state and a set of features by approximating their conditional dependence. This approach to sparsification introduces a conditional dependence between the robot pose and the passive map. The SEIF then adopts a specific realization of the passive landmarks, which, as

[2] The mean is preserved by the sparsification routine that Thrun and colleagues describe in their paper [130] since the authors condition upon $\boldsymbol{\varphi} = \boldsymbol{\mu}_{m^-}$ and not $\boldsymbol{\varphi} = \mathbf{0}$ as the paper states.

a result of this dependence, affects the form of the resulting posterior. In particular, choosing a value other than the mean for the passive map modifies the mean of the SLAM distribution. Even in the case that the mean is preserved, though, the sparsification rule induces inconsistent error bounds in the posterior. In this section, we present a variation on the SEIF rule that similarly enforces conditional independence to impose sparsity without sacrificing the quality of the approximation.

We derive the modified sparsification rule from a factorized version of the posterior, $p\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right)$ much like we use for the SEIF in (3.2). In this case, though, we exploit the initial conditional independence between the robot and the passive landmarks when given the active map. Rather than setting the passive features to a specific instantiation, we make the valid choice of dropping $\mathbf{m}^-$ from the conditional distribution over the pose at the outset. We exploit the conditional independence in (3.8b) and, for convenience, factorize the posterior as

$$
\begin{aligned}
p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right) &= p\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \\
&= p\left(\mathbf{x}_t \mid \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) && \text{(3.8a)} \\
&\overset{\text{C.I.}}{=} p\left(\mathbf{x}_t \mid \mathbf{m}^0, \mathbf{m}^+\right) \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) && \text{(3.8b)} \\
&= \frac{p\left(\mathbf{x}_t, \mathbf{m}^0 \mid \mathbf{m}^+\right)}{p\left(\mathbf{m}^0 \mid \mathbf{m}^+\right)} \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) && \text{(3.8c)}
\end{aligned}
$$

The modified rule proceeds from the theoretically exact formulation in (3.8c) and subsequently approximates the conditional independence of $\mathbf{x}_t$ and $\mathbf{m}^0$, given $\mathbf{m}^+$:

$$
\begin{aligned}
\breve{p}_{\text{ModRule}}\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) &= \frac{p\left(\mathbf{x}_t \mid \mathbf{m}^+\right) \cdot p\left(\mathbf{m}^0 \mid \mathbf{m}^+\right)}{p\left(\mathbf{m}^0 \mid \mathbf{m}^+\right)} \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) && \text{(3.9a)} \\
&= p\left(\mathbf{x}_t \mid \mathbf{m}^+\right) \cdot p\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) && \text{(3.9b)} \\
&= \frac{p_U\left(\mathbf{x}_t, \mathbf{m}^+\right)}{p_V\left(\mathbf{m}^+\right)} \cdot p_D\left(\mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right). && \text{(3.9c)}
\end{aligned}
$$

For convenience, we factorize the modified posterior in terms of p_U, p_V, and p_D. The form of these distributions easily follow from the original SLAM posterior, $p\left(\mathbf{x}_t, \mathbf{m}^0, \mathbf{m}^+, \mathbf{m}^-\right) = \mathcal{N}\left(\boldsymbol{\mu}_t, \Sigma_t\right) = \mathcal{N}^{-1}\left(\boldsymbol{\eta}_t, \Lambda_t\right)$, according to the marginalization and conditioning operations described in Table 2.1. As with the derivation of the SEIF sparsification step, we take advantage of this factorization to arrive at the standard (3.10) and canonical (3.11) forms of the modified posterior.

Covariance Form

$$
\begin{aligned}
\breve{\Sigma}_t &= \left(\mathrm{S}_{x_t,m^+}\Sigma_U^{-1}\mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Sigma_V^{-1}\mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Sigma_D^{-1}\mathrm{S}_{m^0,m^+,m^-}^\top\right)^{-1} && \text{(3.10a)} \\
\breve{\boldsymbol{\mu}}_t &= \boldsymbol{\mu}_t && \text{(3.10b)}
\end{aligned}
$$

where

$$\begin{aligned}
\Sigma_U &= \mathrm{S}_{x_t,m^+}^\top \Sigma_t \mathrm{S}_{x_t,m^+} \\
\Sigma_V &= \mathrm{S}_{m^+}^\top \Sigma_t \mathrm{S}_{m^+} \\
\Sigma_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \Sigma_t \mathrm{S}_{m^0,m^+,m^-}
\end{aligned}$$

Information Form

$$\breve{\Lambda}_t = \mathrm{S}_{x_t,m^+}\Lambda_U \mathrm{S}_{x_t,m^+}^\top - \mathrm{S}_{m^+}\Lambda_V \mathrm{S}_{m^+}^\top + \mathrm{S}_{m^0,m^+,m^-}\Lambda_D \mathrm{S}_{m^0,m^+,m^-}^\top \tag{3.11a}$$

$$\breve{\boldsymbol{\eta}}_t = \mathrm{S}_{x_t,m^+}\boldsymbol{\eta}_U - \mathrm{S}_{m^+}\boldsymbol{\eta}_V + \mathrm{S}_{m^0,m^+,m^-}\boldsymbol{\eta}_D \tag{3.11b}$$

where

$$\begin{aligned}
\Lambda_U &= \mathrm{S}_{x_t,m^+}^\top \Lambda_t \mathrm{S}_{x_t,m^+} - \mathrm{S}_{x_t,m^+}^\top \Lambda_t \mathrm{S}_{m^0,m^-}\left(\mathrm{S}_{m^0,m^-}^\top \Lambda_t \mathrm{S}_{m^0,m^-}\right)^{-1} \mathrm{S}_{m^0,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^+} \\
\boldsymbol{\eta}_U &= \mathrm{S}_{x_t,m^+}^\top \boldsymbol{\eta}_t - \mathrm{S}_{x_t,m^+}^\top \Lambda_t \mathrm{S}_{m^0,m^-}\left(\mathrm{S}_{m^0,m^-}^\top \Lambda_t \mathrm{S}_{m^0,m^-}\right)^{-1} \mathrm{S}_{m^0,m^-}^\top \boldsymbol{\eta}_t \\
\Lambda_V &= \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{m^+} - \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^-}\left(\mathrm{S}_{x_t,m^0,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^-}\right)^{-1} \mathrm{S}_{x_t,m^0,m^-}^\top \Lambda_t \mathrm{S}_{m^+} \\
\boldsymbol{\eta}_V &= \mathrm{S}_{m^+}^\top \boldsymbol{\eta}_t - \mathrm{S}_{m^+}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^-}\left(\mathrm{S}_{x_t,m^0,m^-}^\top \Lambda_t \mathrm{S}_{x_t,m^0,m^-}\right)^{-1} \mathrm{S}_{x_t,m^0,m^-}^\top \boldsymbol{\eta}_t \\
\Lambda_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \Lambda_t \mathrm{S}_{m^0,m^+,m^-} - \mathrm{S}_{m^0,m^+,m^-}^\top \Lambda_t \mathrm{S}_{x_t}\left(\mathrm{S}_{x_t}^\top \Lambda_t \mathrm{S}_{x_t}\right)^{-1} \mathrm{S}_{x_t}^\top \Lambda_t \mathrm{S}_{m^0,m^+,m^-} \\
\boldsymbol{\eta}_D &= \mathrm{S}_{m^0,m^+,m^-}^\top \boldsymbol{\eta}_t - \mathrm{S}_{m^0,m^+,m^-}^\top \Lambda_t \mathrm{S}_{x_t}\left(\mathrm{S}_{x_t}^\top \Lambda_t \mathrm{S}_{x_t}\right)^{-1} \mathrm{S}_{x_t}^\top \boldsymbol{\eta}_t
\end{aligned}$$

The modified sparsification rule reduces the size of the active map by imposing conditional independence between $\mathbf{x}_t$ and $\mathbf{m}^0$. If we look closely at the expression for the information matrix in (3.11a), the term $\mathrm{S}_{x_tm^+}\Lambda_U\mathrm{S}_{x_tm^+}^\top$ populates the links between the robot and the $\mathbf{m}^+$ landmarks only. The off-diagonal elements that pair the robot pose with the $\mathbf{m}^0$ map elements are now zero, which implies that the approximation enforces conditional independence. Additionally, without setting the passive map to a particular instantiation, the modified rule preserves the mean of the original distribution (3.10b). Furthermore, we will show that, unlike the SEIF sparsification strategy, the modified rule yields uncertainty estimates that are nearly identical to those of the original SLAM posterior. Nonetheless, the results demonstrate that the distribution remains slightly overconfident. In the following section, we suggest that this inconsistency is an implicit result of a sparsification strategy that relies on an approximation to the conditional independence between the robot and active landmarks.

Though the modified rule provides a more sound means of enforcing conditional independence, the computational cost required to implement it is significant. In particular, in order to solve for Λ_U and Λ_V, we need to invert the two matrices, $\mathrm{S}_{m^0,m^-}^\top\Lambda_t\mathrm{S}_{m^0,m^-}$ and $\mathrm{S}_{x_t,m^0,m^-}^\top\Lambda_t\mathrm{S}_{x_t,m^0,m^-}$, of size proportional to the number of passive landmarks. The cost of these inversions is cubic in the size of the passive map, and quickly becomes intractable. Consequently, the modified rule is not a viable option when it comes to solving the scalability problem of SLAM. Nonetheless, the modified rule provides insights into sparsification and motivates the investigation into a consistent alternative that *can* meet computational efficiency requirements.

3.3 Discussion on Overconfidence

An important consequence of the SEIF sparsification algorithm is that the resulting approximation to the SLAM posterior significantly underestimates the uncertainty in the state estimate. In this section, we show that this inconsistency is a natural consequence of imposing conditional independence between the robot pose and the $\mathbf{m}^0$ subset of the map. To illustrate this effect, consider a general three state Gaussian distribution parametrized in the standard form (3.12a) and information form (3.12b).

$$
\begin{aligned}
p\left(a,b,c\right) &= \mathcal{N}\left(\begin{bmatrix}\mu_a\\ \mu_b\\ \mu_c\end{bmatrix}, \begin{bmatrix}\sigma_a^2 & \rho_{ab}\sigma_a\sigma_b & \rho_{ac}\sigma_a\sigma_c\\ \rho_{ab}\sigma_a\sigma_b & \sigma_b^2 & \rho_{bc}\sigma_b\sigma_c\\ \rho_{ac}\sigma_a\sigma_c & \rho_{bc}\sigma_b\sigma_c & \sigma_c^2\end{bmatrix}\right) && \text{(3.12a)}\\
&= \mathcal{N}^{-1}\left(\begin{bmatrix}\eta_a\\ \eta_b\\ \eta_c\end{bmatrix}, \begin{bmatrix}\lambda_{aa} & \lambda_{ab} & \lambda_{ac}\\ \lambda_{ab} & \lambda_{bb} & \lambda_{bc}\\ \lambda_{ac} & \lambda_{bc} & \lambda_{cc}\end{bmatrix}\right) && \text{(3.12b)}
\end{aligned}
$$

We would like to sparsify the canonical parametrization by forcing a and b to be conditionally independent given c:

$$
p\left(a,b,c\right) = p\left(a,b \mid c\right)p\left(c\right) \xrightarrow{\text{approx.}} \tilde{p}\left(a,b,c\right) = p\left(a \mid c\right)p\left(b \mid c\right)p\left(c\right).
$$

Recalling the discussion in Section 2.4.1, the approximation is implemented in the canonical form by setting $\lambda_{ab} = 0$. In the standard form, this is equivalent to treating a and b as being uncorrelated in $p\left(a,b \mid c\right)$. The resulting approximation then follows as

$$
\tilde{p}\big(a,b,c\big) = \mathcal{N}\left(\begin{bmatrix}\mu_a\\ \mu_b\\ \mu_c\end{bmatrix}, \begin{bmatrix}\sigma_a^2 & \rho_{ac}\rho_{bc}\sigma_a\sigma_b & \rho_{ac}\sigma_a\sigma_c\\ \rho_{ac}\rho_{bc}\sigma_a\sigma_b & \sigma_b^2 & \rho_{bc}\sigma_b\sigma_c\\ \rho_{ac}\sigma_a\sigma_c & \rho_{bc}\sigma_b\sigma_c & \sigma_c^2\end{bmatrix}\right). \qquad (3.13)
$$

In order for the approximation to be consistent, it is necessary and sufficient that the resulting covariance matrix obey the inequality,

$$
\tilde{\Sigma} - \Sigma = \begin{bmatrix}0 & \left(\rho_{ac}\rho_{bc} - \rho_{ab}\right)\sigma_a\sigma_b & 0\\ \left(\rho_{ac}\rho_{bc} - \rho_{ab}\right)\sigma_a\sigma_b & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \geq 0. \qquad (3.14)
$$

A necessary condition for (3.14) to hold is that the determinant of the upper-left 2×2 sub-block be non-negative [48]. Clearly, this is not the case for every $\rho_{ac}\rho_{bc} \neq \rho_{ab}$. Extending this insight to the SEIF sparsification strategy sheds some light on why enforcing the conditional independence between the robot pose and the $\mathbf{m}^0$ landmarks leads to overconfident state estimates. This helps to explain the empirical results that demonstrate that the modified rule yields a posterior that, while nearly identical to the original, remains slightly inconsistent.

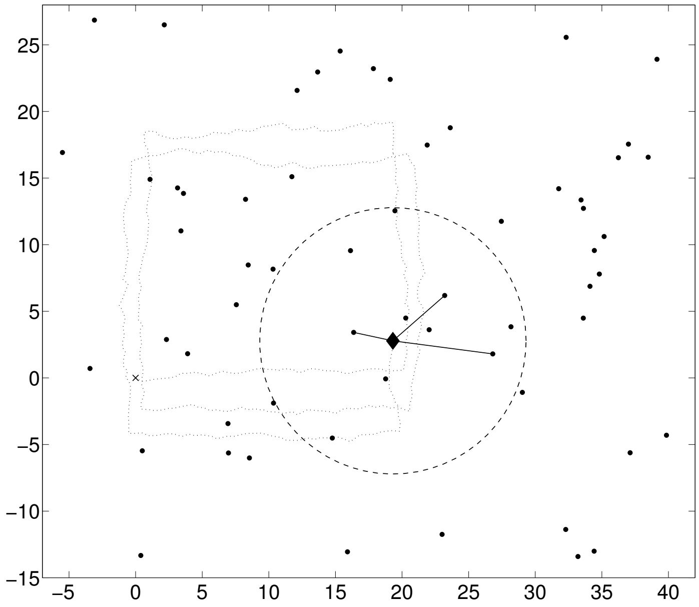


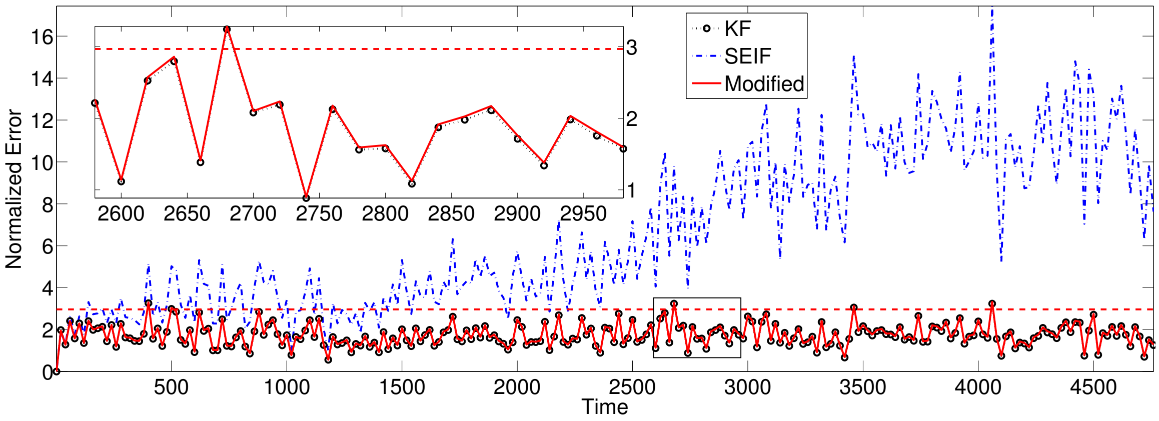
Figure 3-1: A plot of the LG simulation environment. The robot (denoted by the diamond marker) begins at the origin and travels in a series of counterclockwise loops within an environment that consists of 60 point features. The vehicle measures the relative position of a limited number of features that lie within the sensor's FOV, as indicated by the circle.

3.4 Experimental Results

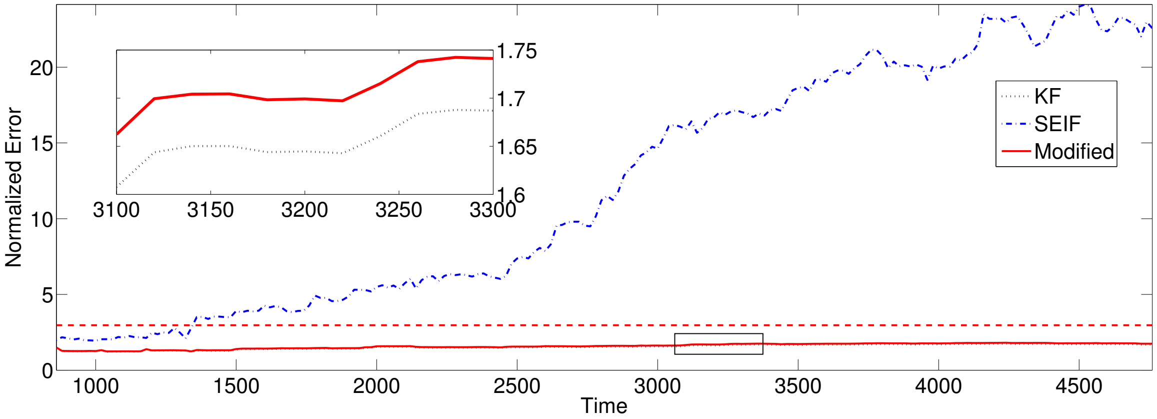
In this section, we experimentally investigate the two sparsification strategies and focus on the implications of approximating conditional independence to achieve sparsity. We compare the two sparsified information filters to the standard Kalman Filter (KF) formulations in the context of two different implementations. The first scenario considers a controlled linear Gaussian (LG) SLAM simulation for which the KF is the optimal Bayesian estimator. The KF serves as a benchmark against which to compare the consequences of the different sparsification strategies. Next, we discuss the performance of the different filters on a real-world nonlinear dataset to better understand their performance in practice.

3.4.1 Linear Gaussian Simulation

In an effort to better understand the theoretical consequences of enforcing sparsity in information filters, we first study the effects of applying the SEIF and modified rule to a synthetic dataset. In this example, the vehicle travels in a series of counterclockwise loops in a roughly 45 × 45 unit environment. Shown in Figure 3-1, the environment consists of 60 uniformly distributed point features. The vehicle observes the relative position of a bounded number of features within a limited FOV. The robot motion is purely translational and evolves according to a linear, constant-velocity model that is corrupted by additive white Gaussian noise. Similarly, the landmark observation data is also subject to additive white Gaussian noise. Appendix A.1 presents additional



(a) Global Vehicle NEES



(b) Global Feature NEES

Figure 3-2: Plots of the *global* normalized estimation error squared (NEES) for the (a) vehicle and (b) one of the features as estimated based upon a series of LG Monte Carlo simulations. The global errors are computed by comparing the direct filter estimates to the ground truth and provide a measure of global consistency for the two sparsification routines. The horizontal line signifies the the 97.5% chi-square upper bound. In the case of both the vehicle and the map, the SEIF sparsification rule induces significant overconfidence while the modified rule better approximates the true distribution.

details concerning the simulation.

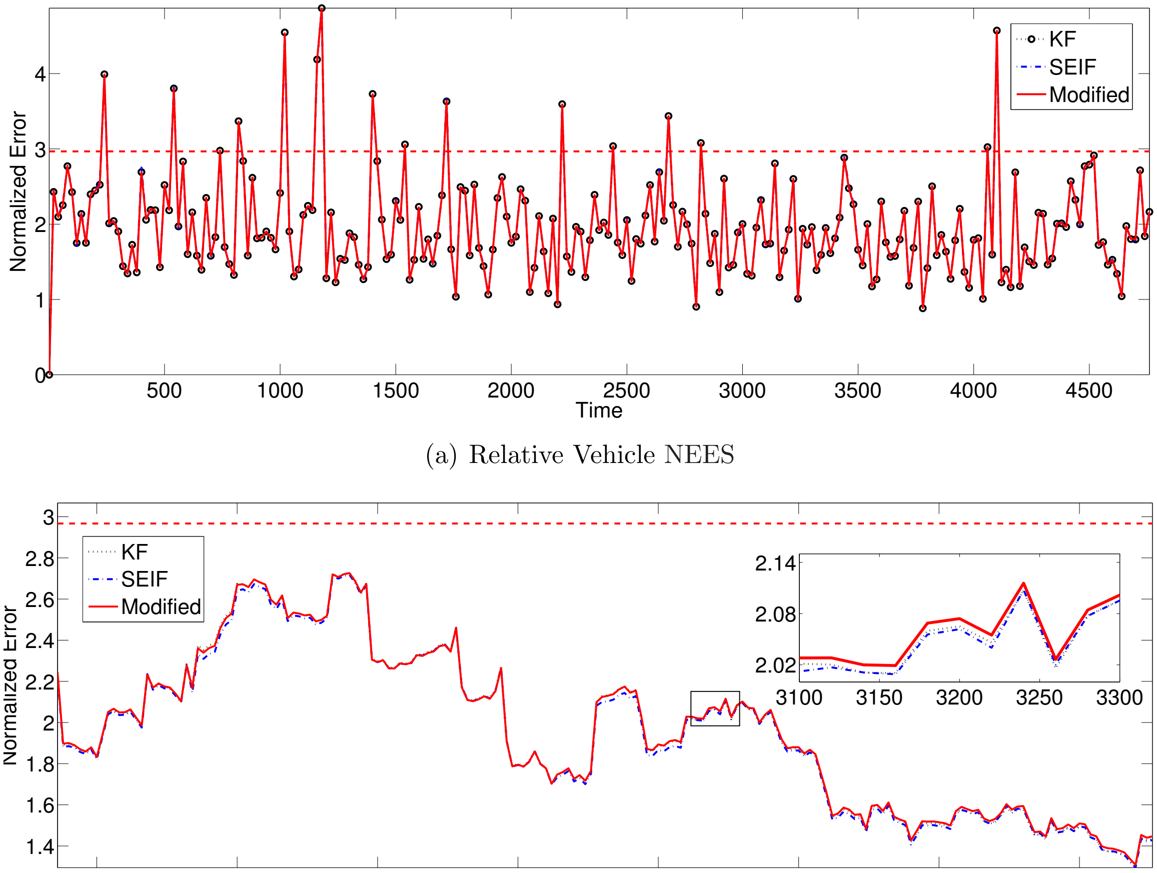
We implement a separate information filter for the SEIF and modified rule and use their corresponding sparsification routines to maintain a bound of $\Gamma_a = 6$ active features (10% of the total number of features). Additionally, we apply the standard Kalman Filter (KF) that, by the linear Gaussian (LG) nature of the simulation, is the optimal Bayesian estimator. Aside from the different sparsification routines, each estimator is otherwise identical.

We analyze the performance of the different sparsification routines based upon a series of Monte Carlo LG simulations. We compare the consistency of the resulting posteriors relative to the true distribution based the normalized estimation error squared (NEES) [5]. As one test of filter consistency, the NEES jointly measures estimator bias, along with the extent of agreement between the estimation error and a filter's corresponding confidence. We evaluate the NEES error based upon a pair of error metrics, the first of which relates the ground truth to the direct output of the filters and provides a measure of *global* error. Figure 3-2(a) compares the global NEES error for the KF, SEIF, and modified rule vehicle position estimates. Similarly, Figure 3-2(b) presents the normalized global errors attributed with a single feature, which represents the typical performance for the rest of the map estimates. The horizontal threshold corresponds to the the 97.5% upper bound for the chi-square test and serves as reference for assessing the consistency of the different filters. Looking at the vehicle pose and landmark scores, the modified rule yields errors that are nearly identical to those of the KF, both with regards to magnitude, as well as behavior over time. In contrast, the SEIF induces global errors that are noticeably larger than the NEES score associated with the true distribution. This implies that the SEIF sparsification routine produces an approximation to the SLAM posterior that is inconsistent with the true distribution. The modified rule better approximates the actual posterior but in agreement with our discussion on overconfidence in Section 3.3, remains inconsistent.

The second NEES error metric considers the state estimate as expressed relative to the first feature that was observed, $\mathbf{m}_1$, via the compounding operation: $\boldsymbol{\xi}_{m_1 i} = \ominus \mathbf{m}_1 \oplus \boldsymbol{\xi}_i$ [121].[3] This metric provides a measure of the *relative* error associated with the filter estimates. Figure 3-3(a) depicts the relative NEES error associated with the vehicle pose for the three filters. In Figure 3-3(b), we plot the relative score for the same representative feature that we employed for the global error. The NEES errors attributed with the modified rule are again nearly indistinguishable from those of the KF. Interestingly, unlike the global estimate errors, the relative SEIF NEES scores are similarly close to the KF errors. This discrepancy suggests that, while the SEIF sparsification routine induces an inconsistent global posterior, it preserves the relative consistency of the state estimates.

The NEES score measures the mean estimation error normalized by uncertainty as an evaluation of an estimator's consistency. We gain further insight into the consequences of sparsification by looking directly at the uncertainty estimates for the two information filters relative to those of the actual posterior. Figure 3-4 depicts a histogram over the map uncertainties that result from the modified rule and SEIF relative to those of the KF. More specifically, we convert the two canonical filter parametrizations to the standard form by inverting the information matrices. We then compute the log of the ratio between the determinants of each feature's covariance matrix sub-block and the covariance submatrix from the true distribution as maintained by the KF. This metric describes the confidence associated with the two sparsified filter estimates: positive values reflect a conservative belief while negative values signify overconfidence. The histogram reveals that the two sparsified uncertainty measures are overconfident with respect to those of the standard KF and, in turn, are inconsistent with the true estimation error. This agrees with our earlier

[3] The compounding operation transforms the state from one coordinate frame to another. We also refer to this process as root-shifting.



(a) Relative Vehicle NEES

(b) Relative Feature NEES

Figure 3-3: The *relative* normalized estimation error squared (NEES) for the (a) vehicle and (b) a representative feature based upon the same series of Monte Carlo simulations that we use to calculate the global error. We compute the relative error by transforming the original random vector with respect to the first feature instantiated in the map: $\boldsymbol{\xi}_{m_1 i} = \ominus \mathbf{m}_1 \oplus \boldsymbol{\xi}_i$. Both plots include a horizontal line that denotes the 97.5% chi-square upper bound. The relative map error for both the SEIF and modified rule is nearly identical to that of the KF, suggesting that the SEIF preserves the relative consistency of the state estimates.

discussion in Section 3.3 on the consequences of approximating conditional independence to achieve sparsity. However, the SEIF sparsification strategy induces a level of overconfidence that is noticeably greater than that of the modified rule. Thus, the high global NEES scores for the SEIF are not so much a consequence of error in the vehicle and map estimates as they are of the overconfidence of the SEIF posterior.

Much like we have seen with the NEES error analysis, the SEIF yields relative uncertainty estimates that are much better behaved. In this case, we again transform the state relative to the first feature added to the map the compounding operation. Note that in the process of expressing the map relative to the first feature, the original world origin is now included as a state element. We compute the confidence

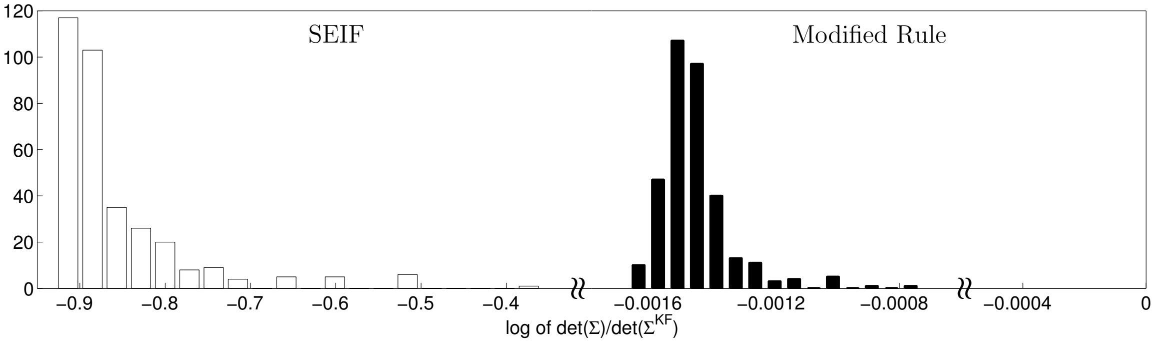


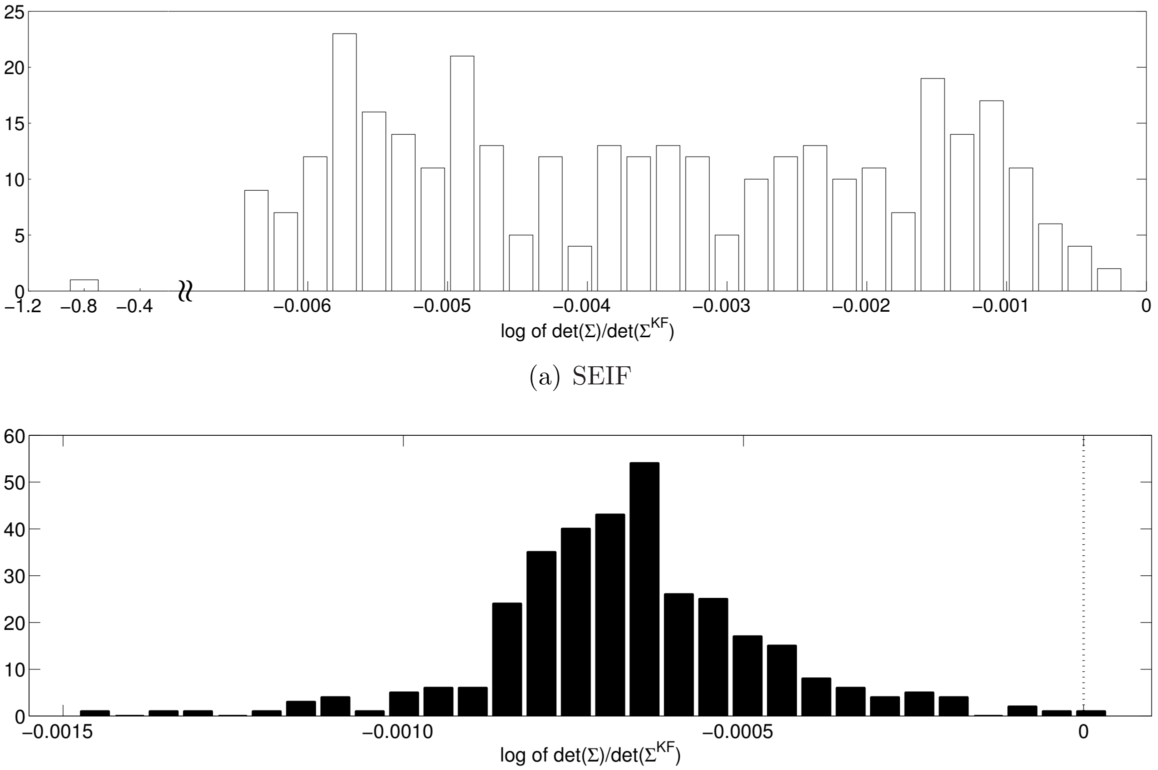
Figure 3-4: Histogram for the LG simulation that depicts the global uncertainties that result from the SEIF (left) and modified rule (right) sparsification strategies as compared to those of the KF. We compute the relative uncertainty for each feature as the log of the ratio between the determinant of the information filter covariance sub-block and the corresponding determinant from the actual distribution as maintained by the KF. Positive values correspond to conservative error estimates while negative log ratios indicate overconfidence. Both sparsification routines yield overconfident map estimates, though the inconsistency of the SEIF is more pronounced.

measures for the SEIF and modified rule relative to the KF based upon covariances associated with the root-shifted state, as before. Figures 3-5(b) and 3-5(a) plot the histograms associated with the modified rule and SEIF sparsification strategies, respectively. Unlike the global (nominal) distribution, the SEIF uncertainty estimates for the relative feature positions are closer to the values from the actual distribution. The one exception is the estimate for the former world origin as expressed in the relative map, which remains overconfident as a result of the global inconsistency of the SEIF. Meanwhile, the modified rule remains slightly overconfident in the relative estimates with confidence levels that are more similar to those of the underlying Gaussian.

The effect of sparsification on the covariance estimates is in-line with what is observed with the normalized errors. Though there is little difference between the three sets of feature position estimates, the normalized errors for the global SEIF map are larger due to the higher confidence attributed to the estimates. In the case of root-shifting the state, the histograms in Figure 3-5 reveal a negligible difference between the relative uncertainty estimates associated with the three filters. Consequently, the uncertainty-based normalization has similar effects on each filter's feature position errors.

3.4.2 Experimental Validation

Simulations are helpful in investigating our findings without having to take into consideration the effects of linearization. However, real-world SLAM applications typically involve nonlinear vehicle motion and perception models, and include noise that is not truly Gaussian. For that reason, we analyze the effects of the two sparsification strategies on a typical nonlinear dataset. As we show, the SEIF and modified rule



(a) SEIF

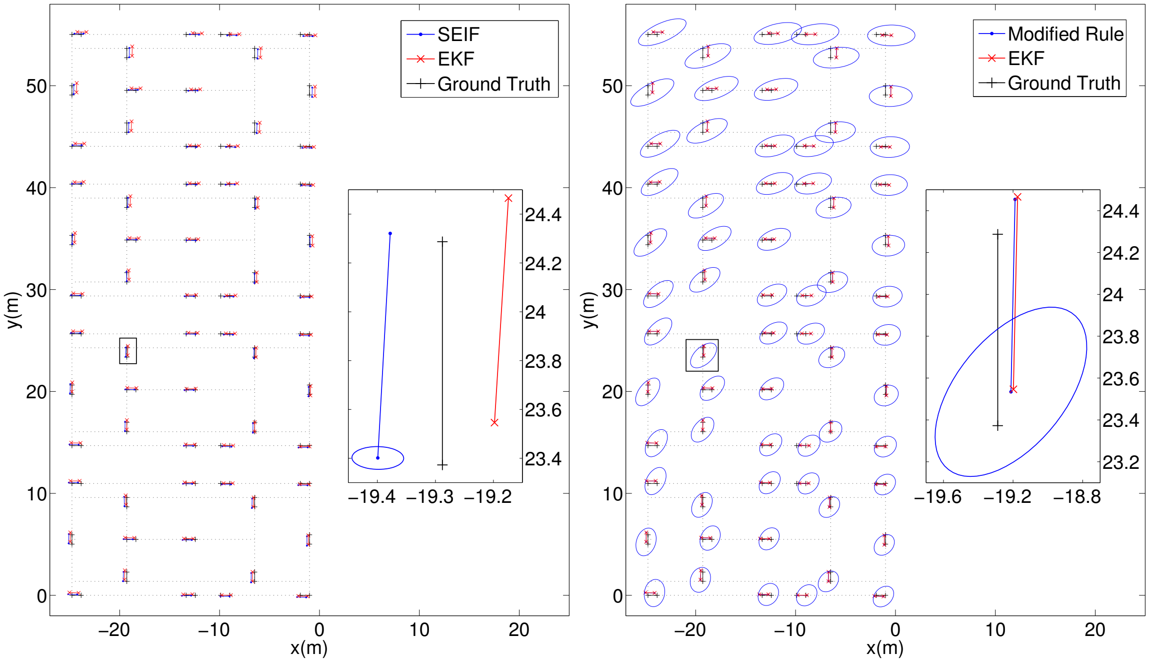
(b) Modified Rule

Figure 3-5: Histograms that show the uncertainties associated with the relative (a) SEIF and (b) modified rule map estimates relative to the baseline KF. We compute the uncertainty ratios based upon the relative covariance estimates that follow from root-shifting the state to the first feature added to the map. Unlike the global estimates, the SEIF is only slightly overconfident and performs similarly to the modified rule. The one outlier in the SEIF histogram corresponds to the representation for the original world origin in the root-shifted reference frame and is a consequence of the inconsistent global representation.

yield posteriors with the same characteristics as those of the linear Gaussian (LG) simulations.

In our experiment, we operated an iRobot B21r wheeled robot in a gymnasium consisting of four adjacent tennis courts. A set of 64 track hurdles were positioned at known locations on the court baselines, which provide a convenient ground truth for the experiment. Figure A-1 within Appendix A presents a photograph of the environment. The vehicle recorded observations of the the relative position of the legs of neighboring hurdles with a SICK laser range finder as we drove it in a series of loops. Wheel encoders measured the vehicle's forward velocity and rate of rotation, which we employ in the time projection step for each filter.

We independently implement two information filters, one that employs the SEIF sparsification routine and a second filter that uses the modified rule to maintain a limit of $\Gamma_a = 10$ active features. As a basis for comparison, we also apply a standard

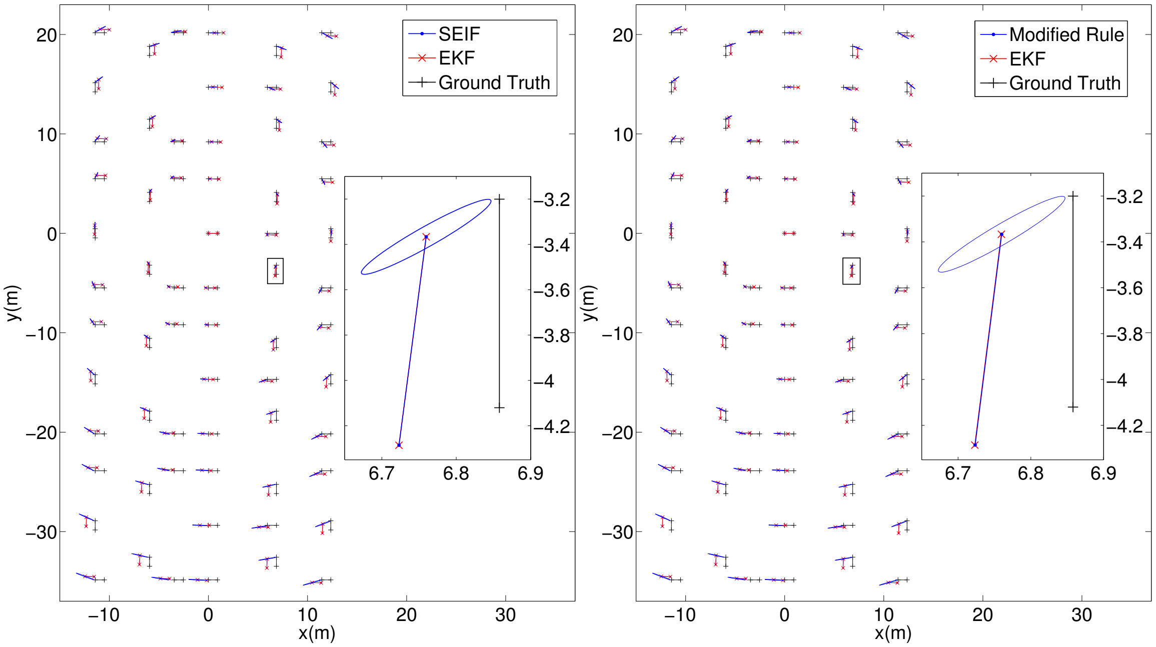


(a) SEIF (b) Modified Rule

Figure 3-6: The final global maps for the (a) SEIF and (b) modified rule, along with the three-sigma uncertainty ellipses. We compare each to the map generated with the standard EKF, as well as the manually-measured ground truth hurdle positions. The SEIF maintains global feature estimates that are significantly overconfident as the uncertainty bounds do not capture the ground truth or the EKF estimates. The modified rule, meanwhile, yields estimates for absolute feature pose and uncertainty which are nearly identical to those of the EKF.

EKF. We treat each hurdle as a single feature that we interpret as a 2D coordinate frame. The model considers one of the two hurdle legs, which we refer to as the "base" leg, to be the origin of this frame and defines the positive x-axis in the direction of the second leg. Features are then parametrized by the translation and rotation of this coordinate frame. Each filter employs a kinematic model for the vehicle motion and treats the forward velocity and rotation rates as control inputs. The measurement model adapts the laser range and bearing observations into a measure of the position and orientation of the hurdle reference frame with respect to the vehicle's body-fixed frame. We solve the data association problem independently in order to ensure that the correspondences are identical for all three filters. For a more detailed explanation of the experiment and the filter implementation, refer to Appendix A.2.

We first consider the posterior over the global state representation that results from the different sparsification routines. Figure 3-6 presents the final global map estimates for the SEIF and modified rule together with the EKF map and the ground truth hurdle positions. The ellipses denote the three-sigma confidence intervals associated with the estimate for the base position for each hurdle. Much like the LG simulation, the final SEIF map estimates exhibit a distinctive degree of overconfi-



(a) SEIF (b) Modified Rule

Figure 3-7: A comparison of the relative maps estimates that result from the (a) SEIF and (b) modified rule, along with the three-sigma uncertainty ellipses. We compute the relative map by expressing the nominal state in the reference frame associated with the first hurdle added to the map. Both the SEIF and modified rule sparsification strategies yield estimates for the relative relationship between features that are nearly identical to those of the EKF.

dence. We see in the inset plot in Figure 3-6(a) that the SEIF uncertainty estimates are overly tight and do not capture either the EKF position estimates or the ground truth. Empirically, this behavior supports the belief that the SEIF sparsification strategy yields global map estimates that are inconsistent. In contrast, the confidence intervals associated with the modified rule are much larger and account for the ground truth and EKF positions. Qualitatively, we also see that the modified rule yields estimates for the feature position and orientation that better approximate the EKF estimates. While the modified rule produces a posterior that remains overconfident with respect to the EKF, it maintains a distribution that better approximates that of the EKF.

As a study of the relative map estimate structure, we transform the state into the reference frame associated with the first hurdle added to the map. The result agrees with the LG analysis in that the quality of the SEIF estimates improves significantly when we consider the relative map relationships. We plot the relative maps for the SEIF and modified rule in Figure 3-7 alongside the root-shifted EKF estimates and the ground truth. The ellipses again denote the three-sigma uncertainty bounds for the two information filters. The SEIF's relative feature pose and uncertainty estimates agree more closely with the ground truth and EKF estimates and are similar to those of the modified rule. The SEIF sparsification strategy seems to preserve the relative consistency of the feature estimates, not only in a controlled simulation, but also with this real-world experiment.

3.5 The Argument for Consistent Sparsification

In their presentation of the Sparse Extended Information Filter (SEIF), Thrun and colleagues [130] provide novel insights into the benefits of a sparse canonical parametrization of the feature-based SLAM Gaussian. Their insight gives rise to the SEIF as a possible solution to the problem of scalability that remains an open issue in SLAM. A delicate issue with the SEIF algorithm is the method by which it enforces the sparsity of a canonical parametrization of the feature-based SLAM posterior that is naturally populated. We have taken a close look at the SEIF sparsification strategy, which controls the population of the information matrix by approximating the conditional independence between the vehicle pose and much of the map. We revealed inconsistencies with the SEIF approximation to conditional independence and proposed the modified rule as a more precise strategy for enforcing conditional independence. A controlled, linear Gaussian (LG) simulation reveals that the SEIF maintains an overconfident posterior and confirms our belief that the SEIF sparsification strategy leads to inconsistent global estimates. Results from a real-world, nonlinear dataset provide empirical evidence that support our claim. Alternatively, the modified rule yields an approximation to the SLAM posterior that is nearly identical to the nominal Gaussian distribution. Unfortunately, the accuracy comes at a significant computational cost as the modified rule is cubic in the size of the map and, therefore, not scalable.

Despite the performance of the modified rule, it too results in a posterior that is overconfident. We argue in Section 3.3 that this inconsistency is implicit in any sparsification routine that approximates conditional independence. In the next chapter, we describe an alternative sparsification strategy that does not rely on such an approximation. Instead, our sparse information filter actively controls the formation of links in the Gaussian Markov random field (GMRF) in a manner that is computationally efficient and preserves the consistency of the distribution.

Chapter 4

Exactly Sparse Extended Information Filters

At this point, we have shown that an open problem in SLAM is that of scaling to large environments. Previous chapters discussed several promising algorithms that address this problem, ranging from submap approaches that break the world into more manageable chunks to pose graph formulations that exploit the efficiency of iterative optimization techniques. Most relevant to the work of this thesis are techniques based upon the canonical form of a Gaussian distribution that exploit a sparse parametrization to achieve computational efficiency. As such, we have devoted much attention to the SEIF algorithm by Thrun and colleagues [130], which has helped to lay the groundwork for information filter SLAM formulations, including the algorithm that we derive in this thesis. The previous chapter presented a detailed analysis of SEIF sparsification process whereby the filter controls the population of the information matrix. We have shown that, as a direct consequence of the sparsification strategy as implemented by the SEIF, the filter maintains overconfident estimates for the vehicle pose and map.

The thesis presents the Exactly Sparse Extended Information Filter (ESEIF), an alternative form of the EIF that achieves the efficiency benefits of a sparse posterior parametrization without sacrificing consistency. This chapter describes the our estimator as a feature-based SLAM algorithm. We derive the filter in terms of a novel sparsification strategy that prevents the population of the information matrix by proactively controlling the creation of links in the Gaussian Markov random field (GMRF). The ESEIF then tracks an alternative form of the SLAM posterior that is inherently sparse. Consequently, the filter avoids the need to force what are naturally non-zero elements of the information matrix to zero. By adopting an efficient, approximate inference strategy, the natural sparsity of the Gaussian distribution enables the ESEIF to perform SLAM filtering at computational cost that is linear in the size of the map. We show both in a controlled simulation, as well as on a pair of real-world datasets, that the ESEIF maintains uncertainty estimates that are conservative relative to the gold standard EKF.

4.1 Sparsification Strategy

We begin with high-level description of the ESEIF sparsification strategy. We present the intuition behind our approach, framing it in the context of the fundamental aspects of Bayesian SLAM filtering. In this way, our goal is to ground the ESEIF sparsification methodology in the natural operation of SLAM filtering.

4.1.1 Conditional Dependence of Filter Estimates

As part of our earlier discussion in Chapter 2, we described SLAM filtering in terms of three fundamental processes: augmentation, roll-up, and measurement updates. Here, we again use this interpretation of filtering to motivate the ESEIF sparsification strategy.

Consider the Gaussian filtering process at time t immediately following a time prediction step. The current robot pose is conditionally dependent upon a subset of the map per the distribution, $p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^{t-1}, \mathbf{u}^t\right)$, and makes a set of feature observations, $\mathbf{z}_t$. We incorporate this evidence into the distribution via a measurement update step and, in the case of new landmarks, add them to the map. The latter process of growing the map instantiates relatively strong links between the robot state and the new features. Similarly, the measurement update strengthens the existing shared information between the vehicle and the re-observed landmarks. Despite the introduction of new information, the density of the information matrix is likely to decrease on account of the conditional independence between the new and old given the robot pose. More importantly, though, the size of the active map will either stay the same or increase in the case that new landmarks are observed.

Next, we project the robot state forward in time, first augmenting the posterior distribution, $p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right)$, to include the predicted pose at time $t+1$. Assuming that the process model is first-order Markov, the new pose is conditionally independent of the map given the previous pose and recent control input. As we have mentioned in earlier chapters, this conditional independence implies that we add a single link in the GMRF between the previous and current robot states, leaving the rest of the graph untouched. In the specific case of our canonical Gaussian approximation to the distribution, we add an additional block row and column to the information matrix that is zero everywhere in the off-diagonal aside for entries between the two poses. At this point, the computational cost to tracking the additional pose is negligible as we have not increased the density of the information matrix.

The final component of the time projection step is to marginalize the previous robot state from the distribution, $p\left(\mathbf{x}_{t+1}, \mathbf{x}_t, \mathrm{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right)$, via the roll-up step. In Chapter 2 we discussed, in detail, the consequences of marginalization on the structure of the Gaussian SLAM distribution, namely the fill-in of the information matrix. The Schur complement (2.25a) instantiates shared information among each feature within the active map, equivalently adding links to the undirected graphical model. Active landmarks remain active as the information that they share with the previous pose is, effectively, transferred to the current robot state. While these links weaken due to the process noise, they persist over the course of subsequent prediction steps, resulting in an active map that never decreases in size.

4.1.2 Actively Control Link Formation

The factor most critical to the population of the information matrix is the size of the active map. While the measurement update step does not immediately contribute to the density of the information matrix, it generally increases the number of active features. The marginalization component of the subsequent time prediction step will then create a maximal clique among what is now a larger set of active landmarks, corresponding to an increase in the density of the information matrix. While edges between the robot pose and map in the GMRF may weaken over time, they will not vanish, resulting in an ever-growing active map and the inherent population of the information matrix. The role of the active map suggests that we can prevent the fill-in of the information matrix by controlling the number of landmarks that are made active as a consequence of measurement updates. One strategy is to actively restrict the number of observations that the filter incorporates in an update step to a finite number of "optimal" measurements. Optimality may be defined, for example, by a reward function that balances information gain as defined by relative entropy [84] with the size of the resulting active map. With this regularization penalty, however, it would be difficult to formulate the reward measure such that it permits exploration, which inherently implies adding new links between the robot pose and map. An additional problem with such an approach is that it disregards measurement information.

We propose an alternative strategy that actively prevents the fill-in of the information matrix while retaining all map observation data. In the same spirit as the aforementioned approach, we control the incorporation of measurement information and, in turn, the addition of links to the GMRF, in order to maintain a sparse information matrix. The ESEIF sparsification process takes the form of a modified measurement update step that we implement periodically in response to the size of the active map. Our strategy partitions the available map observations into two sets. The first includes any observations of passive features (including new landmarks) as well as a limited number of active feature measurements. The filter incorporates this data as in the standard EIF to update the filter and grow the map. This update strengthens links in the graph and increases the number of active landmarks. Next, we marginalize the SLAM distribution over the robot pose, effectively "kidnapping" the robot from the map. As in the roll-up stage of the time projection step, this induces a maximal clique among the set of active features. The ESEIF then relocates the vehicle within the map based upon the remaining set of measurements. The active map now consists only of the landmarks that were used for relocalization and the subsequent time projection produces a limited amount of fill-in.

The fundamental advantage of this sparsification strategy is that it prevents the unbounded growth of the active map. By *periodically* kidnapping the robot, the ESEIF actively controls the population of the information matrix induced by the roll-up step, without ignoring measurement data. Instead, the ESEIF sacrifices some odometry information by relocating the vehicle within the map based only on a subset of measurements. As we describe throughout the remainder of the chapter, this is analogous to tracking an alternative form of the SLAM posterior that is exactly sparse.

4.2 Mechanics of the ESEIF

The Exactly Sparse Extended Information Filter (ESEIF) introduces a novel sparsification strategy that provides for an exactly sparse parametrization of the SLAM posterior. In turn, the ESEIF is able to efficiently track estimates for the robot pose and map that are consistent by exploiting the advantages of sparse SLAM information filters. In this section, we discuss the mechanics of the ESEIF, first explaining the sparsification step in detail. We then briefly describe our implementation of the basic time projection and measurement update steps. Subsequently, we explain an approach for approximate inference on the canonical distribution, which efficiently computes the mean and uncertainty estimates for the robot pose and map needed for linearization and data association. Throughout the section, we elaborate on the computational costs associated with the filtering process.

4.2.1 Sparsification Process: Maintaining Exact Sparsity

The general premise of ESEIF sparsification is straightforward: rather than deliberately breaking links between the robot and map, we maintain sparsity by controlling their initial formation. More specifically, the ESEIF manages the number of active landmarks by first marginalizing out the vehicle pose, essentially “kidnapping” the robot [128]. The algorithm subsequently relocalizes the vehicle within the map based upon new observations to a set of known landmarks. The set of features that were originally active have been made passive and the set of landmarks used for relocalization form the new active map.

The ESEIF sparsification algorithm takes place as needed to maintain the Γ_a bound on the number of active landmarks. Outlined in Algorithm 1, sparsification takes the form of a variation on the measurement update step. For a more detailed description, we consider a situation that would give rise to the GMRF depicted in Figure 4-1. At time t, suppose that the robot makes four observations of the environment, $\mathcal{Z}_t = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_5\}$, three of active features and one of a passive landmark:

$$\begin{aligned} &\text{Active:} \quad \mathbf{z}_1 = \mathbf{h}(\mathbf{x}_t, \mathbf{m}_1),\ \mathbf{z}_2 = \mathbf{h}(\mathbf{x}_t, \mathbf{m}_2),\ \mathbf{z}_5 = \mathbf{h}(\mathbf{x}_t, \mathbf{m}_5) \\ &\text{Passive:} \quad \mathbf{z}_3 = \mathbf{h}(\mathbf{x}_t, \mathbf{m}_3). \end{aligned}$$

Updating the current distribution, $p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^{t-1}, \mathbf{u}^t\right)$, based upon all four measurements would strengthen the off-diagonal entries in the information matrix pairing the robot with the three observed active features, $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_5$. Additionally, the update would create a link to the passive landmark, $\mathbf{m}_3$, the end result being the information matrix and corresponding graphical model shown in the left-hand side of Figure 4-1. Suppose that activating $\mathbf{m}_3$ would violate the Γ_a bound. The subsequent time prediction process would then induce shared information (i.e. a maximal clique) among

```
Time Projection: $\left(\boldsymbol{\eta}_{t-1}^{+}, \Lambda_{t-1}^{+}\right) \longrightarrow \left(\boldsymbol{\eta}_{t}^{-}, \Lambda_{t}^{-}\right)$ ;
Measurement Update $\left(\mathbf{z}_t = \left\{\mathbf{z}_{active}, \mathbf{z}_{passive}\right\}\right)$:
  if $\mathrm{N}_{active} + n\left(\mathbf{z}_{passive}\right) \leq \Gamma_a$ then
      Standard update: $\left(\boldsymbol{\eta}_{t}^{-}, \Lambda_{t}^{-}\right) \xrightarrow{(4.10)} \left(\boldsymbol{\eta}_{t}^{+}, \Lambda_{t}^{+}\right)$ ;
      $\mathrm{N}_{\text{active}} = \mathrm{N}_{\text{active}} + n\left(\mathbf{z}_{\text{passive}}\right)$;
  else
      Partition $\mathbf{z}_t = \left\{\mathbf{z}_\alpha, \mathbf{z}_\beta\right\}$ s.t. $n\left(\mathbf{z}_\beta\right) \leq \Gamma_a$ ;
      (i) Update using $\mathbf{z}_\alpha$: $\left(\boldsymbol{\eta}_{t}^{-}, \Lambda_{t}^{-}\right) \xrightarrow{(4.1)} \left(\bar{\boldsymbol{\eta}}_{t}^{-}, \bar{\Lambda}_{t}^{-}\right)$ ;
      (ii) Marginalize over the robot pose: $\left(\bar{\boldsymbol{\eta}}_{t}^{-}, \bar{\Lambda}_{t}^{-}\right) \xrightarrow{(4.3)} \left(\check{\boldsymbol{\eta}}_{t}^{-}, \check{\Lambda}_{t}^{-}\right)$ ;
      (iii) Relocate the robot using $\mathbf{z}_\beta$: $\left(\check{\boldsymbol{\eta}}_{t}^{-}, \check{\Lambda}_{t}^{-}\right) \xrightarrow{(4.6)} \left(\check{\boldsymbol{\eta}}_{t}^{+}, \check{\Lambda}_{t}^{+}\right)$ ;
      $\mathrm{N}_{\text{active}} = n\left(\mathbf{z}_\beta\right)$ ;
  end
```

Algorithm 1: A description of the ESEIF algorithm. Note that $\mathrm{N}_{\text{active}}$ denotes the number of features that are currently active.

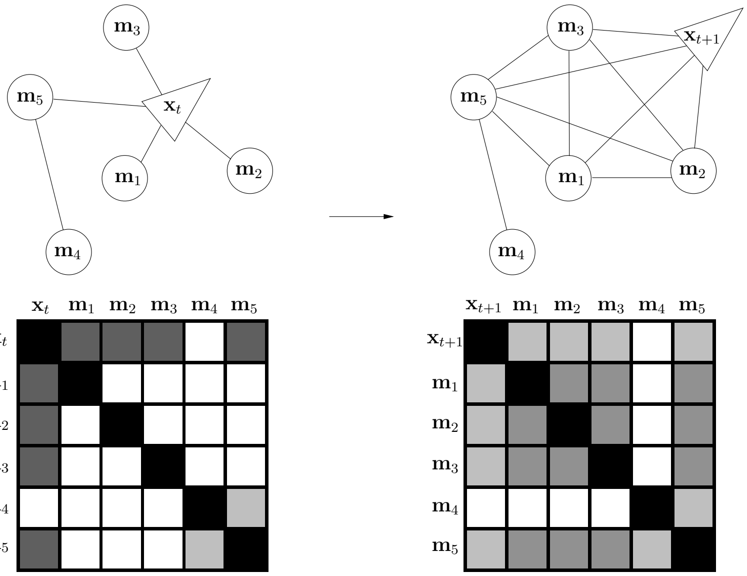

$\mathbf{m}_3$
$\mathbf{m}_5$
$\mathbf{x}_t$
$\mathbf{m}_1$
$\mathbf{m}_2$
$\mathbf{m}_4$
$\mathbf{m}_3$
$\mathbf{x}_{t+1}$
$\mathbf{m}_5$
$\mathbf{m}_1$
$\mathbf{m}_2$
$\mathbf{m}_4$
$\mathbf{x}_t$ $\mathbf{m}_1$ $\mathbf{m}_2$ $\mathbf{m}_3$ $\mathbf{m}_4$ $\mathbf{m}_5$
$\mathbf{x}_{t+1}$ $\mathbf{m}_1$ $\mathbf{m}_2$ $\mathbf{m}_3$ $\mathbf{m}_4$ $\mathbf{m}_5$

Figure 4-1: At time t, the robot observes four features, $\{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_5\}$, three of which are already active, while $\mathbf{m}_3$ is passive. The update strengthens the shared information between vehicle pose and $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_5$ and adds a link to $\mathbf{m}_3$ as we indicate on the left. The next time projection step forms a clique among the robot and these four features and populates the information matrix. The ESEIF sparsification strategy avoids this effect by controlling the number of active landmarks and, in turn, the size of this clique.

the set of active features as we show on the right. The simplest way to avoid this is to disregard the observation of the passive landmark entirely. This approach, though, is not acceptable, since the size of the map that we can build is then dictated by the Γ_a bound. A better option is to proceed with a standard update based upon the complete set of measurements and immediately nullify weaker links by enforcing conditional independence between those features and the robot pose. As we saw in the previous chapter, however, ignoring conditional dependence gives rise to overconfident state estimates. Alternatively, the ESEIF allows us to both incorporate all measurement data and simultaneously maintain the desired degree of sparsity without sacrificing consistency.

In the ESEIF sparsification step, the measurement data is partitioned into two sets, $\mathbf{z}_\alpha$ and $\mathbf{z}_\beta$, where the first set of observations is used to first update the filter and the second is reserved for performing relocalization. Several factors guide the specific measurement allocation, including the number and quality of measurements necessary for relocalization. Additionally, as the landmarks associated with $\mathbf{z}_\beta$ are made active as a result of sparsification, the choice for this set defines the size of the active map and, in turn, the frequency of sparsification. The more measurements that we reserve for relocalization, the more often that the filter will sparsify the posterior. In the case that the bound on the active map size is larger than the number of measurements, we prefer to relocalize the vehicle based upon as many measurements as possible in order to reduce the resulting pose error. Meanwhile, we first incorporate any observations of new landmarks in the initial measurement update step, whereby we grow the map. Of the four measurements available in our example, group that of the passive feature together with one of the active measurements for the update, $\mathbf{z}_\alpha = \{\mathbf{z}_1, \mathbf{z}_3\}$. The remaining two observations are withheld for relocalization, $\mathbf{z}_\beta = \{\mathbf{z}_2, \mathbf{z}_5\}$. We now describe the two components of sparsification.

Posterior Update

We first perform a Bayesian update to the current distribution, $p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^{t-1}, \mathbf{u}^t\right)$, to incorporate the information provided by the $\mathbf{z}_\alpha$ measurements:

$$p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^{t-1}, \mathbf{u}^t\right) = \mathcal{N}^{-1}\left(\boldsymbol{\xi}_t; \boldsymbol{\eta}_t, \Lambda_t\right) \xrightarrow{\mathbf{z}_\alpha=\{\mathbf{z}_1,\mathbf{z}_3\}} p_1\left(\boldsymbol{\xi}_t \mid \left\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\right\}, \mathbf{u}^t\right) = \mathcal{N}^{-1}\left(\boldsymbol{\xi}_t; \bar{\boldsymbol{\eta}}_t, \bar{\Lambda}_t\right).$$

The $p_1\left(\boldsymbol{\xi}_t \mid \left\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\right\}, \mathbf{u}^t\right)$ posterior follows from the standard update equations (2.21) for the information filter.

$$p_1\left(\boldsymbol{\xi}_t \mid \left\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\right\}, \mathbf{u}^t\right) = \mathcal{N}^{-1}\left(\boldsymbol{\xi}_t; \bar{\boldsymbol{\eta}}_t, \bar{\Lambda}_t\right)$$

$$\bar{\Lambda}_t = \Lambda_t + \mathrm{H}^\top \mathrm{R}^{-1} \mathrm{H} \tag{4.1a}$$

$$\bar{\boldsymbol{\eta}}_t = \boldsymbol{\eta}_t + \mathrm{H}^\top \mathrm{R}^{-1}\left(\mathbf{z}_\alpha - \mathbf{h}\left(\boldsymbol{\mu}_t\right) + \mathrm{H}\boldsymbol{\mu}_t\right) \tag{4.1b}$$

The Jacobian matrix, H, is nonzero only at indices affiliated with the robot pose and the $\mathbf{m}_1$ and $\mathbf{m}_3$ landmarks. As a result, the process strengthens the link between the robot and the active feature, $\mathbf{m}_1$, and creates shared information with $\mathbf{m}_3$, which was

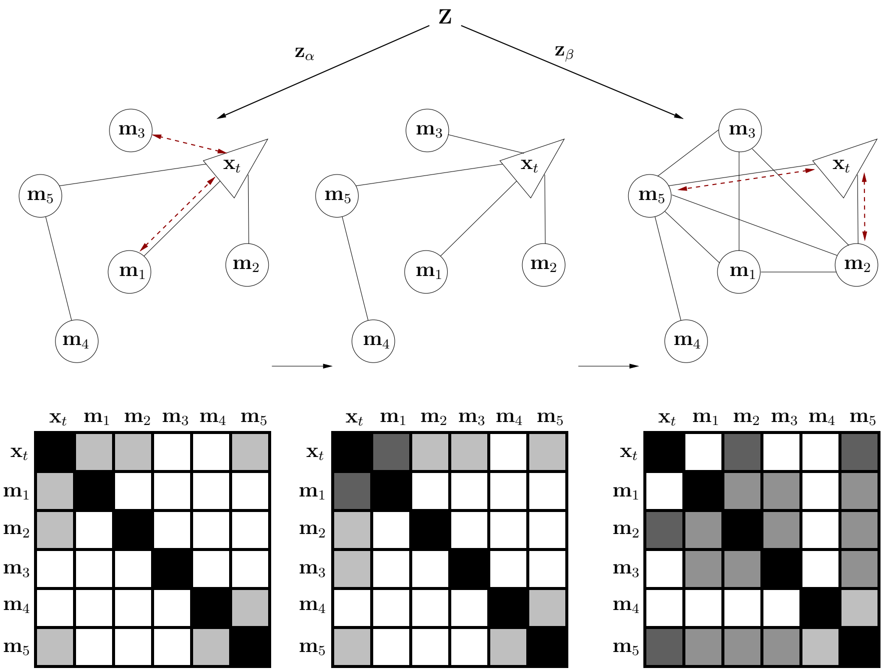


Figure 4-2: A graphical description of the ESEIF sparsification strategy. At time t, the map is comprised of three active features, $\mathbf{m}^+ = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_5\}$, and two passive features, $\mathbf{m}^- = \{\mathbf{m}_3, \mathbf{m}_4\}$, as indicated by the shaded off-diagonal elements in the information matrix. The robot makes three observations of active landmarks, $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_5\}$, and one of a passive feature, $\mathbf{z}_3$. In the first step of the sparsification algorithm, shown in the *left-most* diagram, the ESEIF updates the distribution based upon a subset of the measurements, $\mathbf{z}_\alpha = \{\mathbf{z}_1, \mathbf{z}_3\}$. The result is a stronger constraint between $\mathbf{m}_1$ and the robot, as well as the creation of a link with $\mathbf{m}_3$, which we depict in the *middle* figure. Subsequently, the ESEIF marginalizes out the vehicle pose, leading to connectivity among the active landmarks. The schematic on the *right* demonstrates the final step of sparsification in which the robot is relocated within the map based upon the remaining $\mathbf{z}_\beta = \{\mathbf{z}_2, \mathbf{z}_5\}$ measurements. The result is a joint posterior, $p_{\text{ESEIF}}\big(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\big)$, represented by an exactly sparse information matrix where the size of the active map is controlled.

passive. The middle diagram of Figure 4-2 demonstrates this effect.

As a traditional measurement update step, the computational cost of this component of sparsification is constant-time. In the case that the observation model is nonlinear, linearization requires access to the mean estimate for only the robot pose as well as the $\mathbf{m}_1$ and $\mathbf{m}_3$ landmarks.

Marginalization

Now that a new connection to the vehicle node has been added to the graph, there are too many active features. The ESEIF sparsification routine proceeds to marginalize out the robot pose to achieve the distribution over the map,

$$\begin{aligned} p_2\big(\mathbf{M} \mid \left\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\right\}, \mathbf{u}^t\big) &= \int\limits_{\mathbf{x}_t} p_1\big(\boldsymbol{\xi}_t \mid \left\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\right\}, \mathbf{u}^t\big) d\mathbf{x}_t \\ &= \mathcal{N}^{-1}\big(\mathbf{M}; \check{\boldsymbol{\eta}}_t, \check{\Lambda}_t\big). \end{aligned} \tag{4.2}$$

In order to make the derivation a little clearer, we decompose the canonical expression for $p_1\big(\boldsymbol{\xi}_t \mid \left\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\right\}, \mathbf{u}^t\big)$ into the robot pose and map components,

$$p_1\big(\boldsymbol{\xi}_t \mid \left\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\right\}, \mathbf{u}^t\big) = \mathcal{N}^{-1}\big(\boldsymbol{\xi}_t; \bar{\boldsymbol{\eta}}_t, \bar{\Lambda}_t\big)$$

$$\bar{\boldsymbol{\eta}}_t = \begin{bmatrix} \bar{\boldsymbol{\eta}}_{x_t} \\ \bar{\boldsymbol{\eta}}_M \end{bmatrix} \quad \bar{\Lambda}_t = \begin{bmatrix} \bar{\Lambda}_{x_t x_t} & \bar{\Lambda}_{x_t M} \\ \bar{\Lambda}_{M x_t} & \bar{\Lambda}_{MM} \end{bmatrix}.$$

The information matrix for the marginalized distribution then follows from Table 2.1:

$$p_2\big(\mathbf{M} \mid \left\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\right\}, \mathbf{u}^t\big) = \mathcal{N}^{-1}\big(\mathbf{M}; \check{\boldsymbol{\eta}}_t, \check{\Lambda}_t\big)$$

$$\check{\Lambda}_t = \bar{\Lambda}_{MM} - \bar{\Lambda}_{M x_t} \left(\bar{\Lambda}_{x_t x_t}\right)^{-1} \bar{\Lambda}_{x_t M} \tag{4.3a}$$

$$\check{\boldsymbol{\eta}}_t = \bar{\boldsymbol{\eta}}_M - \bar{\Lambda}_{M x_t} \left(\bar{\Lambda}_{x_t x_t}\right)^{-1} \bar{\boldsymbol{\eta}}_{x_t}. \tag{4.3b}$$

The $\bar{\Lambda}_{M x_t}(\bar{\Lambda}_{x_t x_t})^{-1}\bar{\Lambda}_{x_t M}$ outer product in the Schur complement (4.3a) is zero everywhere except for the entries that pair the active features. Recalling our earlier discussion in Section 2.4.1 on the effects of marginalization for the canonical form, this establishes full connectivity among the active features, as we show in the right-hand side of Figure 4-2. We have essentially transferred the information available in the links between the map and pose to the active landmarks, sacrificing some fill-in of the information matrix. Of course, in contrast to the depiction in the figure, we do not have a representation for the robot pose, which brings us to the final step that we discuss shortly.

The marginalization component of sparsification is computationally efficient. Inverting the robot pose sub-matrix, $\bar{\Lambda}_{x_t x_t} \in \mathbb{R}^{p \times p}$, is a constant-time operation, since p is fixed. The ESEIF then multiplies the inverse by $\bar{\Lambda}_{M x_t} \in \mathbb{R}^{n \times p}$, the sub-block that captures the shared information between the robot and map. With a bound on the number of active landmarks, a limited number of k rows are populated and the matrix product is $\mathcal{O}(kp^2)$. In (4.3a), we then post-multiply by the transpose in $\mathcal{O}(k^2 p)$ time while, in (4.3b) we post-multiply by $\bar{\boldsymbol{\eta}}_{x_t} \in \mathbb{R}^{p \times 1}$, an $\mathcal{O}(kp)$ operation. With the valid assumption that $k \gg p$, the marginalization component of ESEIF sparsification is quadratic in the bounded number of active features and, thus, constant-time.

Relocalization

The sparsification process concludes with the relocalization of the vehicle within the map. We estimate the new robot pose based upon the remaining $\mathbf{z}_\beta$ observations of a set of features that we denote by the random vector $\mathbf{m}_\beta$. This step mimics that of adding landmarks to the map, though, in this case, we define the vehicle pose as a function of these features.

The actual expression for the new pose estimate depends largely on the characteristics of the vehicle's sensor model as well as the nature of the $\mathbf{m}_\beta$ landmarks. For now, we represent this expression in its most general form as a nonlinear function of $\mathbf{m}_\beta$ and the measurement data. We include an additive white Gaussian noise term, $\mathbf{v}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}\big)$, that accounts for model uncertainty and sensor noise, giving rise to the expression in Equation (4.4a). Equation (4.4b) is the first-order linearization over the observed landmarks evaluated at their mean, $\breve{\boldsymbol{\mu}}_{m_\beta}$, from the map distribution (4.2). The Jacobian matrix with respect to the map, G_M, is sparse with nonzero entries only within the columns associated with the $\mathbf{m}_\beta$ landmarks. In turn, (4.4b) requires only the $\breve{\boldsymbol{\mu}}_{m_\beta}$ mean.

$$
\begin{aligned}
\mathbf{x}_t &= \mathbf{g}\big(\mathbf{m}_\beta, \mathbf{z}_\beta\big) + \mathbf{v}_t && \text{(4.4a)}\\
&\approx \mathbf{g}\big(\breve{\boldsymbol{\mu}}_{m_\beta}, \mathbf{z}_\beta\big) + \mathrm{G}_M\big(\mathbf{m} - \breve{\boldsymbol{\mu}}_t\big) + \mathbf{v}_t && \text{(4.4b)}
\end{aligned}
$$

We augment the map state with this new pose, $\boldsymbol{\xi}_t = \big[\mathbf{x}_t^\top \; \mathbf{M}^\top\big]^\top$, and form the joint distribution,

$$
p_{\text{ESEIF}}\big(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\big) = p\left(\mathbf{x}_t \mid \mathbf{m}_\beta, \mathbf{z}_\beta\right) p_2\big(\mathbf{M} \mid \big\{\mathbf{z}^{t-1}, \mathbf{z}_\alpha\big\}, \mathbf{u}^t\big), \tag{4.5}
$$

where the factorization captures the conditional independence between the pose and the remaining map elements. Per the linearization of the relocalized vehicle pose function (4.4b), we approximate the conditional distribution over the pose as Gaussian, $p\left(\mathbf{x}_t \mid \mathbf{m}_\beta, \mathbf{z}_\beta\right) \approx \mathcal{N}\big(\mathbf{x}_t; \mathbf{g}(\breve{\boldsymbol{\mu}}_{m_\beta}, \mathbf{z}_\beta), \mathrm{R}\big)$. The problem of adding the robot pose is fundamentally the same as adding a new feature to the map or augmenting the state as part of the time prediction step (2.24). One can then easily derive the canonical parametrization (4.6) for the joint distribution, $p_{\text{ESEIF}}\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right)$.

$$
p_{\text{ESEIF}}\big(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\big) = \mathcal{N}^{-1}\big(\boldsymbol{\xi}_t; \breve{\boldsymbol{\eta}}_t, \breve{\Lambda}_t\big)
$$

$$
\breve{\Lambda}_t = \begin{bmatrix} \mathrm{R}^{-1} & -\mathrm{R}^{-1}\mathrm{G}_M \\ -\mathrm{G}_M^\top\mathrm{R}^{-1} & \breve{\Lambda}_t + \mathrm{G}_M^\top\mathrm{R}^{-1}\mathrm{G}_M \end{bmatrix} \tag{4.6a}
$$

$$
\breve{\boldsymbol{\eta}}_t = \begin{bmatrix} \mathrm{R}^{-1}\Big(\mathbf{g}\big(\breve{\boldsymbol{\mu}}_{m_\beta}, \mathbf{z}_\beta\big) - \mathrm{G}_M\breve{\boldsymbol{\mu}}_t\Big) \\ \breve{\boldsymbol{\eta}}_t - \mathrm{G}_M^\top\mathrm{R}^{-1}\Big(\mathbf{g}\big(\breve{\boldsymbol{\mu}}_{m_\beta}, \mathbf{z}_\beta\big) - \mathrm{G}_M\breve{\boldsymbol{\mu}}_t\Big) \end{bmatrix} \tag{4.6b}
$$

As a consequence of the sparseness of the Jacobian matrix, G_M, the majority of terms within the $-\mathrm{R}^{-1}\mathrm{G}_M = -\big(\mathrm{G}_M^\top\mathrm{R}^{-1}\big)^\top$ block of the information matrix that link the robot to the map are zero. The landmarks used for relocalization are the only exception, as we show in the right-hand diagram in Figure 4-2 with the robot linked to the $\mathbf{m}_\beta = \{\mathbf{m}_2, \mathbf{m}_5\}$ features but no others. These landmarks now form the active map.

Returning to the general relocalized pose model (4.4), pose estimation depends on the nature of the exteroceptive sensors as well as the map structure. In the case that the measurements are represented by a bijective model, each observation yields a vehicle pose estimate via the inverse measurement function. For example, consider the hurdles dataset that we first described in Section 3.4.2. If we assume that we can resolve the coordinate frame associated with a hurdle observation (i.e. identify the hurdle's base leg), an estimate for the vehicle's position and heading follows from an inverse transformation. Oftentimes, though, the function is not invertible (i.e. injective), such as with a three DOF robot that makes range and bearing measurements to point features in a planar environment. In this case, we need to formulate the pose estimate as a joint function of several observations. Whether or not an observation of a single feature is sufficient to estimate the vehicle state we prefer base the estimate on multiple observations. A "batch" approach to relocalization offers greater robustness to both measurement noise as well as errors that may corrupt individual estimates. However, a larger allocation of measurements to $\mathbf{z}_\beta$ increases the size of the resulting active map and, in turn, affects the frequency of sparsification. We then take both factors into account when partitioning measurement data into the two sets, based upon the desired bound on the number of active landmarks and the nature of the vehicle's sensors.

The ESEIF actively controls the information constraints between the vehicle and the map in a consistent manner, since it does not break (i.e. set to zero) undesired links in order to approximate conditional independence. Instead, the filter marginalizes over the pose, in effect, distributing the information encoded within these links to features in the active map, $\mathbf{m}^+$. The marginalization (4.3) populates the information submatrix associated with $\mathbf{m}^+$, which then forms a maximum clique in the graph. Irrespective of sparsification, this fill-in would otherwise occur as part of the next time prediction step and, with the active map growing ever-larger, would fully populate the matrix. The ESEIF avoids extensive fill-in by bounding the number of active landmarks. When the active map reaches a predetermined size, the ESEIF "kidnaps" the robot, sacrificing temporal information as well as a controlled amount of fill-in. The algorithm then relocalizes the vehicle, creating a new set of active features. Since observations are typically confined to the robot's local environment, these features are spatially close. The active map is built up from neighboring landmarks until the next sparsification. As a result, the ESEIF forms marginalization cliques that resemble submaps that are structured according to robot's visibility and the density of features in the environment.

4.2.2 Core Filter Mechanics

The principle component of the ESEIF that differentiates it from other forms of the information filter is the sparsification strategy, which takes the form of a variation of the measurement update step. Aside for the occasional sparsification step, the ESEIF measurement update and time prediction processes mimic the standard information filter implementations. For completeness, we briefly summarize these filter mechanics that we originally discussed in Section 2.5.

Time Projection Step

We model the vehicle dynamics according to a first-order nonlinear Markov model (4.7a) with additive white Gaussian noise, $\mathbf{w}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{Q}\big)$. A Taylor series expansion about the current mean pose, $\boldsymbol{\mu}_{x_t}$, yields the linear approximation in (4.7b).

$$\begin{aligned}
\mathbf{x}_{t+1} &= \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_{t+1}\big) + \mathbf{w}_t && (4.7\text{a})\\
&\approx \mathbf{f}\big(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}\big) + \mathrm{F}\big(\mathbf{x}_t - \boldsymbol{\mu}_{x_t}\big) + \mathbf{w}_t && (4.7\text{b})
\end{aligned}$$

In Section 2.2.1, we presented time prediction as a two step process in which we first augment the state with the new robot pose, $\mathbf{x}_{t+1}$, and then marginalize over the previous pose, $\mathbf{x}_t$. Below, we present the composition of the augmentation and roll-up processes as a single step that brings the $p\left(\mathbf{x}_t, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right)$ posterior up to date:

$$p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1}\big)$$

$$\bar{\Lambda}_{t+1} = \begin{bmatrix} \mathrm{Q}^{-1} - \mathrm{Q}^{-1}\mathrm{F}\,\Omega\,\mathrm{F}^\top\mathrm{Q}^{-1} & \mathrm{Q}^{-1}\mathrm{F}\,\Omega\,\Lambda_{x_tM} \\ \Lambda_{Mx_t}\,\Omega\,\mathrm{F}^\top\mathrm{Q}^{-1} & \Lambda_{MM} - \Lambda_{Mx_t}\,\Omega\,\Lambda_{x_tM} \end{bmatrix} \qquad (4.8\text{a})$$

$$\bar{\boldsymbol{\eta}}_{t+1} = \begin{bmatrix} \mathrm{Q}^{-1}\mathrm{F}\,\Omega\,\boldsymbol{\eta}_{x_t} + \left(\mathrm{Q}^{-1} - \mathrm{Q}^{-1}\mathrm{F}\,\Omega\,\mathrm{F}^\top\mathrm{Q}^{-1}\right)\left(\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) - \mathrm{F}\boldsymbol{\mu}_{x_t}\right) \\ \boldsymbol{\eta}_M - \Lambda_{Mx_t}\Omega\left(\boldsymbol{\eta}_{x_t} - \mathrm{F}^\top\mathrm{Q}^{-1}\left(\mathbf{f}(\boldsymbol{\mu}_{x_t}, \mathbf{u}_{t+1}) - \mathrm{F}\boldsymbol{\mu}_{x_t}\right)\right) \end{bmatrix} \qquad (4.8\text{b})$$

$$\text{where} \quad \Omega = \left(\Lambda_{x_tx_t} + \mathrm{F}^\top\mathrm{Q}^{-1}\mathrm{F}\right)^{-1}$$

The matrix $\Lambda_{Mx_t} = \Lambda_{x_tM}^\top$ denotes the block matrix that forms the lower-left sub-block of Λ_t and corresponds to the shared information between the map and previous pose. Similarly, $\Lambda_{x_tx_t}$ and Λ_{MM} denote the robot and map sub-blocks along the diagonal.

In Section 2.5.3, we discussed the fundamental characteristics of time prediction in the information form, namely the population of the information matrix and the persistence of the active map. The update to the map information matrix sub-block, $\bar{\Lambda}_{22} = \Lambda_{MM} - \Lambda_{Mx_t}\Omega\Lambda_{x_tM}$, instantiates shared information among the set of active landmarks, $\mathbf{m}^+$, and represents the majority of the matrix fill-in. This particular calculation is quadratic in the number of active features and defines the upper bound on the computational cost of time prediction. As ESEIF sparsification enforces a Γ_a limit on the size of the active map, time projection is constant-time, irrespective of the size of the map, $\mathbf{M}$. This assumes an efficient strategy for estimating the mean vehicle state, which we will describe shortly.

Measurement Update Step

Our description of the ESEIF update step assumes that measurements are nonlinear in the robot pose and landmarks and follow the general form in Equation (4.9a). The additive term, $\mathbf{v}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}\big)$, corresponds to white Gaussian noise and models the contribution of sensor noise and measurement uncertainty. Equation (4.9b) is the first-order linearization about the mean estimate for the robot pose and observed features, with H the sparse Jacobian matrix.

$$\begin{aligned}\mathbf{z}_t &= \mathbf{h}\big(\boldsymbol{\xi}_t\big) + \mathbf{v}_t && \text{(4.9a)}\\ &\approx \mathbf{h}\big(\bar{\boldsymbol{\mu}}_t\big) + \mathrm{H}\big(\boldsymbol{\xi}_t - \bar{\boldsymbol{\mu}}_t\big) + \mathbf{v}_t && \text{(4.9b)}\end{aligned}$$

The ESEIF updates the distribution, $p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1}\big)$, to incorporate this evidence via the standard EIF update step. This process yields the canonical form of the new SLAM posterior,

$$p\left(\mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^{t+1}, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\boldsymbol{\eta}_{t+1}, \Lambda_{t+1}\big)$$

$$\begin{aligned}\Lambda_{t+1} &= \bar{\Lambda}_{t+1} + \mathrm{H}^\top \mathrm{R}^{-1}\mathrm{H} && \text{(4.10a)}\\ \boldsymbol{\eta}_{t+1} &= \bar{\boldsymbol{\eta}}_{t+1} + \mathrm{H}^\top \mathrm{R}^{-1}\left(\mathbf{z}_{t+1} - \mathbf{h}(\bar{\boldsymbol{\mu}}_{t+1}) + \mathrm{H}\bar{\boldsymbol{\mu}}_{t+1}\right) && \text{(4.10b)}\end{aligned}$$

The structure of the environment, along with the limited FOV of the robot's sensors, bound the number of landmarks, m, that the vehicle observes at each time step, i.e. $|\mathbf{z}_{t+1}| = \mathcal{O}(m)$. Consequently, the Jacobian matrix, H, is zero everywhere except for the limited number of $\mathcal{O}(m)$ columns that correspond to the observed features. The additive update to the information matrix, $\mathrm{H}^\top\mathrm{R}^{-1}\mathrm{H}$ in Equation (4.10a) is itself sparse and contributes only to elements that correspond to the robot and observed features. Due to the sparseness of H, this matrix outer-product involves $\mathcal{O}(m^2)$ multiplications. Assuming access to estimates for the mean robot and observed landmark poses, the ESEIF measurement update step is $\mathcal{O}(m^2)$ and, in turn, constant-time irrespective of the overall map size.

4.2.3 Mean Recovery and Data Association

A significant limitation of the ESEIF and other variants of the information filter is that the canonical form does not provide direct access to the mean vector or covariance estimates. As we discussed in our description of the time projection and measurement update steps, the linearization of the motion and observation models requires an estimate for a subset of the mean vector. Data association typically relies on the knowledge of the marginal distribution over the robot state and a subset of the map [128]. Computing this marginal in the information form is equivalent to calculating a sub-block of the corresponding covariance matrix. The ESEIF overcomes these limitations with approximate inference strategies that efficiently perform mean estimation and data association over the SLAM posterior.

Mean Recovery

The sparse information filter provides for a *near* constant-time SLAM implementation. The caveat is, in part, a consequence of the fact that we no longer have access to the mean vector when the posterior is represented in the canonical form. Naïvely, we can compute the entire mean vector as $\boldsymbol{\mu}_t = \Lambda_t^{-1}\boldsymbol{\eta}_t$, though the cost of inverting the information matrix is cubic in the number of states, making it intractable even for small maps.

Instead, we pose the problem as one of solving the set of linear equations

$$\Lambda_t \boldsymbol{\mu}_t = \boldsymbol{\eta}_t \tag{4.11}$$

and take advantage of the sparseness of the information matrix. There are a number of techniques that iteratively solve such sparse, symmetric, positive definite systems of equations, including conjugate gradient descent [118] as well as relaxation-based algorithms, such as Gauss-Seidel [6] and, more recently, the multilevel relaxation adaptation of multigrid optimization [47]. The optimizations can often be performed over the course of multiple time steps, since, aside from loop closures, the mean vector evolves slowly in SLAM. As a result, we can bound the number of iterations required at any one time step [30].

Oftentimes, we are interested only in a subset of the mean, such as during the time projection step, which requires an estimate for the robot pose. We can then consider partial mean recovery [34] in which we partition (4.11) as

$$\begin{bmatrix} \Lambda_{ll} & \Lambda_{lb} \\ \Lambda_{bl} & \Lambda_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_l \\ \boldsymbol{\mu}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_l \\ \boldsymbol{\eta}_b \end{bmatrix} \tag{4.12}$$

where $\boldsymbol{\mu}_l$ is the "local portion" that we want to solve for and $\boldsymbol{\mu}_b$ is the "benign portion" of the map. The benign map typically refers to landmarks outside the robot's local neighborhood. By virtue of the inverse relationship between the strength of feature-to-feature and feature-to-vehicle information constraints and their spatial distance [44], recent measurement updates have only limited effect on the estimates for these landmarks, hence the term. Given an estimate for $\boldsymbol{\mu}_b$, we can reduce the partitioned set of equations (4.12) to an approximate solution for the local mean,

$$\hat{\boldsymbol{\mu}}_l = \Lambda_{ll}^{-1} \left(\boldsymbol{\eta}_l - \Lambda_{lb} \hat{\boldsymbol{\mu}}_b \right). \tag{4.13}$$

Due to the sparsity of Λ_{lb}, this formulation requires only a subset of $\hat{\boldsymbol{\mu}}_b$, corresponding to the Markov blanket for the local map. Assuming that we have an accurate estimate for the mean of this portion of the benign map, this expression (4.13) provides an efficient approximation to the mean that we are interested in.

Data Association

The successful implementation of any SLAM algorithm requires the ability to correctly match observations of the environment with the associated landmarks in the map. The data association problem is often addressed by choosing the feature that best explains the measurement, subject to a threshold that identifies spurious observations. For a particular correspondence, the likelihood follows from the marginal distribution for the particular states associated with the hypothesis (typically the robot pose, $\mathbf{x}_t$, and a single landmark, $\mathbf{m}_i$), $p\left(\mathbf{x}_t, \mathbf{m}_i \mid \mathbf{z}^{t-1}, \mathbf{u}^t\right)$. Unfortunately, the information form is not amenable to computing this marginal from the full joint posterior, since, referring back to Table 2.1, the Schur complement requires the inversion of a large matrix.

Consequently, the traditional approach to data association is not an option for scalable information filters. Instead, Thrun *et al.* [130] approximate the measurement likelihood from a conditional distribution rather than the marginal. Specifically, the SEIF considers the Markov blanket for $\mathbf{x}_t$ and $\mathbf{m}_i$, $\mathrm{MB}(\mathbf{x}_t, \mathbf{m}_i)$, that consists of all states directly linked to either $\mathbf{x}_t$ or $\mathbf{m}_i$ in the GMRF. The SEIF first computes the conditional distribution $p\left(\mathbf{x}_t, \mathbf{m}_i, \mathrm{MB}(\mathbf{x}_t, \mathbf{m}_i) \mid \mathbf{M}', \mathbf{z}^{t-1}, \mathbf{u}^t\right)$ where $\mathbf{M}'$ denotes all state elements not in $\{\mathbf{x}_t, \mathbf{m}_i, \mathrm{MB}(\mathbf{x}_t, \mathbf{m}_i)\}$. This distribution is then marginalized over the Markov blanket to achieve an approximation to the desired marginal, $p\left(\mathbf{x}_t, \mathbf{m}_i \mid \mathbf{M}', \mathbf{z}^{t-1}, \mathbf{u}^t\right)$, which is used to determine the likelihood of the hypothesis. The cost of conditioning on $\mathbf{M}'$ is negligible and does not depend on the size of the map. Once most of the map has been conditioned away, the matrix that is inverted as part of the subsequent marginalization is now small, on the order of the size of the Markov blanket. The resulting distribution has been successfully utilized for data association with the SEIF [79], though it has been demonstrated to yield exaggerated confidence in measurement data. This overconfidence then lead to valid data association hypotheses being ignored and, in turn, the resulting disregard of evidence in the subsequent measurement update step [36].

The marginal is easily determined from the standard parametrization, described by the mean and sub-blocks of the full covariance matrix corresponding to $\mathbf{x}_t$ and $\mathbf{m}_i$. Inverting the information matrix to access the covariance, though, is equivalent to performing the marginalization in the canonical form and is, thus, impractical. Alternatively, Eustice *et al.* [36] propose an efficient method for approximating the marginal that gives rise to a conservative measure for the hypothesis likelihood. The technique stems from posing the relationship, $\Lambda_t \Sigma_t = \mathrm{I}$, as a sparse system of linear equations, $\Lambda_t \Sigma_{\star i} = \mathbf{e}_i$, where $\Sigma_{\star i}$ and $\mathbf{e}_i$ denote the i^{th} columns of the covariance and identity matrices, respectively. They estimate the robot pose joint-covariance, $\Sigma_{\star x_t}$,x online by solving the system of equations with one of the iterative algorithms mentioned for mean recovery. The algorithm combines this with a conservative estimate for the feature covariance to achieve the representation for the marginal covariance. The marginal, which is itself conservative, is then used for data association.

4.3 Experimental Results

This section explores the effectiveness of the ESEIF algorithm in comparison to the SEIF and EKF when applied to different forms of the SLAM problem. We first present the results of a controlled LG SLAM simulation that allows us to compare the different sparsified posteriors with the true distribution as maintained by the Kalman Filter. We then discuss the performance of the sparsified information algorithms on a pair of real-world, nonlinear SLAM problems, including the benchmark Sydney Park outdoor dataset widely popular in the SLAM community.

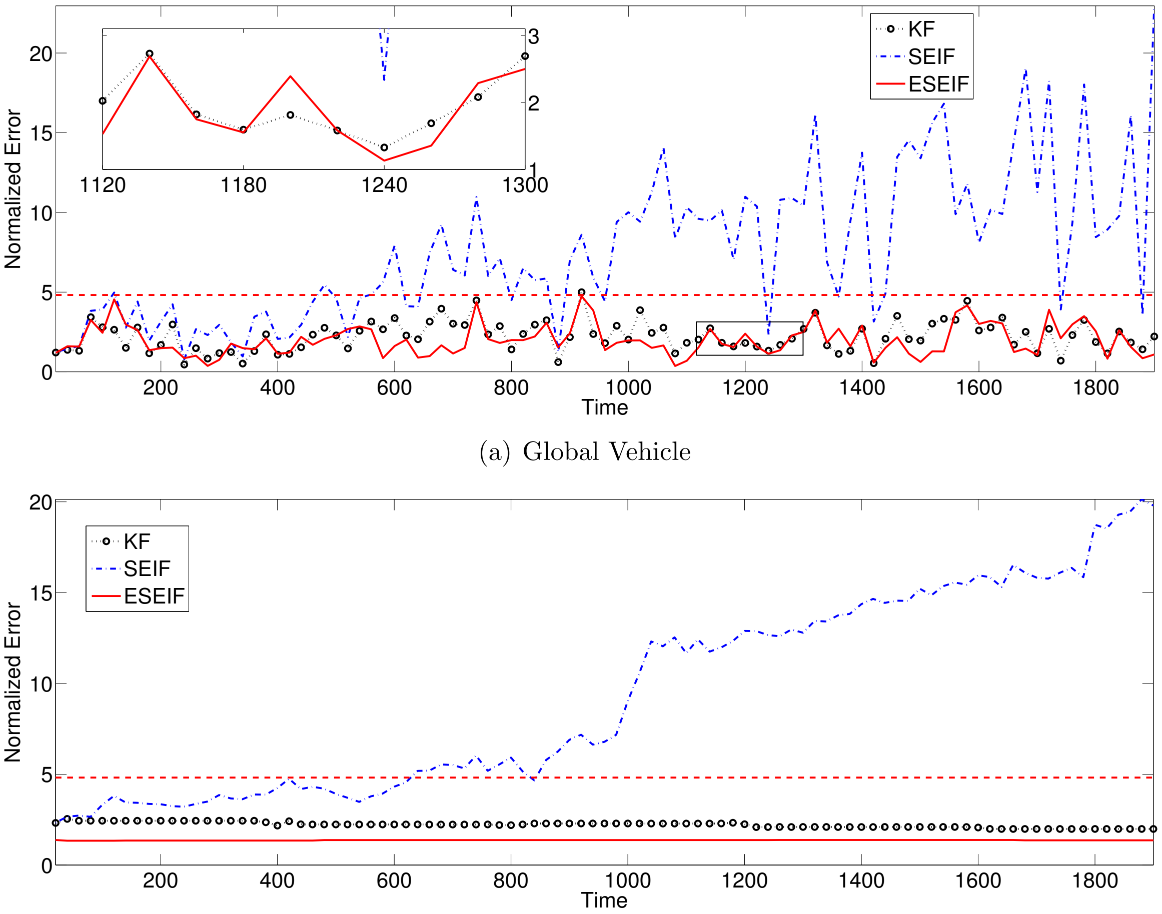
4.3.1 Linear Gaussian Simulation

In an effort to better understand the theoretical consequences of enforcing sparsity in information filters, we first study the effects of applying the different approaches in a controlled simulation. The experiment mimics the linear Gaussian simulations that we use in Section 3.4.1 to compare the SEIF and modified rule sparsification strategies. In this example, the environment is comprised of a set of point features, uniformly distributed over the area. The robot moves translationally according to a linear, constant-velocity model and measures the relative position of a bounded number of neighboring features. Both the measurements, as well as the vehicle motion, are corrupted by additive white Gaussian noise.

We implement the ESEIF and SEIF using their corresponding sparsification routines to maintain a bound of $\Gamma_a = 10$ active features. In the case of ESEIF sparsification, we reserve as many of the measurements as possible for the relocalization component, to the extent that we do not violate the Γ_a bound (i.e. $|\mathbf{z}_\beta| \leq \Gamma_a$). Additionally, we apply the standard Kalman filter that, by the linear Gaussian nature of the simulation, is the optimal Bayesian estimator. Aside from the different sparsification routines, each estimator is otherwise identical.

Our main interest in the LG simulation is to evaluate the effect of the different sparsification strategies on the estimation accuracy. To that end, we perform a series of Monte Carlo simulations from which we measure the normalized estimation error squared (NEES) [5] as an indication of filter consistency. As with our evaluation in Section 3.4.1, we first consider the *global* error between the unadulterated filter estimates for the vehicle and feature positions and their ground truth positions. We compute this score over several simulations and plot the averages in Figure 4-3 for the vehicle and a single landmark. The 97.5% chi-square upper limit for the series of simulations is denoted by the horizontal threshold, which the KF normalized errors largely obey. Figure 4-3(a) demonstrates that the SEIF vehicle pose error is significantly larger than that of the KF and ESEIF, and exceeds the chi-square bound for most of the simulation. The same is true of the estimate for the landmark positions as shown in Figure 4-3(b). This behavior indicates that SEIFs maintain an absolute state estimate that is inconsistent. In contrast, the ESEIF yields global errors for both the vehicle and map that are similar to the KF and pass the chi-square test. This suggests that the ESEIF SLAM distribution is globally consistent.

The second normalized error concerns the accuracy of the relative state elements. We transform the vehicle and map positions into the reference frame associated with the first observed feature, $\mathbf{x}_m$, via the compounding operation, $\mathbf{x}_{m_i} = \ominus\mathbf{x}_m \oplus \mathbf{x}_i$ [121]. We then measure the *relative* error by comparing the transformed map estimates to the root-shifted ground truth positions. The error in the relative estimates for the vehicle and the same feature as in Figure 4-3 are shown in Figures 4-4(a) and 4-4(b), respectively, together with the 97.5% chi-square bound. As we demonstrated as part of our analysis of the SEIF sparsification strategy in Section 3.4.1, the SEIF relative estimates satisfy the chi-square test. Meanwhile, the ESEIF yields relative map errors that are nearly indistinguishable from those of the KF. Furthermore, the normalized errors fall well below the chi-square limit. This behavior suggests that, unlike the



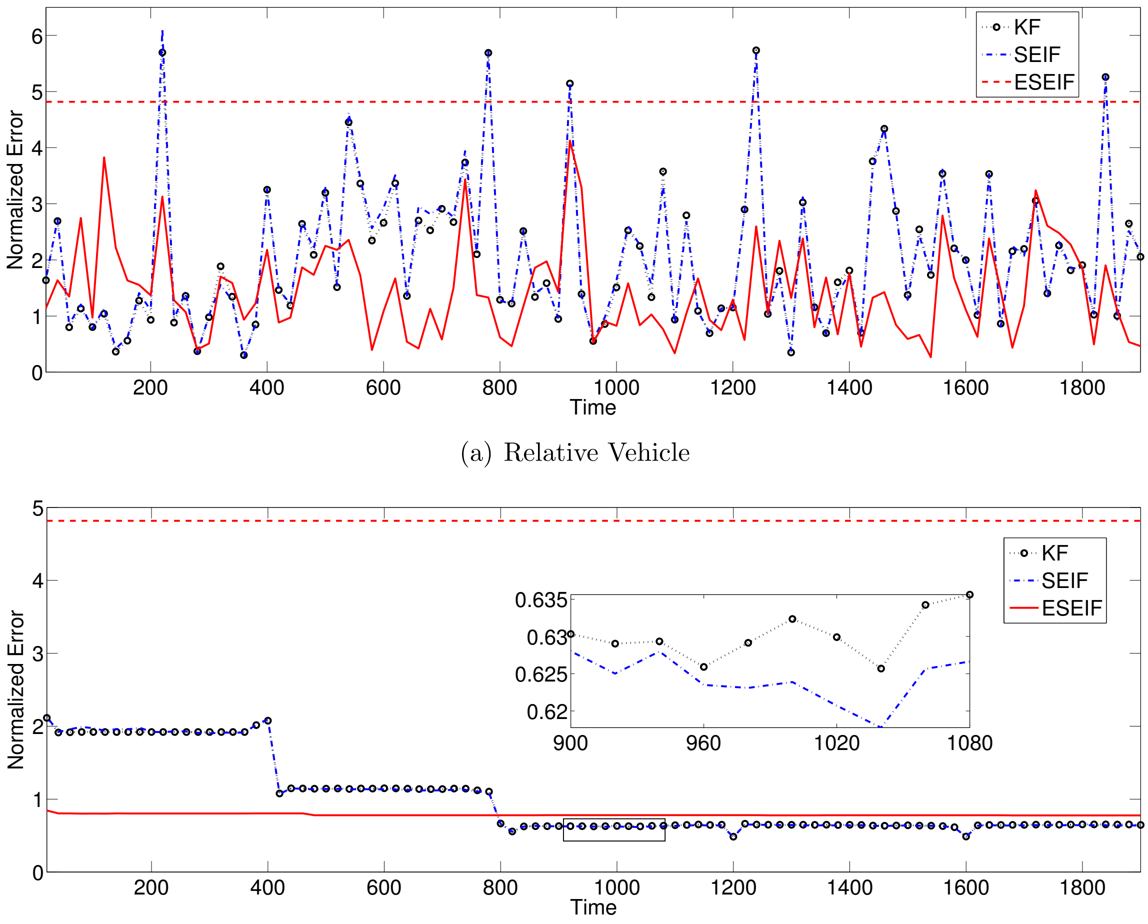
(a) Global Vehicle

(b) Global Feature

Figure 4-3: Plots of the NEES measured based upon a series of Monte Carlo simulations of linear Gaussian SLAM. The *global* errors associated with the estimates for (a) vehicle pose and (b) a single feature representative of the map are computed by comparing the direct filter estimates with ground truth and provide a measure of global consistency. The horizontal threshold denotes the 97.5% chi-square upper bound and serves as a test for the consistency of the different filters. For both the vehicle and the map, the global ESEIF errors satisfy the chi-square limit while those of the SEIF exceed the bound.

SEIF, the ESEIF maintains a posterior that preserves both the global and relative consistency of the map.

The NEES score jointly measures the error in the mean estimate as well as the confidence that the filter attributes to this error. The normalized error for the ESEIF reflects map estimate errors that agree with those of the KF. Additionally, the score is an indication of a posterior belief function that is conservative with respect to the nominal distribution. In the previous chapter, we analyzed this confidence by comparing the filter uncertainty estimates against those of the true distribution as maintained by the KF. We recover the map covariance from the information matrix and, for each landmark, compute the log of the ratio of the covariance sub-block



(a) Relative Vehicle

(b) Relative Feature

Figure 4-4: Plots of the *relative* estimate consistency as measured by the NEES. The error corresponds to the same set of Monte Carlo linear Gaussian simulations that we use to calculate the global NEES. We compute the relative error by transforming the state with respect to the first feature added to the map. The plot in (a) presents the resulting vehicle pose error while (b) demonstrates the relative error for the same feature that we reference in Figure 4-3(b). Both plots include the 97.5% chi-square upper bound (horizontal line) as an indication of estimator consistency. As is the case for the global error shown in Figure 4-3(a), the ESEIF vehicle pose and feature estimates largely satisfy the chi-square bound. This suggests that the filter maintains estimates that are both globally and locally consistent.

determinant to the determinant of the KF sub-block. The KF estimate represents the true distribution and log ratios less than zero signify overconfidence while values greater than zero imply conservative uncertainty estimates. Figure 4-5 presents a histogram plot of these ratios for the two information filters. Our filter maintains uncertainty bounds for the global map estimates that are conservative with respect to the KF. This indicates that, by sacrificing temporal information across poses as part of occasional sparsification, the ESEIF yields a posterior that is conservative relative to the true distribution. Meanwhile, the SEIF uncertainty bounds for the

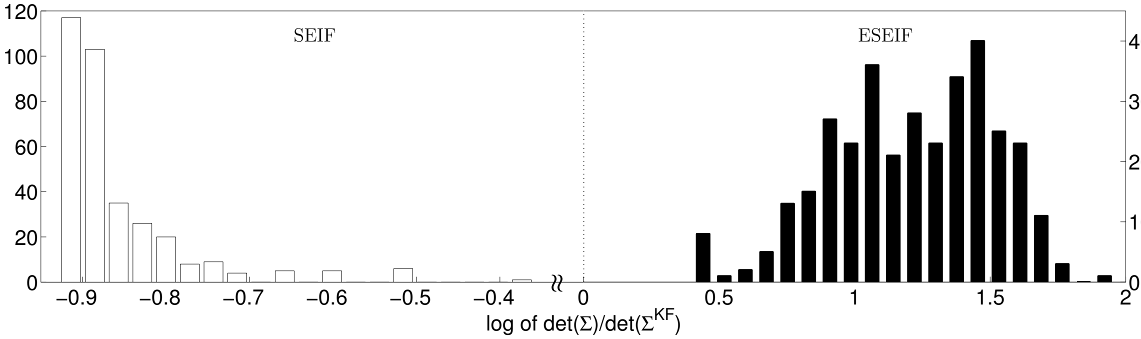
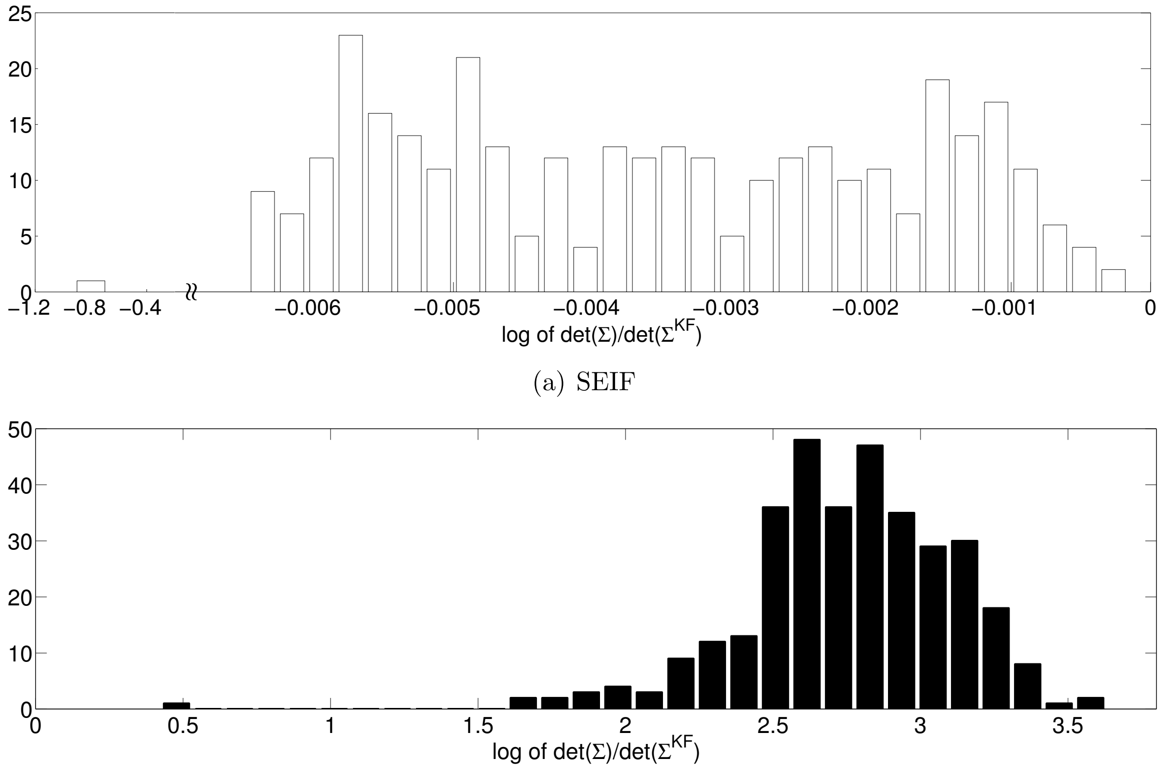


Figure 4-5: Histogram for the LG simulation describing the global map uncertainty maintained by the SEIF (*left*) and ESEIF (*right*) as compared with that of the KF. For each feature, we compute the log of the ratio between the information filter covariance sub-block determinant and the determinant for the actual distribution as given by the KF. Values greater than zero imply conservative estimates for the uncertainty while log ratios less than zero indicate overconfidence. Note that all of the SEIF estimates are overconfident while those of the ESEIF are conservative.

global map are significantly smaller than those of the KF, indicating that the filter is susceptible to overconfidence as a consequence of the sparsification strategy. This agrees with our discussion in Section 3.3 on the inherent implications of enforcing sparsity by approximating conditional independence.

We similarly evaluate the relative uncertainty associated with landmark estimates expressed with respect to the first feature. The histogram in Figure 4-6(b) demonstrates that the ESEIF maintains measures of uncertainty for the relative map that, like the global estimates, are conservative in comparison to the EKF. The one outlier corresponds to the representation for the original world origin within the root-shifted map, whose uncertainty estimate is less conservative. This behavior is in contrast with that of the SEIF relative map estimates. As we the analysis performed in Section 3.4.1 revealed, the histogram in Figure 4-6(a) shows that the SEIF and KF estimates for the relative uncertainty agree much more closely than do the global estimates. The relative position of the world origin is the one exception and exhibits greater overconfidence on account of the inconsistency of the distribution over the global map. Hence, while the SEIF maintains a distribution that is overconfident both for the global and relative map estimates, the ESEIF yields a sparse parametrization of the posterior that remains conservative with respect to the true distribution, in the linear Gaussian case.

Figure 4-7 illustrates the computational benefits of the ESEIF over the KF. Plotted in Figure 4-7(a), the KF update time grows quadratically with the number of states. In contrast, the ESEIF and SEIF updates remain constant-time despite an increase in the state dimension. While this efficiency is inherent to information filter updates, sparseness is beneficial for the prediction step, which is quadratic in size of the map for non-sparse information filters. We see this benefit in Figure 4-7(b) as the prediction time is similar for all three filters. In the case of the KF, the computation time increases linearly with the number of landmarks, albeit with a small



(a) SEIF

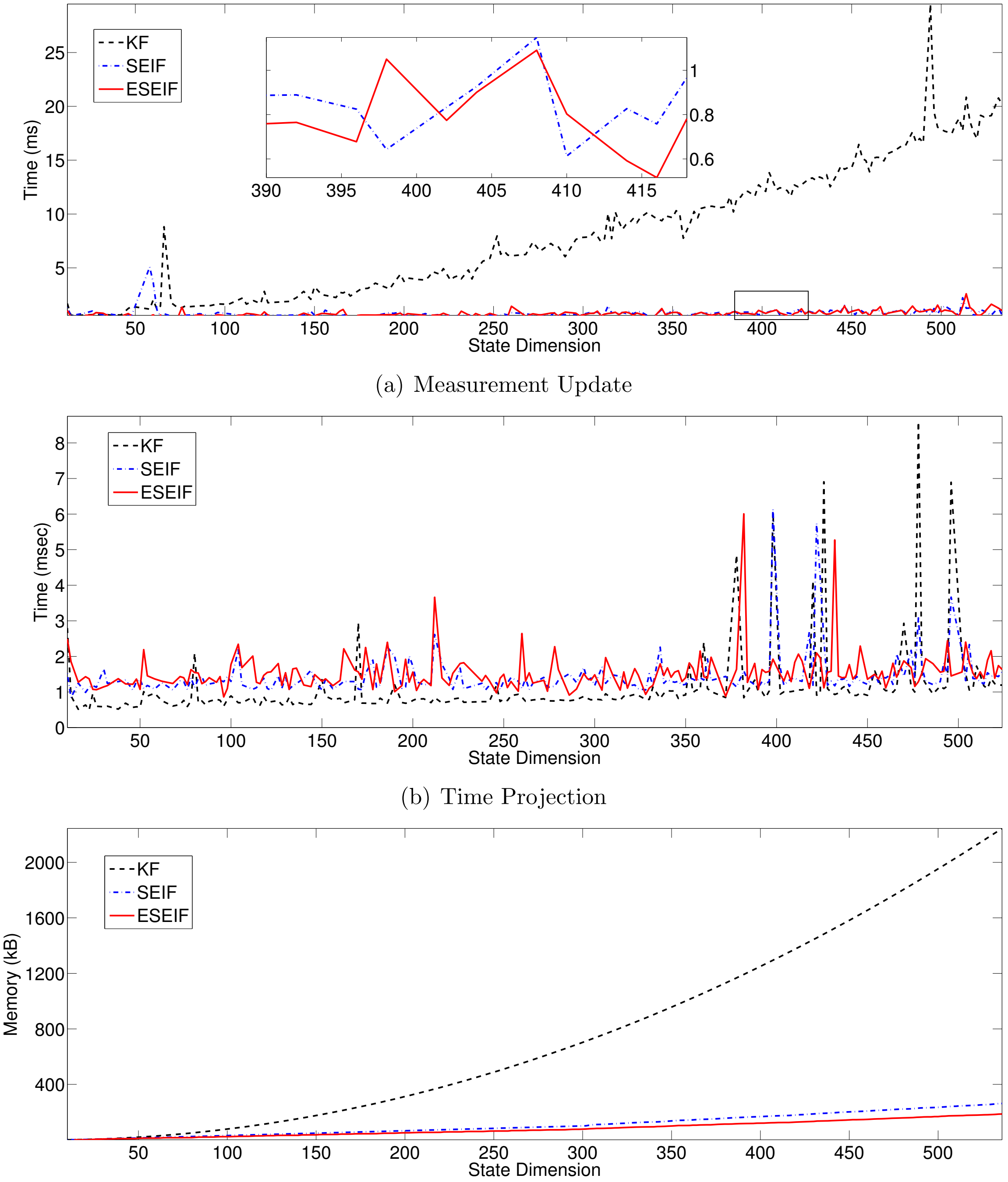
(b) ESEIF

Figure 4-6: The uncertainty attributed to the relative map estimates for the (a) SEIF and (b) ESEIF expressed relative to the optimal KF. The uncertainty ratios are determined as before, in this case based upon the relative covariance estimates that follow from root-shifting the state to the first feature added to the map. While the SEIF relative map estimates remain overconfident, the ESEIF produces a posterior over the relative map that is conservative with respect to the true distribution.

constant factor. Meanwhile, prediction is constant-time for the sparsified information filters as a result of the bounded active map size. Note that the filtering processes do not require knowledge of the mean vector and, thus, both the time prediction and measurement update steps are inherently constant-time.

Additionally, the sparse information matrices impose memory requirements that are considerably smaller than the memory necessary to store the covariance matrix. Consider the density of the three 536×536 matrices at the end of the simulation. The covariance matrix is fully-populated, accounting for the correlations that exist among the entire robot and map state. In contrast, 92% of the terms in the ESEIF information matrix are exactly zero as is 89% of the SEIF matrix. Figure 4-7(c) plots the difference in the memory requirements as a function of the state dimension. The memory for the sparse information matrices reflects the cost of the suboptimal storage-by-index scheme.[1]

[1] The representation stores the matrix as three vectors, one for the non-zero elements and two for their individual row and column indices.



(a) Measurement Update

(b) Time Projection

(c) Memory

Figure 4-7: A comparison of the performance of the three filters for the LG simulation. The (a) update times for the ESEIF and SEIF are nearly identical and remain constant with the growth of the map. In contrast, the KF exhibits the well-known quadratic increase in complexity. The (b) prediction time grows linearly with the size of the map in the case of the KF while those of the SEIF and ESEIF are constant by virtue of the sparsity of the information matrices. The outliers are due to system multitasking. The plot in (c) reveals that the sparse information forms demand significantly less memory than the fully-populated covariance matrix.

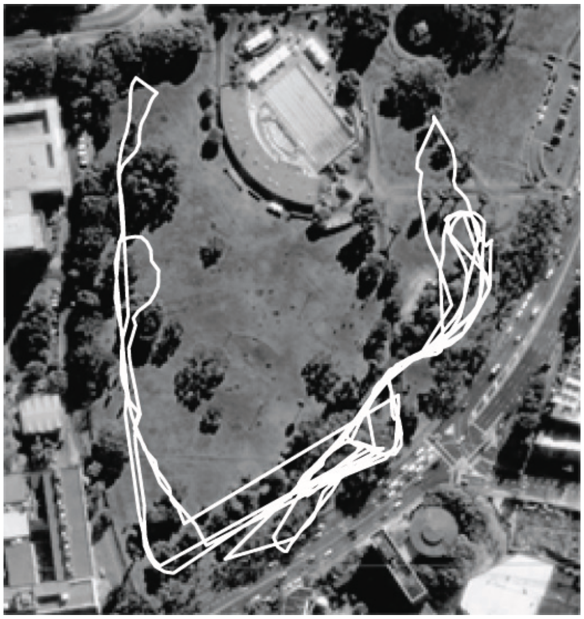


Figure 4-8: An overhead image of Victoria Park in Sydney, Australia, along with a rough plot of the GPS vehicle trajectory. The environment is approximately 250 meters East to West and 300 meters North to South.

4.3.2 Experimental Validation

The linear Gaussian simulations allow us to explore the theoretical behavior of the ESEIF estimator. In particular, we analyze the benefits of a sparsification algorithm that actively controls the formulation of dependence relationships between the robot and map. The results empirically show that the ESEIF provides a sparse representation of the canonical Gaussian while simultaneously preserving consistency. Unfortunately, the simulations are not representative of most real-world applications, which generally involve motion and measurement models that are nonlinear and noise that is non-Gaussian. To study the performance of the ESEIF under these circumstances, we apply it to two nonlinear datasets, along with the SEIF and standard EKF.

Victoria Park Dataset

For the first real-world SLAM problem, we consider the benchmark Victoria Park dataset courtesy of E. Nebot of the University of Sydney [51]. The dataset is widely popular in the SLAM community as a testbed for different algorithms that address the scalability problem [51, 94, 12, 130]. In the experiment, a truck equipped with odometry sensors and a laser range-finder drove in a series of loops within Victoria Park, Sydney. Figure 4-8 presents a bird's-eye view of the park, along with a rough plot of the GPS trajectory. We use a simple perceptual grouping implementation to detect tree trunks located throughout the park among the laser data, which is cluttered with spurious returns. We solve the data association problem offline to ensure that the correspondences are identical for each filter.

We apply the SEIF and ESEIF algorithms together with the EKF, which has been

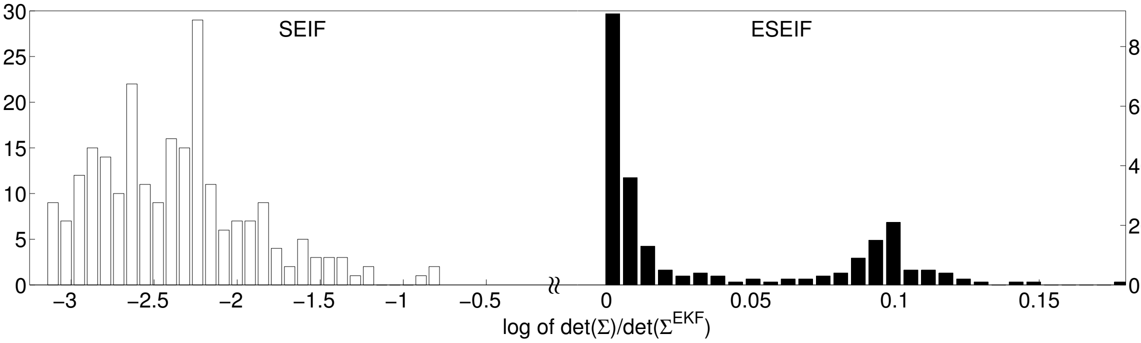
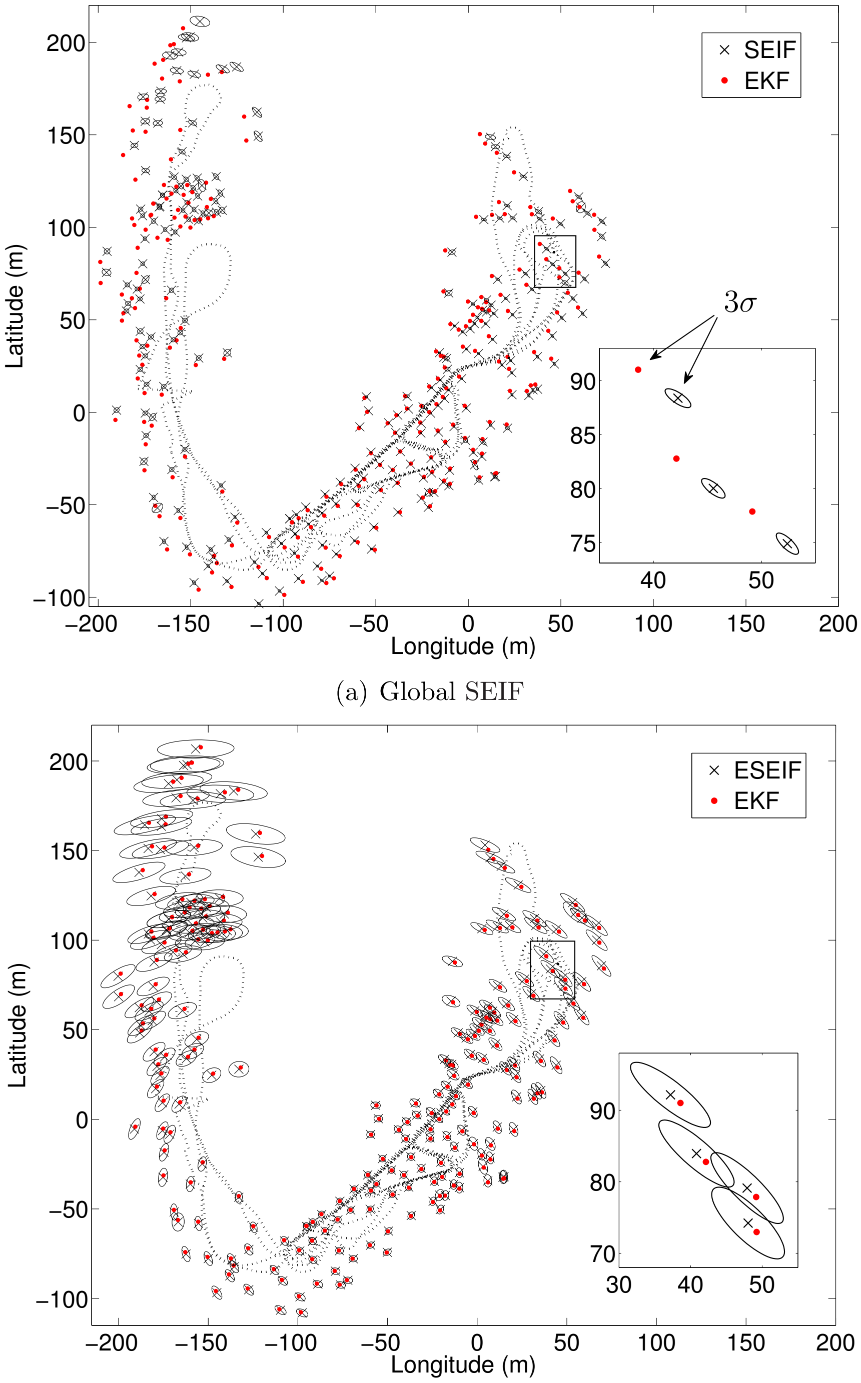


Figure 4-9: Histogram for the Victoria Park dataset comparing the ESEIF and SEIF *global* uncertainty estimates to the results of the EKF. We again use the log of the ratio of the covariance sub-block determinants for each landmark. The ESEIF sparsification strategy yields marginal distributions that are conservative relative to the EKF while the SEIF induces overconfidence.

successfully applied to the dataset in the past [51]. We limit the size of the active map to a maximum of $\Gamma_a = 10$ features for the two information filters. As with the LG simulation, we place a priority on the relocation step when sparsifying the ESEIF, reserving as many tree observations as possible (i.e. no more than $\Gamma_a = 10$) for the sake of adding the vehicle back into the map. Any additional measurements are used to update the filter prior to marginalization. This helps to minimize the influence of spurious observations on the estimate for the relocated vehicle pose.

The final SEIF and ESEIF maps are presented in Figures 4-10(a) and 4-10(b), respectively, along with the estimate for the robot trajectory. The ellipses denote the three sigma uncertainty bounds estimated by the two filters. As a basis for comparison, we plot the map generated by the EKF, which is similar to results published elsewhere. One sees that the ESEIF feature position estimates are very similar to those of the EKF while the SEIF map exhibits a larger deviation. The most obvious distinction between the two maps is the difference in the estimates of filter accuracy as indicated by the uncertainty ellipses associated with each feature. The ESEIF confidence regions capture all of the EKF estimates while the SEIF sparsification strategy induces three sigma confidence estimates that do not account for much of the EKF map. This is particularly evident in the periphery, as we reveal in the inset plot. While not ground truth, the EKF results represent the baseline that the information filters seek to emulate.

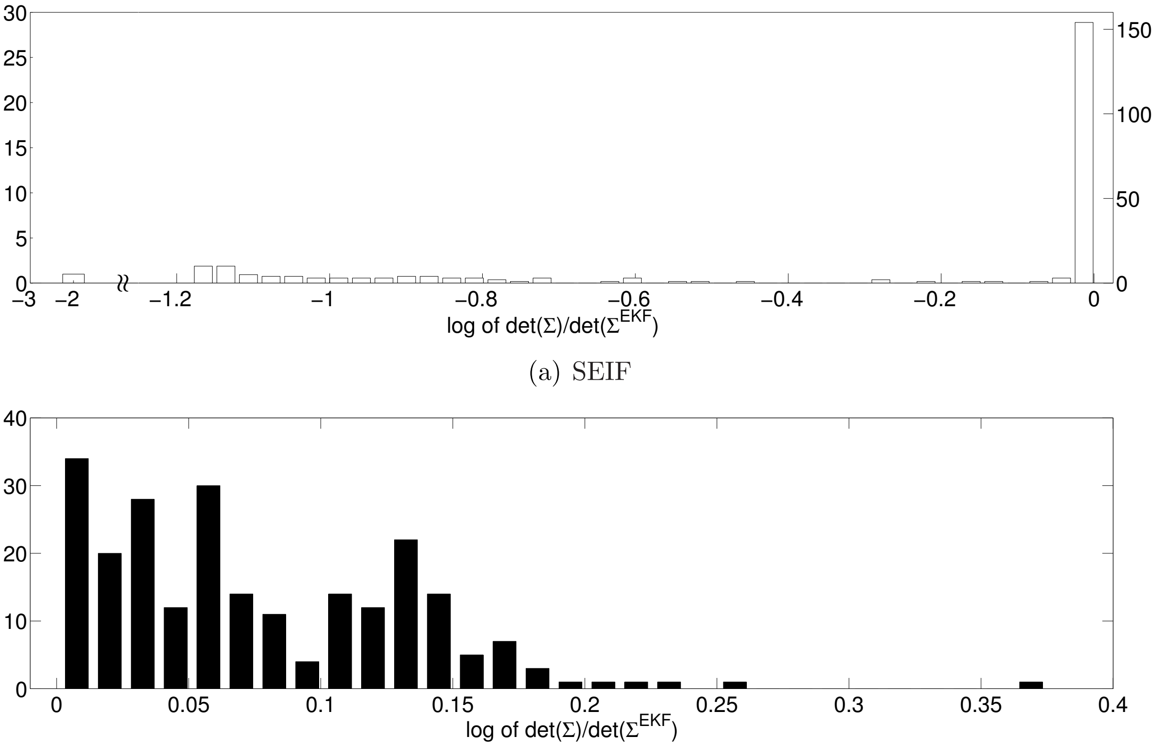
The difference becomes more apparent when we directly compare the uncertainty measures for each feature. Figure 4-9 presents a histogram plot of the log ratio between the global feature covariance determinants for the SEIF and our filter with respect to the EKF determinants. The ESEIF maintains estimates for the global uncertainty that are conservative with respect to the EKF while the SEIF estimates for this dataset are smaller. This is consistent with the linear Gaussian simulation results and empirically suggests that the ESEIF produces a posterior that is conservative with respect to that of the EKF while the SEIF sparsification strategy results in an overconfident distribution.



(a) Global SEIF

(b) Global ESEIF

Figure 4-10: Plots of the *global* vehicle trajectory and feature position estimates, along with the three sigma confidence bounds for the Victoria Park dataset. The global maps generated by (a) the SEIF and (b) the ESEIF are similar to the EKF map. The SEIF uncertainty ellipses, though, are significantly smaller than those of the ESEIF and, in many cases, do not include the EKF feature estimates. The nature of the ESEIF estimates in comparison with those of the EKF empirically supports the claim that ESEIF sparsification does not induce global inconsistency.



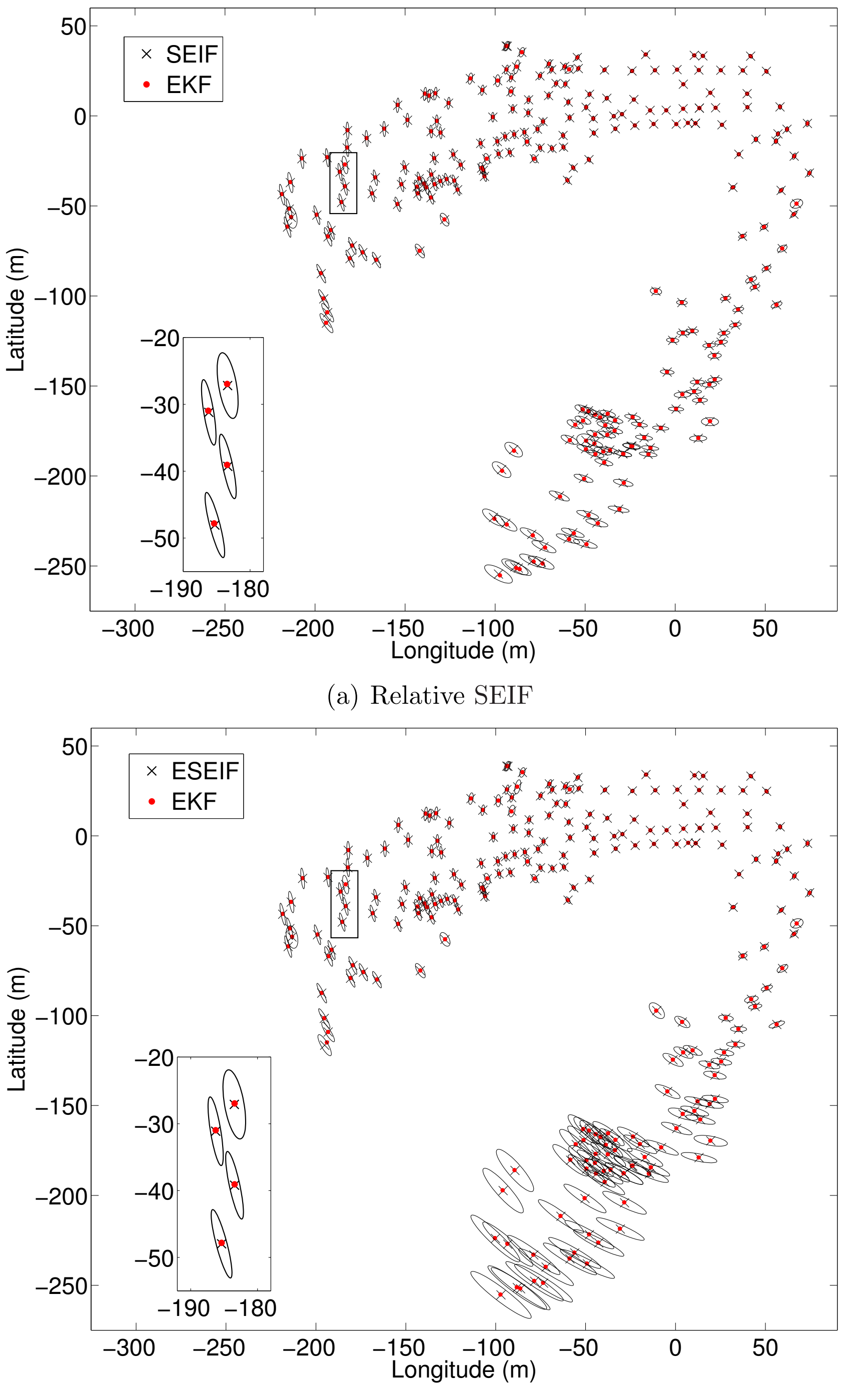
(a) SEIF

(b) ESEIF

Figure 4-11: Histograms for the Victoria Park dataset that compare the *relative* (a) SEIF and (b) ESEIF uncertainty estimates to those of the EKF. We see that the ESEIF estimates remain conservative relative to the EKF and that the SEIF overconfidence persists, but is less severe.

In similar fashion to the LG experiment, we observe contrasting behavior for the relative map that follows from root-shifting the state relative to the vehicle's final pose. The SEIF map shown in Figure 4-12(a) and the ESEIF map plotted in Figure 4-12(b) are both nearly identical to the relative EKF map. Furthermore, the three sigma relative uncertainty bounds maintained by the two filters contain the EKF position estimates. We evaluate the confidence estimates associated with the information filters in Figure 4-11, which presents a pair of histograms over uncertainty relative to that of the EKF. As is the case with the global estimates, the ESEIF maintains a posterior over the relative map that is conservative relative to that of the EKF. Meanwhile, consistent with our analyses in Section 3.4, the SEIF relative estimates better approximate the EKF distribution, but remain slightly overconfident.

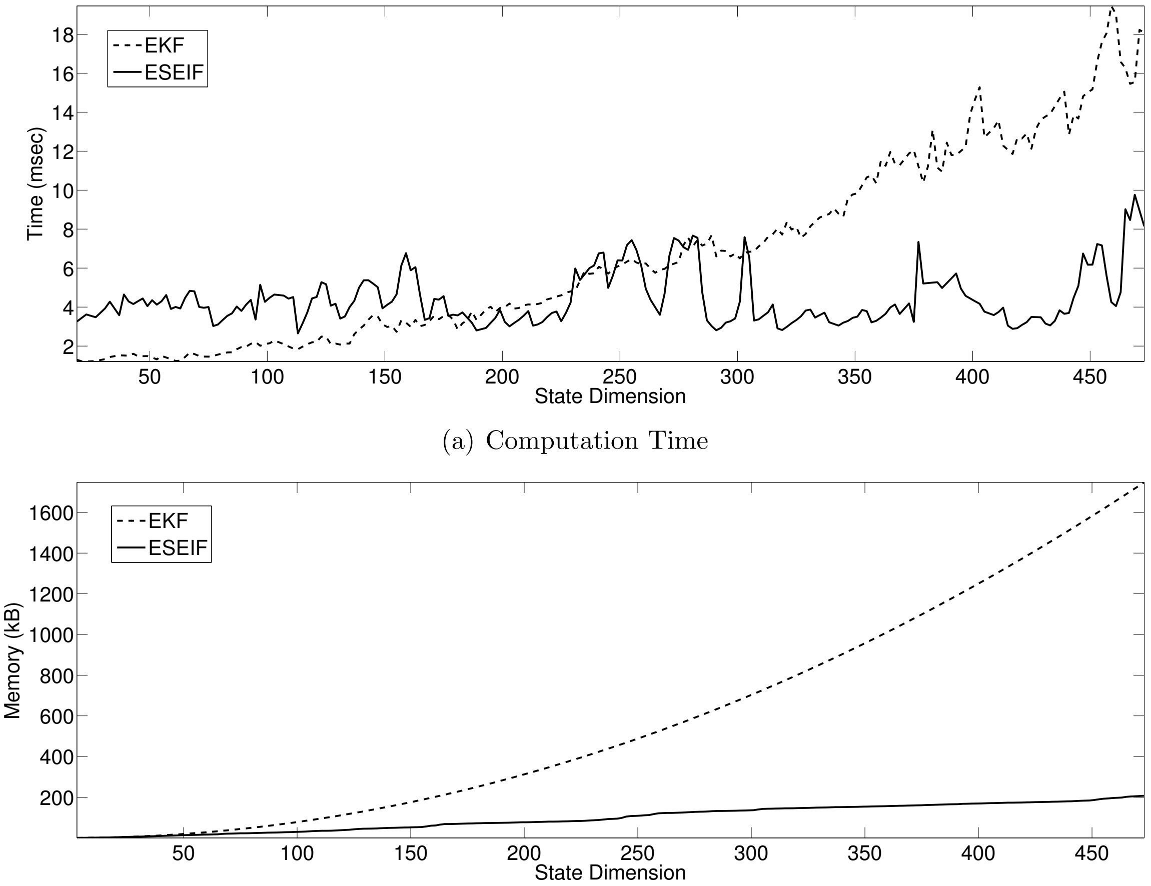
Figure 4-13(a) compares the total time required for the time prediction and measurement update steps for the ESEIF and EKF. We do not include the SEIF performance but note that it is similar to that of the ESEIF. The ESEIF implementation employed partial mean recovery (4.13), solving the full set of equations only upon sparsification. The EKF is more efficient when the map is small (less than 50 landmarks), a reflection of the ESEIF prediction time that is quadratic in the number of



(a) Relative SEIF

(b) Relative ESEIF

Figure 4-12: The *relative* estimates for the vehicle trajectory and map, along with the three sigma confidence bounds for the Victoria Park dataset. We compute the relative estimates by root-shifting the state into the reference frame of the robot at its final pose. Unlike the global estimates shown in Figure 4-10, the relative (a) SEIF and (b) ESEIF feature marginals are similar. The ESEIF uncertainty bounds again capture the EKF landmarks, suggesting that relative estimate consistency is preserved.



(a) Computation Time

(b) Memory

Figure 4-13: Plots of the computational efficiency of the EKF and ESEIF for the Victoria Park dataset. We present (a) the total prediction and update time as a function of state dimension, which includes the cost of mean estimation. The complexity of the EKF increases noticeably with the size of the map while the increase in the ESEIF computation time is more gradual. Employing partial mean recovery, the ESEIF cost is largely a function of the number of active features. The (b) EKF memory requirement is quadratic in the size of the map, yet only linear for the ESEIF.

active features, and the cost of estimating the mean. Yet, as the map grows larger, the quadratic update of the EKF quickly dominates the filtering time of the ESEIF, which varies with the number of active features rather than the state dimension.

The plot in Figure 4-13(b) displays the EKF and ESEIF memory allocations. In order to store the correlations among the map and robot pose, the fully-populated EKF covariance matrix requires quadratic storage space. The ESEIF information matrix, however, is sparse. The matrix is populated along the diagonal and contains a limited number of non-zero, off-diagonal terms that are shared among landmarks and between the robot pose and map. Thus, the ESEIF storage requirement is linear in the size of the map.

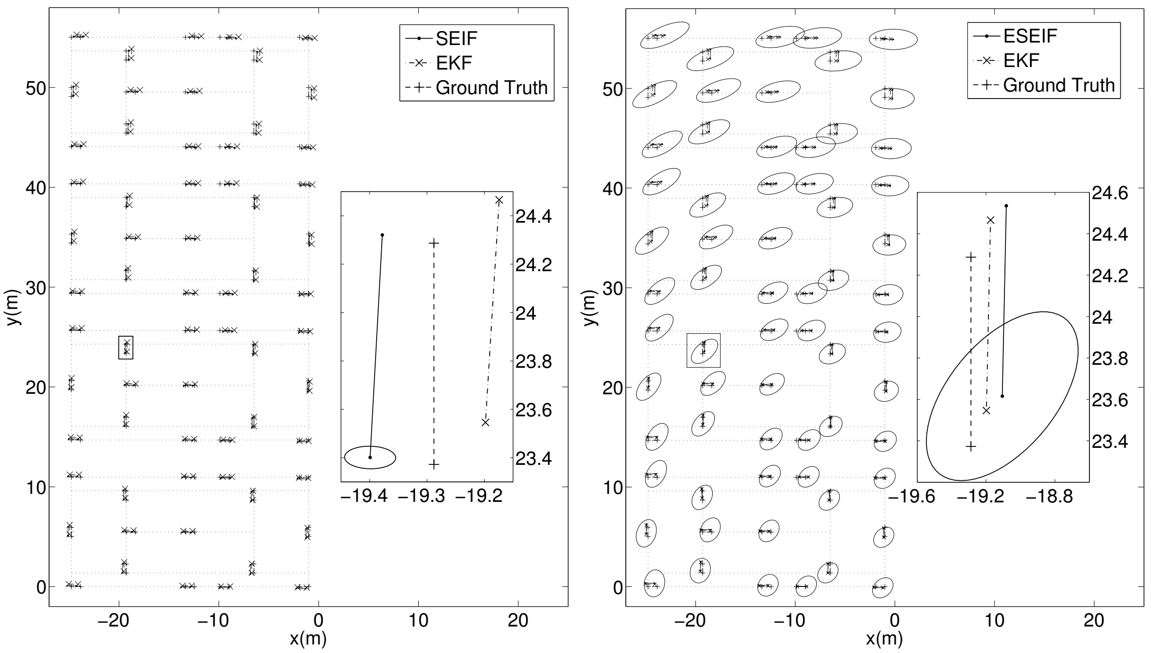
Hurdles Dataset

Section 3.4.2 considered the hurdles dataset to analyze the implications of the SEIF sparsification strategy. We return to this experiment to compare the performance of our ESEIF algorithm. As described earlier, a wheeled robot was driven among 64 track hurdles positioned at known locations along the baselines of four adjacent tennis courts. The vehicle employed a SICK laser scanner to observe the range and bearing to nearby hurdles and encoders to measure the vehicle's forward velocity and rate of rotation. Figure A-1 provides a photograph of the experimental setup.

We again apply the ESEIF and SEIF SLAM algorithms, along with the standard EKF as a basis for comparison. We model each feature as a 2D coordinate frame with the local origin centered on the hurdle's so-called "base" leg and the positive X-axis in the direction of the second leg. Features are then parametrized by the translation and rotation of this coordinate frame. Each filter represents the robot's motion by a kinematic model that includes the encoder-based forward and angular velocity as control inputs. The measurement model is an abstraction of the nominal laser range and bearing observations into a measure of the relative transformation between the vehicle and feature coordinate frames. The data association problem is solved independently, such that the correspondences are identical for all three filters. The maximum number of active landmarks for the three information filters is set at $\Gamma_a = 10$ hurdles. As with the Victoria Park dataset, we prefer to relocalize the vehicle during sparsification with as many measurements as possible and use any surplus observations in the preceding update component. Appendix A.2 presents a detailed explanation of the filter implementation.

We present the final map estimates for the ESEIF and SEIF in Figure 4-14, along with the EKF map and the ground truth feature poses. These maps correspond to the global estimates for feature position and orientation. The ellipses denote the three-sigma uncertainty bounds for the position of each hurdle's base leg. The inset axis within the plot of the ESEIF map is the one exception, where we show the one-sigma bounds for visual purposes. Qualitatively, the ESEIF produces landmark pose estimates that are very similar to those of the EKF as well as the ground truth hurdle positions. The noticeable difference between the two sparsified information filters regards the uncertainty bounds. We again see that the ESEIF confidence estimates account for both the ground truth as well as the EKF map, while the SEIF bounds are too small to capture a majority of the true hurdle positions. This behavior supports our belief that the ESEIF maintains a sparse canonical distribution that is globally consistent with respect to the EKF.

We evaluate the consistency of the filters' relative estimates by transforming the state with respect to the coordinate frame of the first hurdle added to the map. Figure 4-15 compares the relative ESEIF and SEIF map estimates with the EKF map and ground truth. We depict the marginal distribution for each map element with the three-sigma confidence interval. The SEIF yields relative pose estimates that are close to the EKF mean positions, though the uncertainty bounds remain smaller than those of the ESEIF and do not account for many of the ground truth positions. Looking at the ESEIF map, we see that there is very little error between its estimates



(a) SEIF

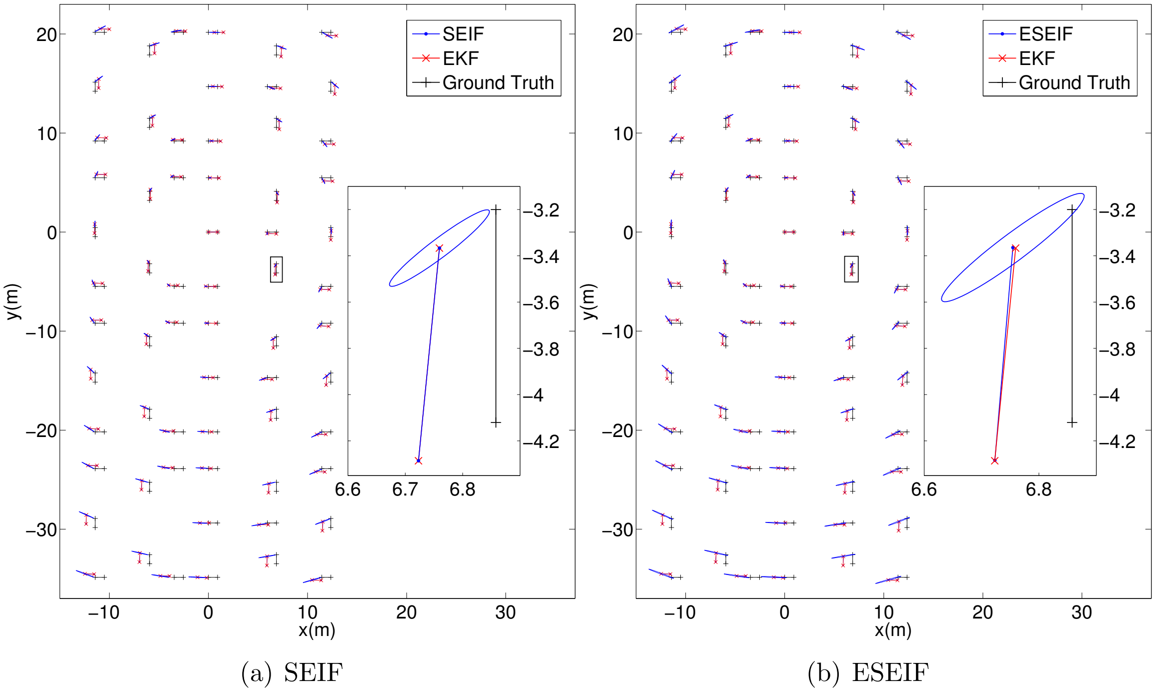
(b) ESEIF

Figure 4-14: The final *global* maps for the hurdles dataset generated with the (a) SEIF and (b) ESEIF compared with the EKF estimates and the ground truth hurdle positions. The ellipses define the three-sigma uncertainty bounds on the location of the base leg of each hurdle. The only exception is the inset plot for the global ESEIF map, where, for aesthetic reasons, we plot the one-sigma uncertainty region. The ESEIF yields marginal distributions that are consistent with the EKF while the SEIF sparsification strategy induces overconfidence.

and those of the EKF. Furthermore, the ESEIF marginals capture the ground truth hurdle positions. In agreement with the LG and Victoria Park analyses, the ESEIF sparsification strategy preserves the relative consistency of the Gaussian model for the marginals.

4.4 Discussion

Over the course of the last two chapters, we have taken a closer look at the SEIF sparsification strategy and, in particular, the consequences on the uncertainty estimates. We presented an alternative algorithm for maintaining sparsity and have shown that it does not suffer from the same overconfidence. In this section, we elaborate on our claims regarding the consistency of the ESEIF. In addition, we draw comparisons between the ESEIF and the D-SLAM algorithm [136], which similarly achieves sparsity while preserving consistency.



(a) SEIF (b) ESEIF

Figure 4-15: The *relative* map estimates for the (a) SEIF and (b) ESEIF as expressed relative to the first hurdles added to the map. The ellipses define the three-sigma uncertainty bounds on the location of the base leg of each hurdle. The ESEIF maintains relative map estimates that are consistent with those of the EKF as well as ground truth.

4.4.1 Estimator Consistency

The results presented in the previous section empirically demonstrate that our filter yields a sparse parametrization of the posterior that is conservative both for the global map as well as the relative landmark estimates. In the linear Gaussian case, this is sufficient to conclude that the ESEIF preserves the consistency of the SLAM posterior for the relative and global representations. On the other hand, as the ESEIF is based upon the dual of the EKF, it is subject to the same convergence issues as the EKF for nonlinear applications [5]. While the results empirically demonstrate that the ESEIF is conservative with respect to the EKF, this does not guarantee that the ESEIF SLAM posterior is a consistent approximation of the true, non-Gaussian distribution. Nonetheless, the algorithm allows us to capitalize on the computational and storage benefits of a sparse information form without incurring additional inconsistency. The EKF has been successfully applied to a wide range of real-world datasets and the ESEIF provides a scalable means of achieving nearly identical estimates.

4.4.2 Comparison with D-SLAM

Wang *et al.* [136] propose a similar algorithm that maintains a sparse canonical parametrization in a consistent manner. The approach decouples SLAM into separate localization and map building problems and addresses them concurrently with different estimators. The D-SLAM considers the map distribution, $p\left(\mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^t\right)$, to be Gaussian and represents it in the canonical form. It then uses an EIF to maintain the information matrix and vector with updates based upon inter-landmark measurements that have been extracted from the robot's observations of the environment. The EIF time projection step is trivial, since the robot pose is not contained in this distribution and, in turn, the information matrix is naturally sparse. The algorithm utilizes two estimators in order to infer the robot pose. One estimator computes the vehicle pose by solving the kidnapped robot problem at each time step, based upon observations of the map. Additionally, D-SLAM implements a standard EKF SLAM process for the robot's local neighborhood that provides a second estimate of pose. To account for unmodeled correlation between the two estimates, they are fused with covariance intersection [64] to achieve a conservative belief over pose. By decoupling the problem in this way, D-SLAM capitalizes on an exactly sparse information matrix without sacrificing consistency.

The key component to maintaining the sparseness of the information matrix follows from the observation that the time projection step for the robot pose causes fill-in. By periodically kidnapping and relocalizing the robot, the ESEIF controls the population of the information matrix. The D-SLAM algorithm takes this one step farther by essentially performing kidnapping and relocalization at each time step. As a result, they sacrifice nearly all information provided by the temporal constraints between successive poses. Additionally, in order to preserve exact sparsity for the map distribution, the algorithm does not incorporate any knowledge of the robot's pose when building or maintaining the map. We believe the D-SLAM estimator to be less optimal as it ignores markedly more information than the ESEIF, which only occasionally disregards temporal links.

Chapter 5

ESEIF for an Underwater Vehicle with an Imaging Sonar

The thesis has analyzed the performance and consistency of the ESEIF algorithm in controlled, linear Gaussian simulations. We then demonstrated the algorithm on a pair of real-world data sets. The results show that the ESEIF yields pose and map estimates similar to those of the EKF, yet at a cost that is better suited to large environments. The performance gains are a direct product of the sparsity of the information matrix. Empirical results suggest that the ESEIF maintains a sparse parametrization of a belief function that is conservative with respect to the EKF in the case that the models are nonlinear and the noise is non-Gaussian. In order to maintain a desired level of sparsity in a consistent manner, the ESEIF relies on there being a sufficient number of observations to relocalize the vehicle. For both the hurdles and the Victoria Park surveys, this isn't a problem due to the environment density and the sensor's field-of-view. We can afford to set a low value for the maximum number of active features and take advantage of the observation frequency to maintain a sparse information matrix. Unfortunately, not every environment is sufficiently feature-rich to relocalize the vehicle on demand. Similarly, the exteroceptive sensors often have a small FOV and limited degrees of freedom that further reduce the times that the ESEIF can sparsify.

This chapter describes the application of the ESEIF to three-dimensional localization and map building with an autonomous underwater vehicle (AUV). In particular, we consider the ship hull inspection problem that we discussed earlier in Section 1.1.2. A vehicle, equipped with an acoustic imaging sonar conducts three-dimensional surveys of the ship's underwater structure.[1] The hull environment consists of both a sparse set of man-made objects including the running gear and weld lines, as well as biological growth. In the context of SLAM, the reduced sensitivity of the sonar reduces the number of available features. This, together with the sonar's reduced FOV complicate the application of feature-based SLAM and, in particular, the ESEIF.

[1] Throughout the chapter, *vehicle* refers to the AUV that performs the survey. The term *ship* denotes the vessel under inspection.

5.1 Ship Hull Inspection Problem

Governments and port authorities worldwide have a pressing need for frequent inspection of marine structures including jetties, walls, and ships. The United States alone has hundreds of ports that service more than 1,100 million tons of international goods per year with a total waterborne cargo of over 2,300 million tons in the year 2000 [131]. Local and national governmental organizations face the difficult challenge of securing such an enormous amount of cargo. In addition to the goods themselves, though, port authorities also must deal with ensuring the security and integrity of the more than 75,000 vessels that transport these goods [131].

Detailed surveys of a ship's hull, whether to identify structural faults or to confirm that the vessel is free of explosives, are important to guaranteeing the safety of the vessels and harbors. The majority of structural inspections of a ship's hull take place annually, while the ship is in drydock. The process of placing a vessel in drydock is time-consuming and costly and imposes an undue burden on the ship's operator if required only for an inspection that could otherwise be performed in water. Meanwhile, security inspections, particularly those of military vessels and ships carrying hazardous cargo, are necessarily performed in-water by a large team of divers. In order to ensure complete coverage, the divers form a line along the ship's width and swim the length of the hull. This process is challenging as the divers are tasked with locating mines as small as 20 cm in diameter on ships that are typically hundreds of meters in length.[2] The shallow water environment typical of most harbors further complicates surveys as the divers are subject to extremely poor visibility, strong currents, and low clearance between the hull and the seabed. Consequently, accurate surveys are dangerous and time-consuming and it becomes difficult to ensure that the entire structure has been inspected.

In light of these challenges, there is a strong desire to perform surveys with underwater vehicles in place of dive teams and as a supplement for costly drydock inspections. In addition to alleviating risk, underwater vehicles offer several advantages over traditional in-water survey techniques. Whether they are autonomous or remotely operated, the vehicles are highly-maneuverable and can be accurately controlled to perform close-range inspection. Additionally, recent advancements in sensor technology, particularly with regards to acoustic imaging sonars, allow vehicles to acquire high-quality imagery of the hull, even in turbid conditions. Leveraging sensor quality with accurate navigation and control, the goal is to then deploy an underwater vehicle in-situ on a vessel and produce accurate and complete maps that describe the location of targets on the hull. Key to achieving the overall goal of autonomous inspection are accurate navigation of the vehicle relative to the hull and the ability to acquire detailed imagery of the structure.

[2] Liquefied natural gas tankers, for example, may be as large as 300 m in length with a draft of 15 m and a beam of 50 m.

5.1.1 A Brief Overview of Underwater Navigation

The key requirements for autonomous underwater inspection are the ability to produce a high-quality map of a ship's hull and to ensure one hundred percent coverage. A requisite element to achieving both goals is an accurate estimate for the vehicle's pose relative to an arbitrary hull over the course of the survey. We briefly discuss the standard techniques that are employed for underwater vehicle localization and describe their limitations in the context of our desire for on-site hull inspection.

Underwater vehicles are typically equipped with an extensive suite of onboard sensors that provide observations of its motion and pose at an update rate of several hertz [137]. Vehicles typically estimate their position by integrating this motion data in the form of angular rate and linear velocity measurements. These dead-reckoned estimates tend to be highly accurate over short timescales, but small noise in the motion data gives rise to errors that grow with time. In order to compensate for the accumulated error, vehicles supplement motion data with drift-free measurements of their depth, attitude, and heading provided by onboard sensors. Meanwhile, vehicles account for drift in 3D position estimates by periodically measuring their position relative to a network of underwater acoustic beacons as part of an long baseline (LBL) navigation framework. Long baseline navigation relies upon acoustic time-of-flight measurements of the range between the vehicle and two or more fixed transponder beacons to triangulate the vehicle's position. Standard 12 Hz LBL systems yield range data at a rate of 0.1 Hz-1.0 Hz with an accuracy around 0.1 m-10 m [137]. The state-of-the-art in underwater navigation fuses these noisy but drift-free observations with higher bandwidth dead-reckoned estimates to achieve localization to within sub-meter accuracy [138].

Standard LBL navigation is well-suited for most underwater applications but is less than optimal in the context of ship hull inspection. In particular, a LBL system requires the deployment of the network of two or more acoustic transponders at the operating site. After estimating the sound velocity profile,[3] the position of each beacon must be calibrated prior to vehicle operation [138]. The vehicle is then confined to operate within the working range of the network. While these distances can be large (5 km-10 km for 12 kHz LBL systems), the localization accuracy degrades with range and, as a result, the performance is greatest closer to the network [138]. These factors do not preclude the use of LBL navigation for ship hull deployments but, together, are in opposition with the desired goal of flexible, in-situ inspections. Furthermore, the environment typical of these surveys complicates LBL time-of-flight position estimates. In particular, the acoustic signals are prone to a greater degree of multipath compared with standard deployments due to the vehicle's close proximity to the surface and the ship's hull.

Alternatively, we describe a navigation strategy that employs localization and mapping to provide a drift-free estimate of the vehicle's position relative to the hull. We utilize the ESEIF algorithm to jointly build the desired map of the hull, which describes the location of both natural and man-made features on the vessel. The

[3] The sound velocity profile represents a measure of the speed of sound at various depths and is necessary to infer range from time of flight data.

filter then exploits subsequent observations of these targets to yield "absolute" vehicle position and heading fixes.

5.1.2 Underwater Imaging Sonar

In addition to accurate navigation, the second capability that is necessary to generate a thorough map of the hull is a high quality imaging sensor. An optical camera provides high resolution imagery at high frame rates and would be the ideal sensor for surveys. Unfortunately, the underwater environment is not conducive to optical imagery, largely due to the rapid attenuation of light with depth [87]. Poor lighting requires that underwater vehicles provide their own external light source, which then becomes a significant fraction of their power budget. Additionally, the turbidity of the water severely limits the imaged field of view. This problem is further compounded by illumination from the external light source, which yields an increase in backscatter with larger fields of view [62]. Despite these complications, though, a number of autonomous vehicles successfully utilize optical cameras, including the Autonomous Benthic Explorer (ABE) [141] and the SeaBED AUV [115, 35], to name a few. These vehicles typically operate at depth, imaging the sea floor. Near the surface, where ship hull inspection takes place, the conditions are arguably more difficult on account of greater turbidity, cast shadows, and high variability lighting [101].

Sonar sensors are not subject to these same factors and offer an alternative for underwater imaging. The advanced, high frequency sonars currently available produce acoustic images that resemble those of standard optical cameras, at near-video frame rates. While the quality of optical imagery remains superior in ideal conditions, sonars are far less sensitive to the limitations that characterize the underwater environment. Imaging sonars rely upon the acoustic illumination of the scene and are not affected by the lighting conditions nor the presence of particulates in the water. As a result, they function equally well in the often turbid water of harbors where sub-meter visibility prohibits the use of traditional cameras. Unlike pinhole cameras that are invariant to scale, sonars directly provide depth information at the sacrifice of some directional ambiguity. These factors have contributed to the recent popularity of imaging sonars, which have been used for tasks that range from building image mosaics [69, 102] to monitoring fish populations [59]. Both applications rely, in particular, on the Dual Frequency Identification Sonar (DIDSON) high frequency sonar developed by the Applied Physics Laboratory at the University of Washington [9].

5.2 Vehicle Platform

The Hovering Autonomous Underwater Vehicle (HAUV) was initially designed and built by Massachusetts Institute of Technology (MIT) Sea Grant with the assistance of Bluefin Robotics as a platform for close-range underwater inspection [21, 60]. The vehicle has been deployed on numerous ship hulls ranging from the USS Salem, a 215 meter heavy cruiser [132] to a 64 meter Sauro-class Italian naval submarine [132].

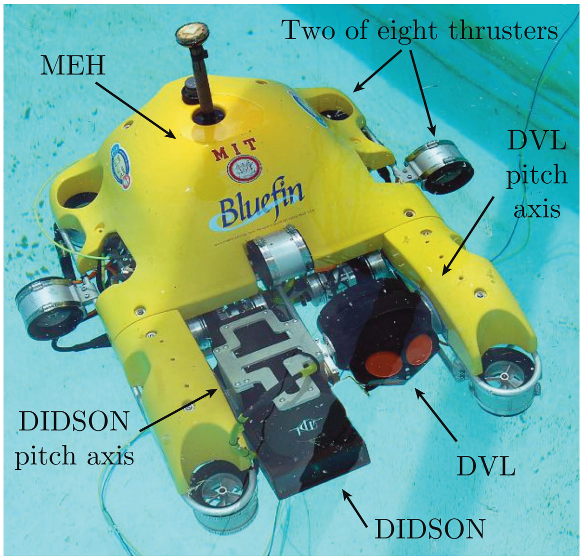


Figure 5-1: Version 1B of the HAUV. The vehicle is roughly 90 cm (L) × 85 cm (W) × 44 cm (H) in overall size and weighs approximately 90 kg. Eight ductless thrusters provide for full maneuverability. At the front of the vehicle are the DIDSON imaging sonar and DVL that can be independently pitched to accommodate changes in hull geometry. The aft section includes the MEH, which houses the IMU, depth sensor, and PC104 stack. An optional fiber optic tether provides real-time topside DIDSON imagery, as well as navigation information.

We conduct our surveys with the second iteration HAUV1B,[4] redesigned through a collaboration between MIT [71] and Bluefin Robotics. Shown in Figure 5-1, the HAUV is relatively small and lightweight compared with other AUVs with dimensions 90 cm (L) × 85 cm (W) × 44 cm (H) and a weight of roughly 90 kilograms. The vehicle is designed as a highly maneuverable inspection platform. The vehicle is equipped with eight brushless DC thrusters that provide a high degree of control for each degree of freedom, allowing the vehicle to conduct forward/backward, lateral, and vertical surveys. The vehicle's primary inspection sensor is a forward-looking DIDSON imaging sonar. Located at the front of the vehicle, the sonar is mounted on its own independently actuated pitch axis, which allows the FOV to be oriented according to the hull geometry. Adjacent to the DIDSON is a Doppler Velocity Log (DVL) that is independently pitched to track the hull. The main electronics housing (MEH) at the aft end houses the primary vehicle computer, along with additional electronics and sensors. A 1.5 kWh lithium polymer battery located below the MEH powers the vehicle.

The HAUV is well-instrumented with a sensor suite that includes an IMU, DVL, compass, depth sensor, and GPS. Table 5.1 outlines the sensor details. The IMU utilizes a ring laser gyro to measure the three-axis, body-referenced angular rates and exhibits a bias of 2°/hour (1σ). Inclinometers yield observations of the vehicle's

[4] Note that throughout the thesis, any mention of the HAUV refers to the second version of the vehicle, unless explicitly stated otherwise.

pitch and roll attitude to within ±0.1° (1σ) accuracy. A magnetic flux-gate compass provides absolute vehicle heading to within a nominal accuracy of 3° (1σ), but is subject to extensive interference induced by the ship during survey. The vehicle measures its three-axis surge, sway, and heave body velocities[5] with a 1200 kHz RDI Workhorse Navigator ADCP DVL that is actuated in pitch to face the hull surface. As a result of the close range, the DVL precision is 0.3 cm/s–0.5 cm/s. In addition to velocity, the DVL records the range to the hull along each of its four beams.

Table 5.1: HAUV sensor details.

Measurement	**Sensor Type**	**Performance**
Depth	Pressure Sensor	Accuracy: ±0.01%
XYZ Linear Velocity	1200 kHz DVL	Precision: 0.3 cm/s–0.5 cm/s, Accuracy: ±0.2%
XYZ Angular Velocity	Ring Laser Gyro	Bias: 2°/hour (1σ)
Roll & Pitch Angles	Inclinometer	±0.1° (1σ)
Heading	Magnetic Compass	Accuracy: ±3° (1σ)

5.2.1 DIDSON Sonar

The DIDSON is the primary sensor used by the HAUV to image underwater structures at close range. The DIDSON is a high-frequency, forward-looking sonar that produces two-dimensional acoustic intensity images at near video frame rates (5 Hz–20 Hz). A novel implementation of beamforming utilizes a pair of acoustic lenses to both focus narrow ensonification beams over the horizontal FOV, as well as to sample the corresponding echos. This yields a set of fixed-bearing temporal signals that correspond to the acoustic return intensities along each beam. By then sampling these profiles, the DIDSON generates a two-dimensional range versus bearing projection of the ensonification echo. Figure 5-2 presents the Cartesian projection of a typical acoustic image of a hull. Shown on the right is a 23 cm diameter cylindrical target and, on the left, a 50 cm rectangular target. The quality is surprisingly detailed for a sonar image, though the resolution is still less than that of an optical camera under suitable conditions. Nonetheless, the quality of detail is sufficiently high to detect targets that are several centimeters in size.

The DIDSON images targets within a narrow field-of-view that spans an angle of 28.8° in the horizontal direction (azimuth), and 12° in the vertical direction (elevation). The range extent depends upon the operating frequency, which trades off lower spatial resolution for increased range as detailed in Table 5.2. In the 1.1 MHz extended-range *detection* mode, the range extends from a minimum of 4.5 meters to a maximum of 40 meters. The acoustic lenses ensonify the 28.8° azimuthal window with 48 distinct beams with a beamwidth of 0.4°, spaced 0.6° apart. The sonar samples each fixed-bearing return along 512 range bins, resulting in an acoustic intensity image with 0.6° resolution in bearing and a range-dependent resolution that varies

[5] Surge, sway, and heave refer to the vehicle's forward, lateral, and vertical motion, respectively.

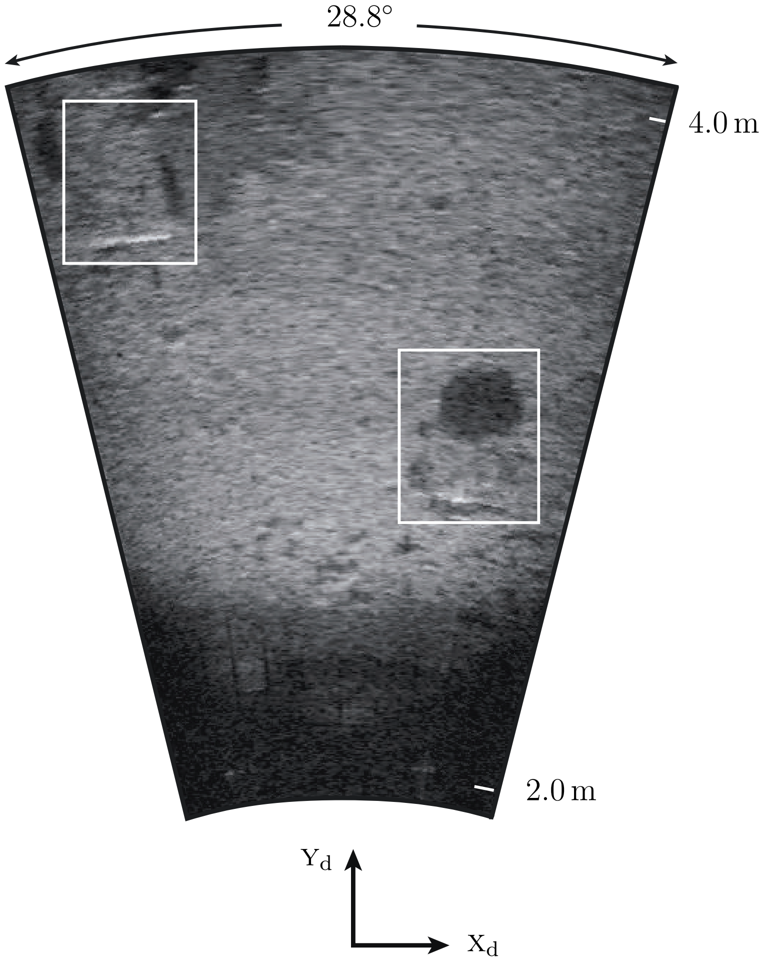
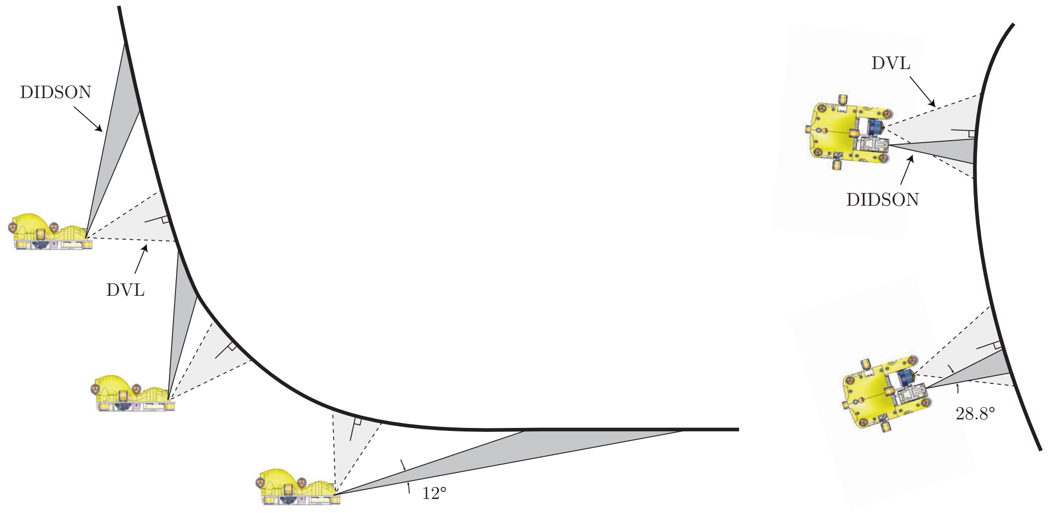


Figure 5-2: The DIDSON acquires two-dimensional acoustic intensity images that are a function of range and bearing that we show mapped into Cartesian space. Visible in the right side of the image is a 23 cm diameter cylindrical target and, on the left, a 50 cm rectangular feature. These targets represent the scale of features that we detect on the hull.

Table 5.2: DIDSON operating modes.

	Detection Mode	**Identification Mode**
Frequency	1.1 MHz	1.8 MHz
Range Extent		
minimum	5.0 m	1.25 m
maximum	40.0 m	10.0 m
Range Resolution		
minimum	8.0 mm	2.0 mm
maximum	78.0 mm	20.0 m
Angular Resolution	0.6°	0.3°



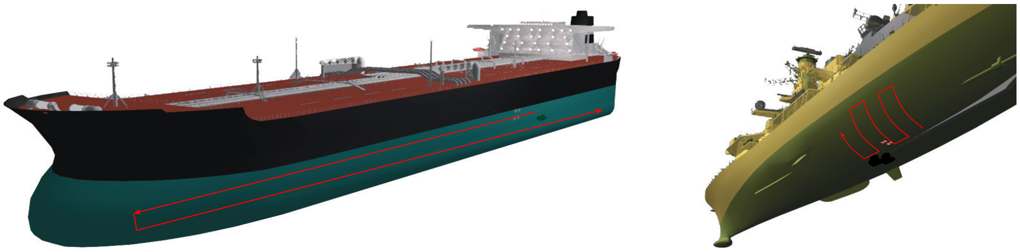
(a) Side View (b) Top View

Figure 5-3: A schematic (not to scale) of hull-relative navigation in which the HAUV maintains a heading orthogonal to the hull surface at a fixed distance. The vehicle servos its pose, as well as the DVL pitch, based upon a planar model for the local hull geometry. The DVL orientation determines the DIDSON's pitch to achieve a suitable viewing angle.

between 8 mm to 78 mm. Alternatively, the 1.8 MHz *identification* mode offers finer resolution over a window that extends from just over 1 meter to 10 meters in range. In this mode, the sonar generates 96 beams of 0.3° horizontal width and produces images with a resolution of 0.3° in bearing and between 2 mm and 20 mm in range. The difference in resolution between the two modes is noticeable when viewing the image in Cartesian coordinates where the pixels are no longer uniform across the image as a consequence of the dependence of resolution on range.

5.2.2 Hull-Relative Survey Pattern

Irrespective of the operating frequency, the sonar's field-of-view is only 12° in elevation. While it can resolve acoustic intensity as a function of range and bearing, it does not disambiguate the elevation angle of the return. Consequently, the DIDSON produces higher resolution images over a larger FOV when the grazing angle relative to the ensonified surface is small. To achieve a suitable viewing geometry, the HAUV conducts surveys as close as possible to the ship, typically on the order of one meter, and pitches the DIDSON to attain a small grazing angle. The pitch angle depends on the local geometry of the hull, which the vehicle models as being locally planar and tracks with the DVL. More specifically, the vehicle estimates the distance to the hull, as well as its hull-relative azimuth and elevation, based upon the ranges to the hull measured along each of the four DVL transducers. The vehicle then navigates at an approximate fixed distance from the vessel while it maintains a perpendicular orientation with respect to the hull. With the DVL servoed to this hull-relative orientation,



(a) Horizontal Survey Pattern

(b) Vertical Survey Pattern

Figure 5-4: The HAUV executes either a horizontal or vertical survey pattern as it inspects each side of the hull in turn. The (a) horizontal survey dictates that the vehicle move laterally along tracklines that extend along the length of the ship. The tracklines are staggered approximately one meter apart from the waterline to the bottom of the hull. When implementing the (b) vertical survey pattern, the vehicle moves in the heave direction along vertical tracklines that extend from the waterline to the bottom of the vessel.

the DIDSON pitches at a slight offset from that of the DVL. The vehicle fine-tunes the sonar's viewing angle based upon the image quality and brightness. Figure 5-3 demonstrates this hull-relative navigation with a schematic on the left that depicts the side view of a survey and, on the right, a top view.

The HAUV inspects each side of the ship according to a predefined, hull-relative survey pattern while maintaining the appropriate viewing geometry. The two standard surveys are comprised of a set of parallel tracklines that extend either vertically or horizontally along one side of the vessel. When following the horizontal survey pattern shown in Figure 5-4(a), the vehicle starts at either the ship's bow or stern at a depth just below the water line. While facing the hull, the vehicle moves laterally along the length of the ship at approximately 20 cm/s and, upon reaching the end of the trackline, dives approximately one meter and continues in the opposite direction along the next trackline. The survey continues in this fashion until the vehicle reaches a particular depth, pre-defined to image the chine (bottom) of the hull. The vehicle executes the mirrored version of the survey pattern in order to inspect the opposing side of the ship. Aside for the short transitions between tracklines, all motion is cross-body (sway). Vertical surveys, depicted in Figure 5-4(b), are much the same with tracklines that extend from the waterline down to the chine. Starting at either the stern or bow, the HAUV heaves downwards until it reaches a certain depth, at which point it transitions laterally and follows the adjacent trackline towards the surface. The vehicle continues in this manner, spanning the ship's length with the set of tracklines and then repeats the same process for the remainder of the hull. The tracklines within both survey patters are spaced to provide roughly 40% overlap between images. Nominally one meter, the actual spacing depends upon the hull geometry and the DIDSON settings.

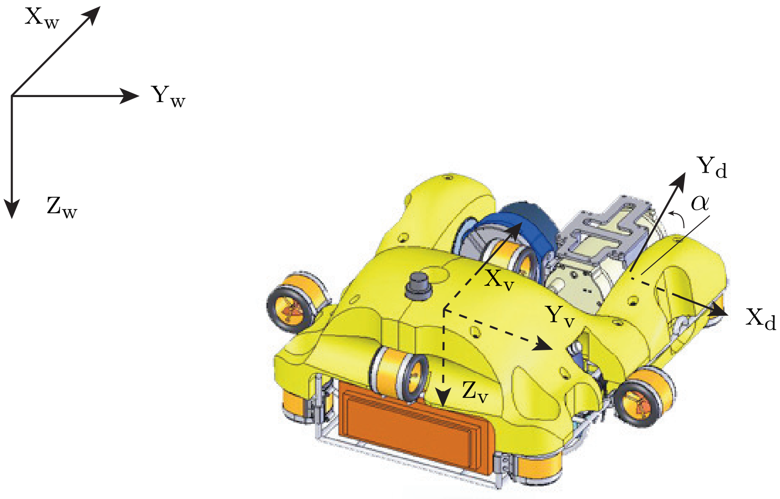


Figure 5-5: We rely on two main coordinate frames to describe the vehicle state. The world frame, $X_wY_wZ_w$ is positioned at the ocean surface with the X_w-axis oriented North and the Y_w-axis East. A separate body coordinate frame is affixed to the vehicle with the X_v-axis directed towards the bow, the Y_v-axis to starboard, and the Z_v-axis down. The $X_dY_dZ_d$ frame at the front of the vehicle denotes the DIDSON coordinate frame with vehicle-relative pitch, α.

5.2.3 Vehicle State Space Model

In modeling the vehicle state, we reference two main coordinate frames, one world frame and one body-fixed frame as shown in Figure 5-5. We assume an inertial world coordinate frame, $X_wY_wZ_w$, located at the ocean surface with the X_w-axis pointing North, the Y_w-axis pointing East, and the Z_w-axis down. This basis serves as the global reference frame for the vehicle pose and the SLAM map.[6] We include a second, body-fixed coordinate frame, $X_vY_vZ_v$, coincident with a stationary point on the vehicle. Consistent with the standard notation [42], X_v points towards the vehicle's bow, Y_v to starboard, and Z_v downwards.

We describe the state of the vehicle by its pose relative to the world frame along with its body-referenced linear and angular velocities. The position of the body-fixed frame, $\mathbf{t}_v = [x\ y\ z]^\top$, along with its orientation, $\boldsymbol{\Theta}_v = [\phi\ \theta\ \psi]^\top$, describe the full six-DOF vehicle pose, $\mathbf{x}_t$. We parametrize the vehicle's orientation as a series of three consecutive rotations, adopting the XYZ-convention for Euler angles in which ϕ denotes vehicle roll, θ designates the pitch, and ψ denotes the vehicle's heading. The state vector also includes the vehicle's linear velocity, $\boldsymbol{\nu}_1 = [u\ v\ w]^\top$, as referenced in the body-fixed frame where u denotes the surge (forward) velocity, v is the sway (lateral) velocity, and w is the heave (vertical) velocity. The vehicle's angular velocity, $\boldsymbol{\nu}_2 = [p\ q\ r]^\top$, consists of the roll, pitch, and yaw rotation rates, respectively, as referenced in the body-fixed frame.

We assign an additional reference frame to the DIDSON. As the sonar samples the

[6] We assume that the ship remains stationary during the inspection. Accommodating a moving vessel would require that we track its motion relative to the inertial reference frame.

acoustic intensity of a three-dimensional scene as a function of range and bearing, the imagery is most naturally parametrized in spherical coordinates. We assign a local coordinate frame to the sonar, $\mathtt{x}_\mathrm{d}\mathtt{y}_\mathrm{d}\mathtt{z}_\mathrm{d}$, that is oriented with the $\mathtt{y}_\mathrm{d}$-axis orthogonal (outwards) to the lens, the $\mathtt{x}_\mathrm{d}$-axis to the right and the $\mathtt{z}_\mathrm{d}$-axis orthogonal to the zero elevation plane. We describe the pitch of the sonar in terms of the rotation of this frame about the $\mathtt{x}_\mathrm{d}$-axis as shown in Figure 5-5. The sonar resolves the range and bearing of the acoustic returns as projections onto the $\mathtt{x}_\mathrm{d}\mathtt{y}_\mathrm{d}$ image plane.

5.3 Navigation with the DIDSON Imaging Sonar

The DIDSON produces high resolution sonar imagery, particularly when operating at close range in the higher-frequency identification mode. The quality of the acoustic imagery is close to that of optical cameras. The similarity suggests that we can treat the sonar as an acoustic camera, leveraging techniques for optical camera-based navigation and mapping for the DIDSON. This section presents an overview of the interpretation of this sonar as a camera, and discusses the corresponding issues as they relate to navigation. In this context, we review some recent work on vision-based techniques for navigation and mapping with the DIDSON. We conclude the section with a description of our SLAM framework, which exploits the sonar as the primary sensor for the ESEIF algorithm.

5.3.1 Acoustic Camera-based Navigation

The sonar-based ship hull inspection problem that we consider here shares a lot in common with the broader field of multiple view geometry. We first discuss work in this area that is particularly relevant to our specific application.

Optical cameras provide rich measurements of the environment, yet, until recently, have seen little use as tools for robotic navigation. The main reason for the limited popularity of cameras is that it is difficult to succinctly parse rich image data and fuse the results with measurements provided by the many other sensors that vehicles rely on for navigation. Recent advances in computer vision, however, have given rise to algorithms that identify salient image data that is amenable for use in a navigation framework. In particular, the scale-invariant feature transform (SIFT) [81] as well as adaptations to Harris corner interest points [56, 90, 91] have introduced image features that are robust to a wide range of changes in scale, illumination, and viewpoint. Consequently, methods exist for detecting and describing interest points that are amenable to re-observations and loop closure.

Once image data has been reduced to a set of interest points, the problem is then to integrate this data in an online navigation framework. The classic formulation of this problem in the computer vision community is that of structure from motion (SFM), which jointly estimates camera position and scene structure based upon interest points common across images [55, 124, 7, 40]. Under the assumption that matching image features correspond to static points in the world, the observations imply constraints on the relative camera motion and the 3D location of these world points. Structure from motion algorithms solve for the set of camera poses and the scene geometry, up to a projective transformation in the case of uncalibrated cameras, that best agree with these constraints. Standard SFM solutions to scene reconstruction [124, 92] operate as batch algorithms, computing the entire set of camera motion and scene geometry states, and are not suitable for online navigation.

Alternatively, a few researchers in the vision community have developed sequential formulations to SFM that provide camera pose and scene estimates in an online fashion [55, 7, 19, 8, 88]. Some algorithms [19, 88] rely on a varying combination of recursive and batch estimation in an attempt to achieve similar performance to bundle adjustment techniques. Harris and Pike [55] utilize a set of Kalman Filters to track the 3D position of points in the scene that they then use to estimate the incremental camera motion. Given a set of matches between image interest points and feature in the "map", their method computes the ego-motion as that which maximizes the joint likelihood of the observation set. Beardsley, Zisserman, and Murray [8] extend upon this approach with work that sequentially updates structure and motion estimates for a range of different problem formulations. The authors consider the case of uncalibrated cameras and describe a method for recovering projective structure. Assuming additional constraints on the camera motion, they describe an algorithm that is capable of estimating structure up to an affine transformation.

Of particular relevance to our work is that of McLauchlan [88], who presents a combined batch/recursive SFM algorithm based upon a novel adaptation to least-squares estimation. McLauchlan considers a non-random, variable-length parameter vector comprised of a set of scene points and camera poses. The algorithm treats each interest point detected within images as noisy measurements over the corresponding scene point and camera pose. He then solves for the structure and motion parameters that maximize the joint measurement likelihood. Under the assumption that the noise is independent and Gaussian, the distribution over the observations is also Gaussian, parametrized by an information matrix that is sparse. The equation that describes the maximum likelihood solution has the same form as that of inference for the canonical Gaussian form (2.18) and the matrix in the least squares expression is the information matrix for the distribution. This matrix is originally sparse and McLauchlan notes that factorization induces fill-in as a consequence of removing measurements (constraints) that model camera motion. The effect is very similar to the density increase that results from marginalizing over robot pose as we discussed in Section 2.5.4.

McLauchlan is concerned with the motion and scene parameters that maximize the likelihood of the measurements. Assuming a uniform prior over these states, the ML estimate is equivalent to the maximum a posteriori (MAP) solution and the algorithm shares a lot in common with the delayed-state EIF [34], as well as with other information filter-based pose graph techniques [27]. In that regard, another option is to treat the camera trajectory as a random vector and track a Gaussian approximation to the distribution over the entire vehicle pose history, conditioned on pose-to-pose camera measurements. Eustice *et al.* [34] propose the Exactly Sparse Delayed-State Filter (ESDSF) whereby they parametrize the Gaussian pose distribution in the canonical form (2.17), in terms of the information matrix and information vector. The ESDSF exploits the epipolar geometry between calibrated image pairs to compute the three angles that parametrize the relative rotation between camera frames, along with the scale-normalized translation. These observations of relative pose serve as primary measurements in an EIF framework. The ESDSF benefits from the authors' fundamental insight that, by maintaining a distribution over vehicle pose history, the corresponding information matrix is *exactly* sparse.[7] The filter is then able to take advantage of this sparsity and track the vehicle pose history in near constant-time as we discussed in Section 2.5.3.

In similar fashion, one can interpret the DIDSON as an acoustic camera and leverage the various tools that exist for optical camera-based navigation. Multiple view techniques, such as SFM or pose graph filters, rely on the epipolar geometry that governs image pairs to establish measurements of the scene structure and the relative camera motion. The imaging geometry that underlies an acoustic sonar, such as the DIDSON, though, is fundamentally different from that of projective cameras. A pair of optical rays for two projective cameras constrain the corresponding scene point and camera centers to lie on the epipolar plane. The nonlinear sonar imaging model, on the other hand, confines the second camera to lie on a torus in three-space, centered about the first camera frame. The corresponding scene point lies on the circular ring that forms the center of the torus tube. The limited FOV in elevation reduces relative camera and scene point locations to a section of the torus. Hence, unlike optical imaging, the epipolar geometry that constrains acoustic camera pairs is highly nonlinear and does not yield a direct observation of the relative transformation between the two poses.

As we describe in detail in Appendix B, we can instead approximate the DIDSON imaging model as a projective geometry and thereby apply standard multiple view techniques. The world point that corresponds to a point in the image is located somewhere along a narrow $|\beta| \leq 6^\circ$ arc at an observed range and bearing. The limited FOV in elevation allows us to approximate the arc projection by an orthographic projection and thereby model the DIDSON as an affine camera. The effect of limited variability in the elevation direction over the scene is analogous to the small depth relief that permits the affine approximation for optical cameras. Given a set of image pairs from two corresponding DIDSON images, we can then exploit affine epipolar geometry to constrain the scene structure as well as the relative pose of the two acoustic cameras. Affine epipolar geometry, though, is invariant to a greater degree of relative camera motion than is the case for perspective projection. In addition to scale ambiguity, the epipolar constraints are unaffected by a rotation of the second camera frame about an axis parallel to the first image as a consequence of the bas-relief ambiguity [70]. Koenderink and van Doorn [70] introduce a novel parametrization of the relative motion that resolves the relative camera position up to a one-parameter family of rotations, thereby isolating the bas-relief ambiguity. Two of the angles that describe this pose-to-pose transformation, along with the relative scale factor between

[7] Recall our earlier discussion in Section 2.5.3 where we demonstrate that the marginalization over old poses induces matrix fill-in. The ESDSF essentially bypasses this component to time projection and, by tracking the entire pose history, preserves the sparsity of the information matrix.

the two images, can be determined directly from an estimate of the fundamental matrix [117].[8]

Given a pair of DIDSON images, then, the affine epipolar approximation provides two observations of the six-DOF transformation between camera frames. The remaining parameter represents the relative distance between the two images and the scene along the projection axis. In the context of the DIDSON model, this direction is the orthographic approximation to the elevation arc. We can not exactly estimate this distance in an absolute or relative manner but can exploit the DIDSON's narrow FOV to restrict its range. The two angular measurements can be fused in a delayed-state Bayesian filter to track vehicle pose, in much the same way as Eustice *et al.* [34] use the five constraints between a pair of perspective views. The success of the latter approach results from the ability to augment the camera-based measurements with observations of absolute attitude and depth provided by onboard sensors with the five relative pose measurements. Unfortunately, with only two observable constraints between each pair of DIDSON images, the framework is less valuable for navigation.

5.3.2 Current Approaches to Sonar-based Ship Inspection

The ability to approximate the DIDSON as an affine camera has not gone unnoticed. At least two other groups have developed algorithms for mapping with an underwater vehicle that use the DIDSON as the primary exteroceptive sensor. Both techniques treat the sensor as an acoustic camera and leverage work from the computer vision community.

Kim *et al.* [69] address the problem of building mosaics of an underwater scene from a collection of DIDSON images. The authors treat the sonar as an affine camera, approximating the acoustic imaging model by an orthographic projection. They assume a planar scene that then induces an affine homography between pairs of images. Image pairs are registered by extracting multi-scale Harris corner features [56] that are matched based upon cross-correlation. They estimate the affine homography between a given image and a reference image in a manner similar to random sample consensus (RANSAC), sampling from the set of feature matches to identify the homography most consistent with the supporting features. Upon estimating the affine homography between consecutive image pairs, the transformations are chained together to compute a mapping that relates each image to a single image plane. They then build the mosaic, using this transformation to map the images onto the reference plane. The mapped images are combined, weighted by their relative illumination, in order to reduce noise and achieve a more uniform illumination pattern across the mosaic.

One limitation of this approach as with other mosaic algorithms is that it relies on the assumption that the scene is well-approximated as planar, which is not generally the case, particularly with regards to hull inspection. At the scale of individual images, the imaging geometry (narrow FOV in elevation, small grazing angle) facilitates the assumption that the scene is locally planar. Imposing this constraint on

[8] The relative scale factor is equal to unity for a pair of orthographic projection cameras.

the entire structure, though, introduces inconsistencies in the mapping that degrade the algorithm's performance as the size of the environment grows. Consequently, the approach does not scale to large environments unless they are sufficiently planar.

Negahdaripour [101] adopts a similar approach to mapping, building a mosaic by estimating the homography between image pairs. The author models the scene as planar and considers both an affine and similarity form of the homography. The latter is more restrictive in that it corresponds to a four parameter transformation between images, in terms of a 2D rotation and translation (isometry) followed by a uniform scaling. The paper briefly describes the mosaicing process, which is based upon an estimate for the similarity homography between successive image pairs from a set of matched Harris corner interest points via RANSAC. The process then builds the mosaic by stitching these homographies together to map each image onto a common plane. As is the case with Kim *et al.*, one drawback of this approach is that the planar scene assumption restricts the algorithm to small environments. Indeed, the paper presents a mosaic constructed from only a handful of images. The error is exacerbated by the similarity approximation to the homography, the accuracy of which depends on the motion of the vehicle between images. While this model may be sufficient for image pairs that are temporally close, it does not hold in general for those that are spatially close but taken from different viewpoints (i.e. images from different tracklines).

5.3.3 Feature-based SLAM with the DIDSON

The two aforementioned algorithms address a similar problem to the one that we are interested in, namely mapping underwater structure with a DIDSON imaging sonar. However, these approaches are concerned with rendering a 2D mosaic rather than a metric map of the scene and do not maintain an estimate of the vehicle pose. The set of image-to-image homographies provide a rank-deficient constraint on the relative vehicle motion. The accuracy of this information in the context of inspecting 3D structure is limited as a result of the planar scene assumption.

One alternative is to generalize the scene to three dimensions and incorporate multiple view geometric constraints analogous to affine structure from motion. Unlike SFM, which provides scene and motion reconstruction up to an affine transformation under our assumed acoustic camera model, we can augment the image data with motion and pose measurements provided by the vehicle's proprioceptive sensors. In particular, measurements of depth, hull-relative velocity, and tilt impose additional constraints on the form of the reconstruction. Fusing this data with the affine epipolar constraints in a Bayesian filter framework then provides an online metric estimate of vehicle pose in much the same vein as the ESDSF [34]. There are a number of benefits to this approach, particularly if we approximate the distribution over poses by a Gaussian. In that case, we can take advantage of what is then a naturally sparse information parametrization to efficiently track the distribution in a scalable manner.

A key component of this "pose graph" approach is establishing correspondence between overlapping image pairs, typically by matching a large number of interest points across the two. With optical imagery, there exist a number of well-established methods for detecting and describing interest points in a manner that is robust to different viewpoints and varying illumination. We have explored similar techniques for DIDSON imagery and have found it difficult to consistently match a sufficient number of features to robustly register image pairs. The fact that the resolution of DIDSON imagery is non-uniform and significantly poorer than that of optical cameras limits robust feature detection, particularly at pixel-level scales. Our experience is consistent with the results of Kim and colleagues, who rely on detections at the third and fourth levels of a Gaussian pyramid to register successive image pairs [69]. This suggests that, while the resolution makes it difficult to identify pixel-scale interest points, it is sufficient to detect larger features within the FOV.

Our approach exploits the capability of the DIDSON in order to detect these large-scale landmarks on the hull. In that regard, we structure the problem in a feature-based SLAM framework whereby we build a three-dimensional map of the environment, using the DIDSON as the primary exteroceptive sensor. Under the true, nonlinear projection model, the sonar imagery provides a measurement of the range and bearing to each target, along with a bound on the elevation. We augment these observations with data from sensors onboard the vehicle that yield measurements of depth, tilt, hull-relative velocity, and angular rates. We fuse this data online, using the Exactly Sparse Extended Information Filter (ESEIF) to track a Gaussian approximation to the distribution over the current vehicle pose and the mapped landmarks. Consequently, we can take advantage of the memory and computational benefits of a sparse information parametrization to better scale to large underwater structures.

5.4 ESEIF for the HAUV

This section describes our approach to feature-based SLAM with the HAUV. We present the algorithm as a form of vision-based localization and mapping filter, employing tools from computer vision in order to use DIDSON image data for estimation with the ESEIF. Adopting a systems-level approach, the algorithm takes advantage of the vehicle's onboard proprioceptive sensor measurements, fusing the available data in a principled manner. As we show, this framework allows us to resolve the ambiguity in the DIDSON imagery to then maintain an online estimate of the six-DOF vehicle pose and map.

At the core of our localization and mapping framework is the ESEIF, which tracks the posterior distribution over the vehicle pose and targets on the hull. The fundamental aspects of the filter are much like the mechanics that we have employed for the hurdles and Victoria Park data sets. As we discuss shortly, the main differences relate to what is now a three-dimensional rather than planar representation of the environment, which requires slightly different filter mechanics. An additional set of subsystems augment the ESEIF and serve largely to transform raw input data into a more salient form for the filter. This input data includes raw vehicle pose and motion measurements, DIDSON imagery of the ship hull, and a set of DVL-measured ranges to the hull. The bulk of the low-level processing of this data is devoted to extracting measurements of the range and bearing to a set of targets, along with an estimate for their elevation relative to the vehicle's reference frame. This component of the system includes a vision-based feature detection process that extracts large-scale blob features from acoustic imagery. Each detection is associated with a range and bearing measurement as well as a bound on its elevation angle with respect to the sonar. We fuse this data with an independent estimate for the local hull geometry in order to disambiguate the elevation. Upon performing data association, ESEIF uses the observations to either update the state estimate or augment the map.

5.4.1 Feature Extraction from Acoustic Imagery

Our objective with underwater acoustic surveys is to demonstrate the effectiveness of the ESEIF algorithm for SLAM in a 3D environment based primarily on sonar imagery. We facilitate this application of localization and mapping by leveraging tools from the vision community to treat the DIDSON as a more traditional sensor. As mentioned earlier, our system relies on the ability to reliably identify large-scale features within images. We then resolve each sonar image into a more succinct set of observations of range and bearing to a comparatively small number of objects in the scene.

Feature Detection

Feature detection is performed by a third-party application developed by SeeByte Ltd. that detects targets within DIDSON imagery as part of a computer-aided detection and classification (CAD/CAC) system [89]. We use the application as a black box tool, but describe its fundamental structure for clarity.

Feature detection relies on a multistage algorithm that identifies, fuses, and subsequently tracks salient image regions. In the first stage, the image is processed in parallel by a number of coarse detection filters, each tuned to respond to different image signatures that are characteristic of discriminating features. One detector segments images into echo, shadow, and background reverberation regions. The detector utilizes k-means clustering to partition the images based on the acoustic return intensity, followed by a Markov random field (MRF) that imposes spatial consistency. A second filter searches for protrusions on the hull by dilating high echos, which can often be attributed to the edge of an object. The bank of detectors also includes filters that identify regions exhibiting high intensity gradients, analogous to large-scale corner features. Each filter is liberal in its detections, resulting in a large number of false positives that is subsequently reduced based upon *a priori* constraints on the size of target observations. The aggregate output of the feature detectors are then combined to form a candidate set of target observations labeled as either shadows or echos. In the case of acoustic imagery, an object's shadow often conveys as much information as its direct return. The next stage of the algorithm attempts to fuse each of these echos with their corresponding shadow by evaluating the similarity of the shadow and echo detections. This fusion step models the similarity of each pair with a Gaussian likelihood that accounts for the relative position and orientation of the shadow. Pairs that meet a likelihood criterion are identified as object hypotheses and passed on to the final tracking stage. This filter tracks detections over several frames, looking for image-to-image object motion that is consistent with the vehicle motion. Objects whose motion is deemed consistent are output as valid observations of a target on the hull.

Computing Feature Observation Data

Consistent with the sonar's imaging geometry, each feature detected within a DIDSON image is associated with an arc in three dimensions that represents the possible source of the acoustic return. This arc is parametrized by a constant range, r, and bearing, θ, along with the elevation, $|\beta| \leq 6°$, corresponding to the DIDSON's FOV with respect to the sonar's reference frame. We consider feature detections to then be a measure of the range, bearing, and elevation of a set of targets on the ship hull. A Gaussian distribution approximates the uncertainty in the range and bearing data, which accounts both for feature detection errors, as well as noise that is induced by the sonar itself. We further assume that the elevation associated with each return is independent of the range and bearing and model the distribution over elevation as uniform over the $|\beta| \leq 6°$ arc.

The sensor model for each target observation, $p\left(\mathbf{z} \mid \mathbf{x}_t, \mathbf{m}\right)$, is then the product of a Gaussian distribution over range and bearing and a uniform distribution over elevation. The ESEIF, though, assumes a fully Gaussian model and does not account for the uniform elevation likelihood. We address this by approximating this uniform distribution by a finite set of particles that span the $|\beta| \leq 6°$ elevation window. The approach resembles that of Davison [24] who utilizes a particle set to model the uniform depth ambiguity of monocular camera rays. Initially, the particles are weighted equally, corresponding to a uniform prior distribution. We subsequently update the weight of each particle based upon an independent estimate for the local hull geometry, which constrains the source of the acoustic return. A separate filter maintains a planar approximation to the local structure based upon DVL observations. The DVL measures the range to the hull along each of its four beams. We fit a plane to the corresponding set of four points via least-squares estimation. The filter treats this as an observation of the local geometry and updates an online estimate of the planar model. Based upon this estimate, the particle tracker identifies the maximum likelihood elevation angle that best supports the hull approximation. We subsequently replace the uniform distribution for elevation angle with a Gaussian model, using the maximum likelihood estimate as the mean. The variance is conservatively set to one third of the elevation. Combining this model for the elevation angle with the direct range and bearing data from the feature detection algorithm, we now approximate each measurement of a target's range, bearing, and elevation as being jointly Gaussian.

5.4.2 ESEIF Architecture

The ESEIF forms the core of the HAUV localization and mapping algorithm, tracking the six-DOF vehicle pose along with the target locations. Fundamentally, the filter is little different from the ESEIF implementation that we employed for the Victoria Park and hurdles applications previously in the thesis. In this case, we take advantage of the acoustic imagery to sparsify the filter, relying on target detections to relocalize the HAUV within the 3D map. The main differences relate primarily to the details of the filter that are specific to this application. Namely, we rely on a modified form of the prediction and measurement update steps largely on account of what is now a six-DOF motion model.

The state vector, $\boldsymbol{\xi}_t = \left[\mathbf{x}_t^\top \; \mathbf{M}^\top\right]^\top$, consists of the AUV state along with a feature-based model of the environment. As described earlier in Section 5.2.3, we model the vehicle state, $\mathbf{x}_t = \left[\mathbf{t}_v^\top \; \boldsymbol{\Theta}_v^\top \; \boldsymbol{\nu}_1^\top \; \boldsymbol{\nu}_2^\top\right]^\top \in \mathbb{R}^{12}$, by its six-DOF position and orientation, $\mathbf{t}_v$ and $\boldsymbol{\Theta}_v$, as well the vehicle's body frame linear and angular velocities, $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$. We describe the map as a set of point features, $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_n\}$, where each $\mathbf{m}_i \in \mathbb{R}^3$ represents the 3D position of a target on the hull.

Time Prediction

We represent the actual vehicle motion by a constant-velocity kinematic motion model. By considering the linear and angular velocities as part of the vehicle state, the continuous-time kinematics can be described according to the general form

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left(\mathbf{x}(t)\right) + \mathrm{G}\mathbf{w}(t) \tag{5.1}$$

where $\mathbf{f}\left(\mathbf{x}(t)\right)$ is a nonlinear, time-invariant function of the vehicle pose and velocity. As a constant velocity model, we assume that the control term is zero, i.e. $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t) \equiv \mathbf{0}) = \mathbf{f}(\mathbf{x}(t))$. The $\mathrm{G}\mathbf{w}(t)$ additive term corresponds to unmodeled or otherwise unknown aspects of the true kinematic model. More specifically, we account for the uncertainty in the velocity component of the model with the noise projection matrix, $\mathrm{G} = [0_{6\times 6} \; \mathrm{I}_{6\times 6}] \in \mathbb{R}^{12\times 6}$, where $0_{6\times 6}$ is a 6×6 matrix of zeros and $\mathrm{I}_{6\times 6}$ is a 6×6 identity matrix. We represent this uncertainty as additive noise and approximate $\mathbf{w}(t)$ as a wide-sense stationary, white Gaussian process with zero mean.

The kinematic motion model in (5.1) is a continuous-time, nonlinear function of the vehicle pose and velocity. In order to perform the time projection step, we then linearize the prediction model about the mean state estimate. Additionally, we convert the model to a discrete-time form for the sake of implementation. Appendix A presents the derivation of the discrete-time, linearized kinematic model, which we represent by the familiar form,

$$\mathbf{x}_{t+1} = \mathrm{F}_t\mathbf{x}_t + \mathrm{B}_t\bar{\mathbf{u}}_t + \mathbf{w}_t, \tag{5.2}$$

where $\bar{\mathbf{u}}_t$ is a function of the mean vehicle state and is analogous to an external control input. The discrete-time expression for the model uncertainty, $\mathbf{w}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{Q}_t\big)$, is approximated by a zero-mean Gaussian random vector with covariance, Q_t.

As before, we view time projection as a two step process of state augmentation followed by marginalization over the previous vehicle state. The ESEIF first grows the state vector to include the new vehicle pose and velocity, $\hat{\boldsymbol{\xi}}_{t+1} = \left[\mathbf{x}_t^\top \; \mathbf{x}_{t+1}^\top \; \mathbf{M}^\top\right]^\top$. Treating the process model as first-order Markov, the corresponding distribution follows from the current posterior, $p\left(\boldsymbol{\xi}_t \mid \mathbf{z}^t, \mathbf{u}^t\right) = \mathcal{N}^{-1}\bigl(\boldsymbol{\eta}_t, \Lambda_t\bigr)$, as in (2.24). For the sake of readability, we restate the canonical Gaussian parametrization for the augmented distribution,

$$p\left(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{M} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\bigl(\hat{\boldsymbol{\eta}}_{t+1}, \hat{\Lambda}_{t+1}\bigr)$$

$$\hat{\Lambda}_{t+1} = \left[\begin{array}{c|cc} \left(\Lambda_{x_t x_t} + \mathrm{F}_t^\top \mathrm{Q}_t^{-1} \mathrm{F}_t\right) & -\mathrm{F}_t^\top \mathrm{Q}_t^{-1} & \Lambda_{x_t M} \\ \hline -\mathrm{Q}_t^{-1} \mathrm{F}_t & \mathrm{Q}_t^{-1} & 0 \\ \Lambda_{M x_t} & 0 & \Lambda_{MM} \end{array}\right] = \left[\begin{array}{c|c} \hat{\Lambda}_{t+1}^{11} & \hat{\Lambda}_{t+1}^{12} \\ \hline \hat{\Lambda}_{t+1}^{21} & \hat{\Lambda}_{t+1}^{22} \end{array}\right] \tag{5.3a}$$

$$\hat{\boldsymbol{\eta}}_{t+1} = \left[\begin{array}{c} \boldsymbol{\eta}_{x_t} - \mathrm{F}_t^\top \mathrm{Q}_t^{-1} \mathrm{B}_t \bar{\mathbf{u}}_t \\ \hline \mathrm{Q}^{-1} \mathrm{B}_t \bar{\mathbf{u}}_t \\ \boldsymbol{\eta}_M \end{array}\right] = \left[\begin{array}{c} \hat{\boldsymbol{\eta}}_{t+1}^1 \\ \hline \hat{\boldsymbol{\eta}}_{t+1}^2 \end{array}\right] \tag{5.3b}$$

The zero off-diagonal elements in (5.3a) express the conditional independence between the map and the new vehicle pose and velocity given the previous vehicle state.

The filter subsequently marginalizes over the vehicle pose and velocity state at time, t, yielding the desired distribution over the state, $\boldsymbol{\xi}_{t+1} = \left[\mathbf{x}_{t+1}^\top \; \mathbf{M}^\top\right]^\top$,

$$p\left(\boldsymbol{\xi}_{t+1} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\bigl(\bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1}\bigr)$$

$$\bar{\Lambda}_{t+1} = \hat{\Lambda}_{t+1}^{22} - \hat{\Lambda}_{t+1}^{21} \left(\hat{\Lambda}_{t+1}^{11}\right)^{-1} \hat{\Lambda}_{t+1}^{12} \tag{5.4a}$$

$$\bar{\boldsymbol{\eta}}_{t+1} = \hat{\boldsymbol{\eta}}_{t+1}^2 - \hat{\Lambda}_{t+1}^{21} \left(\hat{\Lambda}_{t+1}^{11}\right)^{-1} \hat{\boldsymbol{\eta}}_{t+1}^1 \tag{5.4b}$$

The current robot state is now conditionally dependent on the active landmarks.

Measurement Update

The ESEIF incorporates observations of the map and vehicle state into the posterior distribution via the measurement update step. These observations include sonar detections of targets on the hull, $\mathbf{z}_{t+1}^e$, which we interpret as measurements of relative range, bearing, and elevation.[9] The filter treats data from the onboard motion and attitude sensors as observations of the vehicle's state. These proprioceptive measurements, $\mathbf{z}_{t+1}^p$, include hull-relative velocity from the DVL, along with observations of pitch and roll and three-axis angular rates provided by the IMU. We model these

[9] Note the abuse of notation with $\mathbf{z}_{t+1}^e$ where the superscript e denotes exteroceptive measurements and not a measurement time history.

observations as linear functions of the vehicle state,

$$\mathbf{z}^p_{t+1} = \mathrm{H}\mathbf{x}_{t+1} + \mathbf{v}^p_{t+1} \tag{5.5}$$

where $\mathbf{v}^p_{t+1} \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}_p\big)$ denotes white Gaussian measurement noise. Appendix A presents the specific DVL, IMU, and depth sensor measurement models, which represent direct observations of the pose and velocity. The ESEIF assimilates the data into the current conditional distribution, $p\left(\boldsymbol{\xi}_{t+1} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\bar{\boldsymbol{\eta}}_{t+1}, \bar{\Lambda}_{t+1}\big)$ via the standard information filter update step (4.10), which simplifies to

$$p\left(\boldsymbol{\xi}_{t+1} \mid \mathbf{z}^t, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\bar{\boldsymbol{\eta}}_t, \bar{\Lambda}_{t+1}\big) \xrightarrow{\mathbf{z}^p_{t+1}} p\left(\boldsymbol{\xi}_{t+1} \mid \check{\mathbf{z}}^{t+1}, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\check{\boldsymbol{\eta}}_{t+1}, \check{\Lambda}_{t+1}\big)$$

$$\check{\Lambda}_{t+1} = \bar{\Lambda}_{t+1} + \mathrm{H}^\top \mathrm{R}_p^{-1}\mathrm{H} \tag{5.6a}$$

$$\check{\boldsymbol{\eta}}_{t+1} = \bar{\boldsymbol{\eta}}_{t+1} + \mathrm{H}^\top \mathrm{R}_p^{-1}\mathbf{z}^p_{t+1} \tag{5.6b}$$

where the accent on $\check{\mathbf{z}}^{t+1}$ denotes the absence of new exteroceptive measurements. One thing to note is that the matrix, H, is non-zero only for the observed vehicle states. Consequently, the information matrix update (5.6a) does not contribute to any links between the robot state and map. Additionally, the update does not require knowledge of the vehicle mean state due to the linearity of the measurement model.

Shared information between the vehicle and map result from filter updates based upon DIDSON data. The acoustic image detections together with the hull tracking filter yield observations of the relative range, bearing, and elevation to the targets on the hull. The DIDSON measurement model (5.7a) is a nonlinear function of the six-DOF vehicle pose and the landmark location, $\mathbf{m}_i$. We treat noise in the data and model uncertainty as additive noise, $\mathbf{v}^e_{t+1} \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}_e\big)$. Equation (5.7b) is the first-order linearization about the mean robot and feature pose with the sparse Jacobian, H, evaluated at this mean.

$$\mathbf{z}^e_{t+1} = \mathbf{h}(\mathbf{x}_{t+1}, \mathbf{m}_i) + \mathbf{v}^e_{t+1} \tag{5.7a}$$

$$= \mathbf{h}(\check{\boldsymbol{\mu}}_{x_{t+1}}, \check{\boldsymbol{\mu}}_{m_i}) + \mathrm{H}\left(\boldsymbol{\xi}_{t+1} - \check{\boldsymbol{\mu}}_{t+1}\right) + \mathbf{v}^e_{t+1} \tag{5.7b}$$

The ESEIF updates the SLAM canonical form of the distribution to reflect the measurement information through the standard update step,

$$p\left(\boldsymbol{\xi}_{t+1} \mid \check{\mathbf{z}}^{t+1}, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\check{\boldsymbol{\eta}}_t, \check{\Lambda}_{t+1}\big) \xrightarrow{\mathbf{z}^e_{t+1}} p\left(\boldsymbol{\xi}_{t+1} \mid \mathbf{z}^{t+1}, \mathbf{u}^{t+1}\right) = \mathcal{N}^{-1}\big(\boldsymbol{\eta}_{t+1}, \Lambda_{t+1}\big)$$

$$\Lambda_{t+1} = \check{\Lambda}_{t+1} + \mathrm{H}^\top \mathrm{R}_e^{-1}\mathrm{H} \tag{5.8a}$$

$$\boldsymbol{\eta}_{t+1} = \check{\boldsymbol{\eta}}_{t+1} + \mathrm{H}^\top \mathrm{R}_e^{-1}\big(\mathbf{z}^e_{t+1} - \mathbf{h}\left(\check{\boldsymbol{\mu}}_{t+1}\right) + \mathrm{H}\check{\boldsymbol{\mu}}_{t+1}\big) \tag{5.8b}$$

Sparsification

Sparsification in the context of hull inspection is less straightforward. Due to the DIDSON's limited FOV, the vehicle observes only a small number of features at any

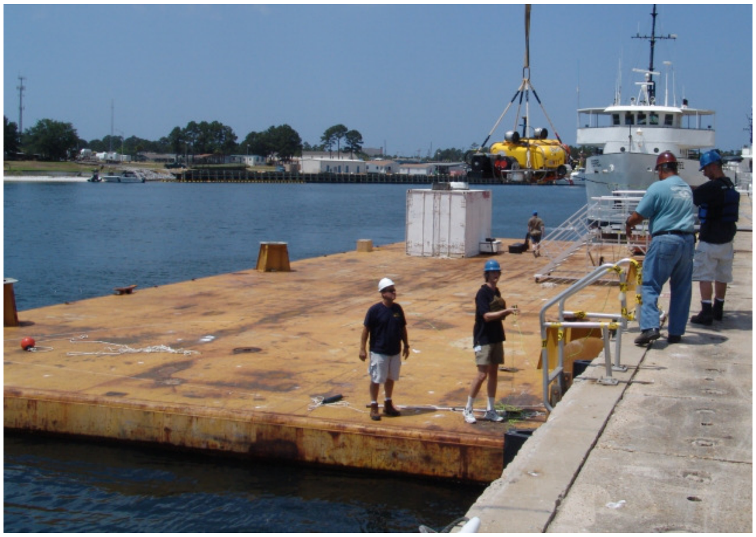


Figure 5-6: The first version of the HAUV being lowered into the water to survey the barge during AUVFest. The barge is 13.4 m from port (side nearest to the pier) to starboard and 36.2 m from bow to stern.

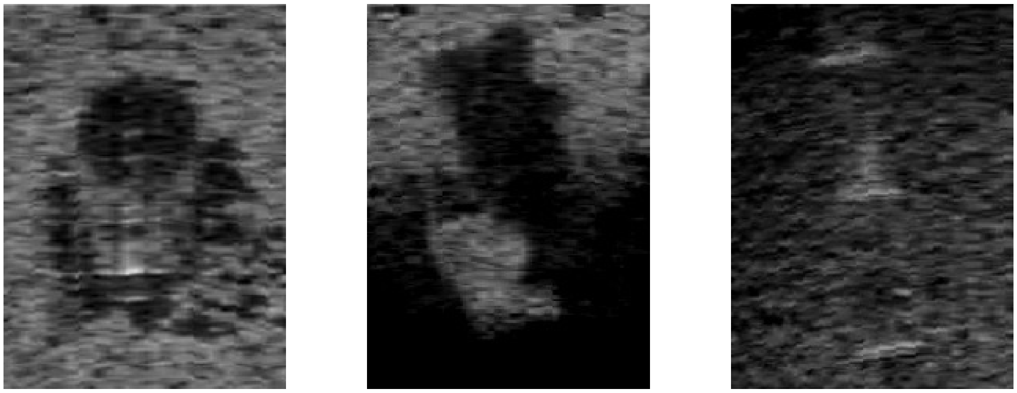
one time. During the sparsification process, we tend to relocalize the vehicle based upon all available observations. We supplement this data with the measurements of the vehicle's velocity, attitude, and depth as measured by the proprioceptive sensors. The marginalization and relocalization components are identical to the forms described in Section 4.2.1 with $\mathbf{z}_\beta = \left\{\mathbf{z}^e_{t+1}, \mathbf{z}^p_{t+1}\right\}$.

5.5 Results

We apply our ESEIF localization and mapping architecture to survey a series of man-made and natural targets located on the hull of a large barge. The vehicle performs an autonomous inspection of the structure independently of our algorithm, which we subsequently apply to post-process the data. In this section, we first describe the experimental setup. We then present the results of applying our system architecture based upon both hand-picked targets as well as features that are automatically detected within the images.

5.5.1 Experimental Setup

Both the first and second versions of the HAUV participated in a series of deployments as part of the 2007 AUVFest at the Naval Surface Warfare Center (NSWC) in Panama City, FL. The focus of the ship hull inspection experiments was a barge of length 36.2 meters and width 13.4 meters that was moored to a pier on its port side, shown in Figure 5-6. Approximately 30 targets were distributed over the underside of the barge and their position measured by a team of divers. Among these targets were several 50 cm × 30 cm rectangular ("ammo box") targets and 23 cm diameter cylindrical ("cake") targets of the form previously shown in Figure 5-2, along with



(a) Cylinder Target (b) Box Target (c) Brick Targets

Figure 5-7: Thumbnail acoustic images of the three types of targets that were manually affixed to the hull.

15 cm × 7 cm brick-shaped targets. Figure 5-7 presents thumbnail DIDSON images of these three targets. In addition to these features, the hull was littered with both man-made as well as natural targets, most of which are clearly visible in the sonar imagery.

Over the course of the experiment, the two vehicles spent more than thirteen hours collecting high-resolution imagery of the entire barge. We consider a survey conducted by version 1B of the HAUV. The 45 minute mission consists of four overlapping horizontal surveys of the bow, stern, port, and starboard sections of the hull. The vehicle starts the mission near the aft-starboard corner of the barge and first surveys most of the stern with the exception of the corners. The vehicle then proceeds to image the port and starboard sides, followed by the bow. The HAUV moves laterally along tracklines that span the width (for the stern and bow surveys) and length (for the starboard and port surveys) of the barge at a velocity of 25 cm/s. Throughout the survey, the DVL is positioned vertically upwards at the hull and the DIDSON is oriented at just over 20° from horizontal to achieve a suitable grazing angle with the hull. Over the duration of the nearly 45 minute mission, the HAUV collected about 4200 acoustic images of the bottom of the barge.

5.5.2 Experimental Results

We consider two different implementations of the sonar-based localization and mapping architecture. In an effort to specifically analyze the performance of the ESEIF in this domain, we first apply the algorithm based upon hand-selected features. This allows us to decouple the effects of the particular feature detector, which is secondary in the thesis. We reduce the batch of roughly 4200 DIDSON images to a set of just over 100 in which we manually identify observations of both natural and synthetic features. Each detection provides a measure of the relative range and bearing to a target on the hull that is subject to the DIDSON 12° elevation ambiguity. We resolve the ambiguity in elevation by independently tracking the local geometry of the hull based upon the range data from the DVL as we described at the end of Section 5.4.1. The resulting measurement data serves as observations of the environment for the ESEIF algorithm.

We implement the ESEIF as described to fuse the motion and pose observations with the manually detected sonar measurements. The filter implements the ESEIF sparsification strategy to maintain a bound of five active features and relocalizes the vehicle within the map based upon all available measurements, i.e. $\mathbf{z}_\beta = \mathbf{z}_t$ and $\mathbf{z}_\alpha = \{\}$. As a basis for comparison, we concurrently apply the localization and mapping algorithm with the standard feature-based EKF estimator in place of the ESEIF.

Figure 5-8 presents a bird's-eye view of the final estimates of the map. The plot compares the map built with the ESEIF with that of the "gold standard" EKF as well as the ground truth target locations as measured by the divers. Both the ESEIF and EKF maps are aligned with the barge's hawse holes[10] based upon a least squares estimate for the transformation. The uncertainty ellipses correspond to the three sigma confidence bounds associated with the ESEIF map estimates. Note that these intervals capture each of the EKF target positions, but not the ground truth location of every feature. We find the same to be true of EKF-based map estimates and believe that the disagreement is largely due to the difficulty in accurately measuring the true position of the targets on the hull. Additionally, the ground truth data indicates that there are several targets that neither the EKF nor the ESEIF-based algorithms incorporate into their respective maps. An inspection of the images associated with these regions of the hull according to the ESEIF pose estimates suggests that these features broke free from the hull. While this does not offer conclusive proof, it is in agreement with divers' claims that targets had broken free.

Meanwhile, we assess the 3D quality of the map based upon the depth of the mapped features. Figure 5-9 presents a side view of the ESEIF map from the barge's port side. Ground truth data regarding the draft profile of the barge is unavailable but, based upon the vehicle's depth measurements and the DVL ranges to the hull, we estimate the draft to be 1.5 m. In comparison, the mapped features exhibit a mean depth of 1.63 m and a standard deviation of 8.7 cm. The synthetic targets are not flush with the hull and their vertical extent largely accounts for this offset.

In order to confirm that the ESEIF sparsification strategy does not induce overconfidence in the state estimates, we compare resulting uncertainty with that of the EKF. The metric is identical to that employed in Sections 3.4 and 4.3. Specifically, we compute the ratio between the determinant of each feature's sub-block of the covariance (inverse information) matrix as maintained by the ESEIF with that of the EKF. On a log scale, a ratio greater than zero implies a conservative estimate for the uncertainty with respect to the EKF while negative ratios suggest overconfidence. We plot a histogram over these ratios in Figure 5-10. As the results described in Section 4.3 reveal for both simulation as well as with experimental data, the plot confirms that the ESEIF preserves estimator consistency relative to the EKF.

[10] The barge is equipped with four 0.5 m diameter openings that extend through the hull, two near the bow and two amidships. We also refer to these as "thru-hulls".

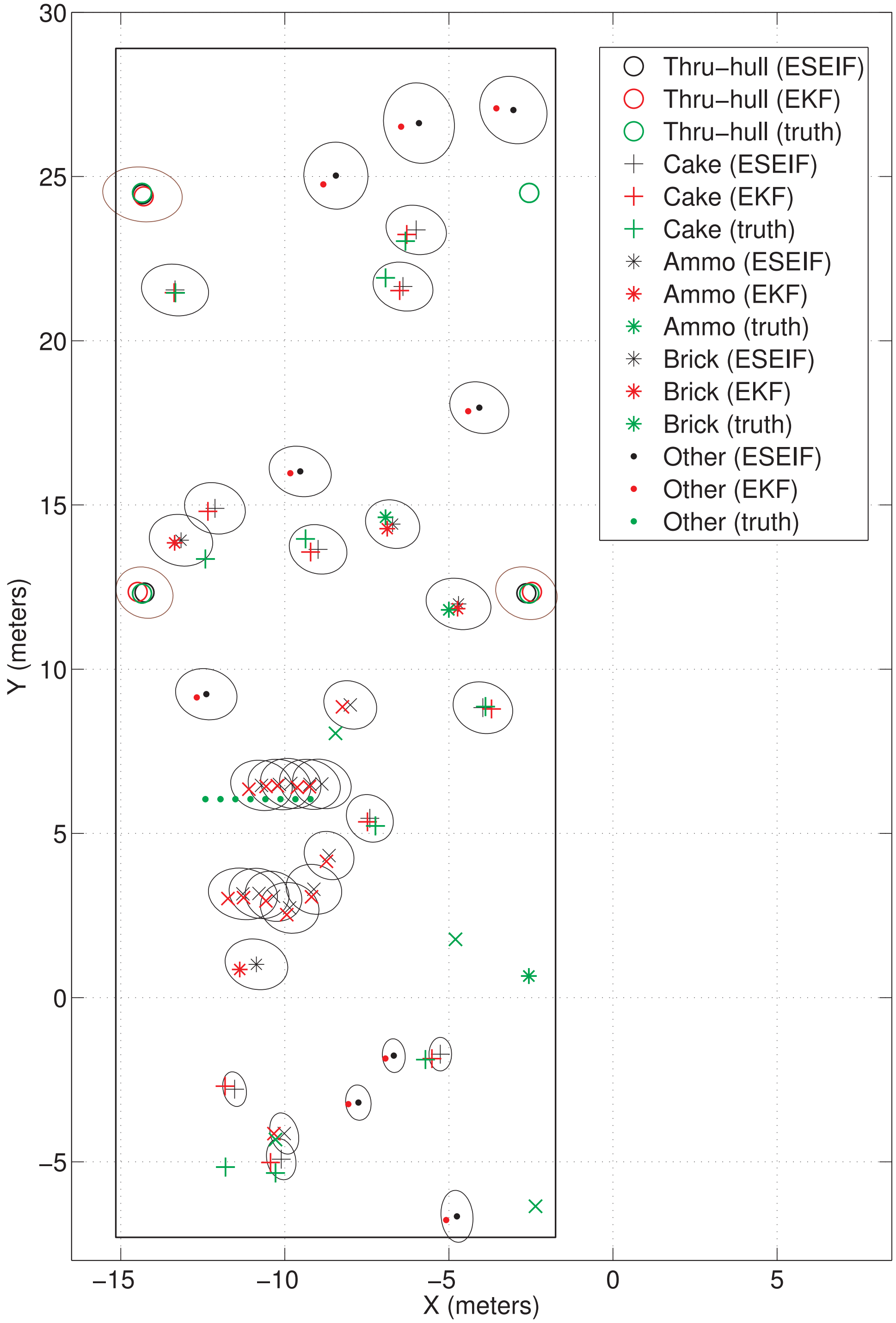


Figure 5-8: Overhead view of the ESEIF map of the barge based upon hand-picked image features. The plot includes the EKF estimates for the feature locations, as well as a measure of ground truth. Targets shown in black comprise the ESEIF map while the EKF map is shown in red and the ground truth in green. The ellipses centered at each feature denote the three sigma uncertainty bounds maintained by the ESEIF.

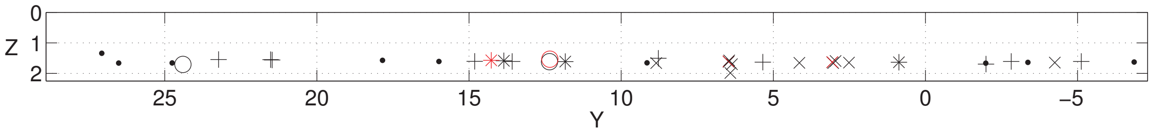


Figure 5-9: A side view of the ESEIF map from the barge's port side. Note that the plot renders features with two colors to help discern between targets that overlap under this projection. While there is no ground truth data regarding the depth of the targets, the DVL ranges to the hull suggest a uniform hull draft of 1.5 m. The mean feature depth as estimated by the filter is 1.63 m with a variance of 8.7 cm. The variation from our DVL-based estimate of the barge's draft is largely due to the three-dimensional structure of the targets, which we model as point features.

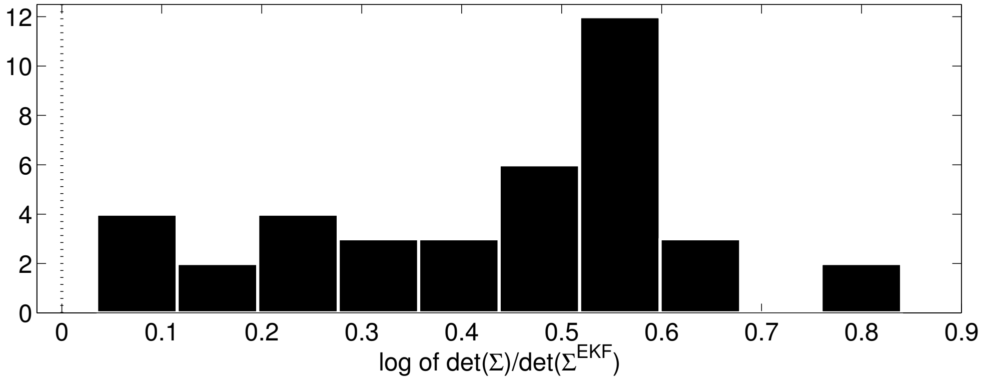


Figure 5-10: A histogram plot comparing the ratio of feature uncertainty as estimated by the ESEIF with that of the EKF. Ratios greater than zero are indicative of conservative confidence intervals while negative values indicate overconfidence.

Our second application of the localization and mapping algorithm considers the same barge survey, but with automatic feature detection. Applied to the DIDSON imagery, the SeeByte computer-aided detection algorithm yields just over 2,000 detections over the course of the survey. The detector fires on both the synthetic, mine-like targets, as well as natural features on the hull, the latter of which make up most of the detections. We then sub-sample this set by a factor of two in order to reduce clutter within images. The result is a list of salient feature detections from which we extract the corresponding range and bearing observations. Note that the detections consist only of pixel locations and are not accompanied by a descriptor. The ESEIF fuses these range and bearing measurements with DVL range to the hull and the vehicle's proprioceptive data as before. In this case, we utilize the sparsification routine in an attempt to maintain a bound of 20 active landmarks. The algorithm again reserves the maximum number of observations to relocalize the vehicle.

Figure 5-11 presents an overhead view of the final ESEIF map, along with the ground truth target positions. For the sake of visualization, we manually identify feature detections that correspond to man-made targets and display them with their corresponding marker as in Figure 5-8. The dark green point features on the right side of the map denote detections of the barge edge. The figure depicts the three sigma uncertainty ellipse for these targets but omits the ellipses associated with the other features merely for aesthetic purposes. Similar to the map generated based upon hand-picked features, many of the ground truth target locations lie within the estimated confidence intervals. One notable exception is the “cake” target (indicated by the plus sign) near the bow of the barge (top of the figure). While the uncertainty ellipse associated with one cake feature captures the ground truth position, the adjacent target lies outside corresponding feature’s confidence interval. Additionally, the filter yields duplicate instantiations for both of these features. This effect is a result of an error in data association.

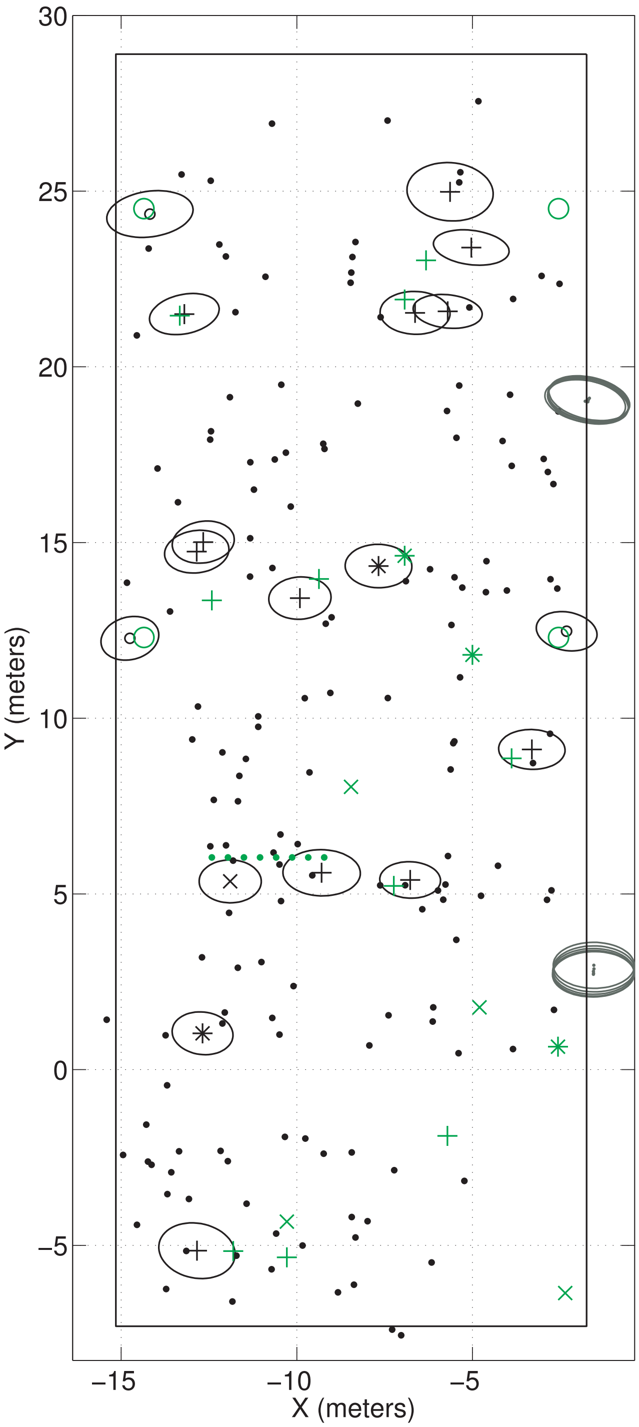


Figure 5-11: A bird's-eye view of the final ESEIF map based upon features automatically detected within the DIDSON images. The plot includes the ground truth target locations as estimated by the divers. For visual clarity, we render the three sigma uncertainty bounds only for mapped features that correspond to the man-made targets. The dark green points with their corresponding ellipses that run down the starboard side denote the edge of the hull.

Chapter 6

Conclusion

There is an increasing demand for robotic vehicles that are capable of sustained, long-term autonomous operation in complex environments, whether on the surface of Mars or underwater. One skill that is necessary to achieve this autonomy is the capability to accurately navigate in large environments that are unknown *a priori*. In the mid-1980's, Simultaneous Localization and Mapping (SLAM) was proposed as a potential solution that actively exploits the structure of the world for localization. The robotics community has since devoted a lot of attention to the SLAM problem. In the process, researchers have helped to answer a number of questions that are fundamental to the coupled problem of localization and mapping. Exemplifying this progress are a number of capable algorithms that have proven successful in environments that range from indoor office-like buildings to highly complex outdoor and underwater settings.

Over the past decade, SLAM has grown from a relatively small discipline to emerge as one of the fundamental problems within robotics science. However, despite extensive contributions towards a solution, many key issues remain unresolved. Foremost among them is the need for an efficient, consistent estimation framework that is capable of easily scaling to large environments. This thesis attempts to answer this question with an efficient variation on the SLAM information filter. We described a feature-based localization and mapping algorithm that leverages the benefits of a sparse parametrization to the posterior, in order to perform estimation in near constant time. The novel aspect of our Exactly Sparse Extended Information Filter (ESEIF) is a sparsification strategy that preserves the consistency of the Gaussian posterior.

6.1 Thesis Contributions

The goal of this thesis is to help answer some open questions that pertain to the localization and map-building problem. To that end, the thesis makes the following contributions to the robotics literature:

- *Sparse approximations of the canonical formulation to the SLAM posterior that impose conditional independence induce inconsistent state estimates.*

We began with an in-depth look at the inherent structure of the information form of the Gaussian SLAM distribution. While the canonical formulation exhibits a natural structure that is relatively sparse in the case of feature-based SLAM, the information matrix is, nonetheless, completely populated. Filtering algorithms that take advantage of the efficiency of a sparse parametrization must approximate the distribution by a posterior that is truly sparse. We derive the SEIF sparsification strategy from the perspective of the independence assertions of the GMRF. The derivation shows that the SEIF approximates the conditional independence between the robot pose and most landmarks in order to enforce a sparse information matrix. We present a detailed analysis of the SEIF sparsification rule through both controlled LG simulations, as well as a pair of nonlinear datasets. The results reveal that the SEIF yields an inconsistent posterior that exhibits overconfident estimates for the map compared with the standard EKF. We present a modified form of the conditional independence approximation that induces far less inconsistency, but sacrifices computational efficiency.

- *The Exactly Sparse Extended Information Filter (ESEIF) provides an alternative formulation to sparse, feature-based SLAM information filters that maintains a sparse representation without inducing an inconsistent posterior.*

  The primary contribution of the thesis is the Exactly Sparse Extended Information Filter (ESEIF) as a scalable localization and mapping framework that maintains consistent state estimates. The ESEIF takes advantage of insights into the canonical formulation to the Gaussian in order to achieve an efficient SLAM information filter that preserves consistency. The principal component of the ESEIF is an alternative sparsification strategy that maintains a modified form of the SLAM posterior that forgoes some temporal information in order to maintain an exactly sparse information matrix. In this manner, the ESEIF employs a distribution that is conservative with respect to the Gaussian approximation of the state distribution. We confirm the consistency of the ESEIF on a set of LG simulations, based upon a comparison with the optimal KF. We further demonstrate the consistency and performance properties of the ESEIF on a pair of benchmark nonlinear datasets.

- *We incorporate the ESEIF in a three-dimensional, six-DOF framework for underwater localization and mapping with an acoustic imaging sonar.*

  The final contribution of the thesis is an extension of the ESEIF algorithm to the problem of ship hull inspection with an AUV equipped with an imaging sonar. We describe a framework for underwater, feature-based SLAM that fuses sonar target detections with observations of vehicle motion and pose provided by onboard sensors. The ESEIF serves as the state estimation engine that maintains consistent estimates for the vehicle pose and the set of targets.

6.2 Assumptions and Failure Modes

The thesis imposes certain assumptions in making these three contributions. These assumptions affect the scope of our claims regarding filter consistency and performance, and give rise to several failure modes that afflict the ESEIF algorithm. In this section, we elaborate upon these assumptions and describe some of the ESEIF's failure modes.

- The ESEIF is a variation on the EKF and is, therefore, subject to the same well-known limitations of the Gaussian approximation to the SLAM posterior. For one, we rely on the assumption that the noise that corrupts the vehicle motion and sensor measurements is Gaussian. In general, though, the noise is non-Gaussian and, in many cases, multimodal, and does not accurately account for systematic modeling errors. Secondly, the Gaussian representation of the posterior depends upon a linear approximation to the process and measurement models, about the current mean estimate. The accuracy of this approximation is sensitive to the nonlinear structure of these models and the quality of the mean estimate, particularly that of the vehicle's heading. Linearization errors degrade the validity of the Gaussian model and can lead to inaccurate covariance (uncertainty) estimates and, over time, an inconsistent posterior [65, 15, 4]. Thus, the consistency of the ESEIF relative to the standard EKF is insufficient to guarantee that the resulting Gaussian approximation to the posterior remains consistent with respect to the true distribution.

- The ESEIF sparsification step relies upon the ability to relocalize the vehicle within the map, as necessary. This assumes that there are a sufficient number of measurements at any one time step to estimate the vehicle pose. While this assumption proved valid in the examples described within the thesis, its validity depends both on the structure of the environment, as well as the nature of the vehicle's exteroceptive sensors. If the distribution of features is sparse, or the sensor's field-of-view is limited, the measurements may not be rich enough for sparsification. In this case, we propose a batch, delayed-state process in which the filter estimates the robot pose over a sequence of time steps.

- The application of the ESEIF to ship hull inspection suffers from a related failure mode. Specifically, our feature-based map parametrization supposes that the hull contains a number of large, high quality targets and that these targets are detectable within sonar imagery. This assumption is typically valid largely due to the marine growth on the ship but, in the case of a relatively clean hull, may be difficult to satisfy. Furthermore, the estimation algorithm assumes that the vessel under survey is stationary during the inspection. In-situ deployments at sea typically invalidate the framework and require that we additionally account for the motion of the ship-relative map reference frame with respect to an inertial coordinate frame.

6.3 Future Research Directions

Several open problems remain, some of which are specific to the ESEIF algorithm or, more generally, to the localization and mapping problem. Others are even broader in scope and concern bigger picture issues related to autonomy. We discuss some of these issues below and offer recommendations for future research.

- A remaining limitation of the canonical parametrization of the Gaussian is the difficulty in performing inference. Mean estimation is equivalent to solving a sparse set of linear equations, which allows us to leverage extensive work in sparse solvers. Data association, on the other hand is not as straightforward. We have discussed approximate strategies for estimating the marginal distributions that establish the likelihood of correspondences, but a more robust solution remains an open problem. This is true not only for the canonical representation, but for the general SLAM problem.

- The ESEIF sparsification strategy introduces a dependence between the vehicle pose and a limited set of neighboring landmarks. Subsequent measurement updates induce additional links to other nearby features and, upon the next sparsification routine, this active map forms a clique in the MRF. This process continues as the vehicle explores the environment. The tendency for cliques to form among neighboring features suggests an overall graph structure that is akin to a submap decomposition of the environment. A detailed comparison between the ESEIF formulation and submap algorithms would provide useful insights into the implications of ESEIF sparsification. For example, if the ESEIF does yield a partitioning of the space, "what is the effect of the bound on the number of active landmarks?", "to what extent does the resulting structure depend upon the vehicle's motion policy?", etc.

- There are several open problems in the context of autonomous ship hull inspection. We have demonstrated the ability to perform SLAM based upon acoustic imagery with an *offline* implementation of the ESEIF in Matlab. The first step towards using the ESEIF as a localization and mapping tool is an online implementation on the HAUV.

  An interesting direction for future research is the coupling of SLAM with planning and control for the HAUV. Hull inspection, particularly in the case of mine detection, demands 100% coverage, together with accurate navigation and mapping in as short a time period as possible. By coupling the SLAM filter with the path planning and control systems, we can better satisfy these constraints. Path planning plays an integral role in the mapping performance of SLAM in terms of coverage rate, overall coverage, and accuracy. At the same time, the vehicle's ability to execute the plan depends directly on the performance of the controller. The SLAM localization estimates, meanwhile, provide pose data that can be exploited to improve the control accuracy.

Appendix A

Implementation Details

Throughout the thesis, we utilize both simulations as well as real-world datasets to analyze the characteristics of the ESEIF. This addendum offers a detailed description of the various filter components, elaborating on the specifics of the measurement and motion models that we employ. We first describe the linear Gaussian (LG) simulation experiments, followed by a description of the models for the 2D Victoria Park and hurdles experiments. We conclude with a detailed discussion of the six DOF underwater vehicle models.

A.1 Linear Gaussian Simulations

Sections 3.4.1 and 4.3.1 utilize a series of linear Gaussian simulations to analyze the effects of the different sparsification routines on the SLAM posterior. Within each simulation, the robot executes a series of counterclockwise loops within an environment that is uniformly distributed with point features. The vehicle motion is purely translational and, thus, linear in the state. The exteroceptive observations consist of measurements of the relative position of neighboring features. We limit the field-of-view to a maximum range of $r_{\max}$, and bound the number of concurrent observations at $m_{\max}$. Table A.1 defines the specific settings for the two sets of simulations.

We represent the two-DOF vehicle and feature states by their Cartesian position, $\mathbf{x}_t = [x_t \; y_t]$ and $\mathbf{m}_i = [x_i \; y_i]$. The linear motion model is of the form,

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathrm{F}\mathbf{x}_t + \mathbf{u}_t + \mathbf{w}_t \\ \begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \mathbf{w}_t,\end{aligned}$$

where $\mathbf{w}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{Q}_t\big)$ is white Gaussian noise. We represent each of the $j \leq m_{\max}$

Table A.1: Linear Gaussian simulation parameter settings.

Parameter	Section 3.4.1	Section 4.3.1
Environment size	45×45	70×70
Number of features	60	375
$r_{\max}$	10	10
$m_{\max}$	3	4
Γ_a	6	6

$$\mathrm{Q}_t = \begin{bmatrix} 0.0225 & 0.010 \\ 0.010 & 0.0225 \end{bmatrix} \qquad \mathrm{R}_t = \begin{bmatrix} 0.040 & 0.010 \\ 0.010 & 0.040 \end{bmatrix}$$

concurrent measurements by the linear model

$$\begin{aligned} {}^{j}\mathbf{z}_t &= \mathrm{H}_j \boldsymbol{\xi}_t + \mathbf{v}_t \\ &= \begin{bmatrix} -\mathrm{I}_{2\times 2} & 0_{2\times l} & \mathrm{I}_{2\times 2} & 0_{2\times p} \end{bmatrix} \boldsymbol{\xi}_t + \mathbf{v}_t \\ &= \begin{bmatrix} x_i - x_t \\ y_i - y_t \end{bmatrix} + \mathbf{v}_t. \end{aligned}$$

The Jacobian matrix, H_j, is non-zero only at columns that correspond to the vehicle pose and the observed landmark. We corrupt the measurements with white Gaussian noise, $\mathbf{v}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}_t\big)$.

Meanwhile, the relocalization function (4.4) that we employ for ESEIF sparsification is a simple extension of the above measurement model,

$$\begin{aligned} \mathbf{x}_t &= \mathrm{G}\mathbf{M}_t + \mathbf{z}_\beta + \mathbf{v}_t \\ &= \begin{bmatrix} 0_{2\times l} & \mathrm{I}_{2\times 2} & 0_{2\times p} \end{bmatrix} \mathbf{M}_t + \mathbf{z}_\beta + \mathbf{v}_t \end{aligned}$$

We specify the motion and measurement noise parameters in Table A.1.

A.2 Hurdles Dataset

The hurdles experiment was conducted in an indoor gymnasium comprised of four adjacent tennis courts, shown within Figure A-1. A total of 64 track hurdles were positioned at various points along the baselines of each court, which allow for an easy measure of their ground truth positions. The overall size of the environment is 57 meters by 25 meters. An iRobot B21r wheeled robot, equipped with a forward-looking SICK laser range finder, was driven in a series of loops throughout the course.

We assume that the robot and map lie in a planar environment. We assign an arbitrary inertial coordinate frame, $\mathrm{x_w y_w z_w}$, and represent the vehicle state as the position, (x_t, y_t), and orientation, θ_t, of a body-fixed reference frame, $\mathrm{x_v y_v z_v}$. The right-handed body-fixed frame is oriented with the $\mathrm{x_v}$-axis aligned with the vehicle's forward direction and $\mathrm{y_v}$-axis to the left. Attached to the body is the exteroceptive sensor coordinate frame, $\mathrm{x_s y_s z_s}$. Figure A-2 provides a general model schematic.

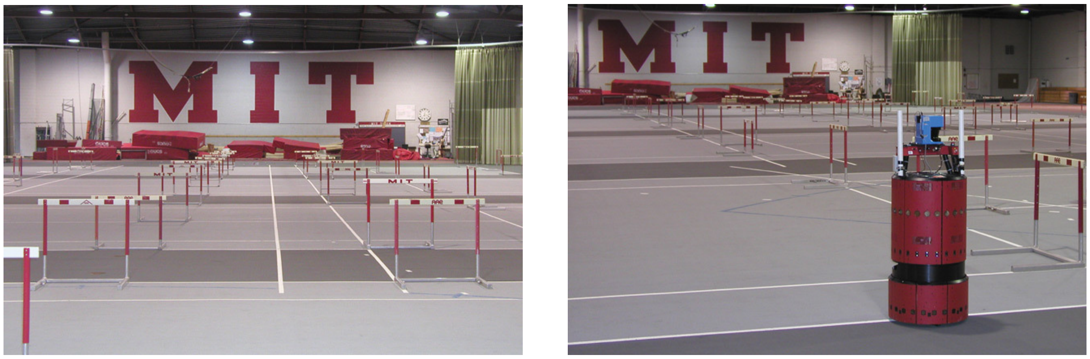


Figure A-1: The experimental setup for the hurdles dataset. The environment consists of 64 track hurdles positioned on the baselines of four adjacent tennis courts. The iRobot B21r robot is equipped with a laser scanner mounted near the base of the vehicle that provides measurements of range and bearing to the hurdle legs within its field-of-view. Note that the second laser scanner affixed at the top of the robot was not used during this experiment.

A.2.1 Motion Model

We describe the motion of the synchronous-drive B21r with a discrete-time, kinematic motion model (A.1). The forward velocity, v_t, and angular velocity, w_t, serve as the control inputs, $\mathbf{u}_t = [v_t \; w_t]^\top$, the former of which is derived from the wheel rotations. The additive term represents uncertainty in the model due to factors such as wheel slip and is modeled as white Gaussian noise, $\mathbf{w}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{Q}_t\big)$.

$$\mathbf{x}_{t+1} = \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_t\big) + \mathbf{w}_t$$

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + v_t \Delta t \cos(\theta_t) \\ y_t + v_t \Delta t \sin(\theta_t) \\ \theta_t + w_t \Delta t \end{bmatrix} + \mathbf{w}_t \tag{A.1}$$

A first-order Taylor's series expansion of the kinematics about the current mean pose estimate, $\boldsymbol{\mu}_{x_t}$, yields the linear approximation to the motion model that we employ for the filter prediction step,

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_t\big) + \mathbf{w}_t \\ &\approx \mathbf{f}\big(\boldsymbol{\mu}_{x_t}, \mathbf{u}_t\big) + \mathrm{F}_{\mathrm{x}}\left(\mathbf{x}_t - \boldsymbol{\mu}_{x_t}\right) + \mathbf{w}_t. \end{aligned} \tag{A.2}$$

The matrix, F_{x}, denotes the Jacobian of the nominal model (A.1) with respect to the vehicle pose, evaluated at the mean,

$$\mathrm{F}_{\mathrm{x}} = \left.\frac{\partial \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_t\big)}{\mathbf{x}_t}\right|_{\mathbf{x}_t = \boldsymbol{\mu}_{x_t}}$$

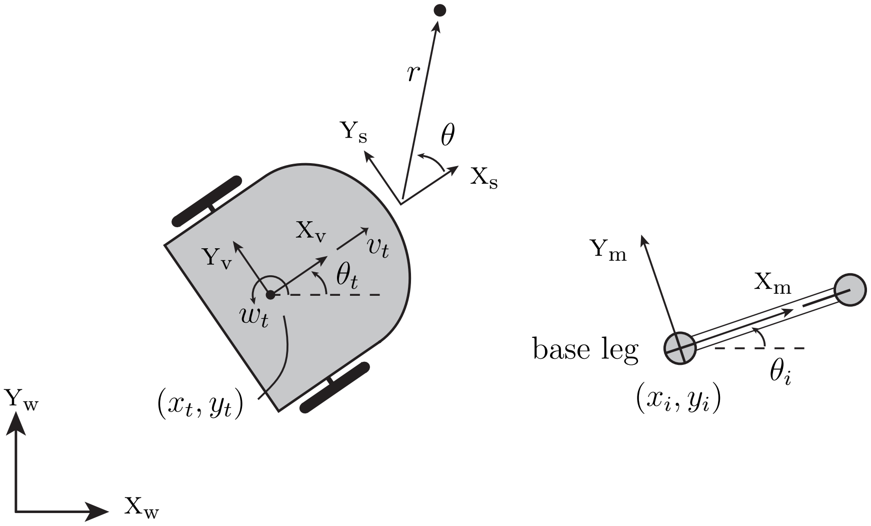


Figure A-2: A schematic of the system model for the hurdles filter implementation. We model the vehicle state by the 2D position and orientation of a body-fixed reference frame, $\mathrm{X_vY_vZ_v}$. The $\mathrm{X_sY_sZ_s}$ frame at the front of the vehicle denotes that of the laser range and bearing sensor. We treat each hurdle as a feature that we parametrize by the position and orientation of a coordinate frame, $\mathrm{X_mY_mZ_m}$ coincident with the hurdle's base leg.

A.2.2 Measurement Model

Our map representation treats each hurdle as a landmark in the planar environment. We model a hurdle as a coordinate frame, $\mathrm{X_mY_mZ_m}$, arbitrarily choosing a "base" leg as the origin while the second leg defines the positive $\mathrm{X_m}$-axis. We parametrize each feature by the 2D position, (x_i, y_i), and orientation, θ_i, of the coordinate frame relative to the world frame, i.e. $\mathbf{m}_i = [x_i \; y_i \; \theta_i]^\top$. Figure A-2 presents a top-view schematic of the feature model.

We abstract laser scan data into measurements of the position and orientation of the hurdle coordinate frame relative to the vehicle's body-fixed frame. In particular, we reduce the noise-corrupted observations of the range and bearing to the two hurdle legs, $\mathbf{z}_a = [z_{r_a} \; z_{\theta_a}]^\top$ and $\mathbf{z}_b = [z_{r_b} \; z_{\theta_b}]^\top$, into the vehicle-relative coordinate frame transformation measurements as

$$\mathbf{z}_{m_i} = \begin{bmatrix} z_x \\ z_y \\ z_\theta \end{bmatrix} = \begin{bmatrix} z_{x_a} \\ z_{y_a} \\ \operatorname{atan2}\left(z_{y_b} - z_{y_a}, z_{x_b} - z_{x_a}\right) \end{bmatrix} \tag{A.3}$$

where

$$\begin{bmatrix} z_{x_a} \\ z_{y_a} \end{bmatrix} = z_{r_a} \begin{bmatrix} \cos\left(z_{\theta_a}\right) \\ \sin\left(z_{\theta_a}\right) \end{bmatrix} \qquad \begin{bmatrix} z_{x_b} \\ z_{y_b} \end{bmatrix} = z_{r_b} \begin{bmatrix} \cos\left(z_{\theta_b}\right) \\ \sin\left(z_{\theta_b}\right) \end{bmatrix}.$$

We model the nominal measurement noise in terms of additive white Gaussian noise, $\bar{\mathbf{v}}_t \sim \mathcal{N}\big(\mathbf{0}, \bar{\mathrm{R}}_t\big)$, that corrupts the range and bearing observations. The corresponding Gaussian approximation to the noise in the relative position and orientation measurement (A.3) follows from the linearization with respect to the original range and bearing pairs.

The expression for the corresponding measurement model (2.6) is given by the nonlinear function,

$$\begin{aligned}\mathbf{z}_t &= \mathbf{h}\big(\boldsymbol{\xi}_t\big) + \mathbf{v}_t \\ &= \begin{bmatrix} \mathrm{R}_v^{w\top}\left(\mathbf{m}_i^w - \mathbf{t}_v^w\right) \\ \theta_t - \theta_i \end{bmatrix} + \mathbf{v}_t,\end{aligned} \tag{A.4}$$

where $\mathrm{R}_v^w \in SO(2)$ and $\mathbf{t}_v^w = [x_t\ y_t]$ denote the rotation and translation from the world frame to the vehicle frame, expressed in the world coordinate frame. The position of the feature in the world frame is denoted as $\mathbf{m}_i^w = [x_i\ y_i]$. The additive term, $\mathbf{v}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}_t\big)$, corresponds to the aforementioned Gaussian approximation to the error.

In order to derive the linearized model, we view the relative measurement as a tail-to-tail compounding operation [121],

$$\mathbf{z}_t = \mathbf{x}_{vm_i} + \mathbf{v}_t = \ominus\mathbf{x}_{wv} \oplus \mathbf{x}_{wm_i} + \mathbf{v}_t.$$

Here, for the sake of consistency with the representation of spatial relationships [121], $\mathbf{x}_{ij}$ represents the pose of state element j relative to the reference frame associated with element i. The linearized measurement equation follows from the nonlinear model,

$$\begin{aligned}\mathbf{z}_t &= \mathbf{h}\big(\boldsymbol{\xi}_t\big) + \mathbf{v}_t \\ &= \ominus\mathbf{x}_{wv} \oplus \mathbf{x}_{wm_i} + \mathbf{v}_t \\ &\approx \mathbf{h}\big(\boldsymbol{\mu}_{x_t}, \boldsymbol{\mu}_{m_i}\big) + \mathrm{H}\big(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\big) + \mathbf{v}_t.\end{aligned} \tag{A.5}$$

The Jacobian of the tail-to-tail operation matrix is

$$\mathrm{H} = \begin{bmatrix}\mathrm{J}_{1\oplus}\mathrm{J}_{\ominus} & 0_{3\times l} & \mathrm{J}_{2\oplus} & 0_{3\times p}\end{bmatrix},$$

Note that the Jacobian is sparse and that the linearization requires knowledge of the mean estimates of only the pose of the robot, $\boldsymbol{\mu}_{x_t}$, and the observed landmark, $\boldsymbol{\mu}_{m_i}$.

A.2.3 Relocalization Model

In the case of the ESEIF, sparsification relocates the robot within the map based upon observations of known landmarks, $\mathbf{z}_\beta = \mathbf{h}\left(\mathbf{x}_t, \mathbf{m}_\beta\right)$. Each observation corresponds to a measure of the relative vehicle-to-feature transformation, $\mathbf{z}_\beta = \mathbf{x}_{vm_\beta}$, of the form in (A.4) and (A.2.2). Inverting this transformation yields a measurement of the vehicle pose via the head-to-head compounding operation [121],

$$\mathbf{x}_{wv} = \mathbf{x}_{wm_\beta} \oplus \big(\ominus\mathbf{x}_{vm_\beta}\big) = \mathbf{x}_{wm_\beta} \oplus \big(\ominus\mathbf{z}_\beta\big). \tag{A.6}$$

We linearize this model for the relocalized pose (4.4) with respect to the mean of the observed feature, $\check{\boldsymbol{\mu}}_{m_\beta}$, and measurement data,[1]

$$\begin{aligned}\mathbf{x_t} &= \mathbf{g}\big(\mathbf{m}_\beta, \mathbf{z}_\beta\big) \\ &= \mathbf{x}_{wm_\beta} \oplus \left(\ominus \mathbf{z}_\beta\right) \end{aligned} \tag{A.7a}$$

$$\approx \mathbf{g}\big(\boldsymbol{\mu}_{m_\beta}, \hat{\mathbf{z}}_\beta\big) + \mathrm{G}_M\big(\mathbf{M}_t - \check{\boldsymbol{\mu}}_t\big) + \mathrm{G}_{z_\beta}\mathbf{v}_t. \tag{A.7b}$$

Note that this representation integrates the noise via the observation of the transformation between the vehicle and hurdle, $\mathbf{x}_{vm_\beta}$. The Jacobian associated with the head-to-head transformation includes two submatrices, one for the Jacobian with respect to the observed features, and the other over the measurements [121],

$$_\oplus\mathrm{J}_\ominus = \begin{bmatrix}\mathrm{J}_{1\oplus} & \mathrm{J}_{2\oplus}\mathrm{J}_\ominus\end{bmatrix} = \begin{bmatrix}\mathrm{G}_{m_\beta} & \mathrm{G}_{z_\beta}\end{bmatrix}.$$

The first term forms the only nonzero component of the Jacobian over the map,

$$\mathrm{G}_M = \begin{bmatrix}0_{3\times l} & \mathrm{G}_{m_\beta} & 0_{3\times p}\end{bmatrix},$$

while we use G_{z_β} to model the contribution of the sensor noise to the uncertainty of the relocated pose. As with the measurement model, the linearization relies only on the mean estimate for the observed features.

Our ESEIF implementation reserves as many hurdle observations for relocalization as possible. Consequently, the sparsification routine typically estimates the robot pose based upon more than one measurement, $\mathbf{z}_\beta = \{{}^i\mathbf{z}_t : i \in \beta\}$. We compute an individual pose estimate for each observation as in (A.6), and model the new pose as the average over these estimates. This is equivalent to the maximum likelihood estimate under the linearized model when the noise in the individual measurements is independent. It is straightforward to compute the corresponding map Jacobian, G_M, which is nonzero at positions that correspond to the set of $\mathbf{m}_\beta$ features.

A.3 Victoria Park Dataset

The Victoria Park dataset serves as a benchmark against which to compare different SLAM algorithms. The dataset is courtesy of E. Nebot at the Australian Centre for Field Robotics at the University of Sydney [51]. During the experiment, a truck was driven among trees in Victoria Park in Sydney, Australia. A SICK laser sensor was mounted at the front, left corner of the car and provides observations of the range and bearing to neighboring tree trunks. The vehicle was equipped with a rotary variable differential transformer and wheel encoders that measured the steering angle and forward velocity, respectively.

We model the environment as planar and represent the state by the vehicle pose along with a set of 2D point features. The vehicle pose corresponds to the (x, y)

[1] Recall our notation in Section 4.2.1 where we represent the "kidnapped" distribution over the map as $\mathbf{M}_t \sim \mathcal{N}^{-1}\big(\check{\boldsymbol{\eta}}_t, \check{\Lambda}_t\big) = \mathcal{N}\big(\check{\boldsymbol{\mu}}_t, \check{\Sigma}_t\big)$.

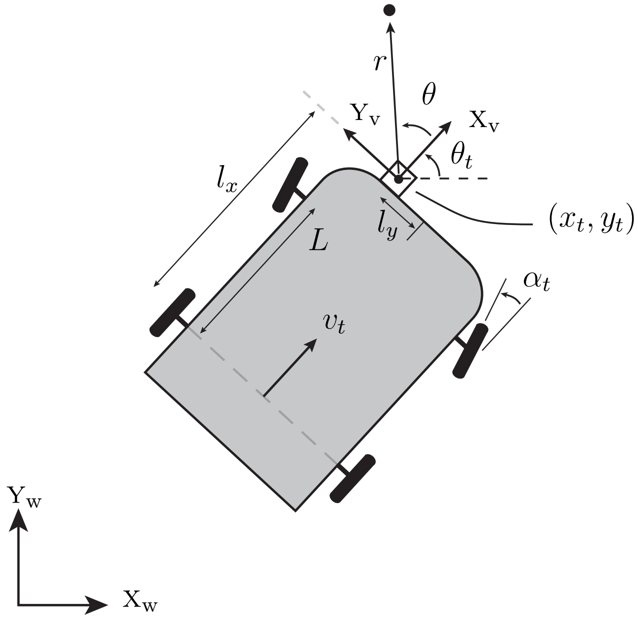


Figure A-3: A diagram of the car model for the Victoria Park dataset. We represent the vehicle state by the 2D pose of a body-fixed coordinate frame, $\mathrm{X_vY_vZ_v}$, coincident with the laser range finder on the left side of the front bumper. The schematic is adapted from that of Guivant *et al.* [51].

position and orientation, θ, of a body-fixed reference frame that is coincident with the sensor coordinate frame. Figure A-3 offers a schematic that describes the model.

A.3.1 Motion Model

The truck moves according to a non-holonomic, Ackerman-steered motion model. We adopt a kinematic representation for the motion, and treat the forward velocity, $\hat{v}_t$, and steering angle, $\hat{\alpha}_t$, as control inputs. The following discrete-time, constant-velocity model describes the motion,

$$\mathbf{x}_{t+1} = \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_t\big) + \mathbf{w}_{\mathrm{m}}$$

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + v_t \Delta t \begin{bmatrix} \cos(\theta_t) - \frac{1}{L}\tan(\theta_t)\big(l_y \sin(\theta_t) + l_x \cos(\theta_t)\big) \\ \sin(\theta_t) + \frac{1}{L}\tan(\theta_t)\big(l_y \cos(\theta_t) - l_x \sin(\theta_t)\big) \\ \frac{1}{L}\tan(\alpha_t) \end{bmatrix} + \mathbf{w}_{\mathrm{m}} \qquad \text{(A.8)}$$

$$v_t = \hat{v}_t + w_v, \quad \alpha_t = \hat{\alpha}_t + w_\alpha$$

where the lengths L, l_x, and l_y are defined in Figure A-3. The kinematic representation includes zero-mean white Gaussian process noise, $\mathbf{w}_{\mathrm{m}} \sim \mathcal{N}\big(\mathbf{0}, \mathrm{Q_m}\big)$, that accounts for uncertainty in the model parameters. We also assume that the velocity and steering control measurements, v_t and α_t, are corrupted by additive noise, $\mathbf{w}_{\mathrm{u}} = [w_v \; w_\alpha]$, where $\mathbf{w}_{\mathrm{u}} \sim \mathcal{N}\big(\mathbf{0}, \mathrm{Q_u}\big)$.

We linearize the model (A.8) about the current mean pose, $\boldsymbol{\mu}_{x_t}$, and noise mean via a Taylor's series expansion. Dropping terms of second order and higher results in the linear approximation,

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_t\big) + \mathbf{w}_{\mathrm{m}} \\ &\approx \mathbf{f}\big(\boldsymbol{\mu}_{x_t}, \hat{\mathbf{u}}_t\big) + \mathrm{F}_{\mathrm{x}}\left(\mathbf{x}_t - \boldsymbol{\mu}_{x_t}\right) + \mathrm{G}_{\mathrm{u}}\mathbf{w}_{\mathrm{u}} + \mathbf{w}_{\mathrm{m}}. \end{aligned} \tag{A.9}$$

The matrix, F_{x}, denotes the Jacobian with respect to the vehicle state evaluated at the mean pose and zero-mean noise. Similarly, G_{u} is the Jacobian with respect to the observed control inputs.

$$\mathrm{F}_{\mathrm{x}} = \left.\frac{\partial \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_t\big)}{\partial \mathbf{x}_t}\right|_{\left(\mathbf{x}_t = \boldsymbol{\mu}_{x_t}, \mathbf{u}_t = \hat{\mathbf{u}}_t\right)} \qquad \mathrm{G}_{\mathrm{u}} = \left.\frac{\partial \mathbf{f}\big(\mathbf{x}_t, \mathbf{u}_t\big)}{\partial \mathbf{u}_t}\right|_{\left(\mathbf{x}_t = \boldsymbol{\mu}_{x_t}, \mathbf{u}_t = \hat{\mathbf{u}}_t\right)}$$

A.3.2 Observation Model

We treat the trees as point features in the 2D environment and extract range and bearing observations from laser scan data. The measurement model for a particular feature, $\mathbf{m}_i = [x_i \; y_i]$, has the simple form,

$$\begin{aligned} \mathbf{z}_t &= \mathbf{h}\big(\boldsymbol{\xi}_t\big) + \mathbf{v}_t \\ &= \begin{bmatrix} \sqrt{\left(x_i - x_t\right)^2 + \left(y_i - y_t\right)^2} \\ \operatorname{atan2}\left(y_i - y_t, x_i - x_t\right) \end{bmatrix} + \mathbf{v}_t, \end{aligned} \tag{A.10}$$

where $\mathbf{v}_t \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}_t\big)$. The linearization about the mean estimate for the robot pose and the landmark position follow as

$$\begin{aligned} \mathbf{z}_t &= \mathbf{h}\big(\boldsymbol{\xi}_t\big) + \mathbf{v}_t \\ &\approx \mathbf{h}\big(\boldsymbol{\mu}_{x_t}, \boldsymbol{\mu}_{m_i}\big) + \mathrm{H}\big(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t\big) + \mathbf{v}_t. \end{aligned} \tag{A.11}$$

A.3.3 Relocalization Model

A single range and bearing measurement to a landmark is insufficient to estimate the three-DOF vehicle pose. We compute the pose based upon observation pairs, ${}^{a}\mathbf{z}_t = [z_{r_i} \; z_{\theta_i}]$ and ${}^{b}\mathbf{z}_t = \left[z_{r_j} \; z_{\theta_j}\right]$. In similar fashion to the hurdle representation, we define a coordinate frame in terms of the two point features corresponding to the observation pair. One arbitrarily chosen landmark defines the origin while the other specifies the direction of the positive X-axis. The filter then interprets the range and bearing measurements as observations of the relative position and orientation of this feature-defined reference frame in the vehicle's body-fixed frame.

Given two point features, $\mathbf{m}_\beta = \{\mathbf{m}_i, \mathbf{m}_j\}$ where $\mathbf{m}_i = [x_i \; y_i]^\top$ and $\mathbf{m}_j = [x_j \; y_j]^\top$, the coordinate frame with the origin coincident with $\mathbf{m}_i$ can be expressed by the spatial relationship,

$$\mathbf{x}_{wm_{ij}} \triangleq \begin{bmatrix} x_{ij} \\ y_{ij} \\ \theta_{ij} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \operatorname{atan2}\left(y_j - y_i, x_j - x_i\right) \end{bmatrix}. \tag{A.12}$$

Similarly, an observation of the range and bearing to the landmark pair is formulated as a measure of the vehicle-relative position and orientation of this frame. This model mimics that of the hurdles dataset and is equivalent to the tail-to-tail compounding operation,

$$\begin{aligned} \mathbf{z}_\beta &= \ominus \mathbf{x}_{wv} \oplus \mathbf{x}_{wm_{ij}} + \bar{\mathbf{v}}_t \\ &= \mathbf{x}_{vm_{ij}} + \bar{\mathbf{v}}_t. \end{aligned}$$

We derive the observation data from the raw measurements of range and bearing to each landmark with the same abstraction that we employ for the hurdles experiment (A.3). The noise term, $\bar{\mathbf{v}}_t \sim \mathcal{N}\big(\mathbf{0}, \bar{\mathrm{R}}_t\big)$, is adapted from the original model (A.10) that treats the noise in range and bearing as Gaussian.

The vehicle relocalization model is equivalent to the head-to-head spatial relationship,

$$\mathbf{x}_{wv} = \mathbf{x}_{wm_{ij}} \oplus \big(\ominus \mathbf{x}_{vm_{ij}}\big).$$

Treating the measurement data, $\mathbf{z}_\beta$, as an observation of the transformation between the vehicle and composite feature frame, $\mathbf{x}_{vm_{ij}}$, yields the model for the relocated vehicle pose,

$$\begin{aligned} \mathbf{x_t} &= \mathbf{g}\big(\mathbf{m}_\beta, \mathbf{z}_\beta\big) \\ &= \mathbf{x}_{wm_{ij}} \oplus \left(\ominus \mathbf{z}_\beta\right) \end{aligned} \tag{A.13a}$$

$$\approx \mathbf{g}\big(\boldsymbol{\mu}_{m_\beta}, \hat{\mathbf{z}}_\beta\big) + \mathrm{G}_M\big(\mathbf{M}_t - \check{\boldsymbol{\mu}}_t\big) + \mathrm{G}_{z_\beta}\bar{\mathbf{v}}_t. \tag{A.13b}$$

A.4 Hovering Autonomous Underwater Vehicle

This section details the model and filter implementation details for the AUV experiments. We first describe the six-DOF vehicle state-space and the various coordinate frames that the filter employs to describe the vehicle's motion and sensing data. We subsequently derive the discrete-time motion model based upon the continuous-time kinematic equations of motion. The section concludes with an overview of the measurement models that describe the different sensors onboard the vehicle.

A.4.1 Vehicle State-Space Model

The underwater survey implementation references two main coordinate frames. The framework assumes an inertial reference frame, $\mathrm{X_w Y_w Z_w}$, that remains fixed at the surface with the $\mathrm{X_w}$-axis pointing North, the $\mathrm{Y_w}$-axis directly East, and the positive $\mathrm{Z_w}$-axis pointing down. This reference frame serves as the world coordinate frame with respect to which we describe the vehicle pose and map. We describe vehicle motion with respect to a second, body-fixed reference frame, $\mathrm{X_vY_vZ_v}$, positioned at the aft end of the vehicle. Consistent with the standard convention for ocean vehicles [42], the $\mathrm{X_v}$-axis points in the direction of forward motion, the $\mathrm{Y_v}$-axis points to starboard, and the $\mathrm{Z_v}$-axis is positive downwards.

The vehicle state consists of its pose with respect to the world frame, together with the vehicle's body-referenced linear and angular velocities.

$$\mathbf{x}_v = \left[\mathbf{t}_v^{w\top}\ \boldsymbol{\Theta}_v^{w\top}\ \boldsymbol{\nu}_1^\top\ \boldsymbol{\nu}_2^\top\right]^\top = \left[x,\ y,\ z,\ \phi,\ \theta,\ \psi,\ u,\ v,\ w,\ p,\ q,\ r\right]^\top$$

The six-DOF vehicle pose is described in terms of the position, $\mathbf{t}_v^w = [x\ y\ z]^\top$, and orientation, $\boldsymbol{\Theta}_v^w = [\phi\ \theta\ \psi]^\top$, of the body-fixed coordinate frame. We adopt the XYZ-convention for Euler angles[2] to define the orientation in terms of the vehicle's roll, ϕ, pitch, θ, and heading (yaw), ψ. In addition to pose, the vehicle state includes the body-relative linear and angular velocities. The linear vehicle velocity, $\boldsymbol{\nu}_1 = [u\ v\ w]^\top$, consists of the forward (surge), u, lateral (sway), v, and vertical (heave), w, velocities. We denote the body-fixed angular rates as $\boldsymbol{\nu}_2 = [p\ q\ r]^\top$ that correspond to the roll, pitch, and yaw rates.

Consistent with the attitude representation, the following rotation matrix relates the vehicle reference frame to the world frame.

$$\begin{aligned}
\mathrm{R}_v^w &= \mathrm{R}_{z,\psi}\mathrm{R}_{y,\theta}\mathrm{R}_{x,\phi} \\
&= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\
&= \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}
\end{aligned} \tag{A.14}$$

A.4.2 Motion Model

This section describes the six-DOF kinematic motion model that is employed for time prediction. We first introduce the continuous-time, constant-velocity equations of motion from which we then derive the linearized state-space equations. Subsequently, we discretize the continuous-time state equation to arrive at a discrete-time approximation to the kinematic motion model. The following derivation mimics that of Eustice [33], who offers a thorough derivation of the stochastic motion model for an underwater vehicle.

[2] The XYZ-convention can be thought of as aligning the world frame with the vehicle frame through a series of rotations about the current (rotating) axes. The world frame is first rotated in yaw about its Z-axis, followed by a rotation in pitch about the current Y-axis, and, subsequently, a rotation in roll about the new X-axis.

Continuous-Time Motion Model

The continuous-time evolution of the vehicle state is described according to a nonlinear, time-invariant state-space equation. We represent the state equation according to a stochastic, constant-velocity model.

$$\dot{\mathbf{x}}_v(t) = \mathbf{f}\left(\mathbf{x}_v(t)\right) + \mathrm{G}\mathbf{w}(t)$$

$$\frac{d}{dt}\begin{bmatrix} \mathbf{t}_v^w(t) \\ \boldsymbol{\Theta}_v^w(t) \\ \boldsymbol{\nu}_1(t) \\ \boldsymbol{\nu}_2(t) \end{bmatrix} = \begin{bmatrix} \mathcal{J}_1\big(\boldsymbol{\Theta}_v^w(t)\big)\boldsymbol{\nu}_1(t) \\ \mathcal{J}_2\big(\boldsymbol{\Theta}_v^w(t)\big)\boldsymbol{\nu}_2(t) \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} + \mathrm{G}\mathbf{w}(t) \tag{A.15}$$

$$\mathcal{J}_1\big(\boldsymbol{\Theta}_v^w(t)\big) = \mathrm{R}_v^w\big(\boldsymbol{\Theta}_v^w(t)\big)$$

$$\mathcal{J}_2\big(\boldsymbol{\Theta}_v^w(t)\big) = \begin{bmatrix} 1 & \sin\big(\phi(t)\big)\tan\big(\theta(t)\big) & \cos\big(\phi(t)\big)\tan\big(\theta(t)\big) \\ 0 & \cos\big(\phi(t)\big) & -\sin\big(\phi(t)\big) \\ 0 & \sin\big(\phi(t)\big)\sec\big(\theta(t)\big) & \cos\big(\phi(t)\big)\sec\theta(t) \end{bmatrix} \qquad \mathrm{G} = \begin{bmatrix} 0_{6\times1} \\ \mathrm{I}_{6\times6} \end{bmatrix}$$

The matrix that describes the rate of change in vehicle position, $\mathcal{J}_1\big(\boldsymbol{\Theta}_v^w(t)\big)$, is the rotation matrix (A.14) that rotates the body-relative linear velocities into the world frame. The additive vector, $\mathbf{w}(t)$, corresponds to wide-sense stationary, zero-mean, white Gaussian noise whose covariance function is $K_{\mathrm{ww}}(t,s) = \mathrm{Q}\delta(t-s)$. Per the projection matrix, G, the noise corrupts only the rate of change for the linear and angular velocities and accounts for error in the constant-velocity assumption.

Prior to deriving the discrete-time form of the motion model, we linearize the state equations about the current mean estimate for the vehicle pose. As our filter operates in discrete time steps, this estimate corresponds to a time, t_k, where $t_k \leq t < t_{k+1}$. We denote the corresponding mean pose as $\boldsymbol{\mu}_{x_v}(t_k)$. Dropping terms of second order and higher from the Taylor's series expansion of the motion model (A.15) yields the linear approximation,

$$\begin{aligned} \dot{\mathbf{x}}_v(t) &= \mathbf{f}\left(\mathbf{x}_v(t)\right) + \mathrm{G}\mathbf{w}(t) \\ &\approx \mathbf{f}\left(\boldsymbol{\mu}_{x_v}(t_k)\right) + \mathrm{F}_{\mathrm{x}_v}\big(\mathbf{x}_v(t) - \boldsymbol{\mu}_{x_v}(t_k)\big) + \mathrm{G}\mathbf{w}(t), \end{aligned} \tag{A.16}$$

where $\mathrm{F}_{\mathrm{x}_v}$ denotes the Jacobian of the state equation, evaluated at the mean,

$$\mathrm{F}_{\mathrm{x}_v} = \left.\frac{\partial \mathbf{f}\left(\mathbf{x}_v(t)\right)}{\partial \mathbf{x}_v(t)}\right|_{\left(\mathbf{x}_v(t) = \boldsymbol{\mu}_{x_v}(t_k)\right)}$$

Rearranging terms results in the familiar form for the linearized state-space equations,

$$\begin{aligned} \dot{\mathbf{x}}_v(t) &\approx \mathrm{F}_{\mathrm{x}_v}\mathbf{x}_v(t) + \left\{\mathbf{f}\left(\boldsymbol{\mu}_{x_v}(t_k)\right) - \mathrm{F}_{\mathrm{x}_v}\boldsymbol{\mu}_{x_v}(t_k)\right\} + \mathrm{G}\mathbf{w}(t) \\ &= \mathrm{F}_{\mathrm{x}_v}\mathbf{x}_v(t) + \mathbf{u}(t_k) + \mathrm{G}\mathbf{w}(t). \end{aligned} \tag{A.17}$$

Above, we adopt the notation employed by Eustice [33], and treat the component,

$$\mathbf{u}(t_k) = \mathbf{f}\left(\boldsymbol{\mu}_{x_v}(t_k)\right) - \mathrm{F}_{\mathrm{x}_v}\boldsymbol{\mu}_{x_v}(t_k)$$

as a form of control input that is fixed for t within the sampling window, $t_k \leq t < t_{k+1}$.

Discrete-Time Motion Model

Equation (A.17) corresponds to the linear approximation to the continuous-time equations of motion. The prediction step of our Bayesian filter relies upon a discrete-time representation for the state-space equations. In particular, the filter implements a motion model that predicts the evolution of the vehicle state at specific instances in time, $t_{k+1} = t_k + \Delta t$, where Δt need not be constant. We derive the discrete-time equivalent of the linearized motion model by discretizing the continuous-time state equation (A.17).

Over each time interval, $t_k \leq t < t_{k+1}$, we assume that the Jacobian, $\mathrm{F}_{\mathrm{x}_v}$, is well-approximated as time-invariant and that the control term, $\mathbf{u}(t_k)$, is also constant (i.e. zero-order hold equivalent). Under these assumptions, the discretization of the continuous-time state-space equation yields the following approximation to the difference equation. The derivation relies upon sampling the solution to the continuous-time state equation (A.17), and is described, in detail, by Ogata [109].[3]

$$\mathbf{x}_v[t_{k+1}] = \bar{\mathrm{F}}_{\mathrm{x}_v}\mathbf{x}_v[t_k] + \bar{\mathrm{B}}\mathbf{u}[t_k] + \mathbf{w}[t_k] \tag{A.18}$$

The discrete-time form of the process matrices are given by

$$\bar{\mathrm{F}}_{\mathrm{x}_v} = e^{\mathrm{F}_{\mathrm{x}_v}\Delta t} \qquad \bar{\mathrm{B}} = e^{\mathrm{F}_{\mathrm{x}_v}t_{k+1}}\int_{t_k}^{t_{k+1}} e^{-\mathrm{F}_{\mathrm{x}_v}\tau}d\tau.$$

The discrete-time representation for the noise term, which is not assumed to be constant over the interval $t \in [t_k, t_{k+1})$ is given by

$$\mathbf{w}[t_k] = \int_{t_k}^{t_{k+1}} e^{\mathrm{F}_{\mathrm{x}_v}(t_{k+1}-\tau)}\mathrm{G}\mathbf{w}(\tau)d\tau = e^{\mathrm{F}_{\mathrm{x}_v}t_{k+1}}\int_{t_k}^{t_{k+1}} e^{-\mathrm{F}_{\mathrm{x}_v}\tau}\mathrm{G}\mathbf{w}(\tau)d\tau.$$

The discretized noise, $\mathbf{w}[t_k] \sim \mathcal{N}\big(\mathbf{0}, \mathrm{Q}[t_k]\big)$, is white and Gaussian with covariance function,

$$\mathrm{Q}[t_k] = \int_{t_k}^{t_{k+1}} e^{\mathrm{F}_{\mathrm{x}_v}(t_{k+1}-\tau)}\mathrm{G}\mathrm{Q}(\tau)\mathrm{G}^\top e^{\mathrm{F}_{\mathrm{x}_v}^\top(t_{k+1}-\tau)}d\tau,$$

where, we additionally model the continuous-time covariance function as constant, $\mathrm{Q}(\tau) = \mathrm{Q}$.

[3] Note that, throughout the remainder of the section, we will abuse our earlier notation and represent discrete-time random vectors and data with their time stamps enclosed within square brackets as opposed to as subscripts, i.e. $\mathbf{w}[t_k]$ in place of $\mathbf{w}_k$.

A.4.3 Observation Models

Since the filter tracks the vehicle's linear and angular velocity as part of the state vector, we treat velocity data from the DVL and IMU as state measurements. Together with attitude observations and DIDSON-based target measurements, we incorporate this data through the appropriate measurement update steps. In this section, we briefly describe the different observation models that the filter employs.

Linear Velocity Measurements

The HAUV is equipped with an RDI Workhorse Navigator DVL that provides measurements of linear velocity along each of its four beams. This radial velocity data can be transformed into an observation of the three-axis vehicle velocities with respect to a reference frame affixed to the DVL, $\boldsymbol{\nu}_1^{\text{DVL}}[t_k] = \left[v_x^{\text{DVL}}[t_k]\; v_y^{\text{DVL}}[t_k]\; v_z^{\text{DVL}}[t_k]\right]^\top$. We transform this velocity into the body-fixed reference frame by applying a sensor-to-vehicle rotation (head-to-tail operation) that is a function of the DVL pitch, $\alpha_{\text{DVL}}[t_k]$.

$$\boldsymbol{\nu}_1^{v}[t_k] = \begin{bmatrix} u[t_k] \\ v[t_k] \\ w[t_k] \end{bmatrix} = \mathrm{R}_{\text{DVL}}^{v}\big(\alpha_{\text{DVL}}[t_k]\big)\boldsymbol{\nu}_1^{\text{DVL}}[t_k]$$

The corresponding observation model is then straightforward,

$$\mathbf{z}_{\text{DVL}}[t_k] = \mathrm{H}_{\text{DVL}}\mathbf{x}_v[t_k] + \mathbf{v}_{\text{DVL}}[t_k] \tag{A.19}$$

$$\mathrm{H}_{\text{DVL}} = \begin{bmatrix} 0_{3\times6} & \mathrm{I}_{3\times3} & 0_{3\times3} \end{bmatrix}.$$

We model the underlying noise as additive, zero-mean, white Gaussian noise that corrupts the four radial velocity measurements. In turn, we approximate the noise in the three-axis DVL linear velocities as Gaussian, $\bar{\mathbf{v}}[t_k] \sim \mathcal{N}\big(0, \bar{\mathrm{R}}_{\text{DVL}}\big)$. The additive noise term, $\mathbf{v}_{\text{DVL}}[t_k]$, corresponds to applying the transformation from the DVL frame to the vehicle frame and exhibits a covariance function that depends upon the sensor's pitch, $\mathbf{v}_{\text{DVL}}[t_k] \sim \mathcal{N}\big(0, \mathrm{R}_{\text{DVL}}^{v}\left(\alpha_{\text{DVL}}[t_k]\right) \cdot \bar{\mathrm{R}}_{\text{DVL}} \cdot \mathrm{R}_{\text{DVL}}^{v\top}\left(\alpha_{\text{DVL}}[t_k]\right)\big)$.

Attitude and Angular Rate Measurements

The IMU is coincident with the vehicle frame and provides observations of the vehicle's roll and pitch, as well as the three-axis body-relative angular rates.

$$\mathbf{z}_{\text{IMU}}[t_k] = \left[\phi[t_k]\; \theta[t_k]\; p[t_k]\; q[t_k]\; r[t_k]\right]^\top$$

The measurement model is linear in the vehicle state,

$$\mathbf{z}_{\text{IMU}}[t_k] = \mathrm{H}_{\text{IMU}}\mathbf{x}_v[t_k] + \mathbf{v}_{\text{IMU}}[t_k] \tag{A.20}$$

$$\mathrm{H}_{\text{IMU}} = \begin{bmatrix} 0_{2\times3} & \mathrm{I}_{2\times2} & 0 & 0_{2\times3} & 0_{2\times3} \\ 0_{3\times3} & 0_{3\times2} & 0 & 0_{3\times3} & \mathrm{I}_{3\times3} \end{bmatrix},$$

where $\mathbf{v}_{\text{IMU}}[t_k] \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}_{\text{IMU}}\big)$ is the zero-mean noise that we attribute to the IMU.

DIDSON Measurements

Features extracted from DIDSON acoustic imagery yield a measure of the range and bearing to targets on the hull, along with a bound on their elevation. We reduce the ambiguity in elevation with the help of an estimate for the local hull geometry. The result is an observation of the range, bearing, and elevation to a target, m_i, as expressed in the sonar's coordinate frame. We model the measurement as an observation with respect to this reference frame, whose origin relative to the world frame follows from the head-to-tail operation, $\mathbf{x}_{ws} = \mathbf{x}_{wv} \oplus \mathbf{x}_{vs}$, where s denotes the sonar sensor. Note that the transformation from the vehicle frame to that of the sonar, $\mathbf{x}_{vs}$, is a function of the sonar pitch pitch, $\alpha_s[t_k]$.

$$\begin{aligned}
\mathbf{z}_s[t_k] &= \mathbf{h}_s\big(\mathbf{x}_{ws}[t_k]\big) + \mathbf{v}_s[t_k] && \text{(A.21a)}\\
&= \mathbf{h}_s\big(\mathbf{x}_v[t_k], \mathbf{m}_i, \alpha_s[t_k]\big) + \mathbf{v}_s[t_k] \\
&\approx \mathbf{h}\big(\boldsymbol{\mu}_{x_v}[t_k], \boldsymbol{\mu}_{m_i}[t_k], \alpha_s[t_k]\big) + \mathrm{H}_s\big(\boldsymbol{\xi}[t_k] - \boldsymbol{\mu}[t_k]\big) + \mathbf{v}_s[t_k] && \text{(A.21b)}
\end{aligned}$$

The last line above corresponds to the linearization of the range, bearing, and elevation measurement with respect to the current mean estimate for the vehicle pose and the landmark location. The additive term signifies zero-mean Gaussian noise, $\mathbf{v}_s[t_k] \sim \mathcal{N}\big(\mathbf{0}, \mathrm{R}_s[t_k]\big)$.

Appendix B

Acoustic Imaging Sonar

The DIDSON ensonifies the scene with a pair of lenses that, operating at 1.8 MHz, individually direct a sequence of 96 transmitted pulses. It then uses the same lenses to focus the acoustic return on a 96 element transducer array. The result is an acoustic intensity image, resolved as a discrete function of range and bearing. The sonar does not disambiguate the elevation angle, and the echos may originate from any point along the constant range and bearing arc that subtends the $|\beta| \leq 6^\circ$ elevation. This ambiguity is analogous to the scale invariance of pinhole camera models, with the additional constraint on the size of the acoustic field-of-view.

B.1 Brief Overview of Multiple View Geometry

The standard approach to resolve observation ambiguity is to image the scene from several vantage points. Multiple view imaging techniques [58], including structure from motion and delayed-state Bayesian filters [34] rely upon observations of the scene that are shared between sets of images. This correspondence establishes the epipolar geometry that relates view pairs, imposing constraints on the relative pose of the corresponding cameras. The linear projection model of pinhole cameras leads to a *fundamental matrix* and *essential matrix* that describe the epipolar geometry for pairs of uncalibrated and calibrated cameras, respectively,[1] Consider a pair of calibrated cameras specified by their normalized camera matrices, $\mathrm{P} = [\mathrm{I}\,|\,\mathbf{0}]$ and $\mathrm{P}' = [\mathrm{R}\,|\,\mathbf{t}]$, where I is the 3×3 identity matrix and $\mathrm{R} \in \mathcal{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ correspond to the rotation and translation of the second camera frame relative to that of the first. Without loss of generality, we assume that the first camera frame is aligned with the world frame.[2] If both cameras image the same world point, $\mathbf{X}$, the corresponding normalized image points, $\mathbf{u} = \mathrm{P}\mathbf{X}$ and $\mathbf{u}' = \mathrm{P}'\mathbf{X}$ satisfy the epipolar constraint,

$$\mathbf{u}'^{\top}\mathrm{E}\mathbf{u} = 0, \tag{B.1}$$

[1] To be correct, the essential matrix specifies the epipolar constraint between pairs of image points that have been normalized by the inverse of the calibration matrix.

[2] These cameras may represent the same physical camera that acquires a pair of images at different points in time from different vantage points.

where $\mathrm{E} = [\mathbf{t}]_\times \mathrm{R} \in \mathbb{R}^{3\times 3}$ is the essential matrix.[3] Unlike the general fundamental matrix, which has seven degrees of freedom, there are only five for the essential matrix. The relative rotation, R, and translation, $\mathbf{t}$, between camera frames account for six degrees of freedom, yet there is a scale invariance as the essential matrix is a homogeneous matrix. It is possible to then estimate the relative rotation and translation up to scale directly from the essential matrix [58].

With only five degrees of freedom, the essential matrix can be estimated from as few as five pairs of image points, in terms of the the zeros of a tenth order polynomial [108]. As with other minimal solutions, though, this five-point algorithm is particularly sensitive to errors in the location of the points within the images, which can be quite high. Alternatively, the eight-point algorithm [80] estimates the essential matrix as the solution to a set of linear equations. Hartley [57] demonstrates that, by first normalizing the image coordinates, the eight-point algorithm tolerates a greater amount of image noise. Typically, feature detection identifies a much larger set of image pairs that, along with errors in point locations, includes erroneous (false) matches. While location inaccuracies degrade estimator precision, false matches induce gross errors in the estimated essential matrix [143]. Most implementations overcome the effects of outliers by sampling solutions from several different sets of point pairs. Hartley and Zisserman [58] describe a multiple step algorithm that first samples from a set of linear solutions for the essential matrix, using RANSAC [39] to identify the largest set of inlier pairs. Next, they iteratively solve for the essential matrix that minimizes the nonlinear reprojection error over this set of inliers. The subsequent step uses the resulting estimate for the epipolar geometry to generate a new set of feature correspondences from the set of interest points. The algorithm then repeats the iterative minimization with this inlier set and continues to refine the essential matrix estimate till convergence.

Given an estimate of the essential matrix for a pair of cameras, one can resolve the second camera matrix, $\mathrm{P}' = [\mathrm{R} \,|\, \mathbf{t}]$, to one of four $(\mathrm{R}, \mathbf{t})$ transformations (modulo scale). Including an additional constraint on the location of an imaged point relative to the two image planes reduces this set to a single transformation up to a scale factor. In similar fashion, Eustice [33] estimates the roll, pitch, and yaw angles that parametrize the relative rotation, $\mathrm{R} = \mathrm{R}\left(\phi, \theta, \psi\right)$, along with the scale-normalized translation, $\mathbf{t}$.

B.1.1 DIDSON Camera Model

The convenient form of the epipolar constraint (B.1) follows directly from the linear projection model of pinhole cameras. With invariance in elevation as opposed to scale, however, the DIDSON imaging geometry obeys a projection model that is nonlinear in the scene coordinates. Consider the schematic in Figure B-1 in which the sonar images a point with spherical coordinates (r, θ, β) relative to the camera frame. If we assume, as before, that the camera frame is aligned with the world reference frame,

[3] Here, $\mathbf{X} \in \mathbb{R}^4$ and $\mathbf{u}, \mathbf{u}' \in \mathbb{R}^3$ are homogeneous coordinates of the world and image points, respectively.

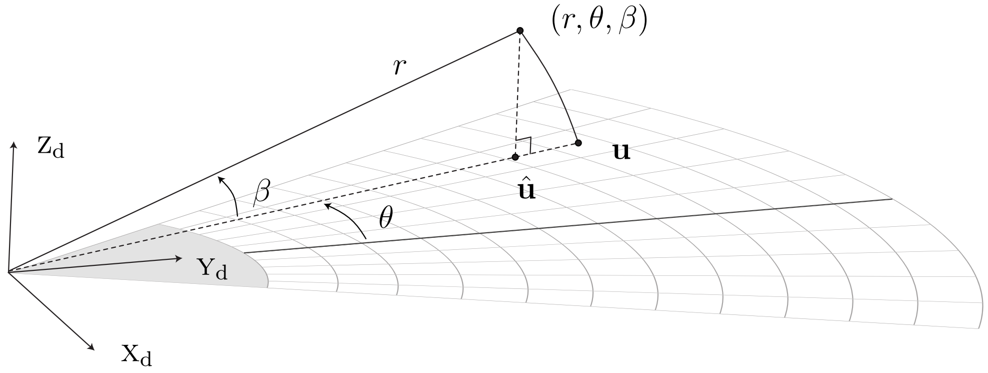


Figure B-1: The DIDSON produces a discrete, two-dimensional acoustic intensity image that resolves the range, r, and bearing, θ, of the echo source. The image is invariant to changes in elevation within the sonar's FOV, i.e. $|\beta| \leq 6°$. It proves beneficial to approximate the image geometry by an orthographic projection. Under this model, the 3D point (r, θ, β) projects to $\hat{\mathbf{u}}$ on the image plane rather than $\mathbf{u}$.

the Cartesian coordinates of the point are

$$\tilde{\mathbf{X}}_c = \begin{bmatrix} \mathrm{X}_c \\ \mathrm{Y}_c \\ \mathrm{Z}_c \end{bmatrix} = \begin{bmatrix} -r\sin\theta\cos\beta \\ r\cos\theta\cos\beta \\ r\sin\beta \end{bmatrix}, \tag{B.2}$$

where we use the tilde accent to denote inhomogeneous vectors. The DIDSON maps the corresponding acoustic return to the inhomogeneous image coordinates, $\tilde{\mathbf{u}}$, according to the nonlinear projection model,

$$\tilde{\mathbf{u}} = \begin{bmatrix} -r\sin\theta \\ r\cos\theta \end{bmatrix} = \begin{bmatrix} \mathrm{X}_c\ (\mathrm{X}_c^2 + \mathrm{Y}_c^2 + \mathrm{Z}_c^2)^{1/2}\,(\mathrm{X}_c^2 + \mathrm{Y}_c^2)^{-1/2} \\ \mathrm{Y}_c\ (\mathrm{X}_c^2 + \mathrm{Y}_c^2 + \mathrm{Z}_c^2)^{1/2}\,(\mathrm{X}_c^2 + \mathrm{Y}_c^2)^{-1/2} \end{bmatrix}. \tag{B.3}$$

This expression takes a more convenient form if we specify the range as

$$r = \left(\mathrm{X}_c^2 + \mathrm{Y}_c^2 + \mathrm{Z}_c^2\right)^{1/2} = \hat{r} + \delta r,$$

where $\hat{r} = (\mathrm{X}_c^2 + \mathrm{Y}_c^2)^{1/2} = r\cos\beta$ is the length of the orthogonal projection on the image plane. The disparity between the range and the projection length is a function of the elevation angle, $\delta r = \hat{r}\left(\sqrt{1+\tan^2\beta} - 1\right)$. Together, this notation yields an alternative representation (B.4a) for the nonlinear projection model.

$$\tilde{\mathbf{u}} = \left(1 + \frac{\delta r}{\hat{r}}\right) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{X}_c \\ \mathrm{Y}_c \\ \mathrm{Z}_c \end{bmatrix} = \sqrt{1+\tan^2\beta} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{X}_c \\ \mathrm{Y}_c \\ \mathrm{Z}_c \end{bmatrix} \tag{B.4a}$$

$$\approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{X}_c \\ \mathrm{Y}_c \\ \mathrm{Z}_c \end{bmatrix} \tag{B.4b}$$

The DIDSON's narrow FOV in elevation suggests that we can ignore the nonlinear aspect of the model (B.4a), and approximate the projection by one that is completely linear. More specifically, the $|\beta| \leq 6°$ bound on the target's elevation corresponds to a tight limit on the nonlinear component as $1 \leq \sqrt{1+\tan^2\beta} \leq 1.0055$ and, equivalently, $0 \leq \frac{\delta r}{\hat{r}} \leq 0.0055$. Consequently, if we make the narrow elevation approximation, $\sqrt{1+\tan^2\beta} \approx 1$, the nonlinear model simplifies to the linear form (B.4b), and we can then view the sonar imaging process as one of orthogonal projection onto the image plane.

The approximation relies on the assumption that the scene's relief in the invariant elevation direction is negligible relative to its extent in other directions. Assuming suitable viewing geometry for the DIDSON (i.e. a small grazing angle), the sonar's narrow FOV supports this assumption. While there is ambiguity in the elevation angle for each acoustic return, this uncertainty is small with respect to range. This relationship is analogous to that of a perspective camera that images a scene with a depth relief that is negligible in comparison to the distance to the camera. If this is the case and the camera has a limited FOV, the perspective camera geometry can be accurately approximated by an affine imaging model [58]. The same is true for the DIDSON acoustic camera, whose narrow elevation component of the FOV allows us to model image formation by the linear approximation (B.4b) with limited error. To better understand this approximation, we express the camera model (B.4) in homogeneous coordinates and consider an arbitrary transformation, $(\mathrm{R}, \mathbf{t})$, from the world frame to the camera's frame. With these changes, we arrive at an orthographic projection model for the DIDSON imaging geometry:

$$\mathbf{u} \approx \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{R} & \mathbf{t} \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix} = \mathrm{P}\mathbf{X}, \qquad \mathrm{P} = \begin{bmatrix} \mathbf{r}_1^\top & t_x \\ \mathbf{r}_2^\top & t_y \\ 0_{3\times 1} & 1 \end{bmatrix} \tag{B.5}$$

The orthographic camera matrix, P, consists of the first two rows of the rotation matrix, $\mathbf{r}_1^\top$ and $\mathbf{r}_2^\top$, and two components of the translation vector. As is the case for orthographic cameras, the DIDSON projection matrix has only five degrees of freedom, namely the three angles that define the rotation relative to the world frame, R, and the x and y components of the translation.

We note that Kim *et al.* [69] and Negahdaripour *et al.* [102] identify much the same approximation of the DIDSON imaging geometry. Taking advantage of the sonar's narrow FOV in elevation and assuming that the sonar ensonifies the environment at a small grazing angle, they treat the scene as being locally planar. This assumption reduces the imaging model to an orthographic projection, $\mathrm{P} = \mathrm{P}\left(\mathrm{X}_c, \mathrm{Y}_c, \mathrm{Z}_c, \mathbf{n}\right)$, that is a non-uniform function of image coordinates and the scene plane normal, $\mathbf{n}$. They then impose the narrow elevation approximation to achieve a linear projection model, $\mathrm{P} = \mathrm{P}(\mathbf{n})$, that is uniform over the image, as above (B.5).

The expression for the DIDSON camera (B.5) models the imaging geometry as an affine transformation followed by orthographic (parallel) projection. An important consequence of this decomposition is that it represents the sonar as an affine camera, which possesses a number of unique properties that differentiate it from standard pinhole cameras. In particular, the camera center associated with affine cameras lies on the plane at infinity. This implies that, unlike perspective projection, the camera preserves parallelism, as parallel lines in the world are also parallel in the image. Similarly, the projection rays are all parallel with one another and perpendicular to the image plane, since we approximate the projection as orthographic. As we discuss next, the parallel projection model of the DIDSON directly affects the constraints that we can impose on camera pairs.

B.1.2 Affine Epipolar Geometry

With a linear approximation for the DIDSON projection model, we can use the same framework that is employed for optical cameras to constrain the relative motion between pairs of acoustic cameras. These constraints follow from the epipolar geometry that describes the relationship between image point pairs. However, inherent differences in the affine epipolar geometry lead to greater ambiguity in the relative camera motion as well as in the scene structure. In addition to the scale ambiguity, it has long been known that two affine views are invariant to a one parameter family of relative transformations that can not be resolved from fewer than three images [61, 70].

The unique nature of the epipolar geometry that governs affine camera pairs can be attributed, in large part, to the fact that the principle plane is the plane at infinity. Consider two acoustic images taken from different vantage points,[4] We assume that the first camera pose is coincident with the global reference frame and let $(\mathrm{R}, \mathbf{t})$ represent the transformation from the first camera pose frame to that of the second pose. Per the linear approximation to the imaging model (B.5), the corresponding orthographic projection matrices are

$$\mathrm{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathrm{P}_2 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.6}$$

The corresponding fundamental matrix for affine camera pairs has the form given in (B.7) (the derivation is straightforward and is described in Hartley and Zisserman [58]). With five non-zero elements, the homogeneous fundamental matrix has four degrees of freedom.

$$\mathrm{F}_\mathrm{A} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{bmatrix} \qquad \begin{array}{ll} a = r_{23} \;\; b = -r_{13} & c = r_{13}r_{21} - r_{11}r_{23} \\ d = r_{13}r_{22} - r_{12}r_{23} & e = r_{13}t_2 - r_{23}t_1 \end{array} \tag{B.7}$$

The two epipoles, which are the right and left null vectors of F_A (i.e. $\mathrm{F}_\mathrm{A}\mathbf{e} = 0$ and $\mathrm{F}_\mathrm{A}^\top \mathbf{e}' = 0$), are given by $\mathbf{e} = [\, b \; -a \; 0\,]^\top$ and $\mathbf{e}' = [\, d \; -c \; 0\,]^\top$. The epipolar lines, $\mathbf{l}$ and

[4] In the context of delayed-state SLAM, we are particularly interested in the case where the same camera images the scene at different points in time. Nonetheless, the subsequent derivation applies just as well to images from arbitrary camera/time pairs.

$\mathbf{l}'$, which relate corresponding points in the first and second image, are all parallel since they intersect at points on the plane at infinity. As a result, the epipolar planes, which intersect the image plane at the epipolar lines, are parallel and, like the projection rays, perpendicular to the image plane.

$$\begin{aligned} \mathbf{e} &= \begin{bmatrix} b & -a & 0 \end{bmatrix}^\top & \mathbf{l} &= \begin{bmatrix} a & b & cx+dy+e \end{bmatrix}^\top \\ \mathbf{e}' &= \begin{bmatrix} d & -c & 0 \end{bmatrix}^\top & \mathbf{l}' &= \begin{bmatrix} c & d & ax+by+e \end{bmatrix}^\top \end{aligned} \tag{B.8}$$

Given corresponding image pairs, $\mathbf{u}_i = [x_i \; y_i \; 1]^\top$ and $\mathbf{u}'_i = [x'_i \; y'_i \; 1]^\top$, the fundamental matrix describes the affine epipolar constraint,

$$\mathbf{u}_i^\top \mathrm{F}_{\mathrm{A}} \mathbf{u}' = 0 \longrightarrow ax_i + by_i + cx'_i + dy'_i + e = 0 \tag{B.9}$$

The unique nature of the affine epipolar geometry, namely the fact that the epipolar planes are parallel, gives rise to invariance in the constraints between camera pairs. In addition to the scale ambiguity, the inferred scene structure and camera motion associated with two affine views is invariant to a one-parameter pencil of transformations [61, 70]. More specifically, given only two views of a scene, parallel projection is subject to two ambiguities that confound the structure and relative camera motion: *Necker reversal* and the *bas-relief* ambiguity [58]. Necker reversal refers to the fact that the parallel projection image of an object that rotates by an angle ρ is identical to that of the mirror object[5] that rotates by $-\rho$. Equivalently, if the object remains stationary and the camera moves, we can not tell whether the camera rotated by ρ about an axis parallel to the image plane, or by $-\rho$ and imaged the mirror object. The bas-relief ambiguity describes the inability to decouple the relationship between the size of the imaged objects in the direction of projection and the extent of their rotation. Under parallel projection, large objects that undergo small rotations yield identical image pairs as shallow objects that rotate by a greater amount. The same holds if the scene is static and the camera rotates.

The bas-relief ambiguity is particularly important in situations where the goal is to exploit the epipolar geometry to estimate the relative motion of the camera. Given two views of a static scene under parallel projection, point correspondence establishes a set of parallel epipolar planes that lie perpendicular to the two image planes. One can imagine rotating the second camera about an axis perpendicular to the epipolar planes while changing the unknown depth relief of the imaged scene points. We demonstrate this ambiguity with Figure B-2, in which we rotate the second camera about a fronto-parallel axis. By simultaneously shortening the line in the scene along the projection ray of the first camera, the two image pairs are invariant to the rotation.

As a consequence of the bas-relief ambiguity, there is a one-parameter family of transformations that describes the relative motion between affine camera pairs [61, 70]. Koenderink and van Doorn [70] present a rotation model that accounts for this invariance by isolating the bas-relief ambiguity. The represent the transformation as

[5] Reflected relative to a plane parallel to the image plane.

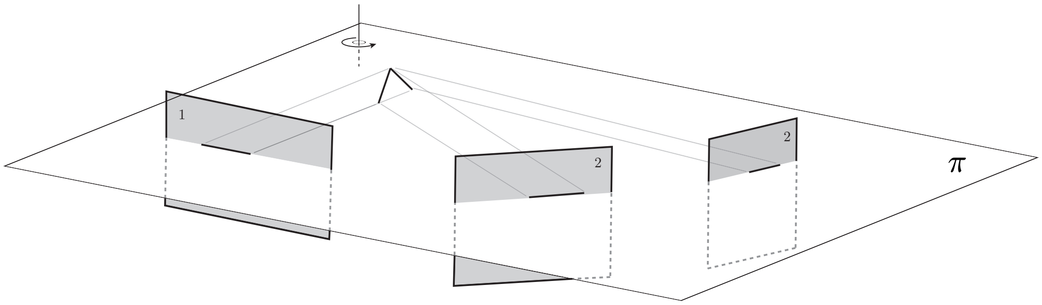


Figure B-2: Pairs of affine camera images are subject to the *bas-relief* ambiguity that describes the coupling between the scene's depth relief and the relative camera rotation. Under parallel projection, a large camera rotation about an axis that lies within a fronto-parallel plane and a shallow scene produces the same pair of images as a smaller rotation but larger depth relief. This effect gives rise to a one parameter family of transformations between affine camera pairs.

a pair of rotations of the base camera frame. They first rotate the camera within the image plane about the z-axis such that the orientation of the epipolar lines is the same within both images. The subsequent rotation takes place about an axis, Φ, that lies at a known orientation in a fronto-parallel plane. This rotation corresponds directly to the bas-relief ambiguity as the distance to the plane and the rotation angle are coupled as part of the one-parameter family of relative motion.

Shapiro *et al.* [117] propose a concise formulation for the Koenderink and van Doorn model in terms of three observable variables, (θ, ϕ, s), and a single free parameter, ρ. The terms θ, ϕ, and ρ parametrize the two rotations while s is the relative scale between cameras. Considering an affine camera pair of the form given in (B.5), they describe the relative rotation as

$$\mathrm{R} = \mathrm{R}\left(\Phi, \rho\right)\, \mathrm{R}\left(\mathrm{z}, \theta\right). \tag{B.10}$$

The matrix $\mathrm{R}\left(\mathrm{z}, \theta\right)$ is the image plane rotation by θ about the z-axis that yields parallel epipolar lines. The camera then rotates by ρ out of the image plane about the Φ axis, which projects onto the image at angle ϕ with respect to the x-axis. The $\mathrm{R}\left(\Phi, \rho\right)$ matrix accounts for this bas-relief rotation.

Shapiro *et al.* relate the one-parameter family of transformations (B.10) to the affine epipolar constraint (B.9). From this constraint, they derive the (θ, ϕ, s) variables as a function of the five parameters, (a, b, c, d, e), that define the fundamental matrix, F_A, in (B.7):

$$\tan\phi = \frac{b}{a} \qquad \tan\left(\phi - \theta\right) = \frac{d}{c} \tag{B.11}$$

Note that we have omitted the relative depth factor, s, as $s = 1$ in the case of orthogonal projection.

One can estimate the fundamental matrix based upon as few as seven paired observations of non-planar structure, one for each degree of freedom. As mentioned with regards to the essential matrix, however, the number of correspondences is much larger and estimators often solve for the fundamental matrix that minimizes the total error associated with the entire set of points. Shapiro *et al.* [117] show the total point-to-point error within both images to be a particularly good metric, in part, because it explicitly accounts for noise in the location of image points.[6] Under the assumption that the noise is independent and Gaussian, the minimum solution is equivalent to the maximum likelihood estimate for the fundamental matrix [58].

Given $n \geq 7$ paired observations of non-planar structure, $(\mathbf{u}_i, \mathbf{u}'_i)$, we can solve for $\mathrm{F_A}$ up to scale. A typical strategy for estimating the matrix is to solve for the set of five parameters, (a, b, c, d, e) that minimize the total error associated with the epipolar constraint equation for each point pair,

$$\left\{\mathbf{u}_i = [x_i \; y_i \; 1]^\top \leftrightarrow \mathbf{u}'_i = [x'_i \; y_i\prime \; 1]^\top\right\} : \; \mathbf{u}_i^\top \mathrm{F_A} \mathbf{u}' = 0 \longrightarrow ax_i + by_i + cx'_i + dy'_i + e = 0.$$

Recalling our earlier discussion on multiple view geometry techniques in Section B.1, an estimate for the fundamental matrix between an acoustic camera pair provides a constraint on their relative pose. In particular, the matrix parameters yield an estimate of the relative rotation between frames (B.10) via the formulation (B.11) that Shapiro *et al.* describe. As we have discussed, the epipolar geometry for optical camera pairs provides five constraints on the six-DOF transformation between frames. In contrast, the epipolar geometry that results from an affine approximation to acoustic cameras yields only two of the six transformation parameters.

[6] When the relative scale is one, as is the case for orthographic projection, the minimum total point-to-point solution is equal to that of the point-to-line error.

Bibliography

[1] Adaptive sampling and prediction (ASAP), Princeton University website. `http://www.princeton.edu/~dcsl/asap/`, 2006.

[2] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, Sydney, Australia, August 2002.

[3] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part ii. *IEEE Robotics and Automation Magazine*, 13(3):108–117, September 2006.

[4] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the EKF-SLAM algorithm. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3562–3568, Beijing, China, October 2006.

[5] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, 2001.

[6] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, 2nd edition, 1994.

[7] P. Beardsley, A. Zisserman, and D. Murray. Navigation using affine structure from motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 85–96, Stockholm, 1994.

[8] P. Beardsley, A. Zisserman, and D. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Robotics Research*, 23(3):235–259, June 1997.

[9] E. Belcher, B. Matsuyama, and G. Trimble. Object identification with acoustic lenses. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 1, pages 6–11, Honolulu, HI, November 2001.

[10] J. G. Bellingham and K. Rajan. Robotics in remote and hostile environments. *Science*, 318(5853):1098–1102, November 2007.

[11] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A.K. Peters, Ltd., Wellesley, MA, 1996.

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

[13] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, April 1977.

[14] J. Castellanos and J. Tardós. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, Boston, MA, 2000.

[15] J. A. Castellanos, J. Neira, and J. Tardós. Limits to the consistency of EKF-based SLAM. In *Proceedings of the Fifth IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 2004.

[16] J. A. Castellanos, J. Tardós, and G. Schmidt. Building a global map of the environment of a mobilee robot: The importance of correlation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1053–1059, Albuquerque, NM, April 1997.

[17] R. Chatila and J. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 138–145, St. Louis, MO, March 1985.

[18] I. Cox and J. Leonard. Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, April 1994.

[19] N. Cui, J. Weng, and P. Cohen. Recursive-batch estimation of motion and structure from monocular image sequences. *CVGIP: Image Understanding*, 59(2):154–170, March 1994.

[20] T. Curtin, J. Bellingham, J. Catapovic, and D. Webb. Autonomous oceanographic sampling networks. *Oceanography*, 6(3):86–94, 1993.

[21] R. Damus, S. Desset, J. Morash, V. Polidoro, F. Hover, C. Chryssostomidis, J. Vaganay, and S. Willcox. A new paradigm for ship hull inspection using a holonomic hover-capable AUV. In *Proceedigns of the First International Conference on Informatics in Control, Automation, and Robotics (ICINCO)*, pages 127–132, Setúbal, Portugal, August 2004.

[22] T. Davis, J. Gilbert, S. Larimore, and E. Ng. A column approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 30(3):353–376, September 2004.

[23] T. Davis and W. Hager. Modifying a sparse cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 20(3):606–627, 1999.

[24] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1403–1410, 2003.

[25] A. Davison and N. Kita. 3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 384–391, Hawaii, December 2001.

[26] J. de Freitas. *Bayesian Methods for Neural Networks*. PhD thesis, Department of Engineering, Cambridge University, Cambridge, U.K., 1999.

[27] F. Dellaert. Square root SAM. In *Proceedings of Robotics: Science and Systems (RSS)*, pages 177–184, Cambridge, MA, June 2005.

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

[29] A. Doucet, N. de Freitas, K. Murphy, and S. Russel. Rao-Blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 176–183, Stanford, CA, 2000.

[30] T. Duckett, S. Marsland, and J. Shapiro. Learning globally consistent maps by relaxation. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3841–3846, San Francisco, 2000.

[31] T. Duckett, S. Marsland, and J. Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12(3):287–300, May 2002.

[32] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part i. *IEEE Robotics and Automation Magazine*, 13(2):99–110, June 2006.

[33] R. Eustice. *Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles*. PhD thesis, Massachusetts Institute of Technology / Woods Hole Oceanographic Institution Joint Program, Cambridge, MA, June 2005.

[34] R. Eustice, H. Singh, and J. Leonard. Exactly sparse delayed-state filters. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2417–2424, Barcelona, Spain, April 2005.

[35] R. Eustice, H. Singh, J. Leonard, and M. Walter. Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. *International Journal of Robotics Research*, 25(12):1223–1242, December 2006.

[36] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Proceedings of Robotics: Science and Systems (RSS)*, pages 57–64, Cambridge, MA, June 2005.

[37] R. Eustice, M. Walter, and J. Leonard. Sparse extended information filters: Insights into sparsification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 641–648, Edmonton, Alberta, Canada, August 2005.

[38] H. Feder, J. Leonard, and C. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18(7):650–668, July 1999.

[39] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis andd automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[40] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In H. Burkhardt and B. Neumann, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 1 of *Lecture Notes in Computer Science*, pages 311–326. Springer-Verlag, June 1998.

[41] J. Folkesson and H. Christensen. Graphical SLAM - a self-correcting map. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 383–390, New Orleans, LA, April 2004.

[42] T. I. Fossen. *Guidance and Control of Ocean Vehicles.* John Wiley & Sons, Inc., New York, NY, 1994.

[43] D. Fox, W. Burgard, S. Thrun, and A. Cremers. Position estimation for mobile robots in dynamic environments. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Madison, WI, July 1998.

[44] U. Frese. A proof for the approximate sparsity of SLAM information matrices. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 331–337, Barcelona, Spain, April 2005.

[45] U. Frese. Treemap: An $\mathcal{O}(\log n)$ algorithm for simultaneous localization and mapping. In C. Freksa, editor, *Spatial Cognition IV*, pages 455–476. Springer–Verlag, 2005.

[46] U. Frese and G. Hirzinger. Simultaneous localization and mapping - a discussion. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 17–26, 2001.

[47] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, 21(2):196–207, April 2005.

[48] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[49] N. Gordon, D. Salmond, and A. Smith. Novel approaches to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2):107–113, April 1993.

[50] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2432–2437, Barcelona, April 2005.

[51] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[52] J. Guivant and E. Nebot. Solving computational and memory requirements of feature-based simultaneous localization and map building algorithms. Technical report, Australian Centre for Field Robotics, University of Sydney, Sydney, 2002.

[53] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in SLAM. In *Proceedings of the 11th International Symposium of Robotics Research (ISRR)*, Sienna, Italy, 2003. Springer.

[54] D. Hähnel, D. Fox, W. Burgard, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 206–211, Las Vegas, NV, October 2003.

[55] C. Harris and J. Pike. 3D positional integration from image sequences. In *Third Alvey Vision Conference*, pages 233–236, 1987.

[56] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, Manchester, U.K, 1988.

[57] R. Hartley. In defense of the 8-point algorithm. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1064–1070, Cambridge, MA, June 1995.

[58] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[59] J. Holmes, G. Cronkite, H. Enzenhofer, and T. Mulligan. Accuracy and precision of fish-count data from a "dual-frequency identification sonar" (DIDSON) imaging system. *ICES Journal of Marine Science*, 63(3):543–555, 2006.

[60] F. Hover, J. Vaganay, M. Elkins, S. Willcox, V. Polidoro, J. Morash, R. Damus, and S. Desset. A vehicle system for autonomous relative survey of in-water ships. *Marine Technology Society Journal*, 41(2):44–55, 2007.

[61] T. Huang and C. Lee. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):536–540, May 1989.

[62] J. Jaffe. Computer modeling and the design of optimum underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):1001–111, April 1990.

[63] M. Jordan. *An Introduction to Probabilistic Graphical Models*. In preparation.

[64] S. Julier and J. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the American Control Conference*, Albuquerque, NM, June 1997.

[65] S. Julier and J. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 4238–4243, Seoul, South Korea, May 2001.

[66] M. Kaess, A. Ranganathan, and F. Dellaert. Fast incremental square root information smoothing. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2129–2134, Hyderabad, India, January 2007.

[67] R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45, 1960.

[68] J. Kim and S. Sukkarieh. Airborne simultaneous localisation and map building. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 406–411, Taipei, Taiwan, September 2003.

[69] K. Kim, N. Intrator, and N. Neretti. Image registration and mosaicing of acoustic camera images. In *Proceedings of the 4th IASTED International Conference on Visualization, Imaging, and Image Processing*, pages 713–718, 2004.

[70] J. Koenderink and A. van Doorn. Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385, 1991.

[71] M. A. Kokko. Range-based navigation of AUVs operating near ship hulls. Master's thesis, Massachusetts Institute of Technology, June 2007.

[72] B. Kuipers and Y. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8(1-2):47–63, 1991.

[73] S. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[74] J. Leonard, H. Durrant-Whyte, and I. Cox. Dynamic map building for an autonomous mobile robot. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems (IROS)*, volume 1, pages 89–96, Ibaraki, Japan, July 1990.

[75] J. Leonard, H. Durrant-Whyte, and I. Cox. Dynamic map building for an autonomous robot. *International Journal of Robotics Research*, 11(4):286–298, April 1992.

[76] J. Leonard and H. Feder. A computationally effecient method for large-scale concurrent mapping and localization. In *Proceedings of the Ninth International Symposium of Robotics Research*, pages 169–176, Salt Lake City, UT, October 1999.

[77] J. Leonard and P. Newman. Consistent, convergent, and constant-time SLAM. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1143–1150, Acapulco, Mexico, August 2003. IJCAI.

[78] J. Leonard, R. Rikoski, P. Newman, and M. Bosse. Mapping partially observable features from multiple uncertain vantage points. *International Journal of Robotics Research*, 21(10):943–975, October 2002.

[79] Y. Liu and S. Thrun. Results for outdoor-SLAM using sparse extended information filters. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1227–1233, Taipei, Taiwan, May 2003.

[80] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[81] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, Corfu, Greece, September 1999.

[82] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, April 1997.

[83] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18(249-275), 1997.

[84] D. J. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, Pasadena, CA, 1992.

[85] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte. An experiment in integrated exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 534–539, Lausanne, Switzerland, September 2002.

[86] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, New York, NY, 1979.

[87] B. L. McGlamery. A computer model for underwater camera systems. In S. Q. Duntley, editor, *Ocean Optics VI, Proc. SPIE*, volume 208, pages 221–231, 1979.

[88] P. McLauchlan. A batch/recursive algorithm for 3D scene reconstruction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 738–743, Hilton Head, SC, June 2000.

[89] P. Mignotte, S. Reed, A. Cormack, and Y. Petillot. Automatic ship hull inspection - the detection of mine-like targets in sonar data using multi-CAD fusion and tracking technologies. In *Proceedings of the International Conference on Detection and Classification of Underwater Targets*, Edinburgh, September 2007.

[90] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 525–531, Vancouver, Canada, July 2001.

[91] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, October 2004.

[92] R. Mohr, F. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 543–548, New York, NY, June 1993.

[93] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598, Edmonton, Canada, 2002.

[94] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1151–1156, Acapulco, Mexico, 2003.

[95] M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 695–701, Washington, DC, May 2002.

[96] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 116–121, St. Louis, MO, 1985.

[97] P. Moutarlier and R. Chatila. An experimental system for incremental environment modelling by an autonomous mobile robot. In *Proceedings of the 1st International Symposium on Experimental Robotics*, pages 327–346, Montreal, Canada, June 1989.

[98] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1015–1021, 1999.

[99] A. G. Mutambara. *Decentralized Estimation and Control for Multisensor Systems.* CRC Press, Boston, MA, 1998.

[100] F. Naderi, D. McCleese, and J. Jordan, Jr. Mars exploration. *IEEE Robotics and Automation Magazine*, 13(2):72–82, June 1996.

[101] S. Negahdaripour and P. Firoozfam. An ROV stereovision system for ship-hull inspection. *IEEE Journal of Oceanic Engineering*, 31(3):551–564, July 2006.

[102] S. Negahdaripour, P. Firoozfam, and P. Sabzmeydani. On processing and registration of forward-scan acoustic video imagery. In *Proceedings of the Second Canadian Conference on Computer and Robot Vision*, pages 452–459, May 2005.

[103] J. Neira and J. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, December 2001.

[104] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1180–1187, Orlando, FL, May 2006.

[105] P. Newman and K. Ho. SLAM - loop closing with visually salient features. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 635–642, Barcelona, Spain, April 2005.

[106] P. Newman and J. Leonard. Pure range-only subsea SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1921–1926, Taiwan, September 2003.

[107] P. M. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem.* PhD thesis, University of Sydney, Australia, 1999.

[108] D. Nistér. An efficient solution to the five-point relative pose problem. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[109] K. Ogata. *Discrete-Time Control Systems.* Prentice-Hall, Upper Saddle River, NJ, 2nd edition, 1995.

[110] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2262–2269, May 2006.

[111] E. Olson, J. Leonard, and S. Teller. Spatially-adaptive learning rates for online incremental SLAM. In *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA, June 2007.

[112] E. Olson, M. Walter, S. Teller, and J. Leonard. Single-cluster spectral graph partitioning for robotics applications. In *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, MA, July 2005.

[113] M. Paskin. Thin junction tree filters for simultaneous localization and mapping. Technical Report UCB/CSD-02-1198, University of California, Berkeley, September 2002.

[114] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, 1988.

[115] O. Pizarro, R. Eustice, and H. Singh. Large area 3D reconstruction from underwater surveys. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 2, pages 678–687, Kobe, Japan, November 2004.

[116] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, September 1951.

[117] L. Shapiro, A. Zisserman, and M. Brady. 3D motion recovery via affine epipolar geometry. *International Journal of Computer Vision*, 16(2):147–182, 1995.

[118] J. Shewchuck. An introduction to the conjugate gradient method without the agonizing pain. Technical Report CMU-CS-94-125, Carnegie Mellon University, August 1994.

[119] R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. The MIT Press, Cambridge, MA, April 2004.

[120] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In O. Faugeras and G. Giralt, editors, *Proceedings of the International Symposium of Robotics Research (ISRR)*, pages 467–474, 1988.

[121] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.

[122] T. Speed and H. Kiiveri. Gaussian Markov distributions over finite graphs. *Annals of Statistics*, 14(1):138–150, March 1986.

[123] C. Stachniss. *Exploration and Mapping with Mobile Robots*. PhD thesis, University of Freiburg, Department of Computer Science, April 2006.

[124] R. Szeliski and S. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. Technical Report CRL 93/3, Digital Equipment Corporation, Cambridge, MA, March 1993.

[125] J. Tardós. Representing partial and uncertain sensorial information using the theory of symmetries. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1799–1804, Nice, France, 1992.

[126] J. Tardós, J. Neira, P. Newman, and J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal Robotics of Research*, 21(4):311–330, April 2002.

[127] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2002.

[128] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[129] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Ng. Simultaneous mapping and localization with sparse extended information filters: Theory and initial results. In *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics*, Nice, France, 2002.

[130] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, July-August 2004.

[131] Transportation Research Board. *Special Report 279: The Marine Transportation System and the Federal Role*. National Research Council, Washington, DC, 2004.

[132] J. Vaganay, M. Elkins, D. Esposito, W. O'Halloran, F. Hover, and M. Kokko. Ship hull inspection with the HAUV: US Navy and NATO demonstrations results. In *Proceedings of OCEANS MTS/IEEE Conference and Exhibition*, volume 1, pages 761–766, Boston, MA, September 2006.

[133] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR380, Cambridge University Engineering Department, August 2000.

[134] R. Volpe. Rover technology development and mission infusion beyond MER. In *IEEE Aerospace Conference*, pages 971–981, Big Sky, MT, March 2005.

[135] C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2918–2924, Washington, DC, May 2002.

[136] Z. Wang, S. Huang, and G. Dissanayake. D-SLAM: Decoupled localization and mapping for autonomous robots. In *Proceedings of the 12th International Symposium of Robotics Research*, San Francisco, CA, October 2005.

[137] L. Whitcomb, D. Yoerger, H. Singh, and J. Howland. Advances in underwater robot vehicles for deep ocean exploration: Navigation, control, and survey operations. In *Proceedings of the International Symposium of Robotics Research (ISRR)*, pages 439–448, Snowbird, UT, October 1999.

[138] L. Whitcomb, D. Yoerger, H. Singh, and D. Mindell. Towards precision robotic maneuvering, survey, and manipulation in unstructured undersea environments. In Y. Shirai and S. Hirose, editors, *Robotics Research – The Eighth International Symposium*, pages 45–54, London, 1998. Springer-Verlag.

[139] S. Williams, G. Dissanayake, and H. Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 406–411, Washington, DC, May 2002.

[140] S. Williams, P. Newman, M. Dissanayake, and H. Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 22–28, San Francisco, CA, April 2000.

[141] D. Yoerger, A. Bradley, M. Cormier, W. Ryan, and B. Walden. High resolution mapping of a fast spreading mid ocean ridge with the autonomous benthic explorer. In *Proceedings of the International Symposium on Unmanned Untethered Subersible Technology (UUST)*, pages 21–31, Durham, NH, August 1999.

[142] D. Yoerger, M. Jakuba, A. Bradley, and B. Bingham. Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *The International Journal of Robotics Research*, 26(1):41–54, January 2007.

[143] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.