Toward Robust Multi-Agent Autonomous Underwater Inspection with Consistency and Global Optimality Guarantees

by

Joshua Gary Mangelson

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in the University of Michigan
2019

Doctoral Committee:

Professor Ryan M. Eustice, Co-Chair
Assistant Professor Ram Vasudevan, Co-Chair
Professor Jessy W. Grizzle
Professor John J. Leonard

Joshua Gary Mangelson

mangelso@umich.edu

ORCID iD: 0000-0002-0550-0368

ACKNOWLEDGMENTS

I have to say, of all the writing that has happened in the last five years, the following two pages are probably the most important. I have had the opportunity and blessing to be able to work with and be taught by an amazing group of people throughout my time at the University of Michigan (UofM) and my life in general. So many of you have made a difference in my life and trajectory and I am thankful to each and every one of you.

First, I want to say thank you to Ryan. You have been an amazing advisor – incredibly understanding, supportive, and patient. I'm not sure why you chose me as a student all those years ago, but I have absolutely loved it. Thank you for introducing me to the wonderful world of mobile and marine robotics. Who ever thought that I could be playing with underwater robots for a living? Thank you for teaching me how to tell a story both with words and visuals. I will always be grateful for your help, support, and friendship.

Next, Ram I want to say thank you for being willing to add that bright eyed kid who came up to you and said, "I want to learn about convex relaxation", to your lab. Thank you for the many hours at the white board and the encouragement over slack and hangouts. Thank you (and Talia) for forcing me to redo the figures over and over again until the colors were perfect and for the many practice talks. I have loved being a part of your lab.

I would also like to thank the other members of my committee, Jessy and John. Thank you for your helpful comments, mentorship, and time. I am particularly thankful to Jessy for your support throughout my time in Robotics at UofM. Our discussions in the classroom and your office have made a significant difference in my time here and I want to thank you for the time and effort you put into your teaching and students. You once said that your kids are your robots, I think its more like your kids are your students.

Thank you to the past teachers and mentors who led me down this path. Mr. King (Greg), thank you for introducing me to robotics and engineering. It has really become one of my true passions. Thank you to my undergraduate advisor Brent, it was under your tutelage that I decided I wanted to get a PhD.

Thank you to everyone in the PeRL and ROAHM labs, (in the order I met them, best I can remember): Arash, Paul, Jeff, Nick, Steven (Parkison), Katie, Stephen (Chavez), Jie, Wolcott(Ryan), Vittorio, Enric, GT, Steve (Vozar), Alex, James, Derrick, Nils, Matthew, Pengcheng, Shreyas, Patrick, Dan, Talia, Shannon, Sean, Basel, Hyongju, Jinsun, Fan, Owen,

Maani, Ali, Lu, Frank, Ray, Daphna, Rohit, and Hannah – each of you have been wonderful. Thank you for the countless hours bouncing ideas back and forth, the much needed feedback, going to lunch, and learning from one another. I look up to you.

Thank you to everyone that made field trials possible: Michael, Greg, Tami, Murph, Pedro, Eric, Tiffany, Paloma, Bing, and all the PeRL underwater folks: Ryan, Jeff, Paul, Steve, Jie, and James. I'll always remember the sushi, manning the tether, the feeling like the land is swimming, and the good italian food. And, I'm not sure I will ever be able to voluntarily eat Subway again.

Thank you to Ross, Katie, Mia, Meghan, Vittorio, and Katelyn, you guys are awesome. I'm thankful to have done this together.

And now on to family – I want to thank my two grandfathers.

My grandpa Veldon was an engineer for all his life. Though he didn't talk much about what he worked on, I am sure his career path influenced my decision to become an engineer and I think it is likely that he has helped me during some of those late nights seeking for a solution to a research problem.

Grandpa Nolan, you are one of my prime role models. Your love for your wife, family, students, and everyone around you is obvious and contagious. One of my most fervent goals is to be the type of professor and husband you are.

Thank you to each of my siblings, both biological and through marriage. You are my best friends in the entire world. I love each of you and I am thankful to have each of you in my family. To Bethany and Paul, especially, thank you for keeping my wife sane through the past five years. And Bethany, I'll let you know if I ever see a mermaid.

Mary and Allan, I love being a member of your family. Thank you for raising such a wonderful daughter and accepting me as an additional son.

Mom, you are one of the strongest people I have ever or ever will know. You are an example to me of perseverance, love, and service and I'm glad my children have you for a grandmother. I'm sure I would never have gotten this far if you hadn't found Dublin Robotics and intervened to help me find something I would love.

Dad, I love you. I'm thankful for your example of hard work and sacrifice. I want to be the type of Father you are. Thank you for getting up everyday, going to work to support your family, and then coming home and switching your focus to us and our lives.

To my three girls – Andi, Linnie, and Isla. Each of you are amazing. I love the Joy, Faith, and Hope you bring to our lives. I am so thankful that I have you forever.

Kaytlin, you are my everything. I love you with all my heart. It has been your support that has kept me going through the past five years. I don't know how many times I felt like maybe I couldn't do it anymore and you were the one to keep me going. I love spending life with you. I am excited to face what comes next with you by my side.

Lastly, I want to thank my Heavenly Father and Savior Jesus Christ. I am absolutely certain I would not have made it through the difficulties of the past five years, in addition to grad-school, in one piece without their peace, assurance, and miracles.

Funding

This work was supported by the Office of Naval Research under award numbers:
N00014-12-1-0092 and N00014-16-1-2102.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ACRONYMS

AUV autonomous underwater vehicle

BSOS bounded degree sum-of-squares

BCH Baker-Campbell-Housedorff

CMU Carnegie Mellon University

DCS Dynamic Covariance Scaling

DOF degree of freedom

DVL Doppler velocity log

EKF extended Kalman filter

EM expectation-maximization

GICP generalized iterative closest point

GPS global positioning system

GkCM group-k consistency maximization

HAUV Hovering Autonomous Underwater Vehicle

IMU inertial measurement unit

iSAM incremental smoothing and mapping

LP linear program

MAP maximum *a posteriori*

MIT Massachusetts Institute of Technology

MLE maximum likelihood estimate

MSE mean-squared error

MAV micro aerial vehicle

NCLT University of Michigan North Campus Long-Term Vision and LiDAR

ONR Office of Naval Research

PCM pairwise consistency maximization

RANSAC random sample consensus

SLAM simultaneous localization and mapping

UofM University of Michigan

UT Unscented Transform

ABSTRACT

Teams of autonomous robotic systems have the potential to have a dramatic positive effect on our society. In the underwater domain specifically, collaborative multi-agent autonomous systems have the potential to lead to significant increases in efficiency, safety, and data quality. However, while autonomous systems have been widely accepted within structured environments such as manufacturing plants and distribution facilities, they have not been nearly as widely adopted in unstructured environments. The primary reason for this is that the reliability of autonomous systems in unstructured environments has not yet reached a level where it is cost and time effective to widely adopt such platforms. One key element of the reliability of autonomous systems is the robustness of navigation and localization algorithms to common failure cases such as outlier measurements, bad initialization, and inaccurate characterizations of uncertainty. Accordingly, this thesis proposes methods for simultaneous localization and mapping (SLAM), multi-agent map merging, trajectory alignment, and uncertainty characterization that seek to address some of these failure cases.

First, we propose an algorithm for robust map merging that takes two pose graphs and a set of potential loop-closures between them and selects a set of those potential loop-closures that can be used to consistently align and merge the two maps. Our proposed algorithm requires no initial estimate of alignment and can handle outlier ratios of over 90%. We take advantage of existing maximum clique algorithms for increased efficiency and show that our algorithm outperforms existing state-of-the-art methods.

Second, we propose an algorithm for localizing a query trajectory to a reference trajectory based solely on low-dimensional data describing the environment around the robotic agent at each position it visited. Our approach takes advantage of convex relaxation techniques to avoid the need for initialization and data association, making it useful in cases where high-dimensional data is unavailable. We compare our proposed method to other existing convex optimization techniques and show that it better enforces a rigid body transformation than other existing methods.

Third, we formulate the planar pose graph SLAM and landmark SLAM problems as polynomial optimization problems and prove that the globally optimal solution to both can always be found by solving a semidefinite program (SDP). Since SDPs are convex, this

enables to guarantee that we can find the true maximum likelihood estimate (MLE) without any initial estimate of the trajectory.

Fourth, we propose a framework for modeling the uncertainty of jointly correlated poses using the Lie algebra of the Special Euclidean group. We then derive first order uncertainty propagation formulas for the pose composition, pose inverse, and relative pose operations when using the proposed framework. We evaluate using both simulated data and data extracted from an existing SLAM dataset and show that our method leads to more consistent uncertainty estimates than commonly used methods. Finally, we release a C++ library implementation of the proposed method.

In summary, this thesis presents four methods for multi-agent map merging, trajectory alignment, globally optimal SLAM, and pose uncertainty characterization that seek to address some of the common failure cases of existing localization and mapping methods. Furthermore, we demonstrate the performance of all of our proposed methods when compared with other methods in the field.

CHAPTER 1

Introduction

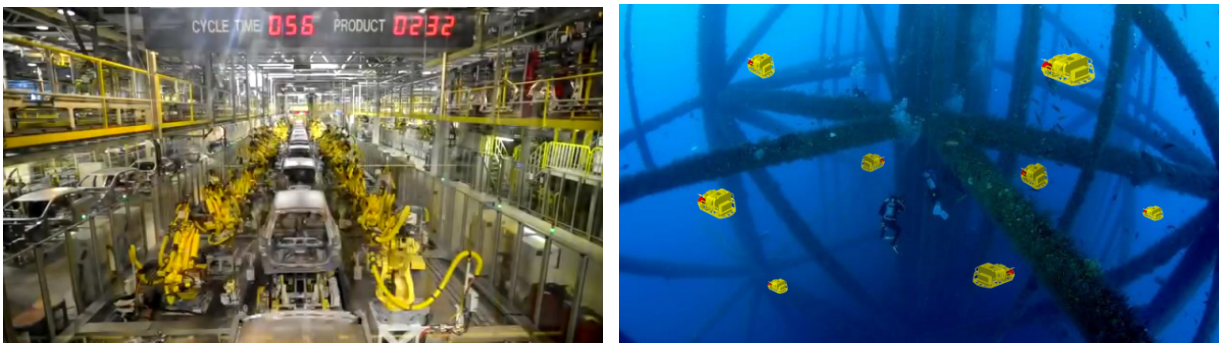
1.1 The Need for Reliable Multi-Agent Autonomy

Teams of cooperative autonomous mobile robotic systems have the potential to make a dramatic difference in our society. Over the past several decades, teams of robotic systems, within structured domains such as the manufacturing and distribution industries, over the past several decades, teams of robotic systems have made dramatic increases in safety, accuracy, and efficiency and are widely adopted across the board.

While robotic teams have led to large changes within these domains, they have an even greater potential to impact our day-to-day lives as they leave structured environments and venture out into the wider world. Potential areas of impact for autonomous systems include the transportation, distribution, and oil/gas industries in addition to disaster relief, security, first response, and scientific exploration/data collection.

If we focus on civil infrastructure inspection alone, teams of cooperative robotic vehicles have an immense potential to increasing public safety. In 2017, the American Society of Civil Engineers estimated that there are 614,387 bridges and about 90,580 dams in the United States. The average age of those bridges and dams are 43 and 56 years respectively. About 9% of bridges are estimated as being, structurally deficient and 17% of dams are classified as having high-hazard potential (ASCE, 2017). Our marine infrastructure is in desperate need of inspection and repair; however, the most common method of inspection is manual visual observation by human divers.

Manual diver inspection is dangerous, costly, and time consuming with a typical inspection task sometimes taking several days to complete. However, if a team of autonomous underwater vehicles (AUVs) were able to collaboratively perform this inspection task, inspection time and cost could be dramatically reduced. In addition, when performing manual inspections, it is often difficult to ensure that the entire structure has been inspected or to provided quantitative results. The data collected by a robotic system could be used to create



(a) Automobile Manufacturing (b) Underwater Infrastructure Inspection

Figure 1.1: Multi-Agent Teams in Structured and Unstructured Environments. Teams of robotic systems have the potential to enable dramatic increases in efficiency, safety, and accuracy. However, robotic systems in structured environments (such as the manufacturing line shown in (a)) have been much more widely accepted than within unstructured environments (such as the potential oil-rig inspection shown in (b)).

a 3D model of the inspected structure and to ensure full coverage.

However, teams of robotic systems in unstructured environments, such as the underwater domain, have not been nearly as widely developed or accepted as those within the manufacturing domain. The primary reason for the delay is that robotic systems in unstructured environments have not yet reached the level of reliability necessary for wide spread adoption.

The highly structured manufacturing/factory environment lends itself to solutions that can easily increase the reliability of these systems because the environment itself can be modified or engineered in ways that simplify the problem. For example, in a manufacturing line, the robotic systems can be fixed in place, eliminating the need to estimate the position of the robotic vehicle in order for it to interact with the environment. Similarly, on a manufacturing line, the objects the robot needs to interact with can also be moved to a predetermined location, eliminating the need for a robotic system to use perceptual sensors to observe its environment (See Fig. 1.1). However, in an unstructured underwater inspection task, each vehicle needs to accurately estimate where it is in space as well as where it is in relation to both the object it is inspecting and the other vehicles it needs to collaborate with.

In addition, the physical proximity of a robotic system to the operator in a manufacturing plant or distribution center lowers the bar of reliability because the system is relatively easily accessible and a failure of the system does not necessarily mean a loss of the system itself. In contrast, an underwater robotic vehicle is relatively inaccessible once a dive has commenced and a failure could result in the loss of the entire vehicle.

I am primarily interested in solving the problems that inhibit the reliability of multi-agent

teams and autonomous systems in real world settings. In the next section, I break down several elements that are crucial to reliable autonomy.

1.2 Elements of Reliable Autonomy

The reliability of a robotic system can be defined by describing four main areas: algorithmic robustness, constraint management, system cognizance, and system interoperability. The next four subsections describe each of these elements in turn.

1.2.1 Algorithmic Robustness and Consistency

An autonomous system must perform several important sub-tasks to successfully achieve its goals, such as: navigation, perception, goal planning, trajectory planning, control, communication, etc. Each of these sub-tasks is usually performed by an algorithm or set of algorithms that takes input information and generates some form of output that is either used by a task farther down the line or results in an action on the environment. An incorrect output can result in a failure of the system.

Algorithmic robustness deals with developing the underlying algorithms such that the set of situations where the output of the algorithm is incorrect is minimized. Algorithmic consistency is focused on developing the underlying algorithms/models in such a way that the output it provides is consistent enough with reality as to not cause a failure of the system.

1.2.2 Constraint Management

The domain the robotic system is operating in as well as the design of the robotic system, induce constraints that must be met if the system is to operate reliably. For example, in the underwater domain, communication is heavily limited because the properties of the water inhibit the propagation of electro-magnetic signals. In order for a team of robotic vehicles to cooperate reliably, the amount of information that is required to pass between them must be lower than the throughput available. Similarly, both an AUV and a micro aerial vehicle (MAV) have a limited amount of battery power that has to be conserved throughout the mission. In addition, to operate reliably, it must be ensured that certain algorithmic processes, such as navigation, planning, and control, run in real-time. Every autonomous system has its own set of constraints that must be managed effectively if the system is to operate reliably.

1.2.3 System Cognizance

System cognizance refers to the level of awareness the system has of itself and the task that it is instructed to carry out, as well as its ability to re-plan or revise its actions part way through a mission. This can be primarily broken into two different sub-elements: system-level awareness and task-level awareness.

At the system-level, a robotic system would be ideally aware of its own system components and architecture to the point where if a system component (either software or hardware) failed, it would be able to recognize the failure and modify its actions so it can continue operating.

At the task-level, ideally a robotic system would be aware of its own capabilities (as well as those of its team members, for a multi-agent system) and the task it has been assigned, such that it can replant on the fly to ensure that the goals of the task are carried out.

1.2.4 System Interoperability

Lastly, system interoperability refers to how effectively the system is able to interact and cooperate with a human operator as well as other agents in a team.

If reliability is interpreted to mean how much trust a human operator is willing to place in the system, then communication between the system and operator is essential. An operator will require consistent feedback in order to be assured that the system is likely to perform the task it has been assigned. Likewise, the instructions from the operator may be vague or high-level and the system will need to consistently interpret them accurately if it is to reliably perform the assigned task.

In addition, if multiple robotic agents are involved, the ability of each individual system to communicate the most essential information between them and to integrate the received information into their own plan of action is integrally tied to the reliability of the system.

1.2.5 The Focus of this Dissertation

Although all four of the listed elements of reliable autonomy are important and in future work I am interested in addressing each of them, the topic of this thesis is primarily focused on the element of algorithmic robustness and consistency, and more specifically, the algorithmic robustness and consistency of the localization and mapping task.

Section 1.3 describes a use case that was used as the primary motivation for this work and Section 1.4 provides a review of historical and modern approaches to localization and mapping in a priori unknown environments for both the single and multi-agent cases.

1.3 The Multi-Agent Ship Hull Inspection Use Case

As described briefly in Section 1.1, reliable teams of collaborative autonomous agents have vast potential to positively impact our society in many domains and areas of application. In particular, in the underwater domain, reliable multi-agent autonomous systems have the potential to dramatically increase safety, decrease cost, and advance scientific knowledge.

Autonomous ship hull inspection is of particular interest to the naval and shipping industries. The need to perform regular inspection of the underwater portion of a ship for damage and security reasons can be quite costly and dangerous for human divers. Multiple autonomous agents working in collaboration with one another to automatically search for damage and react to such occurrences can result in increased efficiency and the removal of human divers from danger.

For the past several years, the Office of Naval Research (ONR) has funded a project focused on developing a robotic system for autonomous ship hull inspection as well as the navigation and mapping algorithms needed to enable its use (Hover et al., 2012). Initially, this project began at Massachusetts Institute of Technology (MIT) and led to the development of the Hovering Autonomous Underwater Vehicle (HAUV) designed by BlueFin Robotics. The HAUV (shown in Fig. 1.2) was specifically designed for ship hull inspection and carries its primary perceptual sensors on a tray that can be rotated as the vehicle moves up and down under the ship such that the sensor payload is always pointing at the surface of the hull being inspected.

Navigational sensors include an on board inertial measurement unit (IMU), precision depth sensor, and Doppler velocity log (DVL) used for dead reckoning. The primary navigational sensor is the DVL which uses four transducers to transmit acoustic pings towards the surface of ship hull. It then uses the measured Doppler shift in the echoed signals to estimate the velocity of the vehicle with respect to the observed surface. This provides an estimate of odometry with respect to the surface of the hull. Perceptual sensors include a periscope camera for use when at the surface, an underwater stereo camera, and a DIDSON imaging sonar.

After the initial platform development, collaboration between MIT, University of Michigan (UofM), and Carnegie Mellon University (CMU) has led to the development of a variety of additional capabilities, including feature based navigation (Hover et al., 2012; Kim and Eustice, 2013), active planning (Chaves et al., 2016; Hollinger et al., 2013), integration of multi-modality (Li et al., 2016), and multi-session capabilities (Ozog et al., 2016).

The system uses Pose Graph simultaneous localization and mapping (SLAM) (Kaess, Ranganathan, and Dellaert, 2008; Kim and Eustice, 2013; Ozog et al., 2016) techniques to

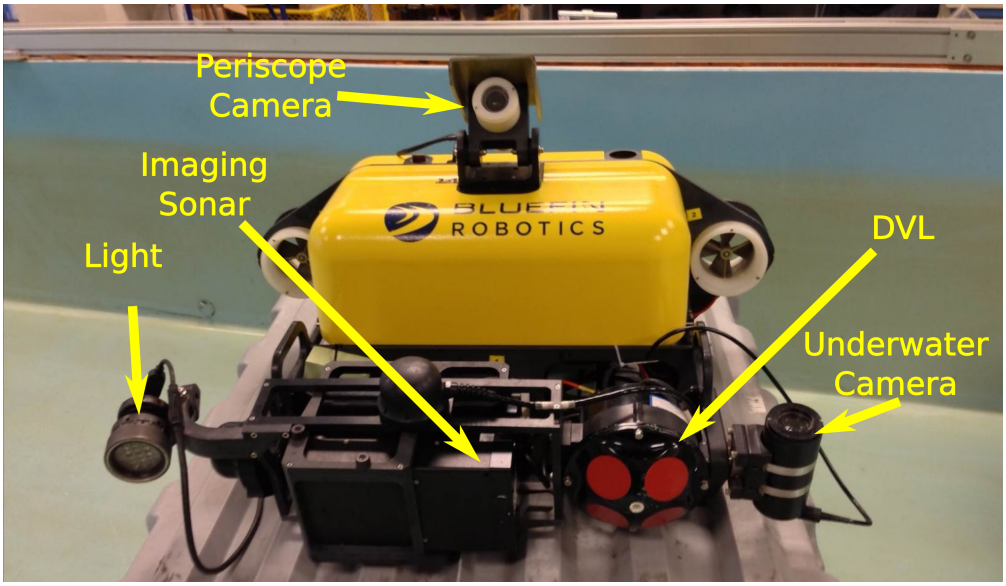
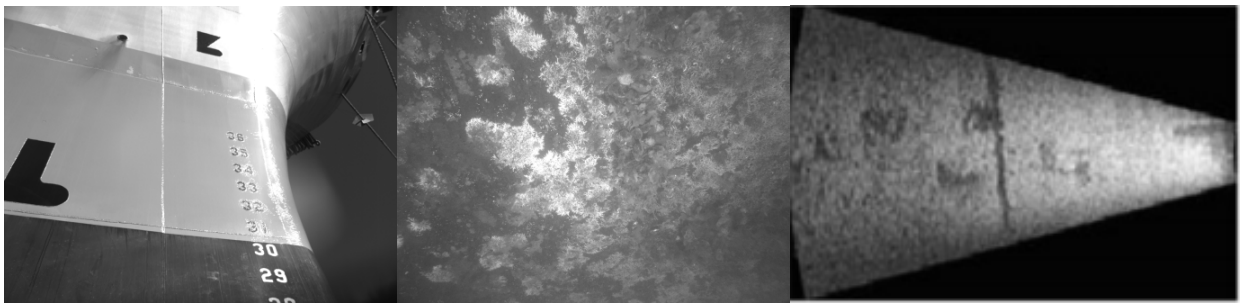


Figure 1.2: BlueFin Hovering Autonomous Underwater Vehicle (HAUV). The HAUV has the ability to hover in place and has a variety of perceptual and navigational sensors including a periscope camera, an underwater stereo camera, a Doppler velocity log, an imaging sonar, an IMU, and a depth sensor. When navigating with respect to the ship hull, the sensor tray rotates to keep the DVL, camera, and sonar pointing perpendicular to the hull at all times.

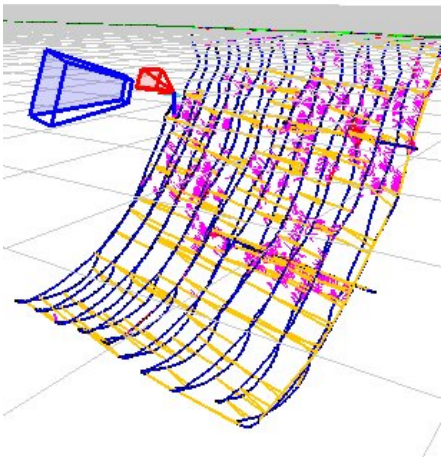


(a) Periscope Image (b) Underwater Image (c) Sonar Image

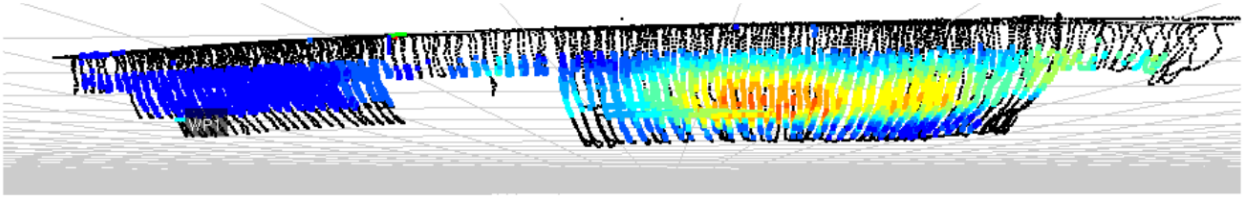
Figure 1.3: Example HAUV Image Data. Perceptual data collected by the periscope camera (a), underwater stereo camera (b) (only master image shown), and imaging sonar (c).

fuse data collected from both navigational and perceptual sensors in order to estimate both the trajectory of the robot and a map of the environment. Sample perceptual data and example maps generated by the system are shown in Fig. 1.3 and Fig. 1.4, respectively.

The majority of recent work at UofM has focused on enabling reliable, multi-agent inspection, through funding under ONR award number N00014-16-1-2102. Towards this goal, we acquired access to a second vehicle and began performing experiments using two agents instead of one. This project served as the primary motivation for the results presented in



(a) USS Mercy Trajectory



(b) USCGC Spencer Curvature Map

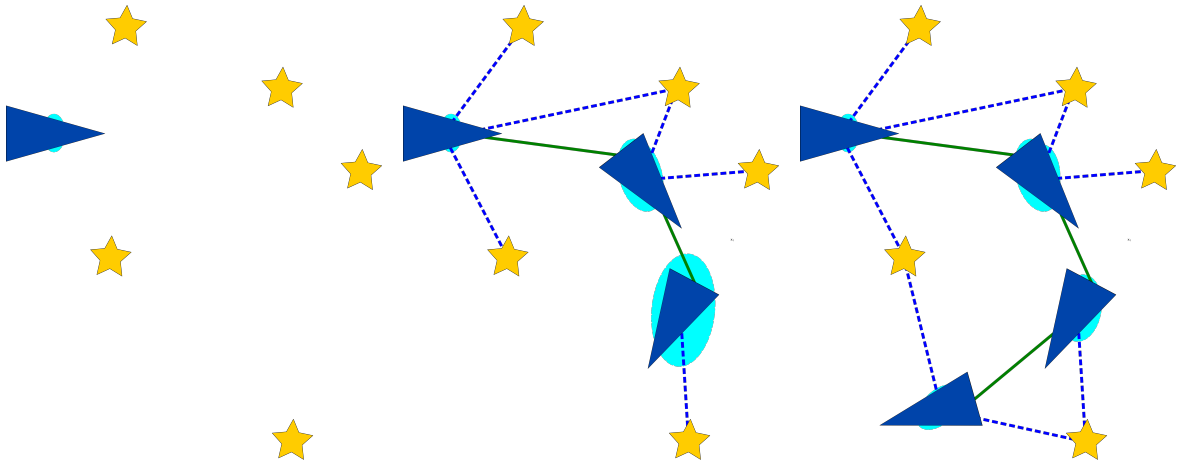
Figure 1.4: Example HAUV Ship Hull Relative Maps. These maps were generated by fusing navigational and perceptual data collected by the HAUV platform. (a) shows a portion of a map created during an inspection of the USS Mercy in San Diego, CA in 2017. (b) shows a map of the USCGC Spencer colored by the estimated curvature of the ship created from data collected in Boston, MA, in 2017.

this thesis.

1.4 Localization in A Priori Unknown Environments

Localization is a fundamental task for any mobile robotic system operating in an unstructured environment. Formally, localization denotes the estimation of a vehicle's orientation and location, or pose, in space, as well as its pose in relation to other agents if part of a team (Thrun, Burgard, and Fox, 2005). This is essential because an accurate understanding of the robot's pose in relation to the environment (and potentially other vehicles) is a necessary prerequisite for safe navigation.

This is made difficult by the fact that all measurements obtained by the robotic system are imperfect and affected by noise, and as a result, the estimated pose of the robotic vehicle is also uncertain. Thus, to ensure safe navigation, it is essential to not only estimate the



(a) Initialization (b) Before Loop-closure (c) After Loop-closure

Figure 1.5: Overview of SLAM. SLAM refers to the problem of simultaneously estimating the robot trajectory and a map of the surrounding environment. (a) shows a robotic vehicle (denoted by the blue triangle) and landmarks (represented by gold stars) surrounding it in the environment. (b) as the robot moves through the environment, it both measures its odometry (shown by the green line) and makes observations of landmarks within the environment (shown by the dashed blue lines). The estimated covariance of each robot pose is shown by the aqua colored ellipses. (c) by re-observing the landmarks previously observed, error caused by noise in the odometry measurements can be corrected and the associated covariance estimate of the trajectory restricted. This "re-observation" of something seen in the past is called a *loop-closure.*

pose of the robotic agent, but to accurately model how uncertain that estimate is. First order uncertainty propagation techniques are often used to model this uncertainty. One of the first papers to describe how this could be done was that of Smith, Self, and Cheeseman, who represented pose uncertainty by assuming a multivariate Gaussian distribution over a pose parameter vector such as transformation coordinates and Euler angles (Smith, Self, and Cheeseman, 1990; Smith and Cheeseman, 1986).

The localization problem can also be complicated by the absence of a prior map. Most often, localization is framed as determining your location with respect to a map of the environment that was estimated previously. However, in many cases, a robotic vehicle may be navigating in an environment that has not been seen before or for which a prior map is not available. Performing localization in these situations is often resolved by simultaneously building a map of the environment and maintaining an estimate of your location with respect to that map. This is referred to as the simultaneous localization and mapping (SLAM) problem.

1.4.1 Simultaneous Localization and Mapping

While a variety of general and application specific methods exist both for localizing a robotic vehicle when given a map or for building a map when given an accurate estimate of the robot's trajectory (Paull et al., 2014; Thrun, Burgard, and Fox, 2005; Wolcott and Eustice, 2017), it is often necessary to simultaneously localize and map an environment without assuming either is given a priori.

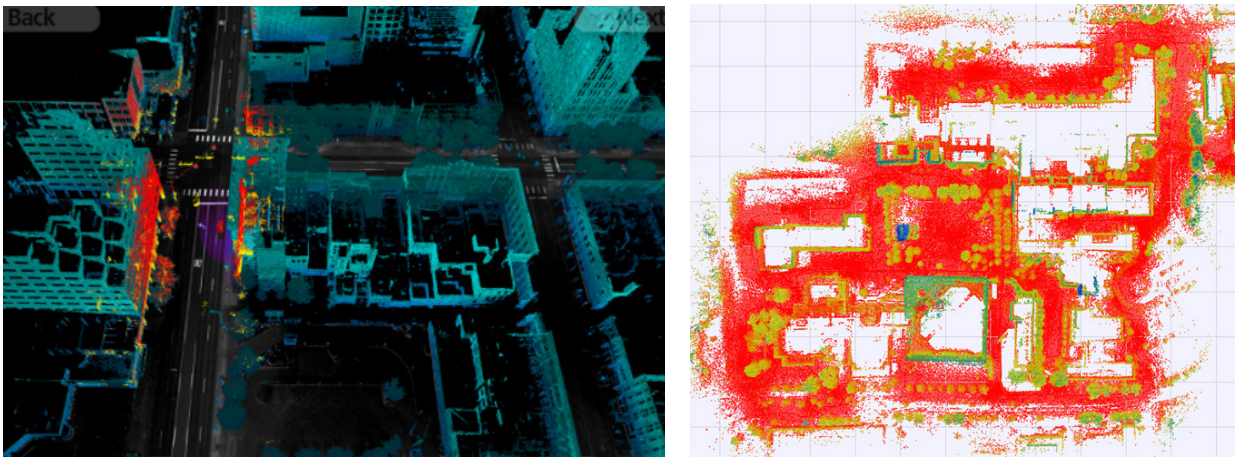
This is done by fusing the robot's measurements of its own motion (odometry) and its observations of the environment to find the set of robotic poses that most consistently match the measurements. While noise in the robot's pose grows over time when fusing odometry measurements alone, when a robotic agent re-observes something it has seen previously, it can use that information to update it's estimate of its position and decrease the uncertainty associated with its estimate. Fig. 1.5 shows an illustration of this process.

In many tasks, including inspection, disaster relief, mapping, and scientific data collection, the goal of the robotic vehicle is often to cover a specified area, collecting data about the environment as it traverses it. In these tasks, the final desired result is to generate a map by projecting the observed data into a common reference frame based on the trajectory estimated by SLAM. In other tasks, online navigation of the robotic vehicle may be the primary need of the system, in which cases, the trajectory or localization of the robotic vehicle is the primary output. Whether the primary goal is mapping or localization, SLAM provides a method for fusing the perceptual and odometric data collected by the robot and estimating the desired quantities.

In the past two decades, SLAM has been extensively studied and significant progress has been made. A variety of methods have been proposed for solving it (Cadena et al., 2016; Dellaert and Kaess, 2006; Kaess, Ranganathan, and Dellaert, 2008; Kümmerle et al., 2011), and it has been integrated into a significant number of real-world products and systems including self-driving cars (Levinson et al., 2011), micro-aerial vehicles (Weiss, Scaramuzza, and Siegwart, 2011), and autonomous underwater vehicles (Hover et al., 2012). Fig. 1.6 shows maps generated using SLAM. The majority of proposed methods can be classified as either filtering or optimization based approaches. In the next three subsections we review the prior literature relating to these two approaches and then review their extension to multi-agent SLAM.

1.4.2 Filtering Based Approaches

Most of the early methods for performing SLAM were based on extensions of the Kalman filter (Dissanayake et al., 2001; Kalman, 1960; Thrun, 2002) and build on the work of Smith,



(a) SLAM Derived Map of Downtown Ann Arbor (b) SLAM Derived Map of UM North Campus

Figure 1.6: Example SLAM Maps. SLAM is a fundamental functionality required for autonomous navigation in a wide variety of scenarios. (a) shows a map of downtown Ann Arbor, MI derived by using Pose Graph SLAM to fuse the data collected by an autonomous vehicle (Wolcott and Eustice, 2017). (b) shows a similar map of the University of Michigan North Campus created by fusing data collected by a segway robot (Carlevaris-Bianco, Ushani, and Eustice, 2015).

Self, and Cheeseman (Smith, Self, and Cheeseman, 1990; Smith and Cheeseman, 1986)Many of these methods employ an extended Kalman filter (EKF) and are termed EKF SLAM methods (Leonard and Durrant-Whyte, 1991; Moutarlier and Chatila, 1989a,b). The majority of these methods track the current state of the robotic vehicle as well as the xy position of landmarks in the environment. At each time step, the state of the robot and each of the landmarks being tracked is updated and the state is augmented for any new landmarks observed. However, EKF SLAM methods suffer from the curse of dimensionality as the number of landmarks increases. In addition, incorrect or unknown data association often causes the system to diverge (Durrant-Whyte and Bailey, 2006; Thrun, Burgard, and Fox, 2005).

Several methods were proposed to handle this data association problem (Bailey and Durrant-Whyte, 2006; Durrant-Whyte and Bailey, 2006; Montemerlo et al., 2002, 2003; Neira and Tardós, 2001). Joint compatibility branch and bound (JCBB) (Neira and Tardós, 2001) uses a branch and bound framework to determine which measurements are jointly compatible with one another. Arguably the most well known is FastSLAM (Montemerlo et al., 2002, 2003), which uses a Rao-Blackwellized particle filter to represent possible trajectories for the robot. Data association and landmark localization is then performed based on each potential trajectory. This creates a map for each trajectory and makes the system more robust to false data association matches. One of the draw backs of this approach, however,

is that because of the nature of the particle filter and the high dimensionality of the state space, as particles are pruned the algorithm quickly forgets its past trajectory. This makes it difficult to perform loop-closure measurements, which are important as they significantly constrain error in the graph (Thrun, Burgard, and Fox, 2005).

A variety of researchers have tried to address the scalability problems of EKF SLAM by leveraging sparsity inherent in the problem. When using the information form parameterization, scalability is significantly improved (Eustice, Singh, and Leonard, 2006; Frese, 2006; McLauchlan and Murray, 1995; Thrun et al., 2004). At first it was observed (Frese, 2005; Thrun et al., 2004) that when using the inverse parameterization of covariance, most of the elements in the matrix were close to zero. A variety of methods tried to approximate those elements as zero, but the approach led to inconsistency in the uncertainty estimate (Eustice, Walter, and Leonard, 2005; Walter, Eustice, and Leonard, 2005). Finally, it was simultaneously observed by several researchers that by not marginalizing out old states and estimating a time discretized version of the whole trajectory, exact sparsity can be obtained, resulting in both consistency and efficiency (Dellaert and Kaess, 2006; Eustice, Singh, and Leonard, 2006).

1.4.3 Factor Graph Based Approaches

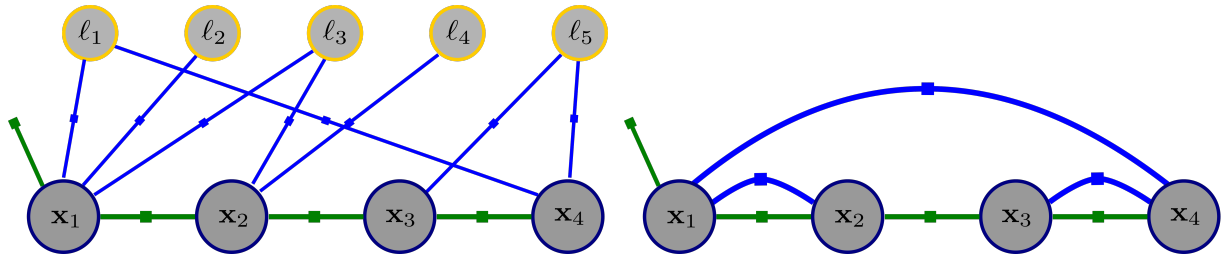
Modern approaches to SLAM are formulated as a maximum *a posteriori* (MAP) estimate of the discretized trajectory (and possibly landmarks), given measurements observed by the robot. The measurements are used to derive probabilistic relationships between the values to be estimated, and then, the values are optimized to minimize deviations from these relationships.

This is usually done by discretizing the trajectory of the robot in time and representing the pose (position and orientation) of the robotic vehicle at each time step as an element of the Lie group SE(2) or SE(3). Then, the MAP estimate of the time discretized trajectory given all of the measurements of the robot is:

$$\hat{\mathbf{X}} = \underset{\mathbf{X} \in \mathrm{SE}(d)^n}{\operatorname{argmax}} \, p(\mathbf{X}|\mathbf{Z}) \tag{1.1}$$

where $\hat{\mathbf{X}}$ is an estimate of time descretized trajectory $\mathbf{X} \in \mathrm{SE}(d)^n = \{\mathbf{x}_i\}_{i=1}^n$ and $\mathbf{Z} = \{\mathbf{z}_j\}_{j=1}^m$ is the set of measurements observed (Cadena et al., 2016).

If we make the assumption that our measurements $\mathbf{z}_j$ are independent and follow the measurement model form $\mathbf{z}_j = h_j(\mathbf{X}) + \omega_j$, where $\omega_j \sim \mathcal{N}(\mathbf{0}, \Sigma_j)$, it is easy to show that



(a) Full SLAM Factor Graph (b) Pose Graph SLAM

Figure 1.7: Factor Graph SLAM. Factor graphs are probabilistic graphical models that are often used to visualize specific SLAM problem instances. Here we show the factor graph representation for the example shown in Fig. 1.5(c). Variable nodes in a factor graph represent values that we want to estimate, and factor nodes represent constraints or observations that relate the variables being estimated. Robot pose nodes are represented by the round grey circles with a blue border. Landmark variables such as landmark position are represented by the round grey circles with a yellow border. Odometry factors are denoted by the green square nodes and perceptual based factors are denoted by the blue square nodes. (a) shows a factor graph representing the full SLAM problem where landmark positions (or the map) is explicitly estimated. (b) shows a pose graph where only robot poses are estimated but perceptual information is used to relate those poses. Pose Graph SLAM increases the sparsity of the problem and is thus often used when real-time computation is required.

(1.1) is equivalent to the following weighted non-linear least squares problem:

$$\hat{\mathbf{X}} = \underset{\mathbf{X} \in \mathrm{SE}(d)^n}{\operatorname{argmax}} \sum_{j=1}^{m} ||\mathbf{z}_j - h_j(\mathbf{X})||^2_{\Sigma_j}, \tag{1.2}$$

with $||\cdot||_\Sigma$ signifying the Mahalanobis distance.

We often visualize this cost function by drawing the associated factor graph probabilistic graphical model. A factor graph is a bipartite graph with variable nodes and factor nodes. Variable nodes represent the variables, $\mathbf{x}_i$, that are to be estimated. Each factor node corresponds to a single term of the cost function and is connected to each of the variable nodes it relates. Fig. 1.7 shows an example factor graph (Dellaert and Kaess, 2006; Eustice, Singh, and Leonard, 2006; Kaess, Ranganathan, and Dellaert, 2008).

One of the earliest works proposing this approach for SLAM was proposed by Lu and Milios (1997). In the last decade, a variety of algorithms for efficiently estimating the solution to (1.2) have been proposed and a variety of open source libraries have been released (Dellaert and Kaess, 2006; Kaess, Ranganathan, and Dellaert, 2008; Kaess et al., 2012; Kümmerle et al., 2011; Rosen et al., 2016). Each of these algorithms take a factor graph as input and try to estimate the solution to (1.2). Many of them solve the graph in a batch process

using algorithms such as Levenburg-Marquardt (Marquardt, 1963), while others have been designed to enable incremental updates (Kaess, Ranganathan, and Dellaert, 2008; Kaess et al., 2012).

It is important to note that the optimization problem posed in (1.2) is non-convex (Boyd and Vandenberghe, 2004). This will be of importance in future sections.

1.4.4 Multi-Agent SLAM

In multi-agent SLAM, each vehicle moves through the environment collecting sensor data about its own movement and the world around it. Then, either the raw data, a subset of that data, or a processed version of that data is passed either between the robots or between the robots and a central processor and the data from the separate robots is fused to generate a global map that can be used for navigation or other purposes. The way this process occurs is dependent on the needs of the application and the vehicles in use. Here we enumerate a subset of the problems that need to be solved in order to facilitate this process.

Methods proposed for multi-agent SLAM problem mostly parallel those for single-agent SLAM (Saeedi et al., 2016). A variety of potential solutions have been proposed, including EKF-SLAM based (Williams, Dissanayake, and Durrant-Whyte, 2002; Zhou and Roumeliotis, 2006), particle filter based (Carlone et al., 2010; Howard, 2006), information filter based (Thrun and Liu, 2005; Webster et al., 2013), and factor graph based (Andersson and Nygards, 2008; Cunningham, Paluri, and Dellaert, 2010; Cunningham et al., 2012; Cunningham, Indelman, and Dellaert, 2013; Kim et al., 2010) solutions.

One important problem in collaborative SLAM is the estimation of the initial transformation between graphs. Initial work assumed this transformation was either given before hand or measured directly at the beginning of the experiment (Thrun, 2001). This assumption was then relaxed. Carlone et al. (2010); Howard (2006); Howard, Matark, and Sukhatme (2002) and Zhou and Roumeliotis (2006) present methods that do not require an initial transformation. Instead, they assume that robots "encounter" one another at least once during the experiment and are able to directly measure their relative pose at the time of the encounter. They use that initial direct measurement to estimate the relative transformation and then merge both past and future data into a local estimate of the global map. However, using a single measurement to estimate relative pose can result in bad results, either because of measurement noise or false observations. In addition, requiring vehicles to encounter one another as a method for deriving their relative offset is not always possible.

A variety of other methods have been proposed that use multiple measurements to minimize error in the estimated relative pose. The following methods use factor graphs to enable

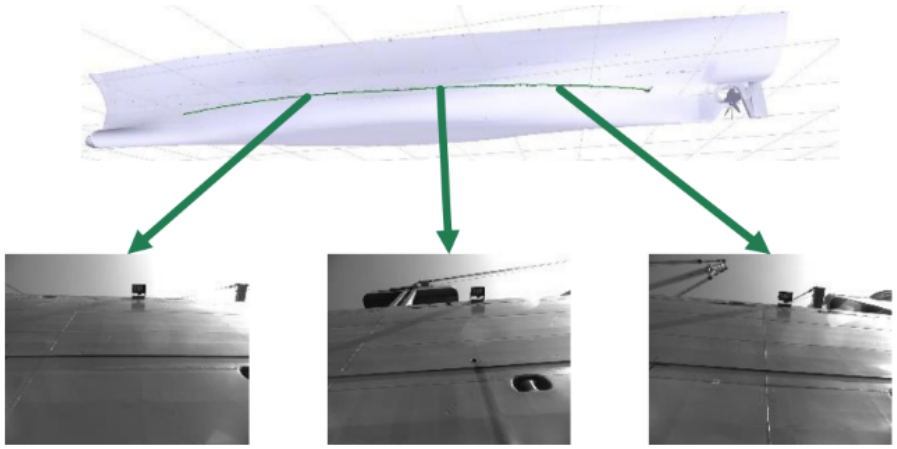
multiple measurements: Andersson and Nygards (2008); Cunningham, Paluri, and Dellaert (2010); Cunningham et al. (2012); Cunningham, Indelman, and Dellaert (2013); and Kim et al. (2010). Andersson and Nygards (2008) introduce a method that attempts to merge maps by incorporating "rendezvous" factors into the graph that represent the measured relative transformation between robots when they encounter one another. Kim et al. (2010) introduces anchor nodes that estimate the transformation between local coordinate frames and the global frame in a way that increases efficiency by minimizing change in linearization point. This formulation also relaxes the assumption that relative pose be derived by observing the other agent. The work proposed by Cunningham et al. in Cunningham, Paluri, and Dellaert (2010); Cunningham et al. (2012) and Cunningham, Indelman, and Dellaert (2013) uses factor graphs to estimate local maps of landmarks and then combines summarized versions of the local maps to create a global map. The use of landmarks as a map representation removes the dependence on direct observation, but it also removes knowledge of individual robot trajectories from the map. Williams, Dissanayake, and Durrant-Whyte (2002) also proposes a method based on EKF-SLAM that merges local maps of landmarks and thus removes the need for vehicles to encounter one another.

1.5 Algorithmically Robust and Consistent Navigation

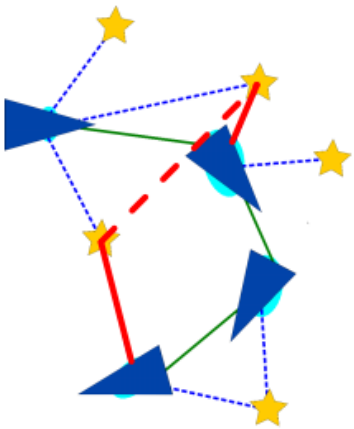
As pointed out in the previous section, a significant amount of work has been done on the localization and mapping problem in both the single and multi-agent cases. However, the majority of existing algorithms can be somewhat unreliable and tend to fail under certain conditions. Some of the most prevalent failure cases include the presence of outlier measurements, bad initialization, and pose parameterizations that lead to inconsistent uncertainty estimates. In the next three subsections we describe these failure cases in more detail.

1.5.1 Outlier Measurements

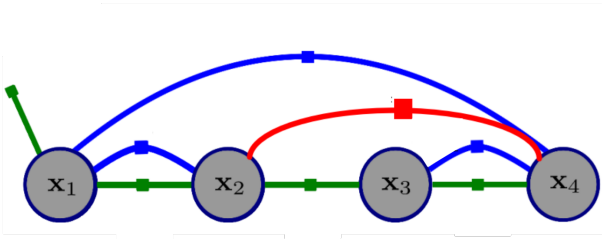
Modern state-of-the-art SLAM methods are generally divided into a front-end and a back-end. The purpose of the front end is to process raw measurement data as it is observed by sensors on board the robot and transform it into a factor graph instance like the one shown in Fig. 1.7. The back-end is then tasked with taking the factor graph as input and optimizing over the poses in the robot's trajectory to find the solution that best matches the observed measurements. This is usually done by solving some form of the optimization problem shown in (1.1). As described in Section 1.4.3, a variety of methods have been proposed to solve this problem and in general, they work very well as long as the factor graph or problem instance



(a) Perceptual Aliasing in Ship-Hull Superstructure



(b) Outlier Landmark Match

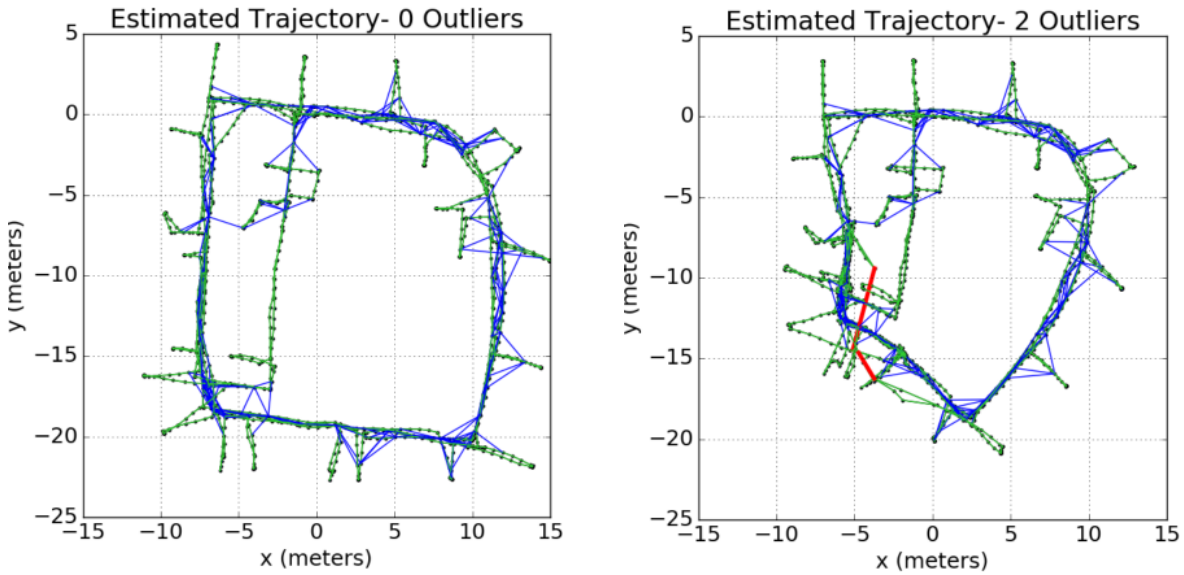


(c) Outlier Loop-closure in Pose Graph

Figure 1.8: Perceptual Aliasing and Outlier Loop-closures. As described in Fig. 1.5, loop-closures are usually made by matching data seen at different time-steps and extracting an implied relationship between the robot pose at those two time steps. However, it is relatively common for multiple locations in a complex environment to look similar to one another. This is referred to as *perceptual aliasing*. (a) shows an example shown in data collected using the HAUV on the USNS Curtiss. When this occurs, it is possible to incorrectly match data seen at one time-step with data at another time-step that in reality do not refer to the same location as shown in (b). This in turn results in an incorrect factor in the pose graph (as shown (c)) that if not handled can drastically distort the final estimated solution.

are well formed. However, because multiple locations in the environment look similar to one another, it is relatively common for false or outlier loop-closures to be proposed by the front end. Fig. 1.8 shows an example of how this could happen.

The addition of even one or two outlier loop-closures to a pose graph can have a devastating affect on the resulting trajectory as shown in Fig. 1.9. The significant susceptibility of



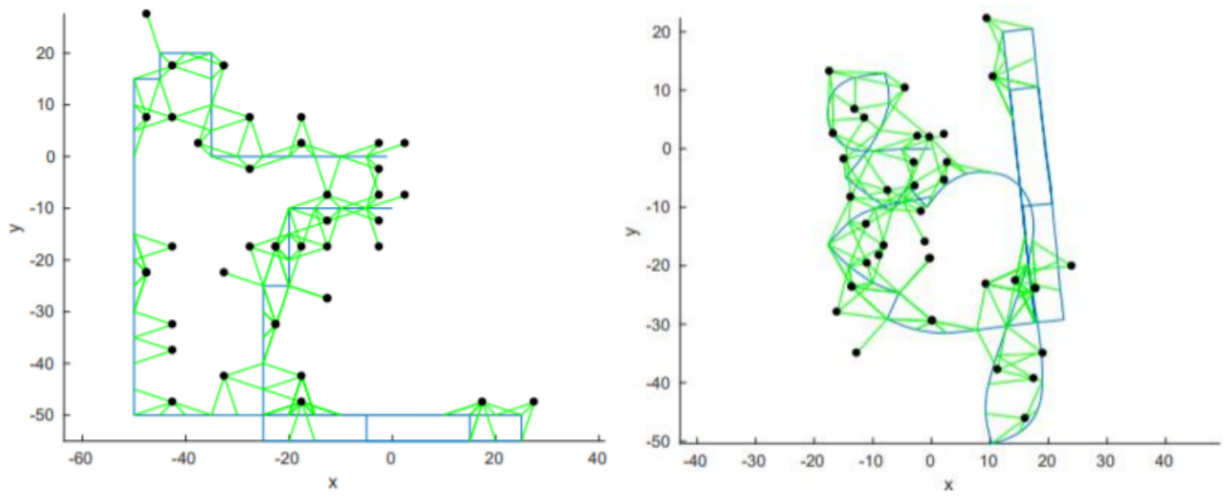
(a) Trajectory with 0 Outliers

(b) Trajectory with 2 Outliers

Figure 1.9: Intel Dataset with/without Outlier Loop-closures. We used iSAM (Kaess, Ranganathan, and Dellaert, 2008) to estimate the solution to the Intel Dataset (Howard and Roy, 2003) with and without outlier measurements. (a) shows the optimal solution with no outliers. (b) shows the solution with two outlier loop-closures. The addition of even a small number of outliers can drastically affect the final solution.

the current state-of-the-art SLAM algorithms to outlier measurements has led to a significant degree of focus within the SLAM community. A variety of methods have been proposed over the years (Agarwal et al., 2013; Latif, Cadena, and José, 2012; Olson and Agarwal, 2013; Sünderhauf and Protzel, 2012), with the primary focus being on single robot graphs. The majority of these methods work by using the residual error of individual factors to determine if a measurement should be trusted. As a result, they require the optimization system to be initialized within a basin of attraction to correctly determine the measurement set.

However, things get more complicated when trying to align and merge graphs collected by multiple robots because a good initial estimate of alignment is not necessarily available. Recently, work by Dong et al. (2015) and Indelman et al. (2016) attempted to handle false measurements in multi-vehicle map merging by using expectation-maximization (EM) to limit the set of trusted measurements to those that are probabilistically likely given an initial transformation. However, EM is known to be only locally optimal and dependent on initialization.



(a) GroundTruth Trajectory and Map

(b) Solution Caught in Local Minimum

Figure 1.10: CityTrees10000 Segment Caught in Local Minimum. An example segement of the CityTrees10000 dataset (Kaess, Ranganathan, and Dellaert, 2008) as solved by Levenburg Marquardt (Marquardt, 1963) caught in a local minimum. (a) shows the groundtruth solution. (b) shows the local minimum solution after being given a bad initialization.

1.5.2 Dependence on Good Initialization

A second common failure case of current SLAM algorithms occurs when the optimization algorithm tasked with finding a solution to (1.2) is given a bad initialization. Since in general, the measurement model h_j is non-linear, the majority of existing algorithms (Dellaert and Kaess, 2006; Kaess, Ranganathan, and Dellaert, 2008; Kaess et al., 2012; Kümmerle et al., 2011) estimate a solution by iteratively alternating between linearizing the problem around a current estimate and solving the corresponding linear, weighted, least-squares problem until a local-minimum is reached.

This can cause problems because the underlying problem is generally non-convex and thus finding a local-minimum does not necessarily guarantee that a global solution corresponding to the solution of (1.1) has been found (See Fig. 1.10). This is really important from a reliability perspective because the solution returned by the SLAM algorithm is usually used for navigation, planning, and coordination with other agents. An incorrect result that is undetected can cause a collision and damage to the system or environment in addition to incorrect mapping or localization results. This is of particular significance in the underwater environment because the robotic system or team needs to be able to operate consistently with little to no supervision.

Recently, a variety of works have looked at incorporating convex optimization theory into algorithms for solving factor graphs. Doing so would enable guarantees of global optimality

and could potentially increase computational efficiency. Rosen, DuHadway, and Leonard (2015) use convex relaxation to find an initial condition for the SLAM problem that can then be used to find the global solution. Carlone et al. (2015) uses lagrangian duality to evaluate the quality of a solution to the SLAM problem and determine if the solution is globally optimal after the fact. Finally, the seminal paper by Rosen et al. (2016) proposes SE-Sync, which uses a semidefinite program (SDP) (defined in Appendix B.2.2) relaxation, and rank restricted Riemannian optimization to efficiently find global solutions in cases where there is non-adverserial noise. Recent work by Briales and Gonzalez-Jimenez (2017) builds on SE-Sync to increase efficiency in the optimization process.

However, all of these methods (Briales and Gonzalez-Jimenez, 2017; Carlone et al., 2015; Rosen et al., 2016; Rosen, DuHadway, and Leonard, 2015) make the restrictive assumption that all measurements are full degree of freedom relative transformations between pose nodes and none of them actually formulate the SLAM problem as a truly convex optimization without relaxation.

1.5.3 Inconsistent Uncertainty Estimates

A final failure case of current navigation and mapping solutions occurs when coordinates-based parameterizations of robot pose are used to characterize uncertainty. Early methods (such as those proposed by Smith, Self, and Cheeseman (1990) and Smith and Cheeseman (1986)), used a vector of Euler angles and translation parameters to represent pose and then assumed this vector of parameters could be represented as a multi-variate Gaussian random variable. However, because of the non-linearities involved in representing rotation via chart coordinates such as Euler angles, a Gaussian distribution cannot accurately model the uncertainty using this parameterization.

However, it has recently been shown that characterizing pose uncertainty in the Lie algebra of the Special Euclidean group leads to more accurate results (Barfoot and Furgale, 2014; Chirikjian, 2011; Kim and Kim, 2017). The work of Barfoot and Furgale (2014), in particular, describes how to characterize the uncertainty of a pose using a mean or nominal value in the Lie group space and a zero-mean, multi-variate Gaussian perturbation variable defined in the Lie algebra. The perturbation variable is then used to perturb the mean value through the use of the exponential function. This enables an accurate characterization of pose uncertainty while taking into account the non-linearity of rotation. Fig. 1.11 depicts a Monte Carlo simulation that shows this improvement in uncertainty characterization accuracy.

However, while Barfoot and Furgale (2014) describe how to model uncertain poses in this manner, they make the assumption that each pose is independent of one another and focus

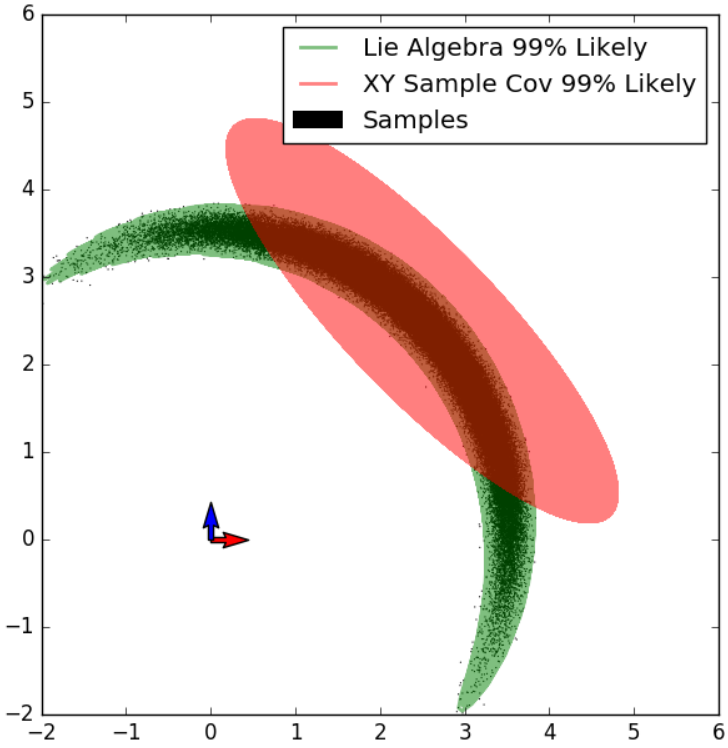


Figure 1.11: Lie Algebra vs Coordinate Based Uncertainty Characterization. A Monte Carlo comparison between the Lie Algebra uncertainty characterization method (proposed in Barfoot and Furgale (2014) and extended in this dissertation) and a coordinates based method (such as the one proposed in Smith, Self, and Cheeseman (1990)).

primarily on uncertainty propagation through the pose composition operation. Meanwhile Smith, Self, and Cheeseman described how to jointly represent correlated poses and described how to perform the pose inverse and relative pose operations in addition to pose composition.

1.6 Dissertation Outline

This thesis aims to increase the robustnessf of single and multi-agent localization and mapping by developing algorithms and methods that address the failure cases described in the prior section. In doing so, we consider four main problems:

- Multi-agent map merging in the presence of outliers – Given two pose graphs collected by robotic agents navigating in partially overlapping environments, we want to align and merge them into a single consistent map, while taking into account the fact that a

large percentage (greater than 80%) of the measurements relating them to one another may be outliers.

- Global trajectory alignment with no initial estimate and low-dimensional data – Given a reference trajectory and a query trajectory (contained within the reference trajectory) and low-dimensional data collected by the agents at each time step along the trajectory, find the rigid body transformation that best aligns them without any initial estimate.

- Globally optimal, initialization independent, planar SLAM – Given a planar Pose Graph SLAM problem, find the globally optimal (and thus true maximum likelihood estimate (MLE)) without any initial guess.

- Accurate characterization of pose uncertainty – Develop a method for accurately manipulating sets of jointly correlated, uncertain pose estimates, while accurately accounting for the structure of the Special Euclidean group.

Towards these problems, we have made the following contributions:

1. A novel framework for selecting a *pairwise internally consistent* set of inter-map loop-closures for robust multi-agent map merging that does not require an initial estimate of alignment and a method for transforming the problem into an instance of the maximum clique problem for increased efficiency.

2. An algorithm for aligning a query trajectory to a reference trajectory based solely on low-dimensional data observed at each pose visited by the vehicles. By using a linear approximation of the $\ell 2$ norm and linear relaxation techniques, we estimate an approximately rigid body transformation using a linear program, requiring no initial estimate or data association.

3. A novel formulation for planar Pose Graph SLAM as a polynomial optimization problem and a proof that guarantees the globally optimal solution can always be found by solving a sparse SDP.

4. A method for characterizing the uncertainty of jointly correlated poses in the Lie algebra of the Special Euclidean group and a derivation of first order uncertainty propagation formulas for the pose composition, pose inverse, and relative pose operations under the proposed framework. Characterizing the uncertainty in the Lie algebra leads to increased uncertainty estimate consistency than coordinate based methods.

The work in this thesis has appeared (or will appear) in the following publications:

Joshua G. Mangelson, Derrick Dominic, Ryan M. Eustice, and Ram Vasudevan. *Pairwise Consistent Measurement Set Maximization for Robust Multi-robot Map Merging.* In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, Australia, May 2018. **IEEE ICRA Best Paper on Multi-Agent Systems.**

Joshua G. Mangelson, Ram Vasudevan, and Ryan M. Eustice. *Communication Constrained Trajectory Alignment For Multi-Agent Inspection via Linear Programming.* In Proceedings of the IEEE/MTS OCEANS Conference and Exhibition, Charleston, SC, USA, Oct 2018. **1st Place in IEEE OCEANS Student Poster Competition.**

Joshua G. Mangelson, Jinsun Liu, Ryan M. Eustice, and Ram Vasudevan. *Guaranteed Globally Optimal Planar Pose Graph and Landmark SLAM via Sparse-Bounded Sum-of-Squares Programming.* In Proceedings of the IEEE International Conference on Robotics and Automation, Montreal, Canada, May 2019.

Joshua G. Mangelson, Maani Ghaffari, Ram Vasudevan, and Ryan M. Eustice. *Characterizing the Uncertainty of Jointly Distributed Poses in the Lie Algebra.* Journal Publication. (In Submission/Under Review).

Other related publications that do not directly include the work presented in this thesis:

Ross Hartley, **Joshua G. Mangelson**, Lu Gan, Maani Ghaffari Jadidi, Jeffrey M. Walls, Ryan M. Eustice, and Jessy W. Grizzle. *Legged Robot State-Estimation Through Combined Forward Kinematic and Preintegrated Contact Factors.* In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, Australia, May 2018.

Joshua G. Mangelson, Ryan W. Wolcott, Paul Ozog, Ryan M. Eustice. *Robust Visual Fiducials For Skin-to-Skin Relative Ship Pose Estimation.* In Proceedings of the IEEE/MTS OCEANS Conference and Exhibition, Monterey, CA, USA, Oct 2016.

Open source C++ library implementations are released at the following links:

- Contribution 1 - https://bitbucket.org/jmangelson/pcm
- Contribution 2 - https://bitbucket.org/jmangelson/cte
- Contribution 4 - https://bitbucket.org/jmangelson/lie

1.7 Dissertation Roadmap

These contributions are discussed in the following chapters:

Chapter 2 We present a method for robust selection of inter-map loop closures in multi-robot simultaneous localization and mapping (SLAM). Existing robust SLAM methods assume a good initialization or an "odometry backbone" to classify inlier and outlier loop closures. In the multi-robot case, these assumptions do not always hold. We present an algorithm called Pairwise Consistency Maximization (PCM) that estimates the largest *pairwise internally consistent set* of measurements. Finding the largest pairwise internally consistent set can be transformed into an instance of the maximum clique problem from graph theory, and by leveraging the associated literature it can be solved in real-time. We evaluate how well PCM approximates the combinatorial gold standard using simulated data. We also evaluate the performance of PCM on synthetic and real-world data sets in comparison with DCS, SCGP, and RANSAC, and show that PCM significantly outperforms these methods.

Chapter 3 We present a system for estimating the alignment between robotic trajectories based solely on low-dimensional data. We propose a method that takes advantage of convex relaxation techniques to determine an alignment between robotic trajectories based on sparse observations of a low-dimensional underlying feature space. We use a linear approximation of the $\ell 2$-norm to approximately enforce that the estimated transformation is an element of SO(2). Because the relaxed optimization problem is linear, we can take advantage of existing convex optimization libraries, which do not require an initial estimate of relative pose. In addition, because the proposed method does not need to perform data association, we can align trajectories using low-dimensional feature vectors and can thus decrease the amount of data that must be transferred between agents by several orders of magnitude when compared to image feature descriptors such as SIFT and SURF. We evaluate the proposed method on simulated datasets and apply it to real-world data collected during autonomous ship hull inspection field trials.

Chapter 4 We formulate the planar Pose Graph and Landmark SLAM problems as polynomial optimization problems. We then describe how the sparse bounded degree sum-of-squares (Sparse-BSOS) hierarchy of SDPs can be used to find its solution. We then prove that the Pose Graph and Landmark SLAM formulations we propose are sum-of-squares (SOS) convex meaning their globally optimal solution can always be found by solving the first SDP in the Sparse-BSOS hierarchy. We evaluate the proposed

method on two existing SLAM datasets and show that while Levenberg-Marquardt (Marquardt, 1963) gets caught in a local minimum, our proposed method does not need an initialization and is able to find the optimal solution.

Chapter 5 We present a framework for modeling the uncertainty of jointly distributed poses while accurately accounting for the structure of the Special Euclidean group and describe how to perform the equivalent of the SSC (Smith, Self, and Cheeseman, 1990) pose composition, pose inverse, and relative pose operations while using the Lie algebra to characterize uncertainty. We evaluate the proposed method and show that it outperforms SSC and other existing methods. We release an open source C++ library of the proposed methods.

Chapter 6 We summarize the contributions of this thesis. Additionally, we explore a few avenues for future work.

Appendix A We briefly define some important concepts from graph theory.

The concepts presented in this appendix is used in Chapter 2.

Appendix B We briefly review convex optimization theory and its benefits. We also define several common types of convex programs that are used in our derivations.

The work presented in this appendix is used in Chapter 3 and Chapter 4.

CHAPTER 2

Consistency Set Maximization for Robust Map Merging

2.1 Introduction

In multi-agent simultaneous localization and mapping (SLAM), multiple robots collect measurements about their own trajectories, the environment, and possibly other robots. To generate an accurate map of the environment, it is necessary to estimate both the local trajectories of the robots as well as the relative offsets (translation and orientation) between the trajectories.

In pose graph SLAM, the map estimation problem is formulated as a factor graph consisting of pose node variables and factor nodes. We often formulate the single robot pose graph SLAM problem as the maximum likelihood estimate (MLE) of the time discretized robot trajectory given odometric and loop-closure measurements (Cadena et al., 2016). Assuming independent measurements and additive Gaussian noise in the measurement and robot process models, this becomes a non-linear, weighted least squares problem and can be solved quickly using available solvers (Kaess, Ranganathan, and Dellaert, 2008; Kümmerle et al., 2011).

However, accurately determining large loop-closure factors is a difficult problem, and least squares can be susceptible to outliers. A variety of methods have been proposed to enhance robustness by disabling factors with high residual error (Agarwal et al., 2013; Latif, Cadena, and José, 2012; Olson and Agarwal, 2013; Sünderhauf and Protzel, 2012), but they either depend on having an odometry backbone of trusted measurements to judge against or a good initial estimate of the relative pose of the vehicles (Pfingsthorn and Birk, 2016). In multi-agent SLAM, this odometry backbone is non-existent, and a prior estimate of relative pose may not exist.

The use of multiple vehicles enables increased scalability and efficiency in the mapping process. However, to accurately fuse their maps, the vehicles must estimate their position

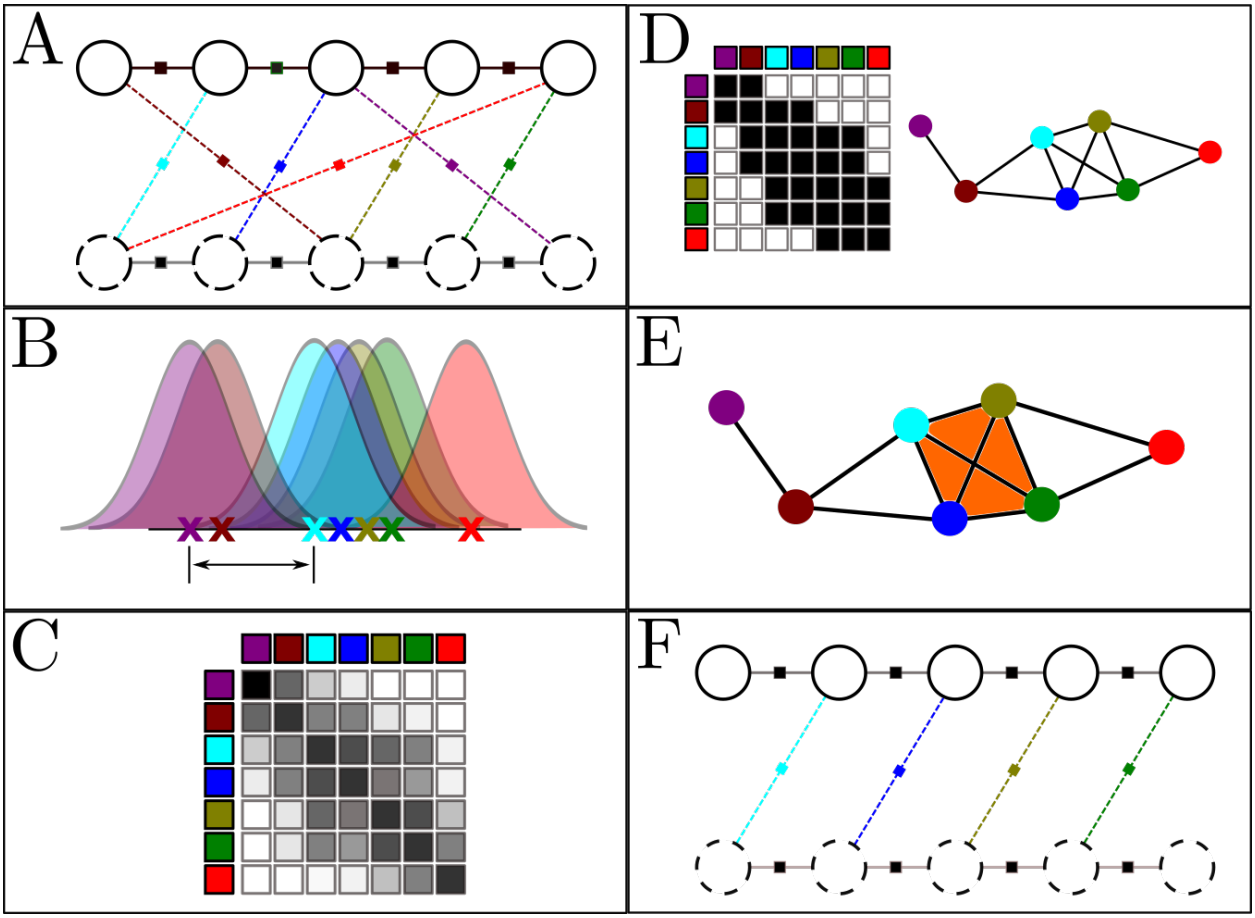


Figure 2.1: Pairwise Consistency Maximization Overview. An illustration of the Pairwise Consistency Maximization (PCM) algorithm for selecting consistent inter-map loop closures measurements. (A) Given two independently derived pose graphs (shown in white and black in step A) and a set of potential loop-closures between them (shown by colored, dotted lines), our goal is to determine which of these inter-robot loop-closures should be trusted. (B) Using a consistency metric such as Mahalanobis distance, we calculate the consistency of each pairwise combination of measurements. (C) We store these pairwise consistency values in a matrix where each element corresponds to the consistency of a pair of measurements. (D) We can transform this matrix into the adjacency matrix for a *consistency graph* by thresholding the consistency and making it symmetric using the maximum consistency when associated elements across the diagonal have differing consistency values. Each node in this graph represents a measurement and edges denote consistency between measurements. Cliques in this graph are *pairwise internally consistent sets*. (E) Finding the maximum clique corresponds to finding the largest pairwise internally consistent set. (F) After determining the largest consistent set, we can robustly merge the two pose graphs using only the consistent inter-map loop closures, allowing us to reject false measurements.

with respect to one another. This process, very similar to the loop-closure problem in single robot SLAM, requires accurately selecting a set of consistent, perception-derived pose

measurements relating the relative position and orientation of the two vehicles and their maps. In this paper, we present a tractable and robust way of selecting which measurements to trust.

Rather than attempt to classify measurements as inliers and outliers, we find the largest consistent set of inter-robot relative pose measurements. We first formulate this problem as a combinatorial optimization problem, which turns out to be an instance of the NP-complete maximum clique problem. We then present a method from graph theory literature that finds the optimal solution for moderately sized problems (including many multi-robot map merging scenarios), as well as a heuristic that can be used to approximate it for larger numbers of measurements.

The contributions of this chapter include the following: *i*) In Section 2.4, a novel definition of *pairwise internally consistent sets* and the formulation of the robust multi-vehicle map merging problem as a combinatorial problem that seeks to find the maximum cardinality set of internally consistent measurements. *ii*) In Section 2.5, a method for transforming the PCM problem into an instance of the maximal clique problem from *consistency graph*, for which several algorithms exist.

The remainder of this chapter is organized as follows. In Section 2.2, we cover related work. In Section 4.3, we present the general formulation of the multi-robot pose graph SLAM problem. In Section 2.3, we outline the steps needed to integrate PCM into a map merging system. In Section 2.7, we evaluate PCM on synthetic and real-world datasets and show that PCM out-performs the state-of-the-art methods in selecting consistent measurements and estimating the merged maps. Finally in Section 2.8, we summarize and conclude the chapter.

2.2 Related Work

Current methods for identifying loop closures often severely limit the number of accepted measurements by setting high likelihood thresholds in an attempt to filter out false positives (Ozog et al., 2016). In pose graph SLAM, inconsistent measurements must be filtered or they may distort the estimated graph.

The detection of outliers is an important problem in both mobile robotics and computer vision and a variety of methods have been proposed to address it. In computer vision, the RANSAC algorithm seeks to determine inlier/outlier sets by iteratively fitting models to samples of the data and counting the number of inliers (Hartley and Zisserman, 2003). However, RANSAC relies on the fact that a point can be designated as an inlier or outlier by measuring its likelihood given a proposed model. However, in multi-robot SLAM, there

is no unique model we can use to determine if a constraint is an inlier. Rather, we need to know if a measurement is consistent with all the other measurements being considered as inliers. In addition, RANSAC and many of its derivatives are sensitive to parameter values that can be difficult to tune.

JCBB is a method that selects measurements by seeking to determine the maximum jointly compatible set (Neira and Tardós, 2001). In the multi-robot map merging problem, however, performing JCBB can be difficult because doing so would require solving the graph for a combinatorial number of measurement combinations to evaluate the likelihood of each measurement given each combination of the other measurements.

There have been several outlier detection methods derived specifically for pose graph SLAM. Switchable constraints (Sünderhauf and Protzel, 2012) proposes the use of of switchable error factors in the SLAM back-end solver that enable pose constraints with high residual error to be "turned off" so that they no longer affect the solution. Dynamic Covariance Scaling (DCS) (Agarwal et al., 2013) builds on the work of Sünderhauf and Protzel (2012) and uses scaling of measurement covariance to more smoothly turn on and off measurement links between poses. Olson and Agarwal (2013) propose Max-Mixtures that uses mixtures of Gaussians to model multiple possible data modes. Realizing, Reversing, Recovering (Latif, Cadena, and José, 2012) iteratively tries to find a set of relative pose measurements that are consistent within themselves. However, each of these methods is designed with a single robot pose graph in mind and fails without a good initialization of relative pose (Agarwal et al., 2013; Olson and Agarwal, 2013; Sünderhauf and Protzel, 2012) or a trusted odometry backbone (Latif, Cadena, and José, 2012).

Recently, several papers have been published directly attempting to solve the "perceptual aliasing" problem of map merging. Perceptual aliasing occurs when multiple locations in the environment look similar enough that perceptual sensors and methods mistake them for the same location. The method presented by Dong et al. (2015) and Indelman et al. (2016) uses expectation maximization to estimate the inliers and outliers in a set of measurements that constrain the relative poses of multiple robots performing cooperative mapping. Our proposed method uses many similar concepts to those presented by Dong et al. (2015), but our solution finds/estimates the maximum cardinality consistent set of inter-robot measurements instead of iteratively refining an initial guess. Generalized graphSLAM (Pfingsthorn and Birk, 2016) handles multi-hypothesis or outlier factors that could be derived from perceptual aliasing by attempting to find the minimum ambiguity spanning tree for the graph. However, their method does not attempt to enforce consistency of measurements and, in the multi-robot case, may select an incorrect measurement if it happens to have the lowest uncertainty.

The two most similar methods to ours are (Carlone, Censi, and Dellaert, 2014) and (Olson, 2009). Our method follows the same approach as Carlone, Censi, and Dellaert (2014) in that we both attempt to find the maximal set of measurements that are consistent with one another. However, their work has only been extended to non-linear measurements in the case of planar SLAM. Single Cluster Graph Partitioning (SCGP) presented in (Olson, 2009; Olson et al., 2005), attempts to determine a single cluster of measurements, given a similarity matrix, by thresholding the elements of the largest eigen-vector. Olson (2009) uses a similar consistency metric to ours, but they do not enforce pairwise internal set consistency as we do.

2.3 Problem Formulation

In our factor graph formulation of SLAM, we denote time discretized versions of the the robot trajectory by $\mathbf{x}_i \in \mathrm{SE}(2)$ or $\mathrm{SE}(3)$. The factors in the graph are derived from the measurements observed by the robot and penalize estimates of the trajectory that make the observed measurement unlikely. We denote measurements that relate the variables $\mathbf{x}_i$ and $\mathbf{x}_j$ by $\mathbf{z}_{ij}$ and call them odometric measurements if i and j are consecutive and loop closure measurements if i and j are non-consecutive in time. The goal of pose graph SLAM is, then, to estimate the most likely value of each pose variable $\mathbf{x}_i$ given the factor measurements $\mathbf{z}_{ij}$. We can formulate the single robot pose graph SLAM problem as the MLE

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmax}}\, P(\mathbf{Z}|\mathbf{X}). \tag{2.1}$$

where, $\mathbf{X}$ is the set of all pose variables $\mathbf{x}_i$, and $\mathbf{Z}$ is the set of all relative pose measurements robot measurements $\mathbf{z}_{ij}$.

In multi-robot SLAM, we also need to estimate the relative transformation between the local coordinate frames of the respective robots. We adopt the method presented by Kim et al. (2010), which proposes the use of an anchor node for each trajectory that encodes the pose of the vehicle's local coordinate frame with respect to some global reference frame. We denote the homogeneous transformation matrix representing this offset by T_a^g and represent measurements relating cross-trajectory poses by $\mathbf{z}_{ij}^{ab}$, where a and b are robot IDs and i and j respectively denote which poses on robots a and b are being related. T_a^g is an element of $\mathrm{SE}(2)$ or $\mathrm{SE}(3)$. $\mathbf{z}_{ij}^{ab}$ is also often an element of $\mathrm{SE}(2)$ or $\mathrm{SE}(3)$ but can be a function of this transformation in general.

In the case of two robots, the SLAM estimation problem becomes

$$\hat{\mathbf{X}}, \hat{\mathbf{T}}^g = \underset{\mathbf{X}, \mathbf{T}^g}{\operatorname{argmax}} P(\mathbf{Z}^a, \mathbf{Z}^b, \mathbf{Z}^{ab} | \mathbf{X}, \mathbf{T}^g), \tag{2.2}$$

where, $\mathbf{X}$ now represents the trajectories of both robots, $\mathbf{Z}^{ab}$ represents the set of all cross-trajectory measurements, $\mathbf{Z}^r$ represents the set of measurements local to robot r, and $\mathbf{T}^g = \{\mathbf{T}^g_a, \mathbf{T}^g_b\}$. This problem can be treated as weighted, non-linear least squares and can be solved efficiently using an array of specialized optimization libraries.

Existing methods do a good job of handling outlier measurements in the local measurement sets $\mathbf{Z}^a$ and $\mathbf{Z}^b$, but not in the inter-robot set $\mathbf{Z}^{ab}$ since generally, no prior estimate of the initial transformation between the robot coordinate frames is available. The focus of this paper is on selecting a subset of the measurements in the inter-robot set $\mathbf{Z}^{ab}$ that we can trust. The next section outlines our approach for doing so.

2.4 Pairwise Consistency Maximization

In this section, we first define a novel notion of consistency and then we use that notion to formulate the selection of inter-robot loop closure measurements as a combinatorial optimization problem that finds the largest consistent set.

2.4.1 Pairwise Consistency

Directly determining if a measurement is an inlier or outlier from the graph itself is unobservable (Carlone, Censi, and Dellaert, 2014). Thus, instead of trying to classify inlier versus outlier, we attempt to determine the maximum subset of measurements that are internally pairwise consistent:

Definition 2.4.1. *A set of measurements $\widetilde{\mathbf{Z}}$ is **pairwise internally consistent** with respect to a consistency metric C and the threshold γ if*

$$C(\mathbf{z}_i, \mathbf{z}_j) \leq \gamma, \quad \forall \quad \mathbf{z}_i, \mathbf{z}_j \in \widetilde{\mathbf{Z}} \tag{2.3}$$

where, C is a function measuring the consistency of measurements $\mathbf{z}_i$ and $\mathbf{z}_j$, and γ is chosen a priori.

This definition of consistency requires that every measurement in the set be consistent with every other measurement in the set with respect to C and γ.

There are a variety of potential choices of a consistency metric depending on the measurement model and the state being observed. For the remainder of this chapter, however, we assume that all inter-robot measurements are relative pose measurements with full degrees of freedom and use the following metric based on (Olson, 2009):

$$C(\mathbf{z}_{ik}^{ab}, \mathbf{z}_{jl}^{ab}) = \left|\left|(\ominus \mathbf{z}_{ik}^{ab}) \oplus \hat{\mathbf{x}}_{ij}^{a} \oplus \mathbf{z}_{jl}^{ab} \oplus \hat{\mathbf{x}}_{lk}^{b}\right|\right|_{\Sigma} \triangleq ||\epsilon_{ikjl}||_{\Sigma_{ikjl}} \tag{2.4}$$

where, we have adopted the notation of (Smith, Self, and Cheeseman, 1990) to denote pose composition using $\oplus$ and inversion using $\ominus$, $||\cdot||_{\Sigma}$ signifies the Mahalanobis distance, and the variables $\hat{\mathbf{x}}_{ij}^{a}$ and $\hat{\mathbf{x}}_{lk}^{b}$ are the current relative pose estimates of the associated poses corresponding to inter-robot measurements $\mathbf{z}_{ik}^{ab}$ and $\mathbf{z}_{jl}^{ab}$.

This choice of metric is useful because it is both easy to compute and follows a chi-squared distribution, giving us a strategy to select the threshold γ without knowledge of the specific dataset. The composition inside the norm of (2.4) evaluates the pose transformation around a loop and should evaluate to the identity transformation in the case of no noise (Olson, 2009). With Gaussian noise, this normalized squared error follows a chi-squared distribution with degree of freedom equal to the number of degrees of freedom in our state variable. By setting γ accordingly, we can determine if the measurements $\mathbf{z}_{ik}^{ab}$ and $\mathbf{z}_{jl}^{ab}$ are consistent with one another.

It should also be noted that pairwise consistency does not necessarily signify full joint consistency. It is possible that a set of measurements can be pairwise internally consistent but not jointly consistent. However, checking full joint consistency is an exponential operation and requires potentially checking every combination of measurements to evaluate their consistency. Finding the maximum cardinality pairwise consistent set is also exponential, but by formulating the problem in this way, we can leverage a body of literature on the maximum clique problem in graph theory that can find or estimate the solution efficiently. In addition, in practice, we observed that testing for pairwise consistency was restrictive enough to filter inconsistent measurements from typical pose graphs with full degree of freedom measurements.

2.4.2 The Maximal Cardinality, Pairwise Consistent Set

Having this definition of pairwise internal consistency allows us to restrict our algorithm to only consider sets of measurements that are pairwise internally consistent; however, due to perceptual aliasing, we may end up with multiple subsets that are pairwise internally consistent. We need to find a way to select between these possible subsets.

The underlying assumption of our method is based on the following two initial assump-

tions:

Assumption 2.4.1. *The pose graphs are derived from multiple robots or the same robot in multiple sessions exploring the same environment.*

Assumption 2.4.2. *The inter-robot relative pose measurements are derived from observations of that environment and the system used to derive them is not biased toward selecting incorrect measurements over correct ones.*

These assumptions fit a large number of multi-robot mapping situations and are reasonable even in perceptually aliased environments whenever a place recognition system does not systematically select the perceptually aliased measurement over the correct ones.

If the above conditions are met than the following can also be safely assumed:

Assumption 2.4.3. *As the number of relative pose measurements increases, the number of measurements in the correct consistent subset will grow larger than those in the perceptually aliased consistent subsets.*

Our goal is, then, to efficiently find the largest consistent subset of $\mathbf{Z}^{ab}$, which we denote by $\mathbf{Z}^*$.

To formalize this, we introduce a binary switch variable, s_u, for each constraint in the set $\mathbf{Z}^{ab}$ and let s_u take on the value 1 if the measurement is contained in the chosen subset and 0 otherwise. Note that there is a single s_u for each measurement $\mathbf{z}_{ij}^{ab} \in \mathbf{Z}^{ab}$; however, for simplicity of notation, we now re-number them with the single index u and denote the corresponding measurement $\mathbf{z}_{ij}^{ab}$ by $\mathbf{z}_u$. Letting $\mathbf{S}$ be the vector containing all s_u, our goal is to find the solution, $\mathbf{S}^*$, to the following optimization problem:

$$\begin{aligned} \mathbf{S}^* &= \underset{\mathbf{S}\in\{0,1\}^m}{\operatorname{argmax}} \ \|\mathbf{S}\|_0 \\ \text{s.t. } &||\epsilon_{uv}||_{\Sigma_{uv}} \, s_u s_v \leq \gamma, \end{aligned} \tag{2.5}$$

where, m is the number of measurements in $\mathbf{Z}^{ab}$, $\mathbf{z}_u$ is the measurement corresponding to s_u, ϵ_{uv} is the associated error term corresponding to measurements $\mathbf{z}_u$ and $\mathbf{z}_v$, and Σ_{uv} is the covariance matrix associated with the error ϵ_{uv}. We refer to this as the PCM problem.

Once found, we can use $\mathbf{S}^*$ to index into $\mathbf{Z}^{ab}$ and get $\mathbf{Z}^*$. This consistent subset of the measurements can then be plugged into any of the existing non-linear least squares based solvers to merge the individual robot maps into a common reference frame. In the next section, we show how this problem can be reformulated into an equivalent problem that has been well studied.

2.5 Solving PCM via Maximum Clique

In this section, we describe how to solve the PCM problem. The goal of PCM to determine the largest subset of the measurements $\mathbf{Z}^{ab}$ that are pairwise internally consistent. This pairwise consistency is enforced by the n^2 constraints listed in (2.5). It is important to note that the norm on the left-hand side of the constraints does not contain any of the decision variables s_i. These distance measures can be calculated in pre-processing, as will be explained in Section 2.6, and combined into a matrix of consistency measures $\mathbf{Q}$, where each element $[\mathbf{Q}]_{uv} = q_{uv} = ||\epsilon_{uv}||_{\Sigma_{uv}}$, corresponds to the consistency of measurement $\mathbf{z}_u$ and $\mathbf{z}_v$. This process is depicted in steps B and C in Fig. 2.1.

We'll now introduce the concept of a *consistency graph*.

Definition 2.5.1. *A **consistency graph** is a graph $G = \{V, \mathcal{E}\}$ where each vertex $v \in V$ represents a measurement and each edge $e \in \mathcal{E}$ denotes consistency of the vertices it connects.*

We can transform the matrix of consistency measures $\mathbf{Q}$ into the adjacency matrix for a consistency graph if we threshold it by γ and make it symmetric by requiring that both q_{uv} and q_{vu} be less than or equal to γ to insert an edge into the graph. An example adjacency matrix and consistency graph are shown in step D of Fig. 2.1.

A *clique* in graph theory is defined as a subset of vertices in which every pair of vertices has an edge between them and the *maximum clique* is the largest such subset of nodes in the graph. A clique of the consistency graph corresponds to a *pairwise internally consistent set* of measurements because every measurement is pairwise consistent with every other measurement in the set. Thus, the solution to the problem defined in (2.5) is the maximum clique of the consistency graph (see step E of Fig. 2.1).

In graph theory, the problem of finding the maximum clique for a given graph is called the maximum clique problem and is an NP-hard problem (Wu and Hao, 2015). The maximum clique problem is also hard to approximate (Feige et al., 1991; Zuckerman, 2006), meaning that finding a solution arbitrarily close to the true solution is also NP-hard. Dozens of potential solutions have been proposed, each of which can be classified as either an exact or a heuristic algorithm. All of the exact algorithms are exponential in complexity and are usually based on branch and bound, while the heuristic algorithms often try to exploit some type of structure in the problem, making them faster, but without guaranteeing the optimal solution (Wu and Hao, 2015).

In 2015, Pattabiraman et al. (2015) proposed a method that aggressively prunes the search tree and is able to find maximum clique solutions for large sparse graphs relatively quickly. They present both an exact algorithm as well as a heuristic version that can be used when the exact algorithm becomes intractable. Though our method could theoretically

use any one of the proposed maximum clique algorithms, we selected the one proposed in (Pattabiraman et al., 2015) because of its simplicity, parallelizability, and open source implementation.

2.6 Robust Multi-Robot Map Merging

Integrating PCM into a map merging system consists of four steps: *i*) Individual map generation *ii*) Consistency calculation *iii*) Pairwise consistency maximization *iv*) Constraint insertion and full map generation

The individual map generation step consists of solving for an estimate of the individual pose graphs. This can be carried out by any pose graph SLAM-based method as long as it is possible to estimate the marginal and cross-covariances of nodes and sets of nodes after a solution has been found. This is important for estimating the Mahalanobis distance matrix.

Using the consistency metric we have selected, the first step in estimating the Mahalanobis distance matrix is to extract the necessary marginal covariances. As explained in Section 2.4.1, we take each pair of potential, inter-robot measurements and trace a loop using the transformations from the underlying graph. To trace this loop, we extract the covariance for every pair of nodes on a given local map that are associated with an inter-robot constraint. Then, using the methods described in (Smith, Self, and Cheeseman, 1990), we calculate the "tail-to-tail" between these nodes allowing us to estimate $\hat{\mathbf{x}}_{ij}^{a}$ and $\hat{\mathbf{x}}_{lk}^{b}$ and their associated covariances. Finally, we use (2.4) to estimate the distances $||\epsilon_{ijkl}||_{\Sigma_{ijkl}}$ needed in (2.5). $\Sigma_{ijkl} = J\Sigma J^{\top}$, where J is equal to the Jacobian of ϵ_{ijkl} with respect to the two measurements and the two estimated local trajectory components evaluated at their means. Σ is equal to the associated block diagonal covariance matrix built up of the covariances of the measurements and pose estimates.

In practice, this calculation is the bottleneck in processing for our algorithm; however, by using analytical Jacobians, parallelization, and incremental updates, we can significantly reduce this calculation time. For more information on calculating the Mahalanobis distance see (Smith, Self, and Cheeseman, 1990).

Once the matrix has been calculated, we convert it to an adjacency matrix and solve the maximum clique problem as explained in Section 2.5 to determine the set of constraints that should be enabled. Finally, we re-solve the graph using only those inter-robot constraints. In practice, when optimizing over SE(3), it helped to incrementally re-build the graph running a batch optimization every few hundred nodes, while for SE(2) it was sufficient to add the inter-robot factors directly to the already existing graph and run a batch optimization.

2.7 Evaluation

In this section, we evaluate the performance of pairwise consistency maximization (PCM) on a variety of synthetic and real-world data-sets. For comparison, we implemented SCGP (Olson et al., 2005), DCS (Agarwal et al., 2013), and random sample consensus (RANSAC) (Hartley and Zisserman, 2003).

We implemented SCGP as described in Olson et al. (2005), with the exception of using an off the shelf eigen-factorization library as opposed to the power method for simplicity. We implemented DCS as described in the original paper, Agarwal et al. (2013), with $\phi = 5$.

We implemented RANSAC by iteratively selecting a single, random inter-map measurement and evaluating the likelihood of the other measurements given the model estimated from the sampled measurement. Because the processing time for this evaluation is so low (given that the Mahalanobis distance evaluations were performed in pre-processing), we exhaustively iterate through all the measurements and evaluate the likelihood of the other measurements with respect to it in turn. We then return the set of measurements that are likely given the sampled point with the largest support. As explained in Section 2.7.2, RANSAC is especially sensitive to the likelihood threshold and does not check pairwise consistency.

For PCM, we present results using the exact maximum clique algorithm (PCM-Exact), as well as the heuristic algorithm (PCM-HeuPatt) as explained in Section 2.5.

2.7.1 Simulated 1D World

First, we simulated a one dimensional world where the robot has a single state variable, x, and receives measurements that are direct observations of that state. We simulate inlier measurements by drawing multiple samples from a Gaussian with a fixed variance and mean x. We simulate both random and perceptually aliased outliers by drawing multiple samples from a single Gaussian with fixed mean and variance and several others from individual Gaussians with random means and variances. We assume the variances are known and are used when computing Mahalanobis distance.

2.7.1.1 Comparison with Combinatorial

For this first experiment, we compare how well PCM-Exact and PCM-HeuPatt approximated the combinatorial gold standard in (2.5). We generated 100 000 sample worlds. On each of these samples, we estimated the pairwise consistent set using the combinatorial solution as well as PCM-Exact, PCM-HeuPatt, SCGP (Olson et al., 2005), and RANSAC (Hartley and

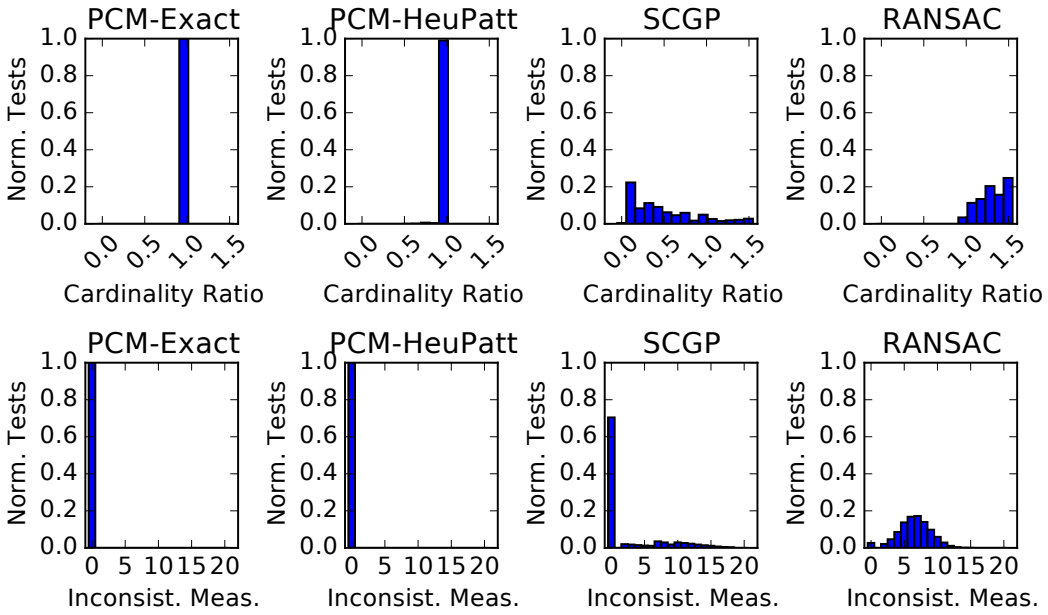


Figure 2.2: Monte-Carlo Approximation Evaluation. Histograms that evaluate how well PCM, SCGP (Olson et al., 2005), and RANSAC (Hartley and Zisserman, 2003) approximate the combinatorial maximum pairwise consistent set in (2.5). The first row of histogram plots shows the size of the measurement set as compared to the maximum consistent set size. The second row of histograms shows the number of inconsistent pairs returned with respect to the set γ threshold on Mahalanobis distance.

Zisserman, 2003).

Fig. 2.2 shows a comparison between these four methods with respect to the combinatorial solution. Both PCM methods enforce consistency of the returned measurements. PCM-Exact returns the same number of points as the combinatorial solution 100 percent of the time, while PCM-HeuPatt returns the same number of points 98.97 percent of the time. SCGP varies significantly in both the number of points returned and the consistency of those measurements. RANSAC also sometimes returns more or less points than the combinatorial solution and also fails to enforce measurement consistency.

Interestingly, RANSAC is especially dependent on threshold value. The threshold value for RANSAC is centered around a single point and thus is not the same as the threshold value for PCM. If the value is set too high, the number of inconsistent measurements increases. If it is set too low, the total number of returned measurements decreases below the optimal. In Fig. 2.2, RANSAC's threshold is set arbitrarily to show a single snapshot.

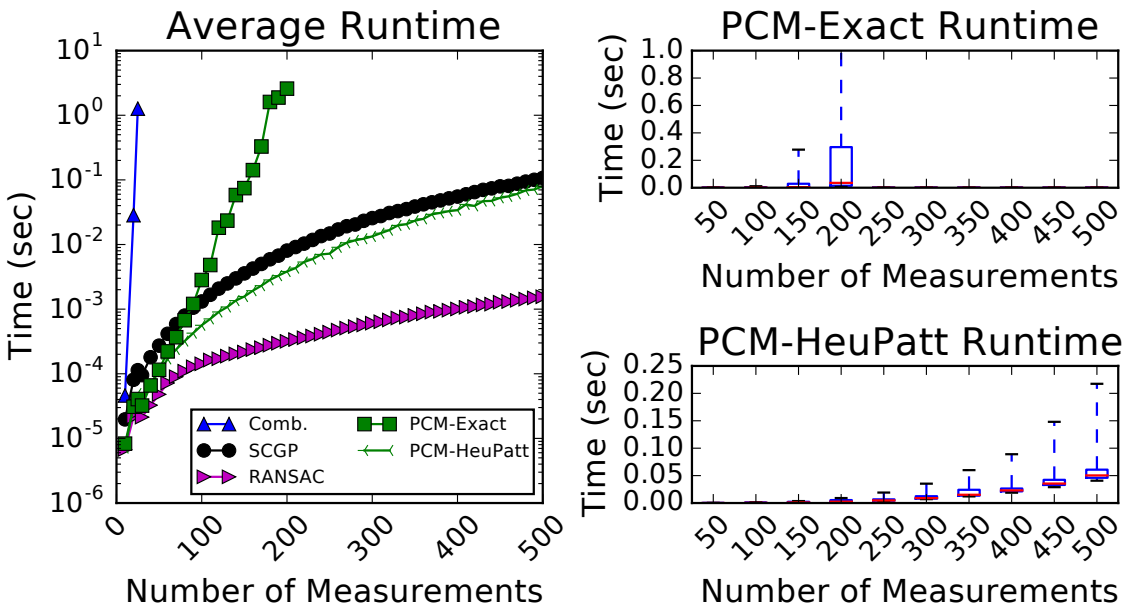


Figure 2.3: PCM Timing Comparison. A plot of the evaluation times of the different methods versus the number of measurements being tested. The combinatorial solution takes exponential time and PCM-Exact takes exponential time in the worst case, while the other methods are polynomial in the number of measurements. (This excludes the time to estimate the distance matrix $\mathbf{Q}$, which is required for all methods.)

2.7.1.2 Timing Comparison

We also used this 1D World to evaluate the timing characteristics of the different algorithms. To test this, we generated 500 sample worlds each, for an increasing number of measurement points. The results are shown in Fig. 2.3. Note, these timing results can be significantly improved through parallelization.

2.7.2 Synthetic 2D Comparison

To test our method's accuracy and consistency on a full SLAM dataset, we took a portion of the City10000 dataset released with incremental smoothing and mapping (iSAM) (Kaess, Ranganathan, and Dellaert, 2008) and split it to form two separate robot trajectories. After removing all factors connecting the two graphs, we generated 81 different versions of this dataset by randomly selecting a subset of the true loop closures between the two graphs to be used as inliers, as well as randomly adding outlier loop closures to the graph. As before, some of the outliers are internally consistent to simulate perceptual aliasing and some are generated randomly with random mean and covariance. In this experiment, the number of inlier loop closures was 15, there were two groups of 5 perceptually aliased outliers, and the

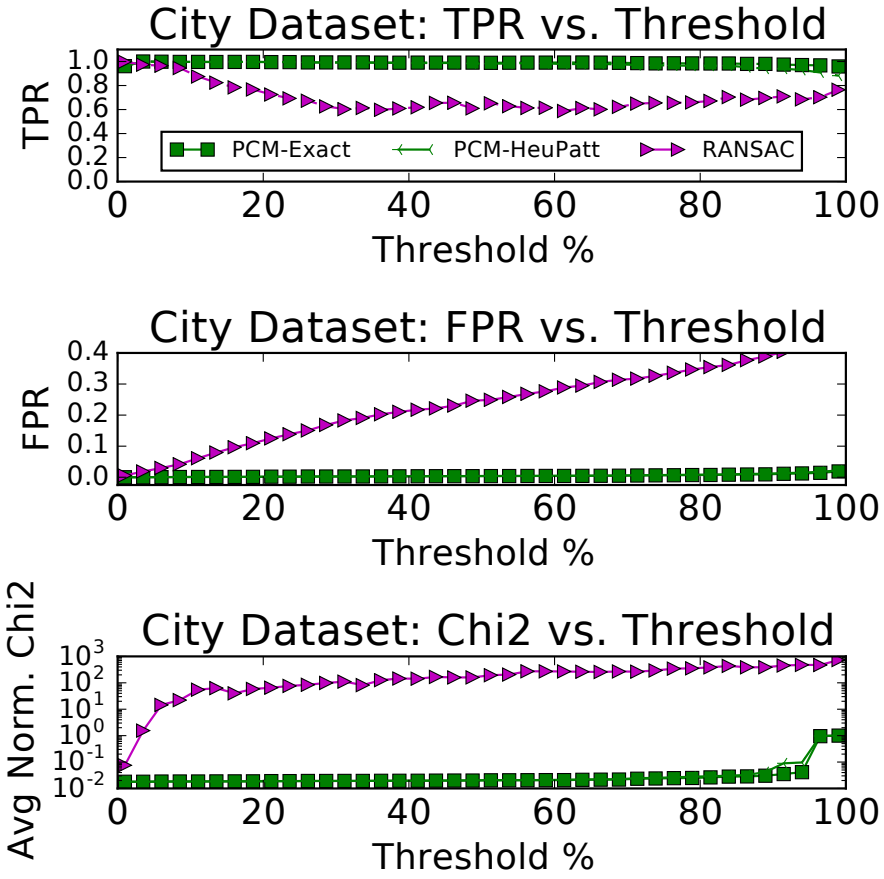


Figure 2.4: City Dataset TPR, FPR, Consistency versus Threshold. Plots of the true positive rate (TPR = TP / (TP + FN)), false positive rate (FPR = FP / (FP + TN)), and average normalized chi-squared value (Chi2) of PCM-Exact, PCM-Heu, and RANSAC versus the threshold value γ. The TPR and FPR can be thought of as the probability of getting a true positive or a false positive. The Chi2 value should be close to zero if the measurements in the graph are consistent.

number of random outliers was 90.

2.7.2.1 Parameter Sweep

Because RANSAC is significantly dependent on the threshold value set, we ran a parameter sweep for the likelihood threshold over all 81 datasets. Fig. 2.4 summarizes this experiment. The true positive rate (TPR) and false positive rate (FPR) of PCM is relatively unaffected by the choice of the threshold parameter as long as it is less than about 85 percent. RANSAC, on the other hand, has a different FPR for each threshold selected and never has an FPR of zero. This is because PCM conservatively evaluates the consistency of each measurement and determines the consistency of a group of measurements as a whole, while RANSAC selects

Table 2.1: Simulated City Dataset Comparison Results. Summary of results from using DCS (Agarwal et al., 2013), SCGP (Olson et al., 2005), RANSAC (Hartley and Zisserman, 2003)(with two different thresholds), and PCM to robustly merge maps generated from a synthetic city dataset. These results are a summary of runs on 81 different generated datasets. We evaluated the mean squared error (MSE) of the two graphs with respect to the non outlier case (NO-OUT). The worst results for each metric are shown in red, the best are shown in **blue**, and the second best shown in **BOLD**.

	Trans. MSE (m^2)		Rot. MSE		Residual		Inliers		Chi2 Value		Eval Time (sec)
	Avg	Std	Avg	Std	Avg	Std	TPR	FPR	Avg	Std	Avg
NO-OUT	0.0	0.0	0.0	0.0	32.320	0.117	1.0	0.0	N/A	N/A	N/A
DCS	183077.917	1194931.105	4.169	3.285	**31.687**	**0.507**	0.0	**0.0**	**0.013**	**< 0.001**	N/A
SCGP	623.278	1278.493	0.648	1.535	237385.743	894303.187	0.668	0.051	96.734	364.427	0.006
RANSAC-1%	**5.688**	**21.976**	**0.009**	**0.040**	185.190	587.590	**0.998**	0.006	0.076	0.239	**< 0.001**
RANSAC-3.5%	183.150	636.441	0.236	0.791	3807.570	18478.340	0.974	0.019	1.552	7.530	**< 0.001**
PCM-Exact-11%	**0.276**	**1.537**	**< 0.001**	**0.003**	**45.057**	**105.385**	**0.997**	**0.001**	**0.018**	**0.043**	**< 0.001**
PCM-HeuPatt-11%	**0.276**	**1.537**	**< 0.001**	**0.003**	**45.057**	**105.385**	**0.997**	**0.001**	**0.018**	**0.043**	**< 0.001**

the largest set of measurements that are likely given a single randomly selected measurement. The last plot shows the average normalized chi-squared value of the residual for the entire graph after solving with the selected factors. This value should be close to zero if the graph is consistent.

The results show that PCM does significantly better at restricting the set of measurements to those that are consistent with one another, decreasing the likelihood of getting a false measurement. This is essential because of the extreme susceptibility of SLAM to false loop closures. PCM-HeuPatt is also almost indistinguishable from PCM-Exact.

2.7.2.2 Accuracy Analysis

To evaluate the accuracy of PCM, we compared its performance on all 81 datasets to SCGP, RANSAC (using the two second to lowest thresholds from Fig. 2.4), and DCS (Agarwal et al., 2013). γ for both PCM-Exact and PCM-HeuPatt was set so that it corresponded to the equivalent of 11% likelihood.

Table 2.1 gives an overall summary of the results. We used the Mean Squared Error (MSE) of the trajectory of the two graphs (with respect to the no-outlier case), the residual, and the normalized chi squared value of the non-linear least squares solver as metrics to evaluate the solution accuracy. The rotation mean-squared error (MSE) was calculated via $\epsilon_{rot} = \frac{1}{n}\sum_i \left|\left|log(R_{i\_true}^{\top} R_{i\_est})\right|\right|_F$ over each pose i and the translation MSE was calculated in the normal manner. For this experiment, all MSE values were calculated with respect to the absolute trajectory value.

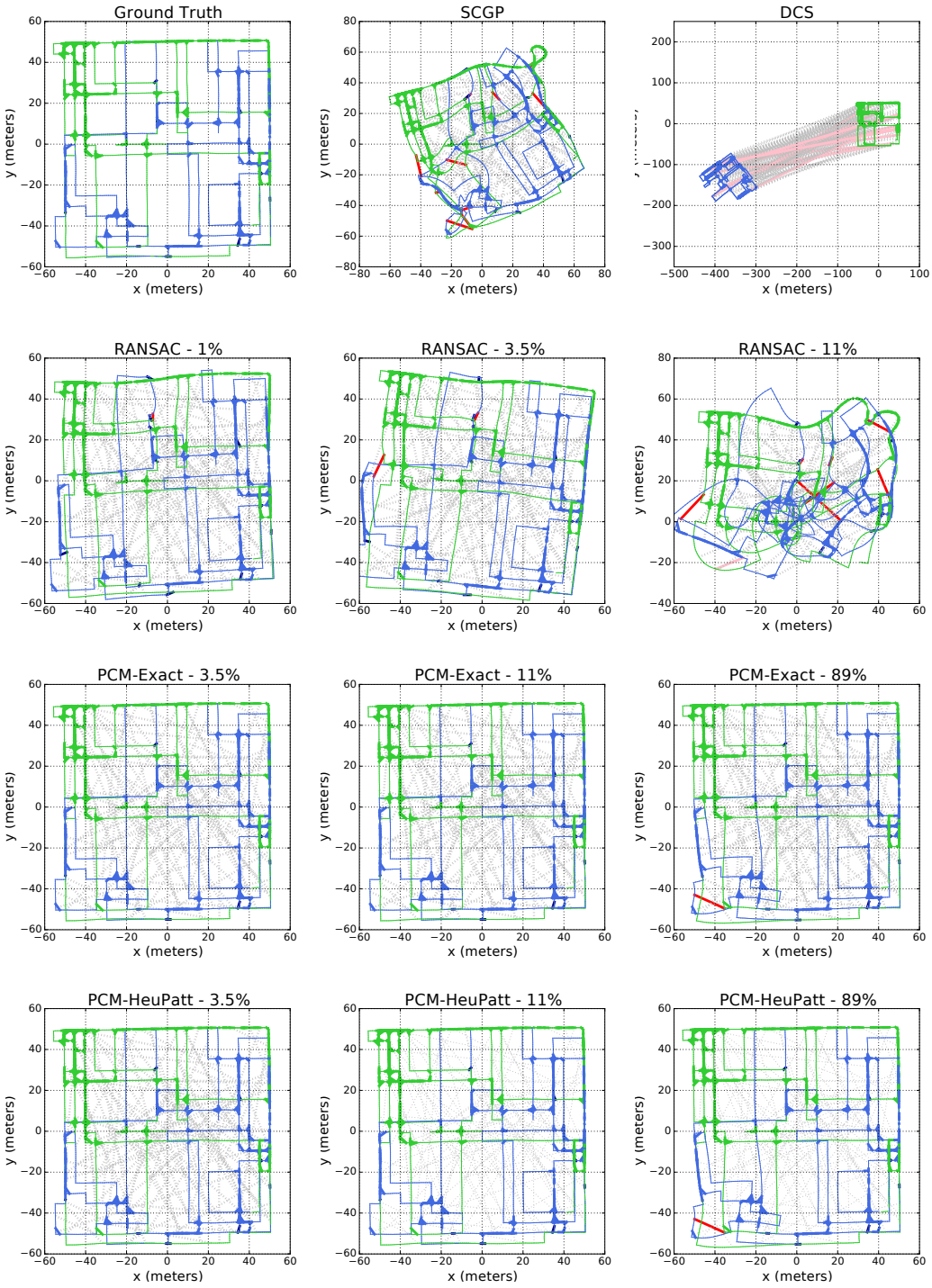


Figure 2.5: Example Plots on Simulated City Datasets. Example plots of the maps estimated by PCM-Exact, PCM-HeuPatt, RANSAC, DCS, and SCGP for one of the generated city datasets. Correctly labeled inlier factors are shown in bold dark blue with correctly disabled outliers shown as dotted gray. Accepted outliers are shown in bold red with disabled inliers shown in pink. PCM significantly outperforms the other methods. When PCM does accept measurements not generated from the true distribution, thay are still consistent with the uncertainty of the local graphs.

Table 2.2: NCLT Comparison Results. Results from using DCS (Agarwal et al., 2013), SCGP (Olson et al., 2005), RANSAC (Hartley and Zisserman, 2003)(with two different thresholds), and PCM to robustly merge segments extracted from two sessions of the NCLT dataset (Carlevaris-Bianco, Ushani, and Eustice, 2015). NO-OUT corresponds to a version with none of the measurements labeled as outliers. We evaluated the MSE of the two graphs with respect to the groundtruth. The worst results for each metric are shown in red and the best are shown in **blue**.

	Rel. Pose MSE		Traj. MSE		Residual	Inliers		Chi2	Evaluation
	Trans. (m^2)	Rot.	Trans. (m^2)	Rot.	Error	TP	FP	Value	Time (sec)
NO-OUT	455.4763	0.0308	0.0501	0.0005	765.072	10	0	0.3428	N/A
DCS	206782.2303	0.7154	0.0502	**0.0005**	**724.061**	0	**0**	**0.2568**	N/A
SCGP	522.2352	**0.0162**	0.0502	**0.0005**	748.351	3	**0**	0.3417	0.0021
RANSAC - 1%	1244.3818	0.0697	0.1036	0.0015	4228.21	**10**	6	1.8643	$<$ **0.0001**
RANSAC - 3.5%	13507.7032	17.4156	0.1146	0.0040	7457.54	**10**	7	3.2795	0.0001
PCM-HeuPatt	**386.6876**	0.0245	**0.0501**	**0.0005**	817.803	**10**	3	0.3635	0.0001

PCM has the lowest trajectory MSE, and DCS has the lowest residual. Note that DCS also has the highest trajectory MSE, which is as expected. DCS seeks to minimize the least squares residual error and depends on a good initialization to determine what measurements are consistent enough to not be turned off. Without this initialization, DCS has no reason to believe that the inter-map factors are not outliers and thus turns off all the inter-map factors in the graph.

RANSAC and both PCM methods take about the same amount of time to find the consistency set once given the matrix $\mathbf{Q}$. The average time to estimate the Mahalanobis distances without the use of analytical Jacobians, parallelization, and incremental updates was 70.8 seconds.

Fig. 2.5 shows example plots of the estimated maps. Both SCGP and RANSAC have trouble disabling all inconsistent measurements. PCM-Heu well approximates PCM-Exact, and both PCM methods do well at disabling inconsistent measurements. When PCM does accept measurements not generated from the true distibution, they are still consistent with the uncertainty of the local graphs.

2.7.3 Real-World Pose-Graph SLAM

We evaluate PCM on the 3D NCLT dataset (Carlevaris-Bianco, Ushani, and Eustice, 2015). The NCLT dataset was collected using a Seqway robot equipped with a LiDAR, Ladybug3 Camera, Microstrain IMU, and GPS. There are 27 sessions in all with an average length of 5.5 km per session.

For our experiment, we took two sessions collected about two weeks apart, removed the first third of one and the last third of the other, and then generated potential loop-closure

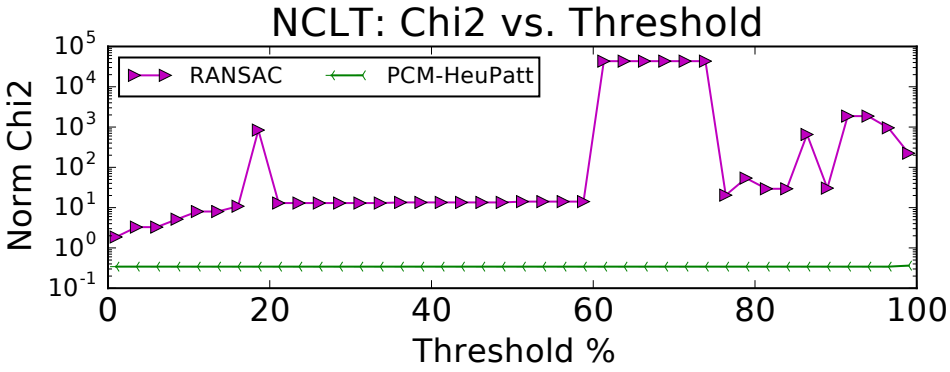


Figure 2.6: PCM and RANSAC Threshold versus Consistency - NCLT. The normalized chi-squared value (Chi2) of the resulting NCLT graph versus the threshold value γ for PCM-HeuPatt and RANSAC. The Chi2 value for RANSAC is never below 1.0 signifying that the selected factors are probabalistically inconsistent, while the Chi2 value for PCM is relatively constant and below 1.0 regardless of threshold.

measurements between the two graphs by aligning every fourth scan on each graph using generalized iterative closest point (GICP) (Segal, Haehnel, and Thrun, 2009) and selecting the match with the lowest cost function. We then labeled these registrations as "inliers" and "outliers" by thresholding the translation and rotation mean squared error of the estimated pose transformations with respect to the ground-truth poses for the dataset derived by performing pose-graph optimization on all 27 sessions. Finally, to increase the difficulty of the dataset, we removed all but one sixteenth of the measurements labeled as "inliers" from the graph, resulting in a graph with 10 "inliers" and 98 "outliers." The GPS measurements and images were ignored.

In this experiment, we compare PCM-HeuPatt with DCS, SCGP, and RANSAC. Fig. 2.6 shows the normalized chi-squared value of the resulting graphs for RANSAC and PCM versus threshold. Table 2.2 provides a comparison of results, and Fig. 2.7 shows the estimated maps. The MSE was calculated using the same method as in the prior section, however in this test we calculated trajectory and map relative pose error separately. The trajectory MSE calculates the error in the estimated relative pose between consecutive nodes allowing us to evaluate graph correctness, while the relative map pose MSE evaluates the offset between the maps.

PCM results in the graph with the best trajectory MSE and the best translational MSE for the relative pose of the two graphs and results in a consistent graph regardless of threshold. It also detects all of the "inlier" measurements as well as three of the measurements labeled as "outliers". DCS once again disables all measurements. SCGP results in a good graph but

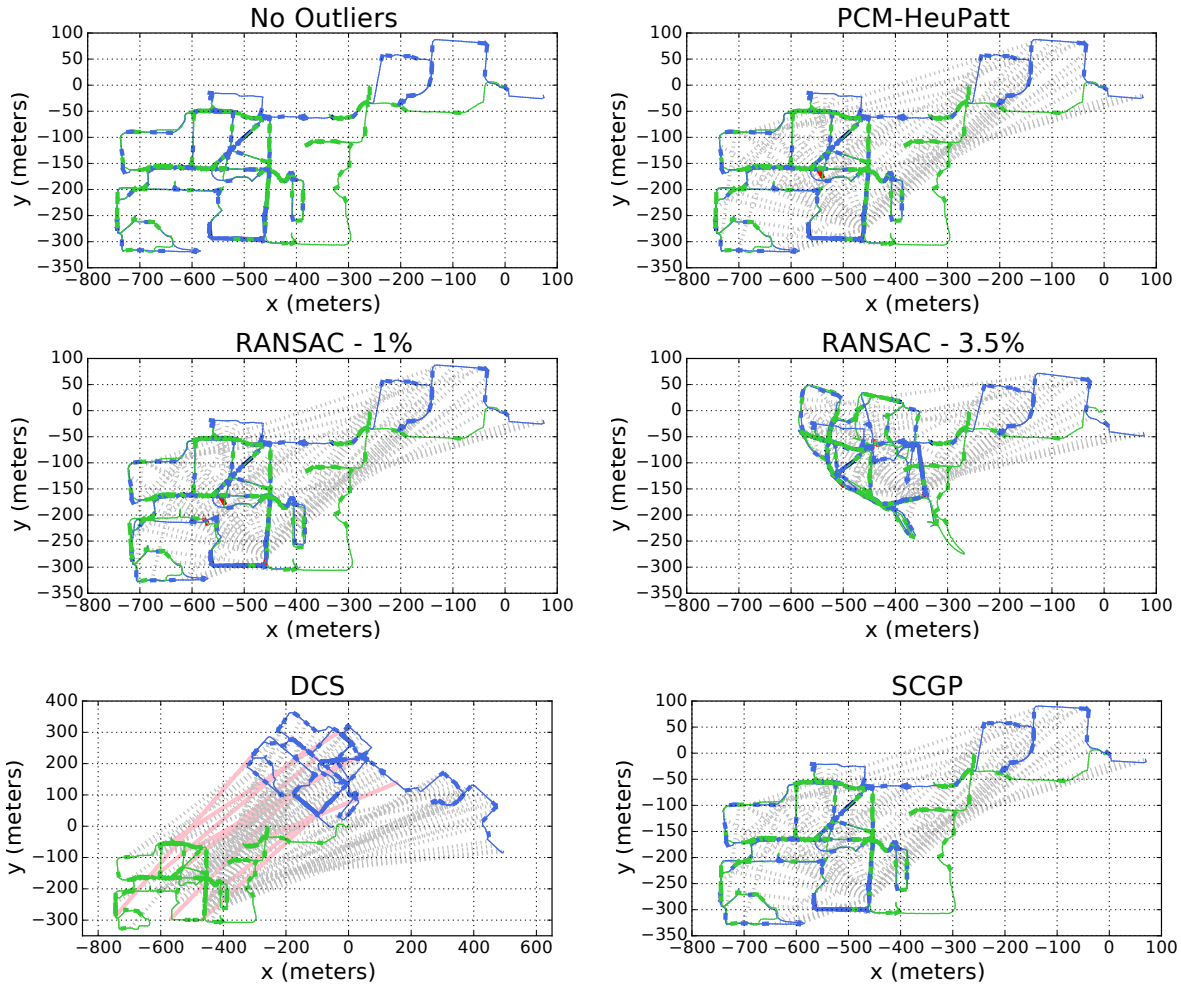
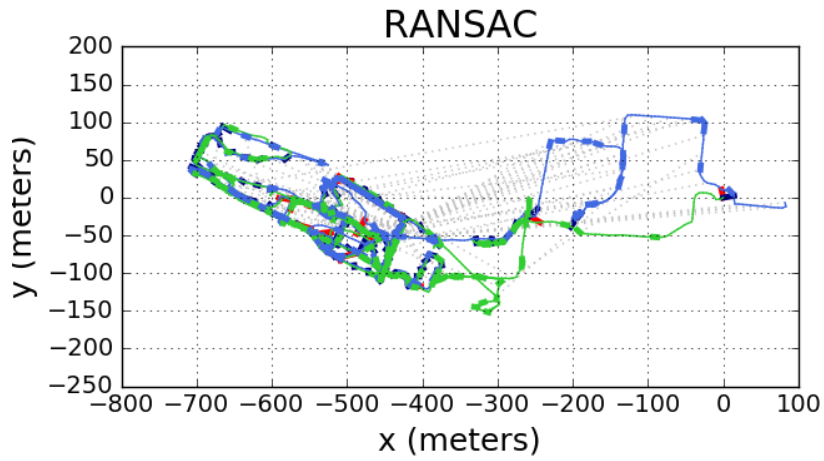


Figure 2.7: Example Plots on NCLT Dataset. Plots of the trajectories of two partial sessions of the NCLT dataset as estimated by PCM-HeuPatt, RANSAC, DCS, and SCGP.

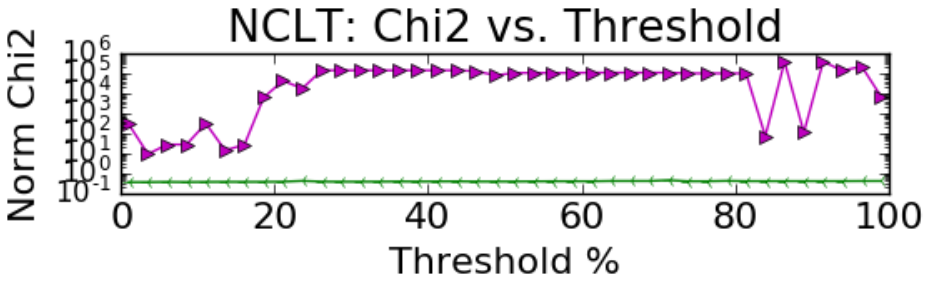
only enables three of the "inlier" measurements, and finally RANSAC (for both of the lowest thresholds tried) enables all "inliers" and several "outliers" and results in an inconsistent graph regardless of the threshold selected.

Note that while in this experiment PCM admits more false positives than in the last experiment, the measurements it accepts are consistent with the "inlier" measurements even though they were labeled as "outliers" (Fig. 2.6). In fact, notice that PCM has a better MSE for the relative map pose then the no outlier (NO-OUT) version of the graph. This suggests that by maximizing the consistent set, PCM is selecting measurements that are actually inliers but were mis-labeled as outliers when compared to the ground-truth. After verification this turned out to be the case.

It is also important to note that although SCGP results in a good graph for this dataset,



(a) RANSAC with Threshold Close to 1%



(b) RANSAC Threshold versus Consistency

Figure 2.8: Additional Evaluation of RANSAC Threshold. Testing for RANSAC with a different sampling of the inlier loop closure measurements. In this case, as the threshold is decreased, the chi squared value of the graph fluctuates as opposed to directly decreasing and even with a low threshold results in an inconsistent graph.

as shown in the earlier experiment, this does not occur in all cases. In addition, it fails to select the maximum consistent set of measurements, this can be catastrophic in the case of perceptual aliasing.

While it may seem that RANSAC will always do well if a low threshold is selected, there is a tradeoff between setting the threshold too low and not getting enough measurements and setting it high and getting outliers. The optimal value for RANSAC's threshold is dependent on the individual measurements observed and varies from dataset to dataset. The threshold γ for PCM has an intuitive probabalistic meaning and can be set a priori. As another example of RANSAC failing regardless of threshold, Fig. 2.8 shows results for running RANSAC on the NCLT dataset if a different sample of inliers is selected.

2.8 Conclusion

In summary, we have developed an algorithm that enables the selection of consistent factors for map merging. The proposed algorithm does not need an initial estimate of relative pose and also is not heavily dependent on threshold parameter values. We've shown that a version of the method restricted to full degree of freedom factors outperforms state of the art methods.

The completed work was published in:

J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–8, Brisbane, Australia, May 2018

A C++ implementation is released at: `https://bitbucket.org/jmangelson/pcm`.

CHAPTER 3

Initialization Agnostic Trajectory Alignment without Data Association

3.1 Introduction

Multi-agent underwater inspection and mapping tasks depend on the ability to determine an alignment between multiple robot maps or trajectories. This is challenging in underwater environments where global positioning systems are unavailable and where acoustic positioning systems require extensive setup and calibration (Hover et al., 2012). Moreover, in fully submersed scenarios, communication is limited to a few bits per second making data transfer a significant system constraint (Walls, Cunningham, and Eustice, 2015).

Existing methods for estimating this alignment rely on either direct measurement of relative pose or the matching of discrete feature points observed by multiple robotic vehicles (Indelman et al., 2016). However, direct relative pose measurement is not always possible (for example in the presence of occlusion) and performing data association requires that feature points transferred between agents be uniquely identifiable. This is usually accomplished through the transfer of high dimensional feature descriptions that often surpass the throughput available in the underwater environment.

This paper proposes a method that efficiently estimates the rigid body transformation between reference and query robot trajectories based on a sparsely sampled underlying feature space. Because we formulate the problem as a convex optimization problem, our method avoids performing data association and decreases the amount of data that needs to be transferred between robotic vehicles by several orders of magnitude. In addition, because our formulation is convex, our proposed method is not dependent on initialization and does not require a prior estimate of the relative transformation between trajectories. Finally, our method is parallelizable and takes advantage of existing commercial optimization libraries to increase the efficiency of the optimization process.

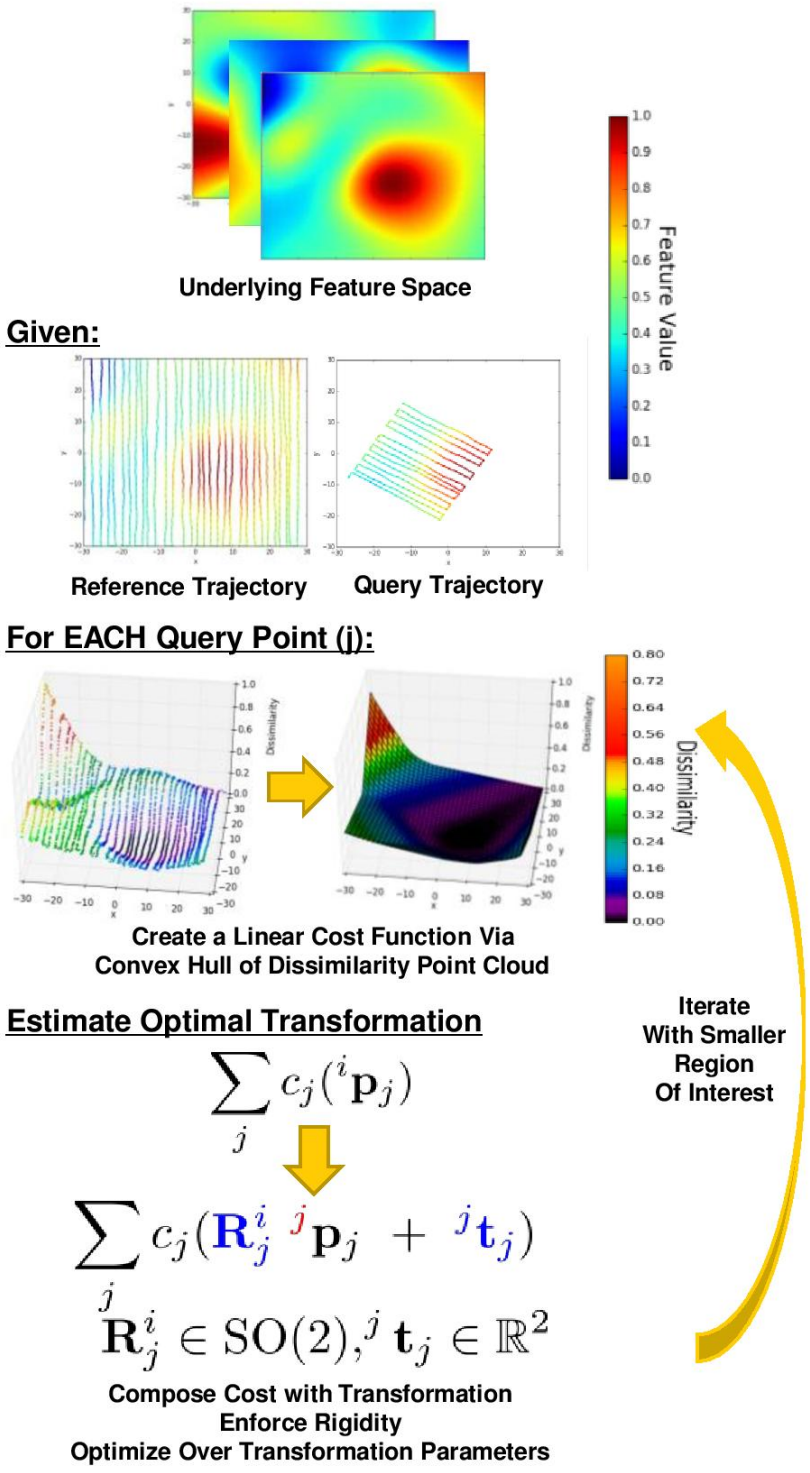


Figure 3.1: Overview of Initialization Agnostic Trajectory Alignment. Given sparse observations of an underlying feature space, we formulate a linear optimization problem that seeks to align a query trajectory with a reference trajectory. We do this by iteratively alternating between creating a linear cost that minimizes feature distance and optimizing over the transformation to minimize that linear cost.

The contributions of this chapter include the following:

1. The development of a system for alignment and localization of robot trajectories that:
   i) Relies only on low-dimensional feature observations.
   ii) Avoids performing data association.
   iii) Does not require an initial alignment estimate.

2. A novel linear approximation based method for approximate optimization over SO(2).

3. A parallelized implementation of the proposed method.

The remainder of this chapter is organized as follows: In Section 3.2, we provide an overview of related areas of work. In Section 3.3, we formalize the trajectory alignment problem. In Section 3.4, we present a method that takes advantage of convex relaxation techniques to generate a linear cost function that is minimized when query feature points are placed near reference feature points with similar value. In Section 3.5, we present a novel method that uses linear programming to approximately optimize over SO(2). In Section 3.6, we provide an outline of the full system and our released implementation. We evaluate the proposed algorithm on simulated datasets in Section 3.7 and apply it to multi-agent ship hull inspection in Section 3.8. Finally, we conclude in Section 3.9.

3.2 Related Work

Multi-agent collaborative mapping has been heavily researched and a variety of methods have been developed for estimating the relative transformation between robot coordinate frames. Early methods assumed that vehicles were able to observe one another directly (Carlone et al., 2010; Howard, Matark, and Sukhatme, 2002; Zhou and Roumeliotis, 2006). These methods relied on a single direct observation to determine the relative pose (position and orientation) of the two agents. Later, maximum likelihood based simultaneous localization and mapping (SLAM) methods enabled the use of multiple observations by estimating the most likely alignment and map given all the observed measurements (Andersson and Nygards, 2008; Cunningham et al., 2012; Dellaert and Kaess, 2006; Eustice, Singh, and Leonard, 2006; Kim et al., 2010). These methods often relax the assumption that vehicles must be able to observe one another directly. Instead, features present in the environment observed by both vehicles are co-registered and used to relate the pose of the two vehicles. More recent methods for determining multi-agent alignment are based on co-registering data in this way (Bonanni, Della Corte, and Grisetti, 2017; Dong et al., 2015; Indelman et al., 2016; Mangelson et al., 2018). While co-registering observed data works well in many cases,

it is dependent on the ability to transmit large amounts of data between vehicles. This work focuses on developing methods that minimize the amount of data that must be transmitted between agents.

Our work is also related to research in terrain based navigation (Carreno et al., 2010; Melo and Matos, 2017; Reis et al., 2017). In the underwater environment, these methods involve the use of a known bathymetric map of the seafloor to improve navigation estimates. These methods, however, are more focused on improving the estimate of a single robot's trajectory than on determining the alignment between multiple trajectories. In addition, these methods require an a prioi map of the environment, while our ultimate goal is to directly align the trajectories of two robotic vehicles that are simultaneously performing an inspection/mapping task in a potentially unknown environment, but unable to directly pass position information between one another.

There has also been significant recent interest in the SLAM and computer vision communities in developing estimation algorithms that leverage convex optimization techniques to avoid the need for an initial guess (Carlone, Censi, and Dellaert, 2014; Li et al., 2014; Mangelson et al., 2018; Rosen, DuHadway, and Leonard, 2015; Saunderson, Parrilo, and Willsky, 2014). This is especially useful in the underwater environment where global positioning system (GPS) is not available and global measurements of position can be hard to come by. In 2014, Li et al. (2014) proposed a method that uses the convex hull of a set of dissimilarity points to estimate the affine transformation that must have occurred in order to transform a set of points observed in one image to a similar set of points observed in another. Their proposed method results in a linear (convex) cost function that can be solved by taking advantage of existing optimization libraries and does not require an initialization. However, to apply their method to the trajectory alignment problem, we need to ensure that the estimated trajectory is rigid as opposed to affine.

Optimization over the group of rigid body transformations is generally non-convex making it hard to guarantee the true optimum. Recent works have investigated convex relaxation based methods for performing optimization over the group of rigid body transformations (the special euclidean group $\mathrm{SE}(d)$) (Rosen, DuHadway, and Leonard, 2015; Saunderson, Parrilo, and Willsky, 2014). Specifically, these methods relax optimization over the set of valid rotation matrices $\mathrm{SO}(d)$ to an optimization over the convex hull of $\mathrm{SO}(d)$. These methods work well in many cases. However, they fail to enforce that the estimated transformation be a valid rigid body transformation. The optimization problem used in our method also takes advantage of convex relaxation techniques, however, we use a linear approximation of the $\ell 2$-norm to add an additional set of constraints to the optimization that collectively enforce that the estimated transformation be approximately rigid. This results in more accurate

transformation estimates than optimization over the convex hull of $\mathrm{SO}(d)$.

3.3 Problem Formulation

In this section, we outline the need for communication constrained trajectory alignment in underwater inspection and then formalize the trajectory alignment problem.

3.3.1 Trajectory Alignment w/o Data Association

In multi-agent inspection tasks, multiple vehicles navigate through the environment collecting information and estimating a map of the structure or scene they are inspecting. Some form of these local maps are then transmitted between vehicles allowing the agents to use the data collected by other vehicles for navigation, path planning, or global map generation. However, before an agent can use the data collected by another agent, it must first determine an alignment between its own local trajectory/map and the trajectory/map received from the other agent. Traditionally, this alignment is determined by matching locations in the environment that are observed by both agents (Carlone et al., 2010; Howard, Matark, and Sukhatme, 2002; Zhou and Roumeliotis, 2006).

However, communicating large amounts of data such as point clouds or high-dimensional image feature vectors between vehicles is often not practical in the underwater domain. While one approach is to limit data transfer by prioritizing data that is most likely to be useful, our approach is to cut out the transfer of this high-dimensional data completely. Instead, we take a discretized version of the robot trajectory and summarize information observed near each individual robot position using a small low-dimensional feature vector. We then align the robot trajectories by trying to find a rigid body transformation that places poses with similar descriptions near one another, without trying to match individual features. Taking this approach allows us to limit the data that must be transferred between vehicles to the discretized set of positions and the associated set of low-dimensional feature vectors.

3.3.2 Convex Trajectory Alignment

Formally, our goal is to align a query trajectory with a reference trajectory based only on low dimensional feature vectors describing the environment at each position visited by the two trajectories.

We denote the positions visited by the reference trajectory by $\{\mathbf{p}_1^a, \cdots, \mathbf{p}_{n_a}^a\}$ and the associated feature vectors by $\{\eta_1^a, \cdots, \eta_{n_a}^a\}$, indexed by i. Similarly, the query trajectory positions and feature vectors are denoted by $\{\mathbf{p}_1^b, \cdots, \mathbf{p}_{n_b}^b\}$ and $\{\eta_1^b, \cdots, \eta_{n_b}^b\}$, respectively,

indexed by j. We then frame the trajectory alignment problem as an optimization that seeks to find a transformation that transforms positions in the coordinate frame of the query trajectory into the coordinate frame of the reference trajectory such that when the points $\{\mathbf{p}_1^b, \cdots, \mathbf{p}_{n_b}^b\}$ are transformed, they lie nearby points in $\{\mathbf{p}_1^a, \cdots, \mathbf{p}_{n_a}^a\}$ with similar feature values. Thus, the feature vectors η only need to describe the local environment as opposed to uniquely identifying a specific point and, as a result, can have a much lower dimension.

Using a formulation similar to (Li et al., 2014), we define a transformation function $T_j^{ab}(\mathbf{\Theta}) : \mathbb{R}^n \mapsto \mathbb{R}^d$ that maps the j-th query point, $\mathbf{p}_j^b$, to a position represented with respect to the reference trajectory coordinate frame. Specifically, we define $T_j^{ab}(\mathbf{\Theta})$ as

$$T_j^{ab}(\mathbf{\Theta}) = \mathbf{R}^{ab}\mathbf{p}_j^b + \mathbf{t}^{ab}, \tag{3.1}$$

where $\mathbf{\Theta} = (\mathbf{R}^{ab}, \mathbf{t}^{ab})$ are the parameters of the function. Together, $\mathbf{R}^{ab} \in \mathrm{SO}(d)$ and $\mathbf{t}^{ab} \in \mathbb{R}^d$ parameterize a global rigid body transformation relating the local coordinate frames of the two vehicles. The points $\mathbf{p}_j^b$ are fixed in the function T_j^{ab} and we thus define n_b transformation functions, one for each feature in the query trajectory.

We also define a function $c_j : \mathbb{R}^d \mapsto \mathbb{R}$ that takes a position represented in the reference trajectory coordinate frame, $\mathbf{p}^a \in \mathbb{R}^d$, and calculates the feature dissimilarity between η_j^b and the given position. As before, because there are n_b query feature points, we define n_b different dissimilarity functions $c_j, j = 1, \cdots, n_b$. Section 3.4 provides more detail on how these functions are defined.

With these definitions, we can formulate the final overall objective function as follows:

$$\underset{\substack{\mathbf{R}^{ab}\in\mathrm{SO}(d)\\ \mathbf{t}^{ab}\in\mathbb{R}^d}}{\text{minimize}} \sum_{j=1}^{n_b} c_j(T_j^{ab}(\mathbf{\Theta})) \tag{3.2}$$

where $c_j(T_j^{ab}(\mathbf{\Theta}))$ is the dissimilarity between the feature vector η_j^b and its new transformed position in the reference trajectory coordinate frame. Solving this optimization problem would allow us to find the transformation that minimizes the dissimilarity between query feature points and their associated positions in the reference trajectory coordinate frame.

Note, that while our formulation is based on that of (Li et al., 2014), their method assumes the transformation is affine, while we restrict it to be an isometric (or rigid body (Hartley and Zisserman, 2003)) transformation. This assumption on their part, results in the associated terms of their cost function being affine and the resulting optimization problem being convex. However, when dealing with physical transformations between coordinate frames, an affine transformation does not represent reality and a rigid body transformation must be used. This makes our resulting optimization problem non-convex and more difficult

to solve. In Section 3.5, we explain a novel method to approximately optimize over SO(2).

3.4 Convex Transformation Estimation via the Lower Convex Hull

In this section, we discuss the formulation of the dissimilarity functions $c_j,\ j = 1, \cdots, n_b$ that measure the feature dissimilarity of the feature vector η_j^b and a position in the reference trajectory coordinate frame. We then compose the dissimilarity functions c_j with the transformation function (3.1) so that we can optimize over the transformation parameters as opposed to individual pose positions. There are two cases that we cover:

1. The 2D case where both maps lie in a 2D world($d = 2$)
2. The known depth case where both maps lie in the 3D world($d = 3$), but one coordinate is known.

We first cover the 2D case which follows directly from (Li et al., 2014). We then generalize this to the case with known depth.

3.4.1 Convex Dissimilarity Function Definition in 2D

Following, Li et al. (2014), we denote the feature dissimilarity between η_i^a and η_j^b by C_{ij}^{ab}. If we use every possibly pairing there are $n_a \times n_b$ of these values and they can be calculated before performing registration via an arbitrary dissimilarity function. However, because we want to find a transformation as opposed to a discrete matching, we create n_b continuous and convex feature dissimilarity functions $c_j,\ j = 1, \cdots, n_b$. Each c_j takes a position represented with respect to the reference trajectory coordinate frame and returns a predicted lower bound on the dissimilarity of that position with respect to the feature vector η_j^b. We derive this function by taking the lower convex hull of the following point cloud:

$$\begin{bmatrix} x_1^a & y_1^a & C_{1j}^{ab} \\ x_2^a & y_2^a & C_{2j}^{ab} \\ \vdots & \vdots & \vdots \\ x_{n_a}^a & y_{n_a}^a & C_{n_a j}^{ab} \end{bmatrix}. \tag{3.3}$$

For a given feature vector η_j^b with $d = 2$, the point cloud (3.3) defines discrete points in a space that relates 2D positions in the reference trajectory coordinate frame to their associated dissimilarity value with respect to η_j^b. Because this space is 3D, we can calculate the

convex hull of these points using an algorithm like the one proposed by Barber, Dobkin, and Huhdanpaa (1996). The lower convex hull with respect to the dissimilarity value dimension is made up of a set of planes (facets) that represent lower bounds on the discrete dissimilarity values C_{ij}^{ab} for a given fixed j. These planes can be found by selecting the planes that have normal vectors with a negative component in the dimension corresponding to dissimilarity. Assuming there are M_j planes in the lower convex hull for feature η_j^b, we can define these planes using the equations $a_m x + b_m y + c_m C_m + d_m = 0$, for $m = 1, \cdots, M_j$, where x and y correspond to the position dimensions and C_m corresponds to the dissimilarity dimension in the point cloud (3.3). We then rearrange these equations to arrive at the plane functions $C_m = r_m x + s_m y + t_m$, for $m = 1, \cdots, M_j$ that calculate the predicted dissimilarity given a specified feature position. Finally, we can now define the feature dissimilarity function c_j as

$$c_j([x, y]^\top) = \max_m (r_m x + s_m y + t_m), \quad m = 1, \cdots, M_j. \tag{3.4}$$

This function is both continuous and convex and its minimization can be transformed into an equivalent linear program (Boyd and Vandenberghe, 2004):

$$\begin{aligned} \underset{u_j, x, y}{\text{minimize}} \quad & u_j \\ \text{subject to} \quad & r_m x + s_m y + t_m \leq u_j, \\ & m = 1, \cdots, M_j. \end{aligned} \tag{3.5}$$

Using the dissimilarity function (3.4) and the linear program (3.5) enables us to efficiently solve for an arbitrary 2D position $[x, y]^\top \in \mathbb{R}^2$ that minimizes the dissimilarity with respect to η_j^b for a given value of j. The next section describes how to compose $c_j(\,\cdot\,)$ with the transformation $T_j^{ab}(\boldsymbol{\Theta})$ which enables us to optimize over the parameters $\boldsymbol{\Theta} = (\mathbf{R}^{ab}, \mathbf{t}^{ab})$.

3.4.2 Composition with the Transformation Function

Our goal is to estimate the rigid body transformation between respective trajectories. As such, rather than estimate the individual locations of points, we estimate the parameters of the transformation, $\boldsymbol{\Theta}$, or more specifically the parameters $\mathbf{R}^{ab}$ and $\mathbf{t}^{ab}$. We do this by minimizing $c_j(T_j^{ab}(\boldsymbol{\Theta}))$ for all $j = 1, \cdots, n_b$, as opposed to $c_j([x, y]^\top)$ directly.

Remembering that $T_j^{ab}(\boldsymbol{\Theta}) \in \mathbb{R}^d$ represents the transformed coordinates of $\mathbf{p}_j^b$, we represent the function that calculates the first coordinate of $T_j^{ab}(\boldsymbol{\Theta})$ by $f_j(\boldsymbol{\Theta})$ and the second coordinate by $g_j(\boldsymbol{\Theta})$. Specifically, define

$$f_j(\boldsymbol{\Theta}) = \mathbf{R}^{ab(1)} \mathbf{p}_j^b + \mathbf{t}^{ab(1)} \tag{3.6}$$

and

$$g_j(\mathbf{\Theta}) = \mathbf{R}^{ab(2)}\mathbf{p}_j^b + \mathbf{t}^{ab(2)}, \tag{3.7}$$

where the notation $^{(i)}$ denotes the i-th row of the given matrix or vector and, again, $\mathbf{p}_j^b$ is fixed for both f_j and g_j.

Using this, we can rewrite (3.5) to be a minimization of $c_j(T_j^{ab}(\mathbf{\Theta}))$ over $\mathbf{\Theta}$ as

$$\begin{aligned} \underset{\substack{u_j\in\mathbb{R} \\ \mathbf{R}^{ab}\in\mathrm{SO}(2) \\ \mathbf{t}^{ab}\in\mathbb{R}^2}}{\text{minimize}} \quad & u_j \\ \text{subject to} \quad & r_m f_j(\mathbf{\Theta}) + s_m g_j(\mathbf{\Theta}) + t_m - u_j \leq 0, \\ & \qquad\qquad m = 1, \cdots, M_j. \end{aligned} \tag{3.8}$$

If the elements of the matrix $\mathbf{R}^{ab}$ are treated as individual elements in $\mathbf{\Theta}$, then both $f_j(\mathbf{\Theta})$ and $g_j(\mathbf{\Theta})$ are affine functions of $\mathbf{\Theta}$, and $r_m f_j(\mathbf{\Theta}) + s_m g_j(\mathbf{\Theta}) + t_m - u_j$ is also an affine function of $\mathbf{\Theta}$ and u_j. This was noted in (Li et al., 2014). However, in our case, the resulting problem (3.8) is non-convex because of the implicit constraint that $\mathbf{R}^{ab} \in \mathrm{SO}(2)$. Section 3.5 proposes a solution to this problem.

The next section explains how this can be generalized to three dimensions when depth is known.

3.4.3 The Known Depth Case

The prior section is defined with a planar world in mind. However, the real world exists in three dimensions. Although the prior section can be generalized to three dimensions, in the underwater domain we can take advantage of the fact that we can accurately measure depth and just estimate translation in the xy-plane and rotation about the z (vertical) axis. Under these assumptions, we can restrict T_j^{ab} to the following:

$$T_j^{ab}(\mathbf{\Theta}) = \begin{bmatrix} \mathbf{R}_z^{ab} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{p}_j^b + \begin{bmatrix} \mathbf{t}_z^{ab} \\ 0 \end{bmatrix}, \tag{3.9}$$

with $\mathbf{R}_z^{ab} \in \mathrm{SO}(2)$, $\mathbf{t}_z^{ab} \in \mathbb{R}^2$, and $\mathbf{p}_j^b \in \mathbb{R}^3$. This restriction simplifies the problem and enables us to work on SO(2) as opposed to SO(3) when relaxing the optimization problem.

In addition, if the feature value varies with depth, then we can limit the reference trajectory feature points that need to be paired with each query feature η_j^b to those with a depth within a threshold γ of the estimated depth of $\mathbf{p}_j^b$. This allows us to decrease the size of the point cloud (3.3) and tighten the lower bounds defined by the planes in the lower convex

hull.

3.5 Ensuring the Transformation is Rigid

The ultimate optimization problem we want to solve, (3.2), is non-convex because of the implicit constraint that $\mathbf{R}^{ab}$ (or equivalently $\mathbf{R}_z^{ab}$) $\in \mathrm{SO}(d)$. This makes it difficult to ensure that the solution obtained is globally optimal without a good initialization. However, this constraint is essential because it ensures that the estimated transformation is rigid as opposed to affine. A general affine transformation allows scaling and distortions of the object in addition to rotation and translation. Physical rigid body transformations on the other hand preserve distance between points and thus do not allow scaling or distortions (Hartley and Zisserman, 2003).

3.5.1 Definition of $SO(2)$ and $\operatorname{conv}\mathrm{SO}(2)$

Formally, $\mathrm{SO}(d)$ is defined as follows:

$$\mathrm{SO}(d) = \left\{\mathbf{R} \in \mathbb{R}^{d\times d} : \mathbf{R}^\top\mathbf{R} = \mathbf{R}\mathbf{R}^\top = \mathbf{I}, \det\mathbf{R} = 1\right\}. \tag{3.10}$$

In the two dimensional case, an alternative definition for (3.10) is:

$$\mathrm{SO}(2) = \left\{\mathbf{R} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \in \mathbb{R}^{2\times 2} : c^2 + s^2 = 1\right\}. \tag{3.11}$$

A variety of recent papers have investigated the use of the convex hull of $\mathrm{SO}(d)$ (Rosen, DuHadway, and Leonard, 2015; Saunderson, Parrilo, and Willsky, 2014):

$$\operatorname{conv}\mathrm{SO}(2) = \left\{\tilde{\mathbf{R}} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \in \mathbb{R}^{2\times 2} : \begin{bmatrix} 1+c & s \\ s & 1-c \end{bmatrix} \succeq \mathbf{0}\right\}. \tag{3.12}$$

These methods relax optimization over $\mathrm{SO}(d)$ to optimization over the smallest convex set of matrices containing it. In this case the set of valid rigid body transformations lies on the border of the convex hull. However, if the minimum of the cost function lies within that convex hull as opposed to outside it or on its border, then these methods still return a transformation that scales and/or distorts the transformed trajectory. While rounding to the nearest rigid body transformation is possible, if the estimated transformation is no where near valid then the rounded solution tends to be inaccurate.

Instead of using the convex hull of $\mathrm{SO}(2)$, we break the problem up into linear subproblems, within which we can use a linear approximation of the $\ell 2$ norm (Barni et al.,

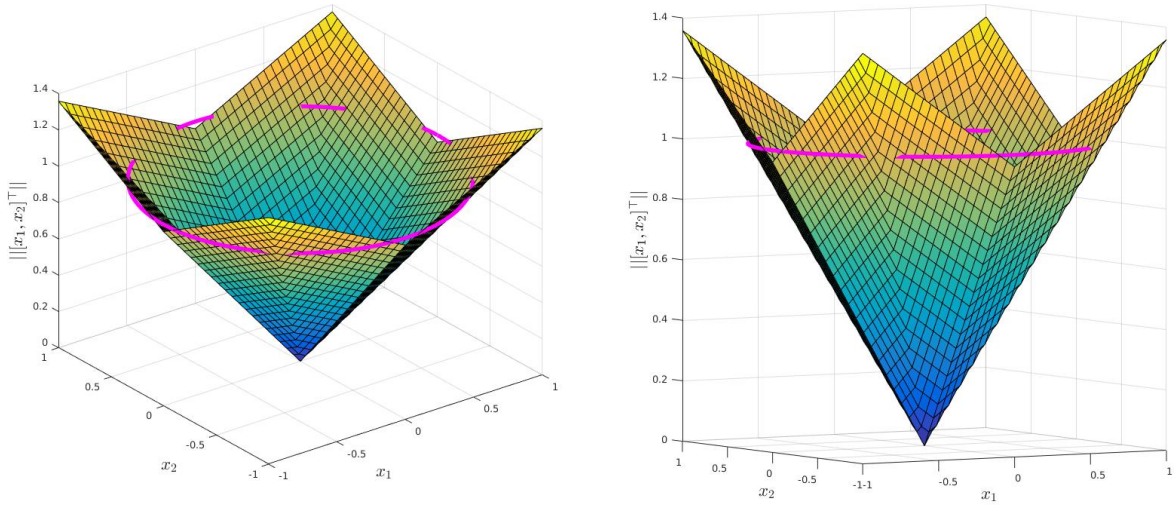


Figure 3.2: Linear Approximation of $\ell 2$. The planes in these plots represent a piece-wise linear approximation of the $\ell 2$-norm of $\mathbf{x} = [x_1, x_2]^\top$ (3.16). The pink circle represents where the $\ell 2$-norm is one, $||\mathbf{x}||_2 = x_1^2 + x_2^2 = 1$.

2000) to enforce that $c^2 + s^2 \approx 1$ and thus that the transformation be approximately rigid.

3.5.2 Linear Approximation of $\ell 2$

Celebi, Celiker, and Kingravi (2011) evaluate several potential linear approximations of the $\ell 2$-norm. According to their evaluation, the approximation presented by Barni et al. (2000) has the lowest maximum error. This approximation states that given a vector $\mathbf{x} = [x_1, x_2, \cdots, x_n]^\top \in \mathbb{R}^n$,

$$||\mathbf{x}||_2 \approx \delta^* \sum_{i=1}^{n} \alpha_i^* x_{(i)}, \tag{3.13}$$

where $(x_{(1)}, x_{(2)}, \cdots, x_{(n)})$ is a permutation of $(|x_1|, |x_2|, \cdots, |x_n|)$ such that $x_{(1)} \geq x_{(2)} \geq \cdots \geq x_{(n)}$, and δ, α_i are parameters optimally given by (See (20) and (21) in (Barni et al., 2000)):

$$\alpha_i^* = \sqrt{i} - \sqrt{i-1}, \quad \delta^* = \frac{2}{1 + \sqrt{\sum_{i=1}^{n} \alpha_i^{*2}}}. \tag{3.14}$$

In the two dimensional case, $\alpha_1^* = 1$, $\alpha_2^* = \sqrt{2} - 1$, and $\delta^* = \frac{2}{1+\sqrt{1+(\sqrt{2}-1)^2}} \approx 0.96044$. In addition, in the two dimensional case, the sorting of $(|x_1|, |x_2|)$ can be implemented via a

maximization term, and (3.13) can be rewritten as

$$||\mathbf{x}||_2 \approx \delta\alpha_1 \max(|x_1|, |x_2|) \; + \\ \delta\alpha_2(|x_1| + |x_2| - \max(|x_1|, |x_2|). \tag{3.15}$$

Implementing max and absolute value in convex optimization problems is not always possible. Instead of actually evaluating the max and absolute values, we can enumerate the possible values and rewrite (3.15) as:

$$\begin{aligned} ||\mathbf{x}||_2 \approx \max(&\delta(\alpha_1 x_1 + \alpha_2 x_2), \\ &\delta(\alpha_1(-x_1) + \alpha_2 x_2), \\ &\delta(\alpha_1(-x_1) + \alpha_2(-x_2)), \\ &\delta(\alpha_1 x_1 + \alpha_2(-x_2)), \\ &\delta(\alpha_1 x_2 + \alpha_2 x_1), \\ &\delta(\alpha_1(-x_2) + \alpha_2 x_1), \\ &\delta(\alpha_1(-x_2) + \alpha_2(-x_1)), \\ &\delta(\alpha_1 x_2 + \alpha_2(-x_1))) \end{aligned} \tag{3.16}$$

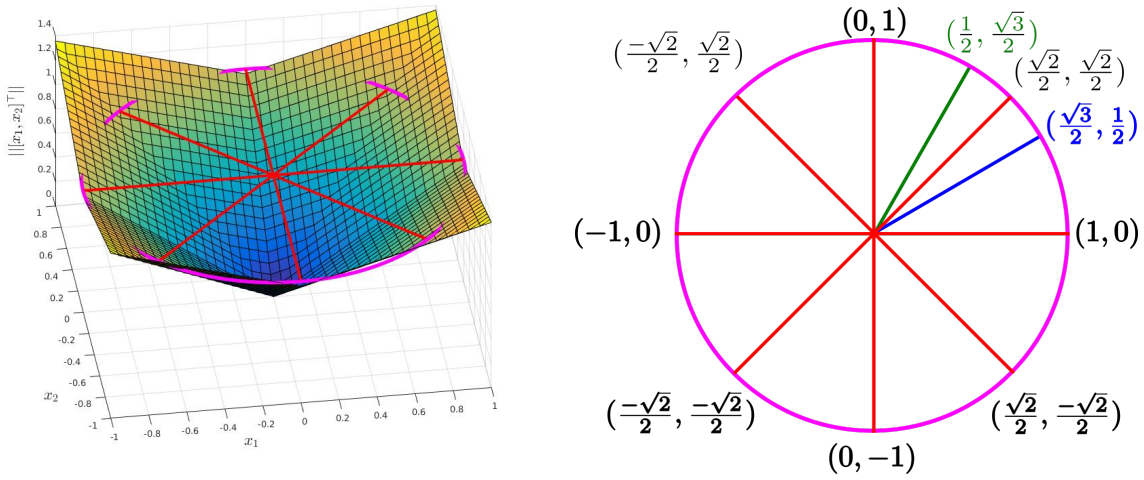
This function is shown plotted in Fig. 3.2.

3.5.3 Breaking the Optimization into Linear Sub-Problems

Using (3.16) to enforce unit norm is still difficult because enforcing the max operation requires simultaneous minimization and maximization; instead, we split the problem into eight sub-problems in a way that enables us to ignore the max operation. Each of the eight terms in (3.16) correspond to a single plane in Fig. 3.2. Additionally, each term is maximal only when the signs and relative values of x_1 and x_2 meet certain conditions. These conditions correspond to sections of the unit circle (Fig. 3.3(b)) and can be enforced with linear constraints on x_1 and x_2.

Specifically, we can enforce that $\mathbf{x}$ lie within any given section by adding the corresponding choice of the following constraints to the optimization problem:

$$\begin{aligned} x_1 \leq x_2 \quad &\text{or} \quad x_2 \leq x_1 \\ x_1 \leq 0 \quad &\text{or} \quad x_1 \geq 0 \\ x_2 \leq 0 \quad &\text{or} \quad x_2 \geq 0. \end{aligned} \tag{3.17}$$



(a) Division into Sections

(b) Unit Circle

Figure 3.3: Problem Division into Eight Sub-problems. By dividing the optimization into eight sub-problems, we no longer need to execute the max operation in (3.16) and each sub-problem then becomes convex. The red lines in (a) demonstrate where this division takes place. These divisions correspond to the unit circle (b) and the enumerated sign/max combinations in (3.16).

Within a given section, only a single plane is maximal and the approximation of $\ell 2$ becomes linear (Fig. 3.3(a)). Thus, we can enforce that our solution be close to unit norm by adding the following linear constraint to the problem:

$$\delta(\alpha_1 x_p + \alpha_2 x_q) = 1 \tag{3.18}$$

where x_p and x_q represent the appropriately selected elements of $\{x_1, -x_1, x_2, -x_2\}$, such that they match the respective maximal term in (3.16). The maximum error of this approximation (3.18) is related to, but likely to be slightly higher than ϵ_{max} as defined in (4) and Table 3 of (Celebi, Celiker, and Kingravi, 2011).

Formulating the problem in this way makes each sub-problem a linear program, and so we can take advantage of existing efficient optimization libraries to solve each sub-problem to a global minimum (ApS, 2018). In addition, because the sub-problems divide the space and optimize over the same linear cost function, the solution to the sub-problem with lowest optimal value is identical to what the solution would be if we were able to constrain (3.16) to be equal to 1. Finally, the sub-problems are independent and thus can be parallelized.

3.6 The Full System

We are now able to create a general system for aligning robot trajectories.

3.6.1 The Final Optimization Problem

We can rewrite the optimization problem (3.2), by combining (3.8), (3.17), and (3.18).

$$
\begin{aligned}
\underset{\substack{u_j \in \mathbb{R} \\ c,s \in \mathbb{R} \\ \mathbf{t}^{ab} \in \mathbb{R}^2}}{\text{minimize}} \quad & \sum_{j=1}^{n_b} u_j \\
\text{subject to} \quad & r_{m_j} f_j(\mathbf{\Theta}) + s_{m_j} g_j(\mathbf{\Theta}) + t_{m_j} - u_j \leq 0, \\
& \qquad m_j = 1, \cdots, M_j, \quad j = 1, \cdots, n_b \\
& \mathbf{R}^{ab} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \\
& c \leq s \quad \text{or} \quad s \leq c \\
& c \leq 0 \quad \text{or} \quad c \geq 0 \\
& s \leq 0 \quad \text{or} \quad s \geq 0 \\
& \delta(\alpha_1 x_p + \alpha_2 x_q) = 1,
\end{aligned}
\tag{3.19}
$$

where the inequality constraints and x_p and x_q are chosen according to the specific sub-problem.

Enumerating the possible combinations of these constraints results in eight sub-problems. Solving all eight sub-problems and then selecting the solution with lowest cost enables us to estimate a transformation to align the two trajectories without an initial estimate of alignment. In addition, this method enables us to avoid performing data association and thus perform alignment based on relatively indistinct, low-dimensional, feature observations. We then iterate over this process with consecutively smaller regions of interest, as explained in Section 4 of (Li et al., 2014), to decrease the number of reference trajectory points used to create the cost function and thus increase accuracy.

3.6.2 Parallel and Feature Agnostic Implementation

We implemented the proposed system in c++. The released implementation uses MOSEK (ApS, 2018) and multiple threads to efficiently solve the linear sub-problems in parallel. The released code can be found at the following link: https://bitbucket.org/jmangelson/cte.

Table 3.1: Summary Statistics for Simulated Alignment Experiment. Comparison of the proposed alignment algorithm with other convex algorithms that use the same alignment framework but enforce structure differently: SOConvHull (Saunderson, Parrilo, and Willsky, 2014), Symmetric (Li et al., 2014), Affine (Li et al., 2014). Standard deviation shown is after removing outliers outside 1.5*IQR, IQR=Inter-Quartile-Range.

	Proposed	SOConvHull	Symmetric	Affine
Rotation Median SE	**0.011**	0.039	0.042	0.338
Rotation Stddev SE	**0.017**	0.096	0.095	0.498
Trans. Median SE (m^2)	**4.796**	10.974	11.601	32.582
Trans. Stddev SE (m^2)	**6.770**	18.219	20.318	57.370
% Approx. Valid Rotations	**100.0**	37.0	25.5	1.0
Avg Runtime (s)	41.444	29.527	25.416	**23.926**

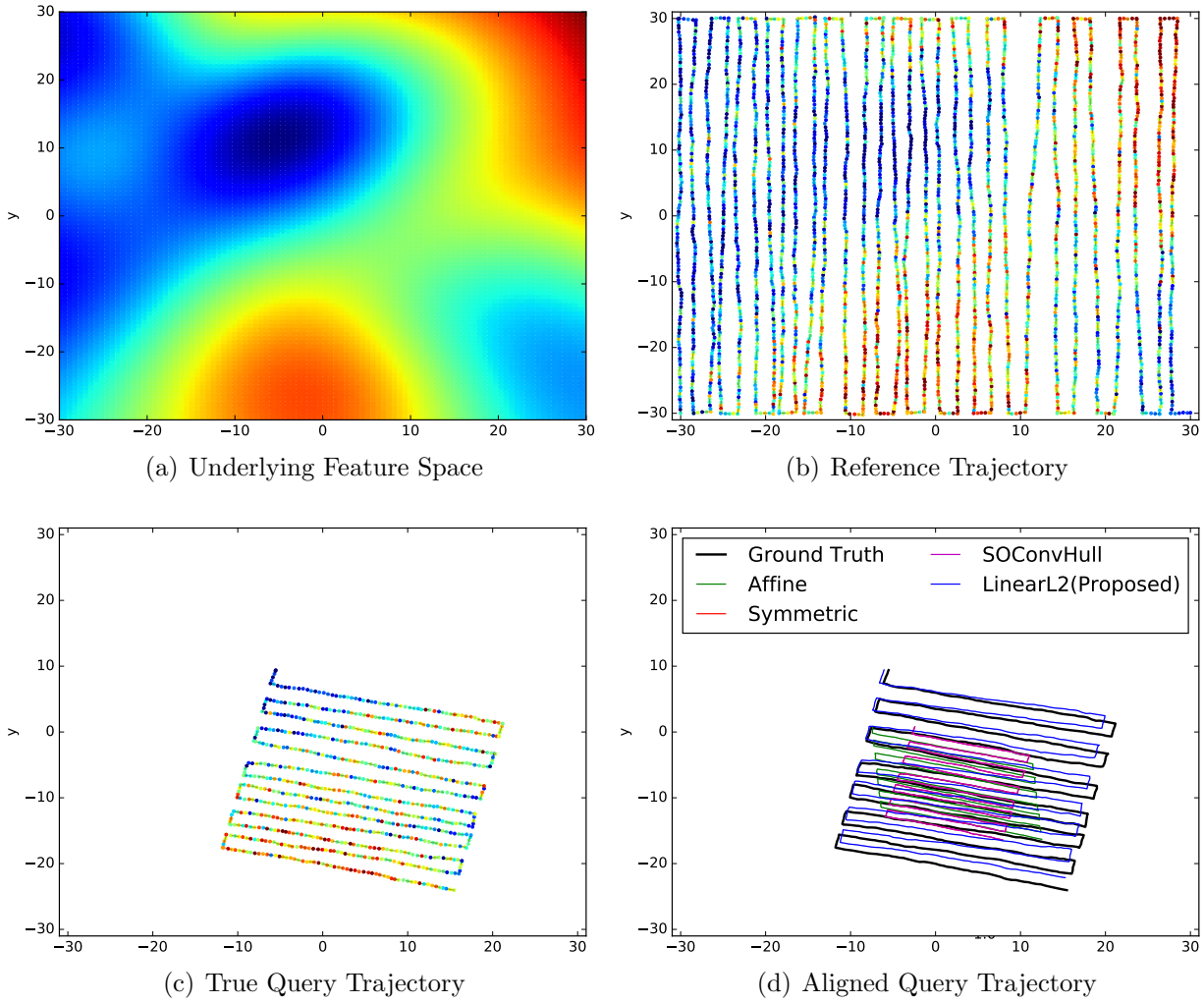
Our implementation and method are agnostic to the underlying feature space. When specifying a problem to be solved, the user provides feature point positions and a function that calculates the feature dissimilarity of a given pair of points. As such, the specifics of the feature space being used and the dissimilarity function are free to be chosen by the user.

In addition to our own proposed method for approximate optimization over SO(2), we also implemented functionality for estimating affine and symmetric transformations (Li et al., 2014), as well as rigid body transformations via conv SO(2) (Saunderson, Parrilo, and Willsky, 2014) for comparison.

3.7 Evaluating our Approximation of SE(2)

To provide quantitative results, we generated 200 synthetic worlds with smoothly varying feature spaces. We then simulated both a reference and a query lawn-mower trajectory within that space and randomly rotated and translated the query trajectory. Finally, we compared the alignment results from a variety of convex alignment methods formulated as in Section 3.4 but with different types of transformation constraints. Specifically, we compared our proposed linear programming-based approach to methods based on (Li et al., 2014) that allowed either affine or symmetric transformations as well as a method that enforced that the estimated transformation lie within the convex hull of SO(2) (Saunderson, Parrilo, and Willsky, 2014). Fig. 3.4 shows a sample simulated trajectory and associated alignment results. The nominal trackline width for this dataset was two meters and the feature vector dimension was three.

Table 3.1 provides a summary of the comparison results. Our proposed method outperforms other methods in all metrics except for runtime. Note that the median translation squared error for the proposed method is 4.796 meters squared while the resolution of the



(a) Underlying Feature Space

(b) Reference Trajectory

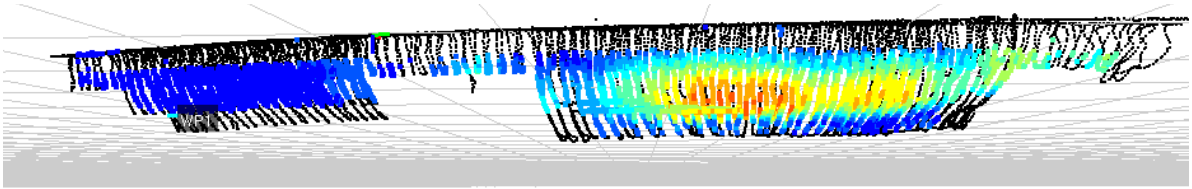
(c) True Query Trajectory

(d) Aligned Query Trajectory

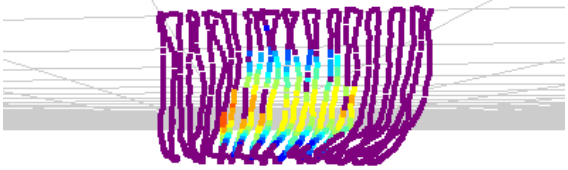
Figure 3.4: Example Simulated Trajectory Alignment. The feature space for a sample simulated trial is shown in (a). The reference and query trajectories are then generated as shown in (b) and (c). To simulate the fact that the relative transformation between trajectories is unknown, we then randomly rotate and translate the query trajectory and use the proposed and evaluated methods to estimate the inverse of the random transformation. The resulting trajectories are shown in (d).

tracklines simulated was two meters, meaning that the median error is only slightly higher than the resolution of the input data.

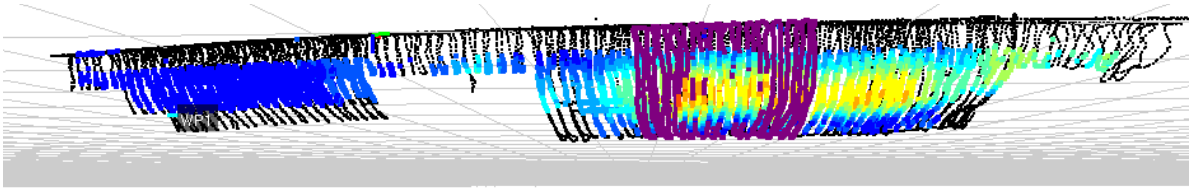
The current implementation uses four threads that independently solve two linear programs each via the optimization library Mosek (ApS, 2018). However, the speed of the algorithm can be additionally improved by increasing the number of cores or by using a faster convex optimization library.



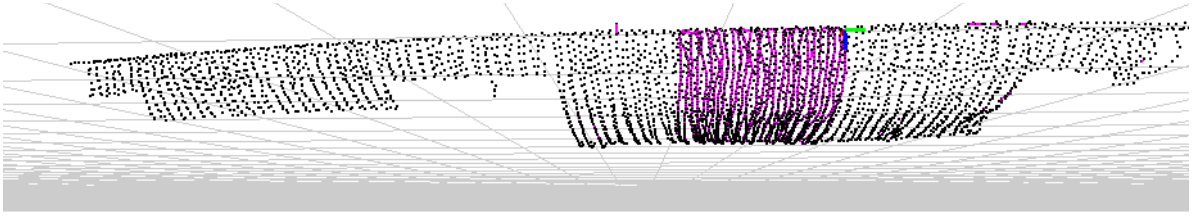
(a) Trajectory 1 Colored by Curvature



(b) Trajectory 2 Colored by Curvature



(c) Alignment Based on Hull Curvature Alone



(d) Ground Truth

Figure 3.5: Ship Hull Trajectories Aligned Based on Curvature. Two trajectories colored by estimated curvature are shown in (a) and (b). The estimated alignment of these two trajectories is shown in (c). The groundtruth alignment is shown in (d). This data was collected during field trials inspecting the hull of the USCGC Spencer in Boston, MA.

3.8 Application to Multi-Agent Autonomous Ship Hull Inspection

We also tested this method on real-world field data collected using a Hovering Autonomous Underwater Vehicle (HAUV) performing autonomous ship hull inspection. We used the sparse range returns of the Doppler velocity log (DVL) to estimate the curvature of the hull. Then, by treating the local curvature of the ship hull as a feature vector, we are able to re-localize to an earlier trajectory using only the DVL. Fig. 3.5 shows an alignment using

this method.

Using the proposed alignment method limits the information that needs to be passed between vehicles to six fixed or floating-point values for each position visited by the agent (including three position coordinates and three curvature feature values), while the throughput needed to transfer image features between agents would be on the order of 1000-10000 fixed or floating-point values per position. Thus, the proposed method results in a decrease in required throughput of approximately 3-4 orders of magnitude.

The final accuracy of the alignment is dependent on the distinctiveness of the features being observed as well as the sampling resolution inherited from the reference trajectory. However, as formulated, the proposed method is independent of the specific feature and thus can be applied to whatever feature set is most distinctive in a given environment, subject to the communication bandwidth available. In addition, alignment can be further refined if desired using higher dimensional data such as imagery once an initial estimate of alignment has been obtained, thus minimizing the amount of image data that must be transferred between agents.

3.9 Conclusion

In this chapter, we propose a method for aligning robot trajectories that is linear and thus does not require initialization. In addition, the proposed method aligns trajectories without performing data association which decreases the amount of information that must be transferred between agents. We compared the existing method to similar convex methods that fail to enforce that the estimated transformation be rigid. We also applied the proposed algorithm to localization in the context of multi-agent autonomous ship hull inspection.

Future work would include: extending the proposed ideas to three dimensions and relaxing the assumption that the query trajectory be contained within the convex hull of the reference trajectory.

The completed work was published in:

J. G. Mangelson, R. Vasudevan, and R. M. Eustice. Communication constrained trajectory alignment for multi-agent inspection via linear programming. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–7, Charleston, SC, USA, October 2018

A C++ implementation is released at: `https://bitbucket.org/jmangelson/cte`.

CHAPTER 4

Guaranteed Globally Optimal Planar Pose Graph and Landmark SLAM

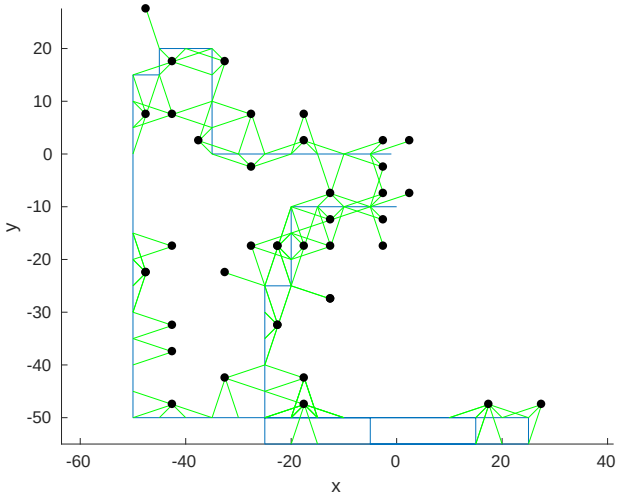
4.1 Introduction

An accurate map of the environment is essential for safe autonomous navigation in the real-world (Cadena et al., 2016). An error in the map has the potential to cause loss of life in self-driving car applications or the loss of millions of dollars of assets when performing underwater or space exploration tasks. Despite the importance of accurate mapping, the majority of algorithms used for simultaneous localization and mapping (SLAM) are prone to local minima and are sensitive to initialization. Troublingly, verification of these maps is either performed by visual inspection or not at all.

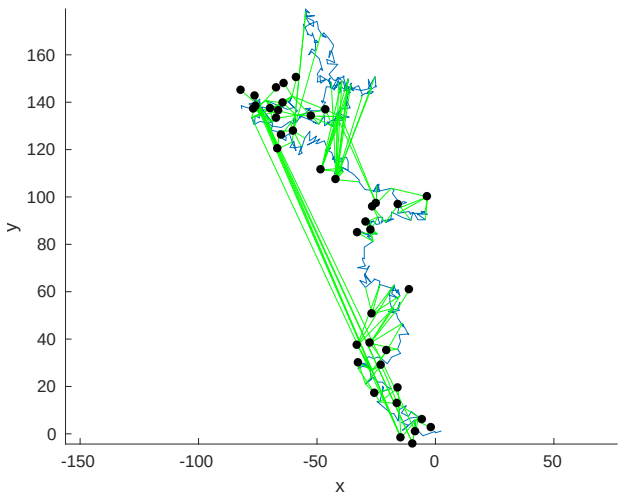
There has been significant recent interest in developing optimization and estimation algorithms that provide mathematical guarantees on whether a computed solution is or is close to the global optimum and is therefore a true maximum *a posteriori* (MAP) estimate of the map (Carlone et al., 2015, 2016; Hu, Khosoussi, and Huang, 2013; Liu et al., 2012; Rosen et al., 2016; Rosen, DuHadway, and Leonard, 2015). These algorithms either use a relaxation or the dual of the original problem to find a solution. As a result, these methods either return an approximate solution, are only able to certify the optimally of a solution after it has been computed, or are only able to return the global solution if the graph meets certain requirements related to limits on measurement noise. In addition, with the exception of Hu, Khosoussi, and Huang (2013), these methods are focused on pose-graph optimization and are unable to handle landmark position measurements or are unable to estimate landmark positions.

As depicted in Figure 4.1(d), the contributions of this paper are the following:

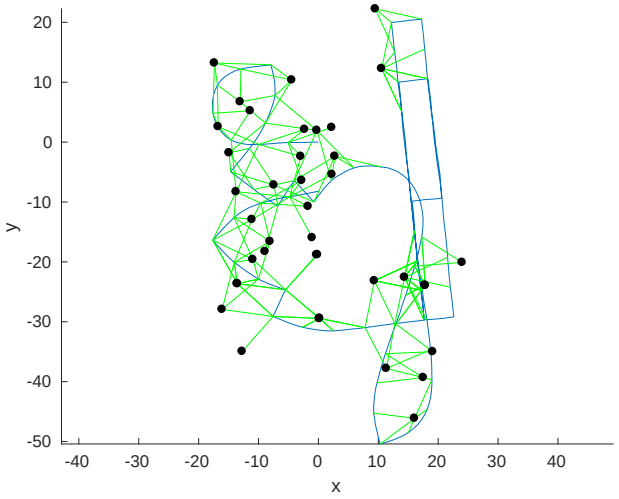
1. We formulate the pose graph and landmark planar SLAM as polynomial optimization programs.



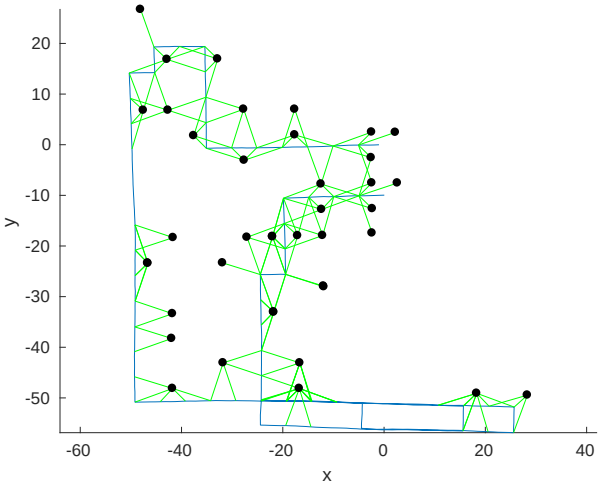
(a) Ground Truth - No Noise



(b) Random Initialization



(c) Levenberg-Marquardt



(d) Global Optimal - SBSOS-SLAM

Figure 4.1: SBSOS-SLAM vs LM on CityTrees10000 430 Node Segment. Estimated Landmark SLAM solution for the first 430 nodes (40 landmarks) of the CityTrees10000 dataset (Kaess, Ranganathan, and Dellaert, 2008). (a) shows the groundtruth pose and landmark positions before noise is added. Levenberg-Marquardt was randomly initialized and becomes trapped in a local minimum. (b) shows the random initialization and (c) shows the LM solution. (d) shows the globally optimal solution found by SBSOS-SLAM. Our algorithm formulates the Pose Graph and Landmark SLAM problems as SOS-convex optimization problems and is guaranteed to find the globally optimal solution without any initialization.

2. We describe how the sparse bounded degree sum-of-squares (Sparse-BSOS) hierarchy of semidefinite programs (SDP) can be used to find its solution (Lasserre, Toh, and Yang, 2017; Weisser, Lasserre, and Toh, 2018).

3. We show that both the pose-graph and landmark SLAM problems are SOS convex meaning that they have a single globally optimal solution that can be found exactly by solving the first step of the Sparse-BSOS hierarchy.

4.2 Related Work

SLAM refers to the problem of estimating the trajectory of a robotic vehicle over time while simultaneously estimating a model of the surrounding environment (Cadena et al., 2016). Initial algorithms used extended Kalman filter and particle filter based methods to simultaneously estimate the position of the robot and the position of observed landmarks in the environment (Bailey and Durrant-Whyte, 2006; Durrant-Whyte and Bailey, 2006; Thrun, Burgard, and Fox, 2005), which we refer to as the Landmark SLAM problem. Since these methods had challenges scaling to larger datasets, researchers began applying information filter and maximum likelihood estimate (MLE) based methods which could exploit sparsity to solve larger instances of the SLAM problem. To improve the sparsity of the problem, research shifted to solving the Pose Graph SLAM problem wherein the landmarks are marginalized out and only the pose of the robot is optimized over at each time step. The majority of modern SLAM algorithms seek to find the MLE of the robot trajectory through the use of nonlinear estimation based techniques (Dellaert and Kaess, 2006; Eustice, Walter, and Leonard, 2005; Kaess, Ranganathan, and Dellaert, 2008; Lu and Milios, 1997). However, the non-linear optimization algorithms used in these methods are dependent on initialization.

Several algorithms leverage theory from the field of convex optimization to overcome this dependence on initialization (Boyd and Vandenberghe, 2004). Optimization over the special euclidean group ($\mathrm{SE}(d)$) has generally been considered a non-convex problem and thus the majority of algorithms rely on some form of convex relaxation to estimate an approximate and sometimes exact solution to the problem. For instance, the Pose Graph and Landmark SLAM problems have been formulated as a non-convex quadratically constrained quadratic program, which was then relaxed into an SDP (Hu, Khosoussi, and Huang, 2013; Liu et al., 2012). Rosen, DuHadway, and Leonard (2015) relaxes optimization over the special orthogonal group ($\mathrm{SO}(d)$) to the convex hull of $\mathrm{SO}(d)$ which can be represented using convex semidefinite constraints. Since each of these methods only provide an approximate solution to the SLAM problem, they are usually only used as an initial stage and their output is then used to initialize a non-linear optimization method (Dellaert and Kaess, 2006; Kaess, Ranganathan, and Dellaert, 2008).

A number of methods take advantage of Lagrangian Duality to convert the Pose Graph SLAM problem into a convex optimization problem that is equivalent to the original opti-

mization problem if the duality gap is zero (Boyd and Vandenberghe, 2004, Section 5.3.2). Carlone et al. (2015) uses Lagrangian Duality to develop a pair of methods to verify if a computed solution is globally optimal. Carlone et al. (2016) applies a similar technique to the planar Pose Graph SLAM problem. SE-Sync, proposed by Rosen et al. (2016), extends this prior work and dramatically increases the scalability of the algorithm by taking advantage of a technique called the Riemannian staircase (Boumal, 2015) enabling efficient optimization over semidefinite matrices if the solution has low-rank. These methods are only guaranteed to find the globally optimal solution if the measurement noise in the problem lies below a critical threshold and are restricted to the case of Pose Graph SLAM where factors are relative pose measurements in SE(d).

This paper shows that the Landmark and Pose Graph SLAM problems can be formulated as SOS convex polynomial optimization problems which can be optimized to the global minimum (without initialization) through the use of Sparse-BSOS optimization.

The convexity of this formulation also allows us to relax the requirement of limited measurement noise and extend the formulation to a broader class of SLAM problems.

4.3 Polynomial Optimization SLAM Formulation

This section formulates the Pose Graph and Landmark SLAM problems as polynomial optimization programs.

4.3.1 Polynomial Optimization

A polynomial optimization program is an optimization problem of the following form (Lasserre, Toh, and Yang, 2017, Section 2.2):

$$f^* = \min_{\mathbf{x}} \{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}\} \tag{4.1}$$

where $f \in \mathbb{R}[\mathbf{x}]$, $\mathbb{R}[\mathbf{x}]$ is the ring of all possible polynomials in the variable $\mathbf{x} = (x_1, \ldots, x_N)$, and $\mathbf{K} \subset \mathbb{R}^N$ is the semi-algebraic set

$$\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^N : 0 \leq g_j(\mathbf{x}) \leq 1, j = 1, \ldots, M\}, \tag{4.2}$$

for polynomials $g_j \in \mathbb{R}[\mathbf{x}], j = 1, \ldots, M$.

4.3.2 Pose Graph SLAM

In planar Pose Graph SLAM, one estimates the pose of the robot, $(\mathbf{R}_i, \mathbf{t}_i) \in \text{SE}(2)$, with respect to a static global reference frame at each time steps $i \in \{1, \ldots, n\}$, by minimizing the error in a set of m_{Rel} relative pose measurements $(\bar{\mathbf{R}}_{ij}, \bar{\mathbf{t}}_{ij}) \in \text{SE}(2)$. The set of available measurements can be represented by the set of edges, $E = \{i_k, j_k\}_{k=1}^{m_{Rel}} \subset \{1, \ldots, n\} \times \{1, \ldots, n\}$, in the corresponding factor graph. We denote the pose of the robot at time step i by the matrix $\mathbf{H}_i = [\mathbf{R}_i | \mathbf{t}_i]$ and the relative pose measurement that relates the pose of the robot at time steps i and j by $\bar{\mathbf{H}}_{ij} = [\bar{\mathbf{R}}_{ij} | \bar{\mathbf{t}}_{ij}]$. We assume that each $\bar{\mathbf{R}}_{ij}$ and $\bar{\mathbf{t}}_{ij}$ are conditionally independent given the true state, that $\bar{\mathbf{R}}_{ij} \sim \text{Langevin}(\mathbf{R}_{ij}, \omega^2_{\mathbf{R}_{ij}})$, and that $\bar{\mathbf{t}}_{ij} \sim \mathcal{N}(\mathbf{t}_{ij}, \Omega^{-1}_{\mathbf{t}_{ij}})$, where $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$ is the true relative pose, $\omega^2_{\mathbf{R}_{ij}}$ is the concentration parameter of the Langevin Distribution, and $\Omega_{\mathbf{t}_{ij}} = \text{blkdiag}(\omega^2_{x_{ij}}, \omega^2_{y_{ij}})$ is the information matrix of $\bar{\mathbf{t}}_{ij}$. Note $\text{blkdiag}(\omega^2_{x_{ij}}, \omega^2_{y_{ij}})$ denotes a block diagonal matrix whose diagonal elements are equal to $\omega^2_{x_{ij}}$ and $\omega^2_{y_{ij}}$.

Under these assumptions, the MLE solution to the planar pose graph SLAM problem is equivalent to:

$$\underset{\mathbf{H}_1, \ldots, \mathbf{H}_n \in \text{SE}(2)}{\text{argmin}} \sum_{(i,j) \in E} \omega^2_{\mathbf{R}_{ij}} \left|\left|\mathbf{R}_j - \mathbf{R}_i \bar{\mathbf{R}}_{ij}\right|\right|^2_F + \left|\left|\mathbf{t}_j - \mathbf{t}_i - \mathbf{R}_i \bar{\mathbf{t}}_{ij}\right|\right|^2_{\Omega_{\mathbf{t}_{ij}}} \tag{4.3}$$

where $||\cdot||_F$ is the Frobenius norm and $||\mathbf{x}||^2_\Omega = \mathbf{x}^\top \Omega \mathbf{x}$ for $x \in \mathbb{R}^2$ (Briales and Gonzalez-Jimenez, 2017; Rosen et al., 2016).

Note that $\mathbf{R}_i \in \text{SO}(2)$ for each $i \in \{1, \ldots, n\}$. SO(2) can be defined as follows:

$$\text{SO}(2) = \left\{ \mathbf{R} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \in \mathbb{R}^{2\times 2} \;\middle|\; c^2 + s^2 = 1 \right\}. \tag{4.4}$$

This definition allows us to parameterize $\mathbf{H}_i$ and $\mathbf{H}_{ij}$ using (c_i, s_i, x_i, y_i) and $(c_{ij}, s_{ij}, x_{ij}, y_{ij})$ respectively,

$$\mathbf{H}_i = \begin{bmatrix} c_i & -s_i & x_i \\ s_i & c_i & y_i \end{bmatrix}, \mathbf{H}_{ij} = \begin{bmatrix} c_{ij} & -s_{ij} & x_{ij} \\ s_{ij} & c_{ij} & y_{ij} \end{bmatrix}, \tag{4.5}$$

as long as we enforce that $c_i^2 + s_i^2 = 1$. To simplify this notation, we define the sets $\mathbf{c} = \{c_1, \ldots, c_n\}$, $\mathbf{s} = \{s_1, \ldots, s_n\}$, $\mathbf{x} = \{x_1, \ldots, x_n\}$, and $\mathbf{y} = \{y_1, \ldots, y_n\}$.

If we evaluate the norms in (4.3) under this parameterization, then (4.3) is equivalent to

$$\begin{aligned} \underset{\mathbf{c},\mathbf{s},\mathbf{x},\mathbf{y}}{\operatorname{argmin}} & \sum_{(i,j)\in E} f_{ij}^{\mathbf{H}}(c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j) \\ \text{s.t.} \quad & c_i^2 + s_i^2 = 1, \quad \forall i \in \{1, \ldots, n\}, \end{aligned} \tag{4.6}$$

with,

$$\begin{aligned} f_{ij}^{\mathbf{H}}(c_i,s_i, x_i, y_i, c_j, s_j, x_j, y_j) = f_{ij}^{Rot}(c_i, s_i, c_j, s_j) + \\ + f_{ij}^{Tran}(c_i, s_i, x_i, y_i, x_j, y_j), \end{aligned} \tag{4.7}$$

where

$$f_{ij}^{Rot}(c_i, s_i, c_j, s_j) = \omega_{\mathbf{R}_{ij}}^2 (c_j - c_i c_{ij} + s_i s_{ij})^2 + \tag{4.8}$$

$$\begin{aligned} & + \omega_{\mathbf{R}_{ij}}^2 (-s_j + c_i s_{ij} + s_i c_{ij})^2 + \omega_{\mathbf{R}_{ij}}^2 (s_j - s_i c_{ij} + \\ & - c_i s_{ij})^2 + \omega_{\mathbf{R}_{ij}}^2 (c_j + s_i s_{ij} - c_i c_{ij})^2, \end{aligned} \tag{4.9}$$

and

$$\begin{aligned} f_{ij}^{Tran}(c_i,s_i, x_i, y_i, x_j, y_j) = \omega_{x_{ij}}^2 (x_j - c_i x_{ij} + \\ + s_i y_{ij} - x_i)^2 + \omega_{y_{ij}}^2 (y_j - s_i x_{ij} - c_i y_{ij} - y_i)^2. \end{aligned} \tag{4.10}$$

Note that the cost is a polynomial in the space $\mathbb{R}[\mathbf{c}, \mathbf{s}, \mathbf{x}, \mathbf{y}]$ and that each individual term $f_{ij}^{\mathbf{H}} \in \mathbb{R}[c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j]$. We can also rewrite $c_i^2 + s_i^2 = 1$ as $0 \leq 1 - c_i^2 - s_i^2 \leq 1$ and $0 \leq 2 - c_i^2 - s_i^2 \leq 1$ for each $i \in \{1, \ldots, n\}$. This parameterization allows us to rewrite (4.6) as a polynomial optimization problem in the form described in (4.1) and (4.2), where $M = 2n$:

$$\begin{aligned} \underset{\mathbf{c},\mathbf{s},\mathbf{x},\mathbf{y}}{\operatorname{argmin}} & \sum_{(i,j)\in E} f_{ij}^{\mathbf{H}}(c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j) \\ \text{s.t.} \quad & 0 \leq 1 - c_i^2 - s_i^2 \leq 1, \ \forall i \in \{1, \ldots, n\}, \\ & 0 \leq 2 - c_i^2 - s_i^2 \leq 1, \ \forall i \in \{1, \ldots, n\}. \end{aligned} \tag{4.11}$$

4.3.3 Landmark SLAM

In Landmark SLAM, one estimates both the pose of the robotic vehicle at each time step, $(\mathbf{R}_i, \mathbf{t}_i) \in \mathrm{SE}(2)$, as well as the position of observed landmarks, $\mathbf{l}_\ell = [l_\ell^x, l_\ell^y]^\top \in \mathbb{R}^2$, for each

$\ell \in \{1, \ldots, w\}$, given both relative pose measurements $(\bar{\mathbf{R}}_{ij}, \bar{\mathbf{t}}_{ij}) \in \mathrm{SE}(2)$, and landmark position observations, $\bar{\mathbf{l}}_{i\ell} = [x_{i\ell}, y_{i\ell}]^\top \in \mathbb{R}^2$, which measure the position of landmark with respect to the local coordinate frame of the robot at the time step that it was observed. Let $L = \{i_k, \ell_k\}_{k=1}^{m_\ell} \subset \{1, \ldots, n\} \times \{1, \ldots, w\}$ identify the set of landmark position measurements where m_ℓ is the number of landmark measurements and let $\mathbf{l_x} = \{l_1^x, \ldots, l_w^x\}$ and $\mathbf{l_y} = \{l_1^y, \ldots, l_w^y\}$. We assume that the relative pose measurements are distributed according to the structure defined in the previous section and that $\bar{\mathbf{l}}_{i\ell} \sim \mathcal{N}(\mathbf{l}_{i\ell}^i, \Omega_{\mathbf{l}_{i\ell}}^{-1})$ where $\mathbf{l}_{i\ell}^i$ is the true position of the landmark ℓ with respect to the true pose $\mathbf{H}_i$ and $\Omega_{\mathbf{l}_{i\ell}} = \mathrm{blkdiag}(\omega_{x_{i\ell}}^2, \omega_{y_{i\ell}}^2)$ is the information matrix of $\bar{\mathbf{l}}_{i\ell}$.

Under these assumptions, the MLE solution to the Landmark SLAM problem can be written as follows:

$$\begin{aligned} \underset{\mathbf{c},\mathbf{s},\mathbf{x},\mathbf{y},\mathbf{l_x},\mathbf{l_y}}{\operatorname{argmin}} & \sum_{(i,j)\in E} f_{ij}^{\mathbf{H}}(c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j) + \\ & + \sum_{(i,\ell)\in L} f_{i\ell}^{Land}(c_i, s_i, x_i, y_i, l_\ell^x, l_\ell^y) \\ \text{s.t.} \quad & 0 \leq 1 - c_i^2 - s_i^2 \leq 1, \ \forall i \in \{1, \ldots, n\}, \\ & 0 \leq 2 - c_i^2 - s_i^2 \leq 1, \ \forall i \in \{1, \ldots, n\}. \end{aligned} \tag{4.12}$$

with,

$$\begin{aligned} f_{i\ell}^{Land}(c_i, s_i, x_i, y_i, l_\ell^x, l_\ell^y) = \omega_{x_{i\ell}}^2 (l_\ell^x - c_i x_{i\ell} + \\ + s_i y_{i\ell} - x_i)^2 + \omega_{y_{i\ell}}^2 (l_\ell^y - s_i x_{i\ell} - c_i y_{i\ell} - y_i)^2. \end{aligned} \tag{4.13}$$

Note that the cost of the optimization problem in (4.12) is a polynomial in the space $\mathbb{R}[\mathbf{c}, \mathbf{s}, \mathbf{x}, \mathbf{y}, \mathbf{l_x}, \mathbf{l_y}]$ while $f_{i\ell}^{Land} \in \mathbb{R}[c_i, s_i, x_i, y_i, l_x, l_y]$. Also note that the constraints are the same as in (4.11) and thus, (4.12) is a polynomial optimization problem of the form defined in (4.1) and (4.2).

4.4 Sparse Bounded Sum-of-Squares Programming

Polynomial optimization problems in general are non-convex, however, they can be approximated and sometimes solved exactly by solving a hierarchy of convex relaxations of the problem (Lasserre, 2009). A variety of such convex relaxations hierarchies exist. This section covers a pair of such hierarchies. The first is called the bounded degree sum-of-squares (BSOS) hierarchy and consists of a sequence of SDP relaxations that can be used

to find the globally optimal solution to small polynomial optimization problems that meet certain conditions (Lasserre, Toh, and Yang, 2017). The second is called Sparse-BSOS and enables us to leverage the sparsity inherent in SLAM problems to solve larger problem sizes than is possible using BSOS (Weisser, Lasserre, and Toh, 2018). We conclude the section by describing the conditions that the cost and constraints that a polynomial optimization must satisfy for the first step of either hierarchy to converge exactly to the global optimum.

4.4.1 Bounded Sum-of-Squares

SOS programming is concerned with finding solutions to polynomial optimization problems as in (4.1). If $\mathbf{x}$ did not have to lie within the semi-algebraic set $\mathbf{K}$, solving the following problem would be equivalent to solving (4.1):

$$t^* = \sup_{t \in \mathbb{R}} \{t \mid f(\mathbf{x}) - t \geq 0, \forall\ \mathbf{x}\}. \tag{4.14}$$

If instead one had constraints g_j that bound the feasible space of the variable $\mathbf{x}$ to $\mathbf{K}$, then one would need to enforce that $f(\mathbf{x}) - t \geq 0,\ \forall\ \mathbf{x} \in \mathbf{K}$. At the same time, one would have to enforce it in a way that enabled $f - t$ to get as close to zero as possible at the optimal solution. Suppose we could optimize over a function h and also strictly enforce that it be non-negative on $\mathbf{K}$. Then, by enforcing that $f(\mathbf{x}) - t - h(\mathbf{x}) \geq 0$, for all x, we would equivalently enforce that $f(\mathbf{x}) - t \geq h(\mathbf{x}) \geq 0$ on $\mathbf{K}$, and we would be able to optimize over h to minimize the gap between f and t on $\mathbf{K}$.

To apply this approach using numerical optimization, one would first need to know whether it was computationally tractable to enforce positivity of h on K. Assuming that $0 \leq g_j(x) \leq 1$ for all $x \in \mathbf{K}$ and $\mathbf{K}$ is compact, one can prove that if a polynomial h is strictly positive on $\mathbf{K}$, then h can be represented as

$$h(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{\alpha, \beta \in \mathbb{N}^M} \lambda_{\alpha\beta} \prod_j \left(g_j(\mathbf{x})^{\alpha_j} (1 - g_j(\mathbf{x}))^{\beta_j} \right), \tag{4.15}$$

for some (finitely many) nonnegative scalars $\boldsymbol{\lambda} = (\lambda_{\alpha\beta})$ (Lasserre, Toh, and Yang, 2017, Theorem 1)[1]. Conversely, any polynomial that can be written in the form defined in (4.15) is also positive on $\mathbf{K}$. This leads to a hierarchy of relaxations in which each relaxation bounds the number of monomial terms used to represent h (Lasserre, Toh, and Yang, 2017, Theorem 2).

[1]Note that the theorem as presented requires the set $\{1, g_1, \ldots, g_M\}$ to generate $\mathbf{K}$, but since $\mathbf{K}$ is compact, one can always add a redundant linear constraint to the set to satisfy this requirement.

Let $N_d^{2M} = \{(\alpha, \beta) | \alpha, \beta \in \mathbb{N}^M, |\alpha| + |\beta| \leq d\}$ where the absolute value denotes the sum and

$$h_d(\mathbf{x}, \boldsymbol{\lambda}) := \sum_{(\alpha,\beta) \in N_d^{2M}} \lambda_{\alpha\beta} \prod_{j=1}^{M} g_j(\mathbf{x})^{\alpha_j} (1 - g_j(\mathbf{x}))^{\beta_j}, \tag{4.16}$$

where $\boldsymbol{\lambda} = (\lambda_{\alpha\beta}), (\alpha, \beta) \in \mathbb{N}_d^{2M}$. By choosing d, one can bound the number of monomial terms that are used to represent h_d and by optimizing over λ, one can optimize over the specific polynomial. By constraining $\boldsymbol{\lambda}$ to be non-negative, one can enforce that h_d be strictly positive on $\mathbf{K}$.

Now one can solve the following optimization problem:

$$t^* = \sup_{t, \boldsymbol{\lambda}} \{t | f(\mathbf{x}) - t - h_d(\mathbf{x}) \geq 0, \forall\ \mathbf{x}, \boldsymbol{\lambda} \geq 0\}. \tag{4.17}$$

However, optimizing over the space of all positive polynomials is computationally intractable. Instead, one can relax the problem again and optimize over the space of SOS polynomials up to a fixed degree since SOS polynomials are guaranteed to be positive and can be represented using a positive semidefinite matrix (Lasserre, 2009, Chapter 2). Let $\Sigma[\mathbf{x}] \subset \mathbb{R}[\mathbf{x}]$ represent the space of SOS polynomials and let $\Sigma[\mathbf{x}]_k \subset \mathbb{R}[\mathbf{x}]_{2k}$ represent the space of SOS polynomials of degree at most $2k$. By fixing $k \in \mathbb{N}$, one arrives at the following BSOS family of convex relaxations: indexed by $d \in \mathbb{N}$:

$$q_d^k = \sup_{t, \boldsymbol{\lambda}} \{t | f(\mathbf{x}) - t - h_d(\mathbf{x}) \in \Sigma[\mathbf{x}]_k, \boldsymbol{\lambda} \geq 0\}. \tag{4.18}$$

Each of these optimization programs can be implemented as an SDP and provides a lower bound on the solution to (4.17). Additionally, it can be shown that under certain assumptions as $d \to \infty$, $q_d^k \to f^*$ (Lasserre, Toh, and Yang, 2017, Theorem 2). While this is useful for small problems, as the number of variables increases or for larger values of d and k, the runtime and memory usage of the optimization makes the use of this method infeasible (Lasserre, Toh, and Yang, 2017, Section 3). To address this challenge, we take advantage of sparsity in the optimization problem to dramatically scale problem size.

4.4.2 Sparse Bounded Sum-of-Squares

The Sparse-BSOS hierarchy takes advantage of the fact that for many optimization problems, the variables and constraints exhibit structured sparsity. It does this by splitting the variables in the problem into p blocks of variables and the cost into p associated terms, such that the

number of variables and constraints relevant to each block is small (Weisser, Lasserre, and Toh, 2018).

Given $I \subset \{1, \ldots, N\}$, let $\mathbb{R}[\mathbf{x}; I]$ denote the ring of polynomials in the variables $\{x_i : i \in I\}$. Specifically Sparse-BSOS assumes that the cost and constraints satisfy the following assumption:

Assumption 4.4.1 (Running Intersection Property (RIP))**.** *There exists $p \in \mathbb{N}$ and $I_\ell \subseteq \{1, \ldots, N\}$ and $J_\ell \subseteq \{1, \ldots, M\}$ for all $\ell \in \{1, \ldots, p\}$ such that:*

- *$f = \sum_{\ell=1}^{p} f^\ell$, for some $f^1, \ldots, f^p$, such that $f^\ell \in \mathbb{R}[\mathbf{x}, I_\ell]$ for each $\ell \in \{1, \ldots, p\}$,*
- *$g_j \in \mathbb{R}[\mathbf{x}, I_\ell]$ for each $j \in J_\ell$ and $\ell \in \{1, \ldots, p\}$,*
- *$\cup_{\ell=1}^{p} I_\ell = \{1, \ldots, N\}$,*
- *$\cup_{\ell=1}^{p} J_\ell = \{1, \ldots, M\}$,*
- *for all $\ell = 1, \ldots, p-1$, there is an $s \leq \ell$ such that $(I_{\ell+1} \cap \cup_{r=1}^{\ell} I_r) \subseteq I_s$.*

In particular, I_ℓ denotes the variables that are relevant to ℓ-th block and J_ℓ denotes the associated relevant constraints. Intuitively, these blocks allow one to enforce positivity over a smaller set of variables which can reduce the computational burden while trying to solve this optimization problem.

We can use these definitions to define the Sparse-BSOS hierarchy that builds on the hierarchy defined in the previous section. Let $N^\ell := \{(\alpha, \beta) : \alpha, \beta \in \mathbb{N}_0, \text{supp}(\alpha) \cup \text{supp}(\beta) \subseteq J_\ell\}$, where $\mathbb{N}_0$ is the set of natural numbers including 0 and $\text{supp}(\alpha) := \{j \in \{1, \ldots, M\} : \alpha_j \neq 0\}$. Now let $N_d^\ell := \{(\alpha, \beta) \in N^\ell : \sum_j (\alpha_j + \beta_j) \leq d\}$, with $d \in \mathbb{N}$ and let

$$h_d^\ell(\mathbf{x}, \boldsymbol{\lambda}^\ell) := \sum_{(\alpha,\beta) \in N_d^\ell} \lambda_{\alpha\beta}^\ell \prod_{j=1}^{M} g_j(\mathbf{x})^{\alpha_j} (1 - g_j(\mathbf{x}))^{\beta_j}, \tag{4.19}$$

where $\boldsymbol{\lambda}^\ell \in \mathbb{R}^{|N_d^\ell|}$ is the vector of scalar coefficients $\lambda_{\alpha\beta}^\ell$. h_d^ℓ is again positive on $\mathbf{K}$ as long as the elements of $\boldsymbol{\lambda}$ are positive. If we again fix $k \in \mathbb{N}$, we can define a family of optimization problems indexed by $d \in \mathbb{N}$ as shown in (4.20), where $d_{\max}$ is defined on page 7 of (Weisser, Lasserre, and Toh, 2018).

This hierarchy of relaxations is called the Sparse-BSOS hierarchy and each level of the hierarchy can be implemented as an SDP. In addition, if RIP is satisfied, $0 \leq g_j(x) \leq 1$ for all $x \in \mathbf{K}$ and $\mathbf{K}$ is compact, then as $d \to \infty$, the sequence of optimization problems defined in (4.20) also converges to f^*(Weisser, Lasserre, and Toh, 2018, Theorem 2). In addition a

$$q_d^k = \sup_{\substack{t,\boldsymbol{\lambda}^1,\ldots,\boldsymbol{\lambda}^p,\\ f^1,\ldots,f^p}} \Big\{ t | f^\ell(\mathbf{x}) - h_d^\ell(\mathbf{x}, \boldsymbol{\lambda}^\ell) \in \Sigma[\mathbf{x}; I_\ell]_k, \ell = 1, \ldots, p, \tag{4.20}$$

$$f(\mathbf{x}) - t = \sum_{\ell=1}^{p} f^\ell(\mathbf{x}), \boldsymbol{\lambda}^\ell \in \mathbb{R}^{|N_d^\ell|} \geq 0, t \in \mathbb{R}, f^\ell \in \mathbb{R}[\mathbf{x}; I_\ell]_{d_{max}} \Big\}.$$

rank condition can be used to detect finite convergence (Weisser, Lasserre, and Toh, 2018, Lemma 4). Importantly in particular cases, one can show that this optimization problem can be solved exactly when $d = 1$.

4.4.3 Sum-of-Squares Convexity

A polynomial $f \in \mathbb{R}[\mathbf{x}]$ is said to be SOS-convex when the Hessian matrix $\mathbf{x} \mapsto \nabla^2 f(\mathbf{x})$ is an SOS matrix polynomial, that is, there exists an $a \in \mathbb{N}$ such that $\nabla^2 f = \mathbf{L}\mathbf{L}^\top$ for some real matrix polynomial $\mathbf{L} \in \mathbb{R}[\mathbf{x}]^{n \times a}$ for some a. A polynomial optimization problem of the form shown in (4.1) and (4.2) is said to be SOS-convex if the polynomials f^ℓ and $-g_j$ are SOS-convex for every $\ell \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, M\}$. Importantly, when $k \in \mathbb{N}$ is fixed, the polynomials f^ℓ and g_j are at most degree $2k$ for each $\ell \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, M\}$, and the optimization program is SOS-convex, then the Sparse-BSOS hierarchy defined in (4.20) converges at $d = 1$ (Weisser, Lasserre, and Toh, 2018, Theorem 3).

4.5 SBSOS-SLAM

We take advantage of the Sparse-BSOS relaxation hierarchy to solve the Pose Graph and Landmark SLAM problems. In this section, we prove that both these problems are SOS-convex as formulated in (4.11) and (4.12). Then, we talk about how we can enforce the RIP. Finally, we end with a discussion of implementation.

4.5.1 SOS-Convexity of Pose Graph and Landmark SLAM

To prove SOS-Convexity of the Pose Graph and Landmark SLAM problems, we use the following result:

Lemma 4.5.1. *(Ahmadi and Parrilo, 2009, Lemma 2.2) A polynomial $f \in \mathbb{R}[\mathbf{x}]$ is SOS-convex if and only if the polynomial $\mathbf{z}^\intercal \nabla^2 f \mathbf{z}$ is a sum of squares in $\mathbb{R}[\mathbf{x}; \mathbf{z}]$.*

Using this result, one can show the following by noting the linearity of the Hessian:

Lemma 4.5.2. *If $f, g \in \mathbb{R}[\boldsymbol{x}]$ are SOS-convex, then $f + g$ is SOS-convex.*

We use these results to prove the following:

Theorem 4.5.3. *The Pose Graph SLAM problem defined in* (4.11) *can be solved exactly using the Sparse-BSOS optimization problem defined in* (4.20) *when $k = 1$ and $d = 1$.*

Proof. To prove this result, we show that the Pose Graph SLAM problem is SOS-convex and use (Weisser, Lasserre, and Toh, 2018, Theorem 3). Note that $-g_i(c_i, s_i)$ can either be $c_i^2 + s_i^2 - 1$ or $c_i^2 + s_i^2 - 2$, which both produce the following Hessian:

$$\nabla^2(-g_i) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix}^\mathsf{T} \tag{4.21}$$

Therefore $-g_i$ is SOS-convex for all i. To show the cost function is SOS-convex based on Lemma 4.5.2, it suffices to show $f_{ij}^{\mathbf{H}}(c_i, s_i, x_i, y_i, c_j, s_j, x_j, y_j)$ is SOS-convex, as can be seen by looking at each of its elements.

- Let $\gamma_1(c_i, s_i, c_j) = \omega_{\mathbf{R}_{ij}}^2(c_j - c_i c_{ij} + s_i s_{ij})^2$:

$$\nabla^2\gamma_1 = \begin{bmatrix} \sqrt{2}\omega_{\mathbf{R}_{ij}} c_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} s_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} \end{bmatrix} \begin{bmatrix} \sqrt{2}\omega_{\mathbf{R}_{ij}} c_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} s_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} \end{bmatrix}^\mathsf{T} \tag{4.22}$$

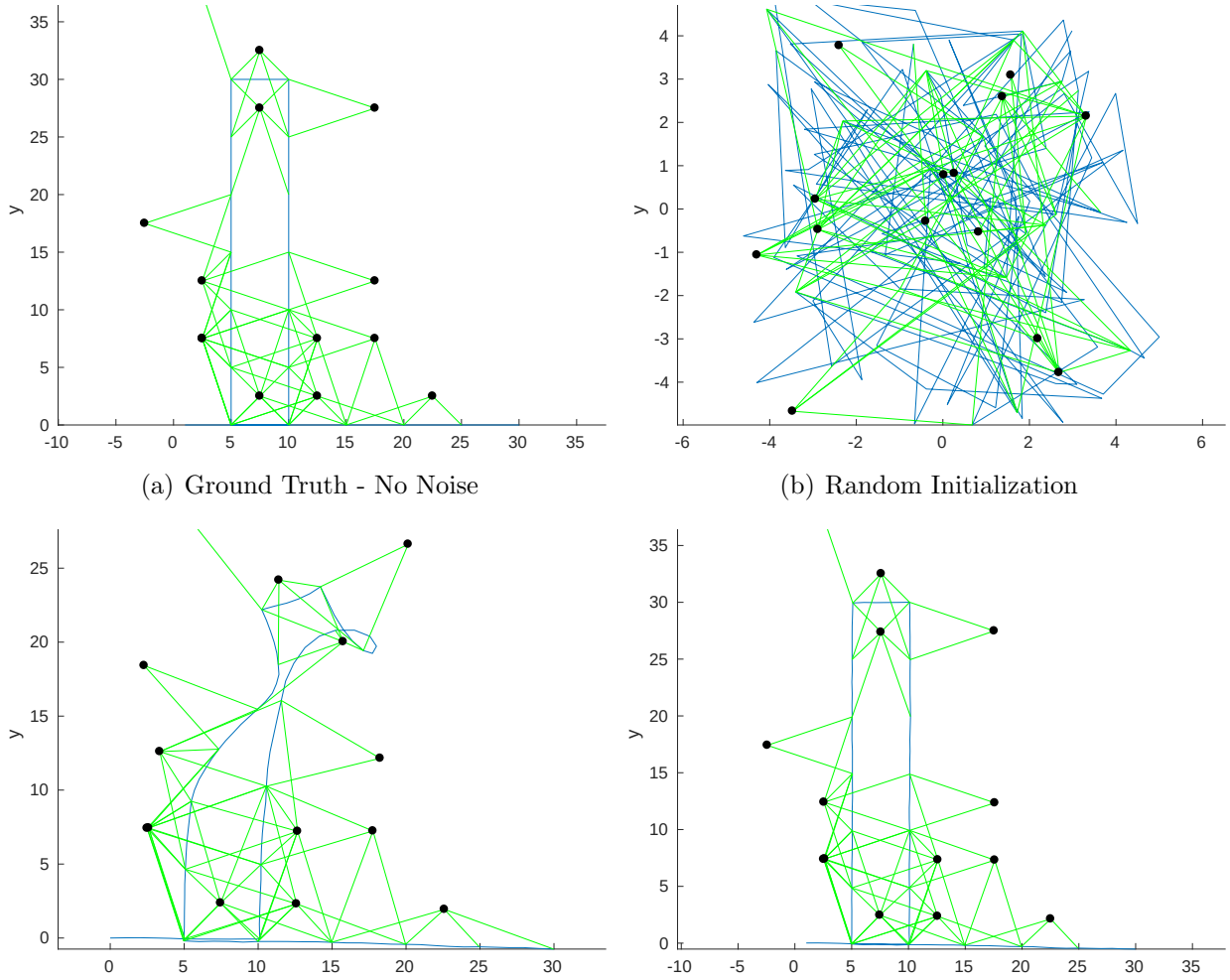
- Let $\gamma_2(c_i, s_i, s_j) = \omega_{\mathbf{R}_{ij}}^2(-s_j + c_i s_{ij} + s_i c_{ij})^2$:

$$\nabla^2\gamma_2 = \begin{bmatrix} \sqrt{2}\omega_{\mathbf{R}_{ij}} s_{ij} \\ \sqrt{2}\omega_{\mathbf{R}_{ij}} c_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} \end{bmatrix} \begin{bmatrix} \sqrt{2}\omega_{\mathbf{R}_{ij}} s_{ij} \\ \sqrt{2}\omega_{\mathbf{R}_{ij}} c_{ij} \\ -\sqrt{2}\omega_{\mathbf{R}_{ij}} \end{bmatrix}^\mathsf{T} \tag{4.23}$$

- Let $\gamma_3(c_i, s_i, s_j) = \omega_{\mathbf{R}_{ij}}^2(s_j - s_i c_{ij} - c_i s_{ij})^2$ and $\gamma_4(c_i, s_i, c_j) = \omega_{\mathbf{R}_{ij}}^2(c_j + s_i s_{ij} - c_i c_{ij})^2$. Notice that $\gamma_3 = \gamma_2$ and $\gamma_4 = \gamma_1$, thus both γ_3 and γ_4 are also SOS-convex.

- Let $\gamma_5(c_i, s_i, x_i, x_j) = \omega_{x_{ij}}^2(x_j - c_i x_{ij} + s_i y_{ij} - x_i)^2$:

$$\nabla^2\gamma_5 = \begin{bmatrix} \sqrt{2}\omega_{x_{ij}} x_{ij} \\ -\sqrt{2}\omega_{x_{ij}} y_{ij} \\ \sqrt{2}\omega_{x_{ij}} \\ -\sqrt{2}\omega_{x_{ij}} \end{bmatrix} \begin{bmatrix} \sqrt{2}\omega_{x_{ij}} x_{ij} \\ -\sqrt{2}\omega_{x_{ij}} y_{ij} \\ \sqrt{2}\omega_{x_{ij}} \\ -\sqrt{2}\omega_{x_{ij}} \end{bmatrix}^\mathsf{T} \tag{4.24}$$



(a) Ground Truth - No Noise

(b) Random Initialization

(c) Levenberg-Marquardt

(d) Global Optimal - SBSOS-SLAM

Figure 4.2: SBSOS-SLAM vs LM on CityTrees10000 100 Node Segment. Sample estimated Landmark SLAM solution for 100 nodes of CityTrees10000 dataset (Kaess, Ranganathan, and Dellaert, 2008).

- Let $\gamma_6(c_i, s_i, y_i, y_j) = \omega_{y_{ij}}^2 (y_j - s_i x_{ij} - c_i y_{ij} - y_i)^2$:

$$\nabla^2 \gamma_6 = \begin{bmatrix} \sqrt{2}\omega_{y_{ij}} y_{ij} \\ \sqrt{2}\omega_{y_{ij}} x_{ij} \\ \sqrt{2}\omega_{y_{ij}} \\ -\sqrt{2}\omega_{y_{ij}} \end{bmatrix} \begin{bmatrix} \sqrt{2}\omega_{y_{ij}} y_{ij} \\ \sqrt{2}\omega_{y_{ij}} x_{ij} \\ \sqrt{2}\omega_{y_{ij}} \\ -\sqrt{2}\omega_{y_{ij}} \end{bmatrix}^\intercal \tag{4.25}$$

Since we can enlarge the domains of all γ_i's into $[c_i; s_i; c_j; s_j; x_i; x_j; y_i; y_j]$ and $f_{ij}^{\mathbf{H}} = \sum_{i=1}^{6} \gamma_i$, we then conclude by Lemma 4.5.2 that $f_{ij}^{\mathbf{H}}$ is SOS-convex. □

By using a proof similar to Theorem 4.5.3, one can then show:

Theorem 4.5.4. *The Landmark SLAM problem defined in* (4.12) *can be solved exactly using the Sparse-BSOS optimization problem defined in* (4.20) *when* $k = 1$ *and* $d = 1$.

4.5.2 Satisfying the Running Intersection Property

The nature of the SLAM problem exhibits a large amount of sparsity (Dellaert and Kaess, 2006; Eustice, Walter, and Leonard, 2005; Kaess, Ranganathan, and Dellaert, 2008). However, to take advantage of the guarantees incumbent to the Sparse-BSOS hierarchy, we need to satisfy the RIP. An odometry chain forming the backbone of a SLAM graph inherently satisfies this property; however, incorporating loop-closures can make satisfying this assumption challenging. A better grouping increases the sparsity of the optimization problem and leads to faster solutions but finding the optimal selection of blocks I_ℓ is NP-hard. We used the heuristic algorithm proposed by Smail (2017) to generate a sequence of variable groupings for the current implementation.

4.5.3 Implementation and Computational Scaling

For the experiments and development presented in this paper, we modified the code base released with (Weisser, Lasserre, and Toh, 2018) to formulate the problem as defined earlier on in the chapter. In addition, we modified the code to convert the problem to a format where we could use the SDP solver within the optimization library Mosek (ApS, 2015).

SOS programming optimizes over polynomials which can become ill-conditioned when optimization occurs over a large domain (Weisser, Lasserre, and Toh, 2018, Section 4). Data can be scaled to address this problem; however, if the optimization problem still remains poorly scaled then the optimization solver will warn the user that the problem cannot be satisfactorily solved.

4.6 Experimental Proof of Concept

We evaluated the proposed convex SLAM algorithm on the CityTrees10000 (Kaess, Ranganathan, and Dellaert, 2008) and Manhattan3500 (Olson, Leonard, and Teller, 2006) datasets by breaking the problem into sequences of 100 nodes and solving those graphs for which Mosek (ApS, 2015) did not run into numerical instabilities. For each sequence (taken individually), we used the proposed SBSOS-SLAM methodology to find the globally optimal

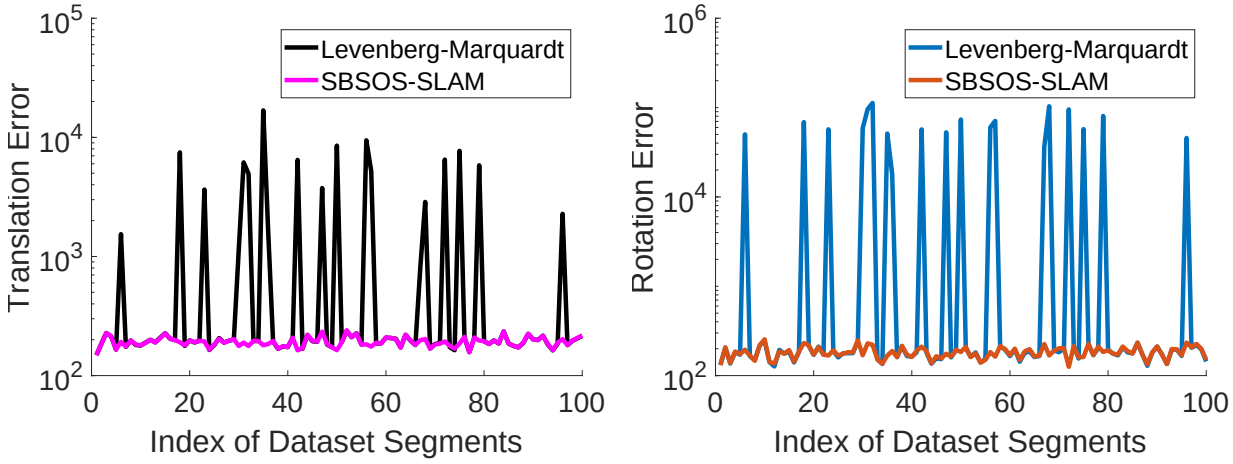


Figure 4.3: Pose Error Comparison on CityTrees10000. Translational and rotational error verses groundtruth for the CityTrees10000 dataset.

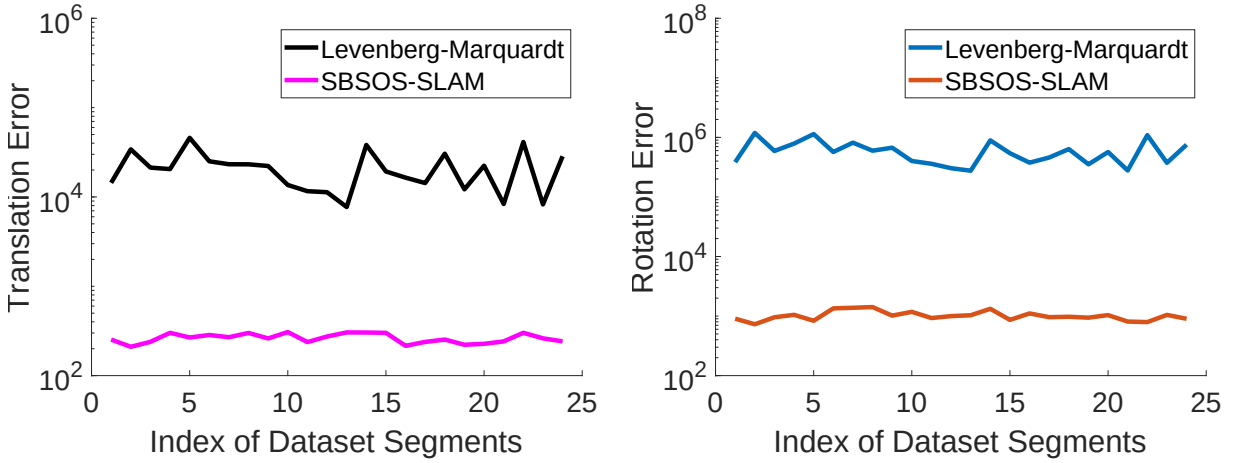
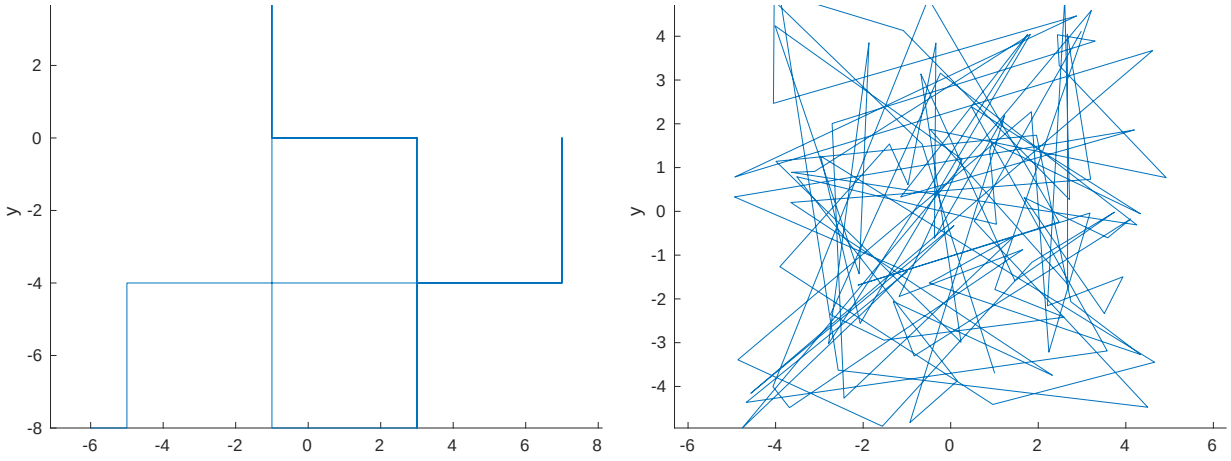


Figure 4.4: Pose Error Comparison on Manhattan3500. Translational and rotational error verses groundtruth for the Manhattan3500 dataset.

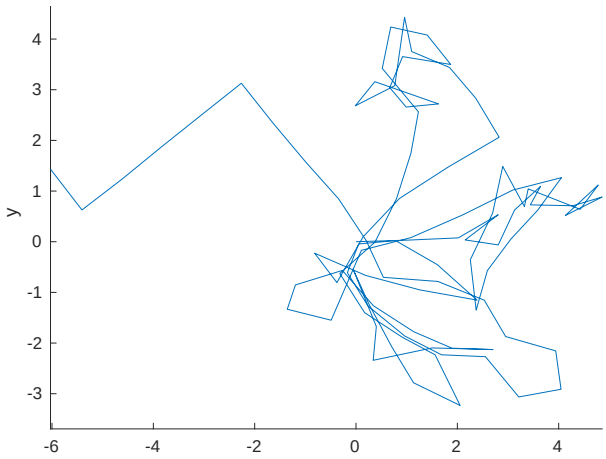
solution to the respective MLE problem defined in (4.11) and (4.12). For comparison, we also initialized Levenberg-Marquardt with a random initialization.

The median solve time for SBSOS-SLAM was 20.5507 seconds for the Manhattan3500 dataset and 161.8387 sec for the CityTree10000 dataset compared to less than a second on average for Levenberg-Marquardt. However, since our current implementation is based in Matlab and we are not attempting to satisfy the running intersection property optimally in these initial experiments, we believe there are a variety of extensions that can be made to improve scalability.

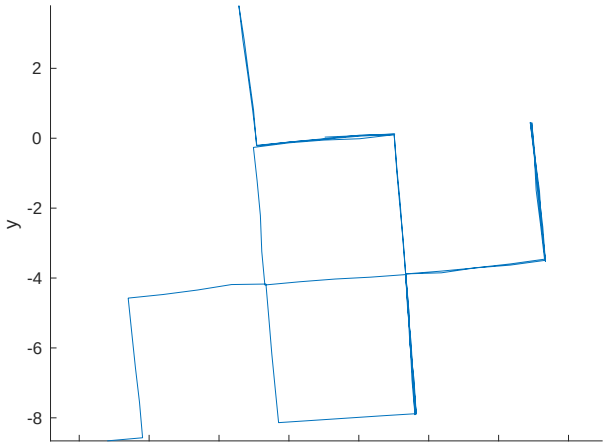


(a) Ground Truth - No Noise

(b) Random Initialization



(c) Levenberg-Marquardt



(d) Global Optimal - SBSOS-SLAM

Figure 4.5: SBSOS-SLAM vs LM on Manhattan3500 100 Node Segment. Sample estimated Pose Graph SLAM solution for 100 nodes of Manhattan3500 dataset (Olson, Leonard, and Teller, 2006).

We show several example plots where Levenberg-Marquardt gets stuck in a local minima while SBSOS-SLAM is guaranteed to find the global minimum and does not require initialization (Fig. 4.2, Fig. 4.5). Fig. 4.3 and Fig. 4.4 show that our proposed algorithm results in significantly less error than Levenberg-Marquardt.

4.7 Conclusion

In this chapter, we proposed an algorithm called SBSOS-SLAM that formulates the planar Pose Graph and Landmark SLAM problems as polynomial optimization programs that are

sum-of-squares convex. As such, we are able to guarantee optimality of the solution without any need for initialization.

In addition, because we formulate the SLAM problem as a polynomial optimization program, there is potential to extend this work by generalizing the formulation to three-dimensional environments and arbitrary measurement models such as range-only measurements, while still providing guarantees of optimality.

The completed work has been accepted and will be published in:

J. G. Mangelson, J. Liu, R. M. Eustice, and R. Vasudevan. Guaranteed globally optimal planar pose graph and landmark slam via sparse-bounded sum-of-squares programming. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–8, Montreal, Canada, May 2019

CHAPTER 5

Characterizing the Uncertainty of Jointly Distributed Poses via Lie Theory

5.1 Introduction

An accurate characterization of robot pose (location and orientation) uncertainty is essential to robust long-term autonomy because planning and safety decisions are often predicated on their value (Thrun, Burgard, and Fox, 2005). For example, an over-confident position estimate could potentially result in a self-driving car crossing out of its lane or an underwater vehicle colliding with a submerged structure. On the other hand, under-confidence can lead to slow or sluggish behavior.

One of the first papers to characterize pose uncertainty of coordinate frame relationships represents the relative pose of objects using a multivariate Gaussian parameter vector and associated covariance matrix (Smith and Cheeseman, 1986). This paper was later extended by Smith, Self, and Cheeseman (1990) by representing multiple uncertain spatial relationships as a *stochastic map* which could be used to evaluate the uncertainty of any given pose with respect to any other. They also proposed several operations (such as the relative pose operation shown in Fig. 5.1) that enable the extraction of additional information not directly estimated, along with first order coordinate-based methods for propagating uncertainty through these operations. For brevity, the operations proposed in (Smith, Self, and Cheeseman, 1990) are often referred to by the initials of the paper's authors (SSC).

Although it is well-understood that the rigid body transformation (or the motion group of $\mathbb{R}^3$) is described by the three dimensional (3D) Special Euclidean group, SE(3), (Murray, Li, and Shankar Sastry, 1994; Spong, Hutchinson, and Vidyasagar, 2005), the uncertainty of these transformations is often modeled in local coordinates leading to inconsistencies in the estimation problem (Huang and Dissanayake, 2007) or the loss of monotonicity in uncertainty propagation (Rodríguez-Arévalo, Neira, and Castellanos, 2018). Wang and Chirikjian (2008)

and Long et al. (2012) were able to overcome these problems by representing each pose using *exponential coordinates* located in the Lie algebra of the SE(d).

Barfoot and Furgale (2014) were then able to show that propagation computations could be simplified by modeling the uncertainty directly in the Lie algebra and then using the exponential map to induce a distribution in the group space. Since the Lie algebra is a vector space, a small perturbation term can be modeled as zero-mean Gaussian noise in $\mathbb{R}^6$ and then used to perturb a mean (or nominal) pose in the group space. Barfoot and Furgale (2014) then build on this by deriving first and second order uncertainty propagation for the pose composition operation when the associated poses are independent. Our approach for modeling the uncertainty of a set of poses is similar; however, we drop the independence assumption, since the poses estimated by SLAM are rarely independent, and additionally describe the additional operations of pose inversion and relative pose extraction (See Fig. 5.1 and Fig. 5.3).

The main contributions of this chapter are as follows:

1. we present a framework that describes how to represent jointly correlated poses while using the Lie algebra to characterize uncertainty;

2. we derive the equivalent of the SSC operations under the proposed framework;

3. we describe how to convert from alternative uncertainty characterization parameterizations to the proposed framework (including Lie algebra covariance extraction from a MLE solution); and,

4. we release an accompanying C++ library implementation along with examples also presented here.

The remainder of this chapter is organized as follows: Section 5.2 provides a brief introduction to the Special Euclidean group and some necessary concepts from Lie group Theory. Section 5.3 provides a summary of the SSC uncertainty representation framework and its associated operations. Section 5.4 describes how to use the Lie algebra to characterize uncertainty for jointly distributed poses. Sections 5.5, Section 5.6, and Section 5.7 describe the derivation of the pose composition, pose inversion, and relative pose operations, respectively, while characterizing uncertainty on the Lie algebra. Section 5.8 describes how to convert from a coordinate based representation of uncertainty to the Lie algebra based representation as well as how to extract an estimate of pose uncertainty from a MLE solution. Section 5.9 describes an experimental evaluation of the proposed methods. Section 5.10 describes the implementation of the released library. Finally, Section 5.11 concludes the chapter.

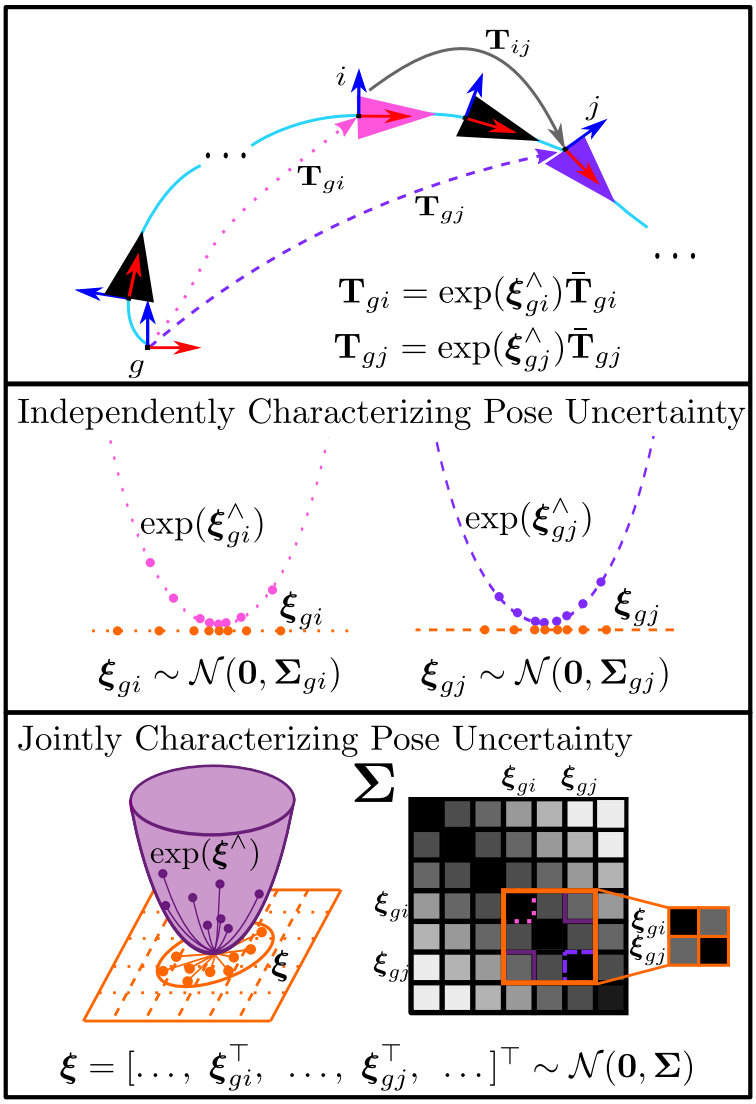


Figure 5.1: Joint vs Independent Lie Algebra Uncertainty Propogation. State-of-the-art Pose-Graph SLAM algorithms estimate the pose (depicted as i and j in the top illustration) of a robotic vehicle at each time step with respect to a fixed coordinate frame, g, which is denoted by $\mathbf{T}_{gi}$ and $\mathbf{T}_{gj}$, respectively. After solving SLAM, it is often necessary to extract additional information by performing a variety of operations such as pose composition, pose inversion, and relative pose estimation, while accurately propagating uncertainty. An example of the relative pose operation $\mathbf{T}_{ij}$ is shown at the top of this figure. Recent work has shown that characterizing uncertainty as Gaussian random variables $(\boldsymbol{\xi}_{gi}, \boldsymbol{\xi}_{gj})$ in the Lie algebra of the Special Euclidean group (shown in the middle section of the above figure) leads to increased consistency (Barfoot and Furgale, 2014); however, this approach focuses primarily on pose composition while assuming that the underlying poses are independent. Typically, the poses estimated from SLAM are heavily correlated (Dissanayake et al., 2001). This paper proposes a framework for jointly characterizing the uncertainty of a set of correlated poses in the Lie algebra space (shown in the bottom section of the above figure). It then describes how to perform the pose composition, pose inverse, and relative pose operations within this framework.

5.2 The Special Euclidean Group and Lie Theory

Estimation of the relative pose (position and orientation) between objects or coordinate frames in space is a common problem in robotic navigation, perception, and manipulation. Formally, we represent 3D relative pose transformations as elements of the *Special Euclidean group*. This section provides a brief introduction to the *Special Euclidean group* and relevant aspects of Lie group theory that are important during the subsequent derivation and discussion.

5.2.1 The Special Euclidean group

The *Special Euclidean group*, or SE(d), represents the space of homogeneous transformation matrices or the space of matrices that apply a rigid body rotation and translation to points in $\mathbb{R}^d$ (represented in homogeneous form). Formally, in three dimensions, SE(3) is defined as follows:

$$\mathrm{SE}(3) := \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \middle| \mathbf{R} \in \mathrm{SO}(3), \mathbf{t} \in \mathbb{R}^3 \right\}, \tag{5.1}$$

where SO(3) is the *Special Orthogonal group* is the space of valid rotation matrices:

$$\mathrm{SO}(3) := \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} | \mathbf{R}\mathbf{R}^\top = \mathbf{I}_3, \det \mathbf{R} = 1 \right\}, \tag{5.2}$$

and $\mathbf{I}_d$ is the identity matrix of dimension d. A variety of methods have been developed for parameterizing these objects such as Euler angles or quaternions (Spong, Hutchinson, and Vidyasagar, 2005).

Both SE(3) and SO(3) are matrix *Lie groups*, meaning they are smooth manifolds that also satisfy the formal definition of a mathematical group (Baker, 2012; Tapp, 2016) with the standard matrix multiplication operation. Intuitively, this means that while the general group is non-linear, each of these groups can be locally approximated using a Euclidean vector space. Additionally, for any given point on the manifold, consider the set of all paths on the manifold that pass through that point. The set of all velocities (both in terms of direction and speed) of those paths at the given point form a vector space called the tangent space. The tangent space centered at the identity is called the *Lie algebra*. This relationship is depicted in Fig. 5.2.

The Lie algebra of SO(3) is denoted $\mathfrak{so}(3)$ and is the space of skew-symmetric 3×3 matrices (Chirikjian, 2011; Tapp, 2016):

$$\mathfrak{so}(3) := \left\{ \boldsymbol{\omega} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{\omega}^\top = -\boldsymbol{\omega} \right\} \tag{5.3}$$

This space is isomorphic to $\mathbb{R}^3$ since skew-symmetric matrices have zeros on the diagonal entries and the lower entries are completely identified by three upper entries (hence $\dim \mathfrak{so}(3) = 3$). Therefore, it is very convenient to work with $\mathbb{R}^3$ instead. Note, we can use the $\wedge$ operator to take an element of $\mathbb{R}^3$ and transform it to an element of $\mathfrak{so}(3)$:

$$\boldsymbol{\phi}^\wedge := \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathfrak{so}(3), \tag{5.4}$$

where $\boldsymbol{\phi} \in \mathbb{R}^3$. The $\vee$ operator denotes the inverse of $\wedge$.

Similarly, the Lie algebra of SE(3), or $\mathfrak{se}(3)$, is defined as follows:

$$\mathfrak{se}(3) := \left\{ \begin{bmatrix} \boldsymbol{\omega} & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \;\middle|\; \boldsymbol{\omega} \in \mathfrak{so}(3), \boldsymbol{\rho} \in \mathbb{R}^3 \right\}. \tag{5.5}$$

This space of matrices is isomorphic to $\mathbb{R}^6$, and we overload the $\wedge$ operator to convert between the Euclidean vector and matrix forms:

$$\boldsymbol{\xi}^\wedge := \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathfrak{se}(3), \tag{5.6}$$

where $\boldsymbol{\xi} \in \mathbb{R}^6$ and $\boldsymbol{\rho}, \boldsymbol{\phi} \in \mathbb{R}^3$.

Understanding the relationship between the group and algebra spaces can enable one to leverage the fact that the algebra is a vector space. The next few subsections cover some important concepts from Lie group Theory that we need in the rest of the chapter. In doing so, we use $\mathcal{G}$ to represent a given Lie group and $\mathfrak{g}$ to represent its associated Lie algebra.

5.2.2 The Exponential Map

The Lie algebra, $\mathfrak{g}$, represents the tangent space of the manifold at the identity. However, given a specific tangent vector, we may want to convert it to its associated transformation in the group space $\mathcal{G}$. The *exponential map*, $\exp : \mathfrak{g} \to \mathcal{G}$, (which can be defined in closed form for SE(3)), enables us to perform this conversion,

$$\exp(\boldsymbol{\xi}^\wedge) = \sum_{k=0}^{\infty} \frac{(\boldsymbol{\xi}^\wedge)^k}{k!} = \mathbf{I}_4 + \boldsymbol{\xi}^\wedge + \frac{(\boldsymbol{\xi}^\wedge)^2}{2} + \cdots. \tag{5.7}$$

The *logarithm map*, $\log : \mathcal{G} \to \mathfrak{g}$, on the other hand, enables us to go the other direction from an action/transformation in the group space to the velocity that would have induced

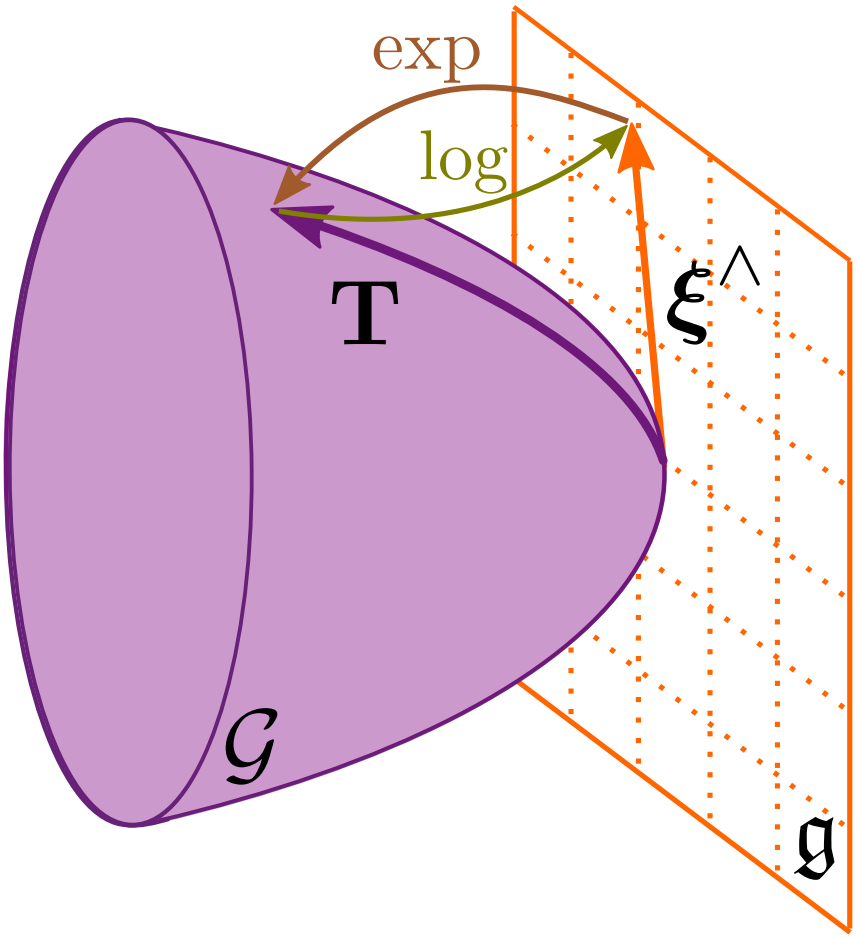


Figure 5.2: The Relationship Between the Lie Group and Algebra. The Lie algebra, $\mathfrak{g}$, is the tangent space to the Lie group, $\mathcal{G}$, centered at the identity element. The Lie algebra represents the space of all possible velocities a particle at a given point on the group could take. The *exponential map* maps velocities in the Lie algebra, $\boldsymbol{\xi}^\wedge \in \mathfrak{g}$, to their associated action in the Lie group, $\mathbf{T} \in \mathcal{G}$, and the *logarithm map* performs the inverse operation.

it,

$$\log(\mathbf{T}) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(\mathbf{T} - \mathbf{I}_4)^k}{k}. \tag{5.8}$$

This relationship is visualized in Fig. 5.2.

5.2.3 The Adjoint Action

Assuming $\mathbf{T} \in \mathcal{G}$ and $\boldsymbol{\xi} \in \mathfrak{g}$, the adjoint action of $\mathbf{T}$ on $\boldsymbol{\xi}$, or $\mathrm{Ad}_{\mathbf{T}}(\boldsymbol{\xi})$ is defined as follows:

$$\mathrm{Ad}_{\mathbf{T}}(\boldsymbol{\xi}) := \mathrm{Ad}_{\mathbf{T}}\, \boldsymbol{\xi} = \log(\mathbf{T} \exp(\boldsymbol{\xi}) \mathbf{T}^{-1}). \tag{5.9}$$

The adjoint action describes the effect that transforming to the group space then applying a transformation on the left and its inverse on the right has on an element of the Lie algebra. This gives rise to the following property, which we use later on:

$$\mathbf{T}\exp(\boldsymbol{\xi}) = \exp(\mathrm{Ad}_{\mathbf{T}}\,\boldsymbol{\xi})\mathbf{T} \tag{5.10}$$

5.2.4 The Baker-Campbell-Hausdorff (BCH) Formula

Finally, we also want to characterize the effect that multiplication in the group space has on the Lie algebra. More specifically, suppose we want to compute the Lie algebra element, $\boldsymbol{\xi}_{ac} \in \mathfrak{g}$, that is generated by taking the logarithm of the product of the exponential of two Lie algebra elements, $\boldsymbol{\xi}_{ab}, \boldsymbol{\xi}_{bc} \in \mathfrak{g}$. The *Baker-Campbell-Hausdorff (BCH)* formula describes this relationship purely in the Lie algebra space without requiring the application of the exponential or the logarithm:

$$\begin{aligned}\boldsymbol{\xi}_{ac} &= \log(\exp(\boldsymbol{\xi}_{ab})\exp(\boldsymbol{\xi}_{bc})) \\ &= \boldsymbol{\xi}_{ab} + \boldsymbol{\xi}_{bc} + \frac{1}{2}[\boldsymbol{\xi}_{ab}, \boldsymbol{\xi}_{bc}] + \\ &+ \frac{1}{12}([\boldsymbol{\xi}_{ab}, [\boldsymbol{\xi}_{ab}, \boldsymbol{\xi}_{bc}]] + [\boldsymbol{\xi}_{bc}, [\boldsymbol{\xi}_{bc}, \boldsymbol{\xi}_{ab}]]) + \cdots,\end{aligned} \tag{5.11}$$

where $[\,\cdot\,,\,\cdot\,]$ is the Lie bracket of $\mathfrak{g}$ (Barfoot, 2017, (7.18)). For brevity, in the special case where we consider elements in SE(3), we adopt the notation of (Barfoot and Furgale, 2014):

$$\boldsymbol{\xi}_i^{\curlywedge} := \begin{bmatrix} \boldsymbol{\rho}_i \\ \boldsymbol{\phi}_i \end{bmatrix}^{\curlywedge} = \begin{bmatrix} \boldsymbol{\phi}_i^{\wedge} & \boldsymbol{\rho}_i^{\wedge} \\ \mathbf{0} & \boldsymbol{\phi}_i^{\wedge} \end{bmatrix}, \tag{5.12}$$

which allows us to write the BCH formula as:

$$\begin{aligned}\boldsymbol{\xi}_{ac} = \boldsymbol{\xi}_{ab} + \boldsymbol{\xi}_{bc} + \frac{1}{2}\boldsymbol{\xi}_{ab}^{\curlywedge}\boldsymbol{\xi}_{bc} + \frac{1}{12}\boldsymbol{\xi}_{ab}^{\curlywedge}\boldsymbol{\xi}_{ab}^{\curlywedge}\boldsymbol{\xi}_{bc} + \frac{1}{12}\boldsymbol{\xi}_{bc}^{\curlywedge}\boldsymbol{\xi}_{bc}^{\curlywedge}\boldsymbol{\xi}_{ab} \\ - \frac{1}{24}\boldsymbol{\xi}_{bc}^{\curlywedge}\boldsymbol{\xi}_{ab}^{\curlywedge}\boldsymbol{\xi}_{ab}^{\curlywedge}\boldsymbol{\xi}_{bc} + \ldots.\end{aligned} \tag{5.13}$$

5.2.5 Defining Random Variables over Poses

Finally, as discussed in (Barfoot and Furgale, 2014), one can define random variables for SE(3) according to

$$\mathbf{T}_\ell := \exp(\boldsymbol{\xi}_\ell^{\wedge})\bar{\mathbf{T}}_\ell \tag{5.14}$$

where $\bar{\mathbf{T}}_\ell \in \mathrm{SE}(3)$ is a 'large' noise free value and $\boldsymbol{\xi}_\ell \in \mathbb{R}^6$ is a 'small' noisy perturbation (using the nomenclature of Barfoot and Furgale (2014)). Two examples of this noisy perturbation are depicted in the middle row of Fig. 5.1. By defining $\boldsymbol{\xi}_\ell$ to be a zero-mean Gaussian random variable $\boldsymbol{\xi}_\ell \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\ell)$ in the Lie algebra, we induce a probability distribution function over SE(3) that is parameterized with a mean $\bar{\mathbf{T}}_\ell \in \mathrm{SE}(3)$ and a covariance $\boldsymbol{\Sigma}_\ell$ defined in the Lie algebra (Barfoot and Furgale, 2014).

We use these properties in Sections 5.4, 5.5, 5.6, and 5.7 to propose a framework for modeling jointly correlated poses and to derive uncertainty propagation formulas for the operations in Fig. 5.3. However, we first review the SSC coordinate based method.

5.3 Review of SSC

The stochastic map, proposed by Smith, Self, and Cheeseman (1990), consists of multiple uncertain spatial relationships that are treated as jointly Gaussian, multivariate random variables and parameterized using a mean vector of positions and Euler angles and an associated covariance matrix. Smith, Self, and Cheeseman (1990) also proposes three operations that can extract information from this map that may not have been directly estimated. These three operations are pose composition (head-to-tail in (Smith, Self, and Cheeseman, 1990)), pose inverse, and relative pose estimation (tail-to-tail in (Smith, Self, and Cheeseman, 1990)), as shown in Fig. 5.3. This section reviews the pose representation and SSC operations formulated in (Smith, Self, and Cheeseman, 1990).

5.3.1 Pose Representation and the *Stochastic Map*

Under SSC notation, the relative transformation between the coordinate frame i and j, or the pose of coordinate frame j with respect to frame i, is denoted by

$$\mathbf{x}_{ij} = [x_{ij}, y_{ij}, z_{ij}, \phi_{ij}, \theta_{ij}, \psi_{ij}]^\top, \tag{5.15}$$

where x_{ij}, y_{ij}, and z_{ij} are position coordinates and ϕ_{ij}, θ_{ij}, and ψ_{ij} are Euler angles that encode the position and orientation of the coordinate frame j with respect to the frame i. However, because each relative transformation is derived from noisy measurements, our estimate is uncertain and we do not know the true value of $\mathbf{x}_{ij}$. Instead, we track a nominal mean value of its parameters and an associated 6×6 covariance matrix, $\hat{\mathbf{x}}_{ij}$ and $\boldsymbol{\Sigma}_{\mathbf{x}_{ij}}$, respectively.

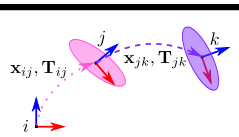
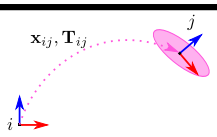
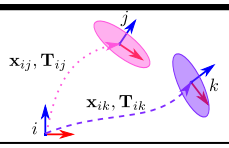
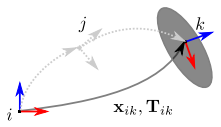
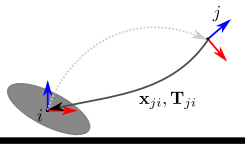
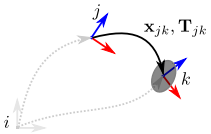
	Pose Composition	Pose Inverse	Relative Pose
Input			
Output			
SSC [5]	$\mathbf{x}_{ab} = [x_{ab}, y_{ab}, z_{ab}, \phi_{ab}, \theta_{ab}, \psi_{ab}]^\top$		
	$\mathbf{x} = [\mathbf{x}_{ij}^\top, \mathbf{x}_{jk}^\top]^\top \sim \mathcal{N}(\hat{\mathbf{x}}, \hat{\mathbf{\Sigma}})$	$\mathbf{x}_{ij} \sim \mathcal{N}(\hat{\mathbf{x}}_{ij}, \mathbf{\Sigma}_{\mathbf{x}_{ij}})$	$\mathbf{x} = [\mathbf{x}_{ij}^\top, \mathbf{x}_{ik}^\top]^\top \sim \mathcal{N}(\hat{\mathbf{x}}, \hat{\mathbf{\Sigma}})$
	$\hat{\mathbf{x}}_{ik} = \hat{\mathbf{x}}_{ij} \oplus \hat{\mathbf{x}}_{jk}$ $\mathbf{\Sigma}_{\mathbf{x}_{ik}} \approx J_{\oplus}(\hat{\mathbf{x}}) \mathbf{\Sigma} J_{\oplus}(\hat{\mathbf{x}})^\top$	$\hat{\mathbf{x}}_{ji} = \ominus \hat{\mathbf{x}}_{ij}$ $\mathbf{\Sigma}_{\mathbf{x}_{ji}} \approx J_{\ominus}(\hat{\mathbf{x}}_{ij}) \mathbf{\Sigma}_{\mathbf{x}_{ij}} J_{\ominus}(\hat{\mathbf{x}}_{ij})^\top$	$\hat{\mathbf{x}}_{jk} = (\ominus \hat{\mathbf{x}}_{ij}) \oplus \hat{\mathbf{x}}_{ik}$ $\mathbf{\Sigma}_{\mathbf{x}_{jk}} \approx J_{\ominus\oplus}(\hat{\mathbf{x}}) \mathbf{\Sigma} J_{\ominus\oplus}(\hat{\mathbf{x}})^\top$
[1]	$\mathbf{T}_{ab} = \exp(\boldsymbol{\xi}_{ab}^\wedge) \bar{\mathbf{T}}_{ab} \qquad \bar{\mathbf{T}}_{ab} \in \mathrm{SE}(3), \quad \boldsymbol{\xi}_{ab}^\wedge \in \mathfrak{se}(3)$		
	$\boldsymbol{\xi}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{ij})$ $\boldsymbol{\xi}_{jk} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{jk})$	–	–
	$\bar{\mathbf{T}}_{ik} \triangleq \bar{\mathbf{T}}_{ij} \bar{\mathbf{T}}_{jk}$ $\mathbf{\Sigma}_{ik} \approx \mathbf{\Sigma}_{ij} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} \mathbf{\Sigma}_{jk} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}$		
Proposed	$\mathbf{T}_{ab} = \exp(\boldsymbol{\xi}_{ab}^\wedge) \bar{\mathbf{T}}_{ab} \qquad \bar{\mathbf{T}}_{ab} \in \mathrm{SE}(3), \quad \boldsymbol{\xi}_{ab}^\wedge \in \mathfrak{se}(3)$		
	$\boldsymbol{\xi} = [\boldsymbol{\xi}_{ij}^\top, \boldsymbol{\xi}_{jk}^\top]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{ij} & \mathbf{\Sigma}_{ij,jk} \\ \mathbf{\Sigma}_{ij,jk}^\top & \mathbf{\Sigma}_{jk} \end{bmatrix}$	$\boldsymbol{\xi}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{ij})$	$\boldsymbol{\xi} = [\boldsymbol{\xi}_{ij}^\top, \boldsymbol{\xi}_{ik}^\top]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{ij} & \mathbf{\Sigma}_{ij,ik} \\ \mathbf{\Sigma}_{ij,ik}^\top & \mathbf{\Sigma}_{ik} \end{bmatrix}$
	$\bar{\mathbf{T}}_{ik} \triangleq \bar{\mathbf{T}}_{ij} \bar{\mathbf{T}}_{jk}$ $\mathbf{\Sigma}_{ik} \approx \mathbf{\Sigma}_{ij} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} \mathbf{\Sigma}_{jk} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} +$ $+ \mathbf{\Sigma}_{ij,jk} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}^\top + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} \mathbf{\Sigma}_{ij,jk}^\top$	$\bar{\mathbf{T}}_{ji} \triangleq \bar{\mathbf{T}}_{ij}^{-1}$ $\mathbf{\Sigma}_{ji} \approx \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ij} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top$	$\bar{\mathbf{T}}_{jk} \triangleq \bar{\mathbf{T}}_{ij}^{-1} \bar{\mathbf{T}}_{ik}$ $\mathbf{\Sigma}_{jk} \approx$ $\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ij} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ik} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top +$ $- \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ij,ik} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top - \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ij,ik}^\top \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top$

Figure 5.3: Summary of Lie Algebra/SSC Uncertainty Propogation Operations. Summary of the pose composition, pose inverse, and relative pose operations and their corresponding uncertainty propagation methods as proposed by Smith, Self, and Cheeseman (1990), Barfoot and Furgale (2014), and in this chapter. The indices i, j, and k correspond to specific coordinate frames of the robotic vehicle at different time steps or locations. SSC (Smith, Self, and Cheeseman, 1990) parameterizes the pose of frame b with respect to frame a using a vector of Euler angle and translation parameters $\mathbf{x}_{ab}$. They then model this vector of parameters as being drawn from a multivariate Guassian distribution with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{\Sigma}$. Under this model, they derive first order uncertainty propagation formulas for the pose composition, pose inverse, and relative pose operations. However, this coordinate based parameterization is unable to accurately model pose uncertainty because the parameter vector $\mathbf{x}_{ab}$ is not truly Gaussian. Barfoot and Furgale (2014) instead parameterize the pose of frame b with respect to frame a, using a mean element of the Special Euclidean group, $\bar{\mathbf{T}}_{ab}$, and an uncertain perturbation or noise parameter $\boldsymbol{\xi}_{ab}^\wedge$ defined in the Lie algebra $\mathfrak{se}(3)$. This enables them to model $\boldsymbol{\xi}_{ab}$ using a Gaussian distribution and accurately take into account the non-linear structure of the group. However, (Barfoot and Furgale, 2014) assumes the poses are independent and focuses primarily on the pose composition operation. The **primary contribution** of this chapter is the extension of the method presented by Barfoot and Furgale (2014) to jointly correlated poses as well as the derivation of the pose inverse and relative pose operations while taking advantage of the Lie algebra to characterize uncertainty.

The *stochastic map*, proposed in (Smith, Self, and Cheeseman, 1990), treats all of the uncertain transformations as jointly Gaussian multivariate random variables by stacking them into a single state vector $\mathbf{x}$ and tracking the mean and covariance of that state. Assuming we have n relative transformations we want to track, if we index them from $a = 1, \ldots, n$, then $\mathbf{x}$ and its associated mean and covariance are:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}, \hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_n \end{bmatrix}, \hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{x}_1} & \cdots & \boldsymbol{\Sigma}_{\mathbf{x}_1\mathbf{x}_n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\mathbf{x}_n\mathbf{x}_1} & \cdots & \boldsymbol{\Sigma}_{\mathbf{x}_n} \end{bmatrix} \tag{5.16}$$

where $\mathbf{x}$ is a vector of length $6n$, $\boldsymbol{\Sigma}_{\mathbf{x}_a}$ is the 6×6 covariance matrix of the relative transformation $\mathbf{x}_a$ and the off diagonal blocks of the form $\boldsymbol{\Sigma}_{\mathbf{x}_a\mathbf{x}_b}$ are the respective 6×6 cross covariance matrices.

5.3.2 Pose Composition (Head-to-Tail)

Suppose we are given a noisy observation of a robot's relative pose between time steps i and j ($\mathbf{x}_{ij}$) and another observation of its relative pose between time steps j and k ($\mathbf{x}_{jk}$), we may want to calculate the relative pose between time steps i and k ($\mathbf{x}_{ik}$) by composing the observations $\mathbf{x}_{ij}$ and $\mathbf{x}_{jk}$ (see Fig. 5.3). The SSC *head-to-tail* operation is a nonlinear function $f_{\oplus} : \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^6$ that takes the parameter vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{jk}$ as input and outputs the parameter vector $\mathbf{x}_{ik}$ that results from composing the respective homogeneous transformation matrices. The $\oplus$ operator denotes this operation:

$$\mathbf{x}_{ik} \triangleq \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} = f_{\oplus}(\mathbf{x}_{ij}, \mathbf{x}_{jk}). \tag{5.17}$$

The mean and covariance of the resulting pose are estimated up-to first order as

$$\hat{\mathbf{x}}_{ik} = \hat{\mathbf{x}}_{ij} \oplus \hat{\mathbf{x}}_{jk} \tag{5.18}$$

and

$$\boldsymbol{\Sigma}_{\mathbf{x}_{ik}} \approx J_{\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{jk})\ \hat{\boldsymbol{\Sigma}}\ J_{\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{jk})^{\top} \tag{5.19}$$

where $J_{\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{jk})$ is the Jacobian of $f_{\oplus}$ at $\hat{\mathbf{x}}_{ij}$ and $\hat{\mathbf{x}}_{jk}$, and

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{x}_{ij}} & \boldsymbol{\Sigma}_{\mathbf{x}_{ij}\mathbf{x}_{jk}} \\ \boldsymbol{\Sigma}^{\top}_{\mathbf{x}_{ij}\mathbf{x}_{jk}} & \boldsymbol{\Sigma}_{\mathbf{x}_{jk}} \end{bmatrix}. \tag{5.20}$$

5.3.3 Pose Inverse

Suppose we are given a robotic vehicle that is tasked with navigating through an *a priori* unknown environment after which it must return to the origin location. While it is common to represent the robotic vehicle's current location with respect to the origin using estimation theory, it may be useful to instead characterize the pose of the origin with respect to the local robot coordinate frame. Formally, given an uncertain estimate of the pose of coordinate frame j with respect to frame i ($\mathbf{x}_{ij}$), we want to determine the pose of frame i with respect to frame j ($\mathbf{x}_{ji}$). This amounts to finding the vector $\hat{\mathbf{x}}_{ji}$ that corresponds to the inverse of $\hat{\mathbf{x}}_{ij}$ (in terms of homogeneous transformation matrices) and representing the uncertainty with respect to this new frame of reference.

The SSC *pose inverse* operation is a nonlinear function $f_{\ominus} : \mathbb{R}^6 \to \mathbb{R}^6$ that takes a pose and computes the inverse of the pose as a homogeneous transformation matrix:

$$\mathbf{x}_{ji} \triangleq \ominus \mathbf{x}_{ij} = f_{\ominus}(\mathbf{x}_{ij}). \tag{5.21}$$

The mean and covariance of the resulting pose are estimated up-to first order as

$$\hat{\mathbf{x}}_{ji} = \ominus \hat{\mathbf{x}}_{ij} \tag{5.22}$$

and

$$\boldsymbol{\Sigma}_{\mathbf{x}_{ji}} \approx J_{\ominus}(\hat{\mathbf{x}}_{ij}) \; \boldsymbol{\Sigma}_{\mathbf{x}_{ij}} \; J_{\ominus}(\hat{\mathbf{x}}_{ij})^{\top} \tag{5.23}$$

where $J_{\ominus}(\hat{\mathbf{x}}_{ij})$ is the Jacobian of $f_{\ominus}$ at $\hat{\mathbf{x}}_{ij}$.

5.3.4 Relative Pose (Tail-to-Tail)

Finally, the SSC *tail-to-tail* operation takes uncertain estimates of two coordinate frames with respect to a single origin frame and evaluates the relative pose between them. More succinctly, given $\mathbf{x}_{ij}$ and $\mathbf{x}_{ik}$, find $\mathbf{x}_{jk}$. This operation is useful when using a method such as pose graph simultaneous localization and mapping (SLAM) for navigation since the robot pose at multiple time steps is often estimated with respect to a single fixed coordinate frame.

The SSC *relative pose* operation is a nonlinear function $f_{\ominus\oplus} : \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^6$ that is defined by first applying the inverse and then the head-to-tail operation:

$$\mathbf{x}_{jk} \triangleq (\ominus \mathbf{x}_{ij}) \oplus \mathbf{x}_{ik} = f_{\ominus\oplus}(\mathbf{x}_{ij}, \mathbf{x}_{ik}). \tag{5.24}$$

The mean and covariance of the resulting pose up-to first order are estimated as

$$\hat{\mathbf{x}}_{jk} = (\ominus \hat{\mathbf{x}}_{ij}) \oplus \hat{\mathbf{x}}_{ik} \tag{5.25}$$

and

$$\boldsymbol{\Sigma}_{\mathbf{x}_{jk}} \approx J_{\ominus\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{ik})\ \hat{\boldsymbol{\Sigma}}\ J_{\ominus\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{ik})^{\top} \tag{5.26}$$

where $J_{\ominus\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{ik})$ is the Jacobian of $f_{\ominus\oplus}$ at $\hat{\mathbf{x}}_{ij}$ and $\hat{\mathbf{x}}_{ik}$, and

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{x}_{ij}} & \boldsymbol{\Sigma}_{\mathbf{x}_{ij}\mathbf{x}_{ik}} \\ \boldsymbol{\Sigma}^{\top}_{\mathbf{x}_{ij}\mathbf{x}_{ik}} & \boldsymbol{\Sigma}_{\mathbf{x}_{ik}} \end{bmatrix}. \tag{5.27}$$

5.4 Jointly Characterizing Uncertainty in the Algebra

While the SSC operations proposed in (Smith, Self, and Cheeseman, 1990) have been used widely since they were introduced, over the past decade Lie algebra based methods have been shown to provide a more accurate characterization of uncertainty (Barfoot and Furgale, 2014; Kim and Kim, 2017). However, while recent years have seen an increase in use of Lie algebra based methods for uncertainty propagation (Forster et al., 2017; Hartley et al., 2018; Wheeler et al., 2018), existing methods assume that individual measurements are independent (Barfoot and Furgale, 2014), which may not be the case when the underlying pose estimates are derived from a SLAM solution. We now describe how this assumption can be dropped and poses can be modeled as jointly correlated within the Lie algebra space.

Assume that we have n uncertain poses $\{\mathbf{T}_1, \ldots, \mathbf{T}_n\}$, each of which is defined according to (5.14). If we assume that the poses are statistically independent, then we can parameterize this distribution of poses with the set of associated mean and covariance matrices $\{\bar{\mathbf{T}}_1, \boldsymbol{\Sigma}_1, \ldots, \bar{\mathbf{T}}_n, \boldsymbol{\Sigma}_n\}$. However, if the uncertainty associated with the set of poses is correlated, then this can be modeled by concatenating the set of perturbation vectors $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n\}$ into a single vector $\boldsymbol{\xi}_{1:n} \in \mathbb{R}^{6n}$ and representing the uncertainty of the distribution using a single covariance matrix:

$$\boldsymbol{\xi}_{1:n} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \vdots \\ \boldsymbol{\xi}_n \end{bmatrix}, \quad \boldsymbol{\Sigma}_{1:n} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \ldots, & \boldsymbol{\Sigma}_{1,n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}^{\top}_{1,n} & \ldots & \boldsymbol{\Sigma}_n \end{bmatrix}, \tag{5.28}$$

where $\boldsymbol{\xi}_{1:n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{1:n})$. Thus, we can parameterize the distribution of n poses using the set of each of their means

$$\bar{\mathbf{T}}_{1:n} = \{\bar{\mathbf{T}}_1, \ldots, \bar{\mathbf{T}}_n\} \tag{5.29}$$

and the covariance matrix $\boldsymbol{\Sigma}_{1:n}$. Note that this only extends because we define the uncertainty in the Lie algebra. The next three sections describe how to apply the composition, inverse, and relative pose operations on uncertain poses represented in this manner.

5.5 Jointly Distributed Pose Composition

This section describes how to compose two uncertain poses whose associated perturbation vectors are jointly Gaussian in the Lie algebra as described in the last section. This is the Lie group-based or coordinate free equivalent to the SSC head-to-tail operation described in Section 5.3.2.

5.5.1 Pose Composition Operation Derivation

Suppose we have two uncertain poses $\mathbf{T}_{ij}$ and $\mathbf{T}_{jk}$ with perturbations $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{\xi}_{jk}$ that are jointly Gaussian in the Lie algebra with covariance matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ij} & \boldsymbol{\Sigma}_{ij,jk} \\ \boldsymbol{\Sigma}_{ij,jk}^{\top} & \boldsymbol{\Sigma}_{jk} \end{bmatrix}.$$

Our goal is to find the mean and covariance of $\mathbf{T}_{ik}$, $\{\bar{\mathbf{T}}_{ik}, \boldsymbol{\Sigma}_{ik}\}$. Under the standard group multiplication operation:

$$\mathbf{T}_{ik} = \mathbf{T}_{ij}\mathbf{T}_{jk}. \tag{5.30}$$

Following the random variable definition in (5.14) and using the property described in (5.10),

$$\begin{aligned} \exp(\boldsymbol{\xi}_{ik}^{\wedge})\bar{\mathbf{T}}_{ik} &= \exp(\boldsymbol{\xi}_{ij}^{\wedge})\bar{\mathbf{T}}_{ij}\exp(\boldsymbol{\xi}_{jk}^{\wedge})\bar{\mathbf{T}}_{jk} \\ &= \exp(\boldsymbol{\xi}_{ij}^{\wedge})\exp((\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}\boldsymbol{\xi}_{jk})^{\wedge})\bar{\mathbf{T}}_{ij}\bar{\mathbf{T}}_{jk} \end{aligned} \tag{5.31}$$

where $\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}$ is the matrix form of the adjoint action of $\bar{\mathbf{T}}_{ij}$ on $\mathfrak{se}(3)$. Letting

$$\bar{\mathbf{T}}_{ik} \triangleq \bar{\mathbf{T}}_{ij}\bar{\mathbf{T}}_{jk}, \tag{5.32}$$

gives us

$$\exp(\boldsymbol{\xi}_{ik}^{\wedge}) = \exp(\boldsymbol{\xi}_{ij}^{\wedge})\exp((\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}\boldsymbol{\xi}_{jk})^{\wedge}). \tag{5.33}$$

We can now use the BCH formula (5.13) to show that

$$\boldsymbol{\xi}_{ik} = \boldsymbol{\xi}_{ij} + \boldsymbol{\xi}'_{jk} + \frac{1}{2}\boldsymbol{\xi}_{ij}^{\curlywedge}\boldsymbol{\xi}'_{jk} + \frac{1}{12}\boldsymbol{\xi}_{ij}^{\curlywedge}\boldsymbol{\xi}_{ij}^{\curlywedge}\boldsymbol{\xi}'_{jk} + \frac{1}{12}\boldsymbol{\xi}'^{\curlywedge}_{jk}\boldsymbol{\xi}'^{\curlywedge}_{jk}\boldsymbol{\xi}_{ij} + \\ - \frac{1}{24}\boldsymbol{\xi}'^{\curlywedge}_{jk}\boldsymbol{\xi}_{ij}^{\curlywedge}\boldsymbol{\xi}_{ij}^{\curlywedge}\boldsymbol{\xi}'_{jk} + \dots \tag{5.34}$$

where $\boldsymbol{\xi}'_{jk} = \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}\boldsymbol{\xi}_{jk}$. Computing the covariance $\boldsymbol{\Sigma}$ amounts to evaluating $E[\boldsymbol{\xi}_{ik}\boldsymbol{\xi}_{ik}^{\top}]$. Multiplying out up-to fourth order, we get:

$$E[\boldsymbol{\xi}_{ik}\boldsymbol{\xi}_{ik}^{\top}] \approx E\Bigg[\underbrace{\boldsymbol{\xi}_{ij}\boldsymbol{\xi}_{ij}^{\top} + \boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^{\top}_{jk}}_{\text{2nd Order Diag. Terms}} + \underbrace{\boldsymbol{\xi}_{ij}\boldsymbol{\xi}'^{\top}_{jk} + \boldsymbol{\xi}'_{jk}\boldsymbol{\xi}_{ij}^{\top}}_{\text{2nd Order Cross Terms}} + \tag{5.35}$$

$$+\frac{1}{12}((\boldsymbol{\xi}_{ij}^{\curlywedge}\boldsymbol{\xi}_{ij}^{\curlywedge})(\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^{\top}_{jk}) + (\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^{\top}_{jk})(\boldsymbol{\xi}_{ij}^{\curlywedge}\boldsymbol{\xi}_{ij}^{\curlywedge})^{\top} + \\ + \quad (\boldsymbol{\xi}'^{\curlywedge}_{jk}\boldsymbol{\xi}'^{\curlywedge}_{jk})(\boldsymbol{\xi}_{ij}\boldsymbol{\xi}_{ij}^{\top}) + (\boldsymbol{\xi}_{ij}\boldsymbol{\xi}_{ij}^{\top})(\boldsymbol{\xi}'^{\curlywedge}_{jk}\boldsymbol{\xi}'^{\curlywedge}_{jk})^{\top}) + \tag{5.36}$$

$$+\underbrace{\frac{1}{4}\boldsymbol{\xi}_{ij}^{\curlywedge}(\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^{\top}_{jk})\boldsymbol{\xi}_{ij}^{\curlywedge\top} +}_{\text{4th Order Diagonal Terms}}$$

$$+\frac{1}{12}((\boldsymbol{\xi}_{ij}\boldsymbol{\xi}'^{\top}_{jk})(\boldsymbol{\xi}_{ij}^{\curlywedge\top}\boldsymbol{\xi}_{ij}^{\curlywedge\top}) + (\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}_{ij}^{\top})(\boldsymbol{\xi}'^{\curlywedge\top}_{jk}\boldsymbol{\xi}'^{\curlywedge\top}_{jk}) + \tag{5.37}$$

$$+\underbrace{(\boldsymbol{\xi}_{ij}^{\curlywedge}\boldsymbol{\xi}_{ij}^{\curlywedge})(\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}_{ij}^{\top}) + (\boldsymbol{\xi}'^{\curlywedge}_{jk}\boldsymbol{\xi}'^{\curlywedge}_{jk})(\boldsymbol{\xi}_{ij}\boldsymbol{\xi}'^{\top}_{jk}))}_{\text{4th Order Cross Terms}}\Bigg].$$

This derivation is identical to the one proposed in (Barfoot and Furgale, 2014) until the last step. Since (Barfoot and Furgale, 2014) assumes the individual poses are independent, the cross terms in (5.35) and (5.37) are zero and the evaluation of the 4th order terms in (5.36) is simplified. If this assumption is true and $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{\xi}'_{jk}$ are independent of one another, as is the case when $\mathbf{T}_{ij}$ and $\mathbf{T}_{jk}$ are independent measurements of consecutive robot motion, then the cross terms in (5.36) and the covariance $\boldsymbol{\Sigma}$ can be calculated up to 4th order as described in (Barfoot and Furgale, 2014). However, if the poses $\mathbf{T}_{ij}$ and $\mathbf{T}_{jk}$ are correlated, as is often the case when they are derived from the solution of a maximum likelihood estimate (MLE) problem such as Pose Graph SLAM or in the case of wheel slip, the cross terms must be included and evaluation of the fourth order terms in (5.36) and (5.37) becomes more difficult[1]. If the correlation is not taken into account, then the result of the pose composition operation under-approximates the true distribution and consistency is lost, as shown in Fig. 5.4.

[1]If increased accuracy is needed, then Isserlis Theorem can be applied to evaluate the fourth order terms in (5.36) and (5.37).

A first order estimate of covariance (second order in the perturbation variables) can be obtained be evaluating

$$\begin{aligned} E[\boldsymbol{\xi}_{ik}\boldsymbol{\xi}_{ik}^{\top}] \approx & E[\boldsymbol{\xi}_{ij}\boldsymbol{\xi}_{ij}^{\top}] + E[\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^{\top}_{jk}] + \\ & + E[\boldsymbol{\xi}_{ij}\boldsymbol{\xi}'^{\top}_{jk}] + E[\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}_{ij}^{\top}], \end{aligned} \tag{5.38}$$

resulting in

$$\begin{aligned} \boldsymbol{\Sigma}_{ik} \approx & \boldsymbol{\Sigma}_{ij} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}\boldsymbol{\Sigma}_{jk}\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}^{\top} + \\ & + \boldsymbol{\Sigma}_{ij,jk}\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}^{\top} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}\boldsymbol{\Sigma}_{ij,jk}^{\top}. \end{aligned} \tag{5.39}$$

Thus, first order pose composition on the Lie algebra can be performed for correlated poses using (5.32) for mean propagation and (5.39) for covariance propagation. As noted previously, this is equivalent to the first order method presented in (Barfoot and Furgale, 2014) with the exception of the two additional cross terms in (5.39) needed to model cross correlation.

5.5.2 Accurately Modeling Group Structure

While both the SSC operations presented in (Smith, Self, and Cheeseman, 1990) and the Lie algebra based methods presented in this chapter (and Σ_{2nd} in (Barfoot and Furgale, 2014, (55))) are first order approximation methods (in terms of covariance), characterizing rotation uncertainty in the Lie algebra space results in a more accurate characterization of pose uncertainty. This is because Euler angle parameterization is a chart and does not cover the entire manifold whereas the Lie algebra inherently captures the group structure and can model any arbitrary group element. In addition, since the Lie algebra is a vector space (Tapp, 2016), modeling the uncertainty as a Gaussian distribution is convenient and well-defined.

To demonstrate this, we performed an experiment similar to the one proposed in (Barfoot and Furgale, 2014). We generated a sequence of N noisy pose transformations of the form $\mathbf{T}_{ab} = \exp(\boldsymbol{\xi}_{ab}^{\wedge})\bar{\mathbf{T}}_{ab}$, with

$$\boldsymbol{\xi}_{ab} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{ab}), \tag{5.40}$$

$$\boldsymbol{\Sigma}_{ab} = \mathrm{diag}([0.001\sigma_t, 1e^{-5}\sigma_t, 1e^{-5}, 1e^{-5}, 1e^{-5}, 0.003\sigma_r]), \tag{5.41}$$

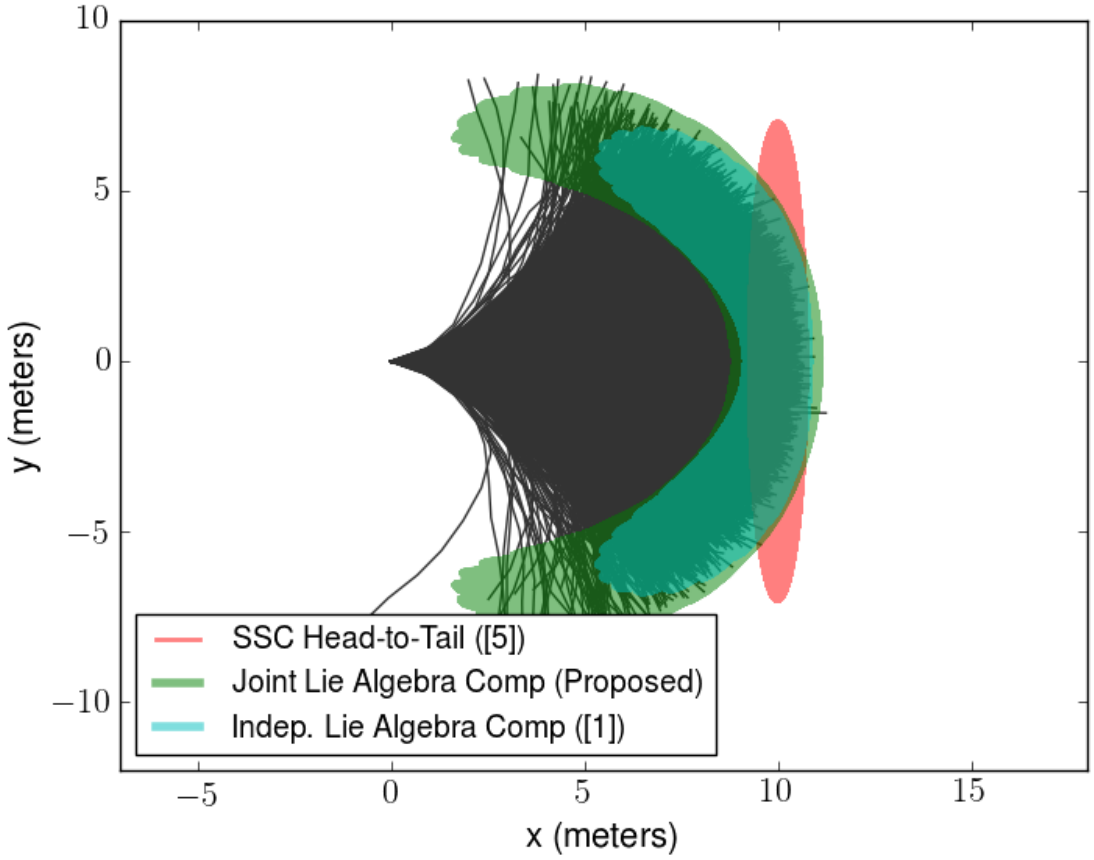


Figure 5.4: Example Pose Composition Comparison. Plots of 10000 sample trajectories each made up of 10 noisy pose transformations. The 95% likely uncertainty ellipse predicted by first order uncertainty propagation through the SSC head-to-tail operation is shown in red, while a representation of the flattened 95% likely uncertainty position ellipsoids predicted by the Lie algebra pose composition methods when correlation is and is not taken into account are shown in green and cyan, respectively.

and

$$\bar{\mathbf{T}}_{ab} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.42}$$

where σ_t and σ_r are scaling parameters for the translation and rotation noise respectively. In addition, each consecutive perturbation variable $\boldsymbol{\xi}_{ab}$ was drawn such that it was correlated with the perturbation variable before it with correlation coefficient ρ.

We then composed these transformations end-to-end and repeated the process to generate

10000 sample trajectories. These Monte-Carlo simulation results were then used to evaluate the uncertainty predicted by the SSC head-to-tail operation as well as the Lie-algebra based pose composition method derived in this section, both with and without correlation being taken into account. A top down view of one such experiment with $N = 10$, $\sigma_t = 5$, $\sigma_r = 5$, and $\rho = 0.4$ is shown in Fig. 5.4.

As expected, the Lie algebra based method significantly outperforms SSC head-to-tail because it takes into account the structure of the SE(3) group. In addition, it is easily seen that dropping the cross covariance terms results in an under-approximation of the true covariance, if positive correlation is indeed present. A more thorough investigation of this experiment is described in Section 5.9.1.

In the next two sections, we provide a derivation of the inverse and relative pose operations on the Lie algebra, which as far as we know has not been previously published.

5.6 The Pose Inverse Operation

The pose inverse operation corresponds to a change in reference frame. Given an uncertain pose distribution $\mathbf{T}_{ij}$ that represents the pose of coordinate frame j with respect to frame i, we want to find the inverse $\mathbf{T}_{ji}$ that represents the pose of coordinate frame i with respect to frame j. The distribution of an inverse pose can be derived as follows (Eade, 2017):

$$\begin{aligned}\mathbf{T}_{ji} &= \mathbf{T}_{ij}^{-1} \\ &= \bar{\mathbf{T}}_{ij}^{-1} \exp(-\boldsymbol{\xi}_{ij}^{\wedge}) \\ &= \exp((-\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\xi}_{ij})^{\wedge}) \bar{\mathbf{T}}_{ij}^{-1} \\ &= \exp(\boldsymbol{\xi}'_{ij}) \bar{\mathbf{T}}_{ij}^{-1}\end{aligned} \tag{5.43}$$

where $\boldsymbol{\xi}'_{ij}$ is the original perturbation $\boldsymbol{\xi}_{ij}$ transformed by the negative adjoint of $\bar{\mathbf{T}}_{ij}^{-1}$. Due to the linearity of the adjoint operation, we can represent the inverse distribution $\mathbf{T}_{ji}$ with mean and covariance

$$\bar{\mathbf{T}}_{ji} = \bar{\mathbf{T}}_{ij}^{-1} \tag{5.44}$$

and

$$\boldsymbol{\Sigma}_{ji} = \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\Sigma}_{ij} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top}. \tag{5.45}$$

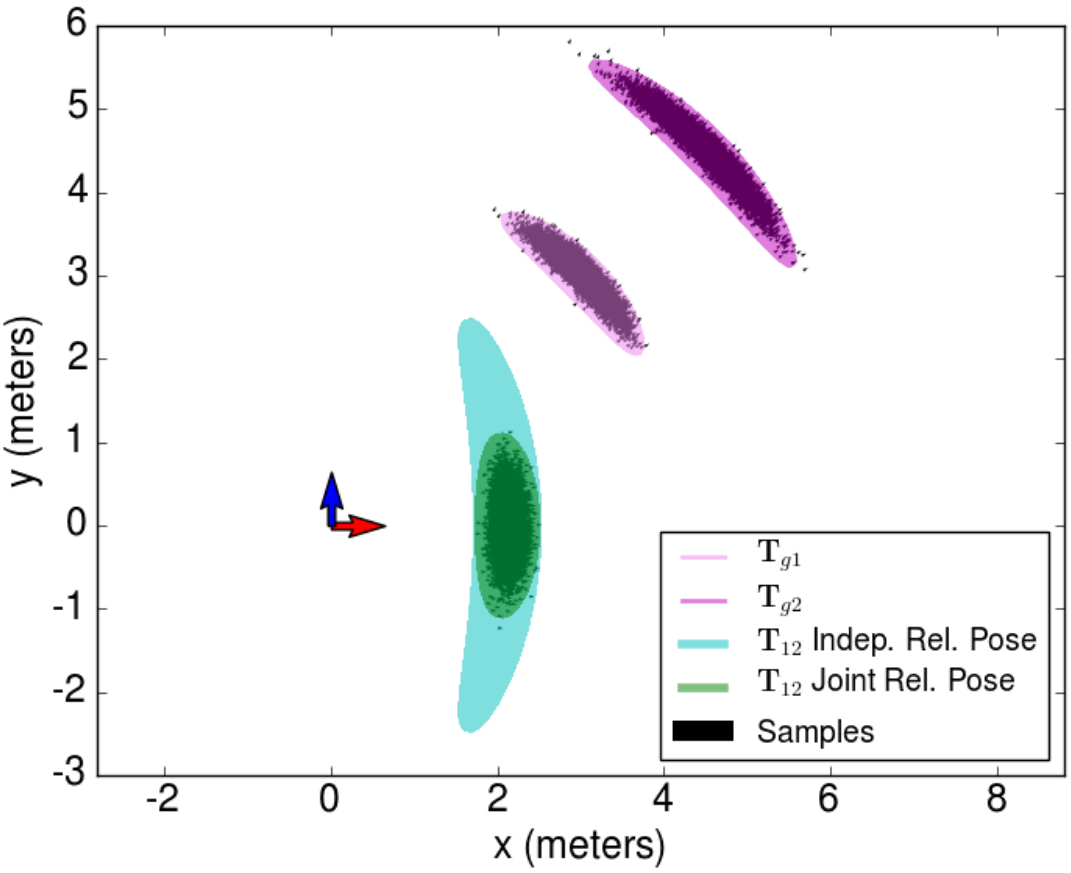


Figure 5.5: Example Relative Pose Operation Comparison. Example results from performing the relative pose operation while taking into account/ignoring correlation. Dropping the correlation terms in (5.51) can lead to over-estimation of uncertainty. In this example, we used our proposed method to estimate the relative pose $\mathbf{T}_{12}$ between two correlated poses $\mathbf{T}_{g1}$ and $\mathbf{T}_{g2}$. A flattened view of the predicted 95% likely uncertainty ellipsoids that result when correlation is ignored ((5.51) without the cross terms) and taken into account ((5.51) with all terms) are shown in cyan and green respectively. For this plot, $\alpha = 1$.

5.7 The Relative Pose Operation

This section combines the derivations from the previous two sections to formulate a first order method for characterizing the uncertainty of the relative pose operation or the coordinate-free equivalent to the SSC tail-to-tail operation described in Section 5.3.4.

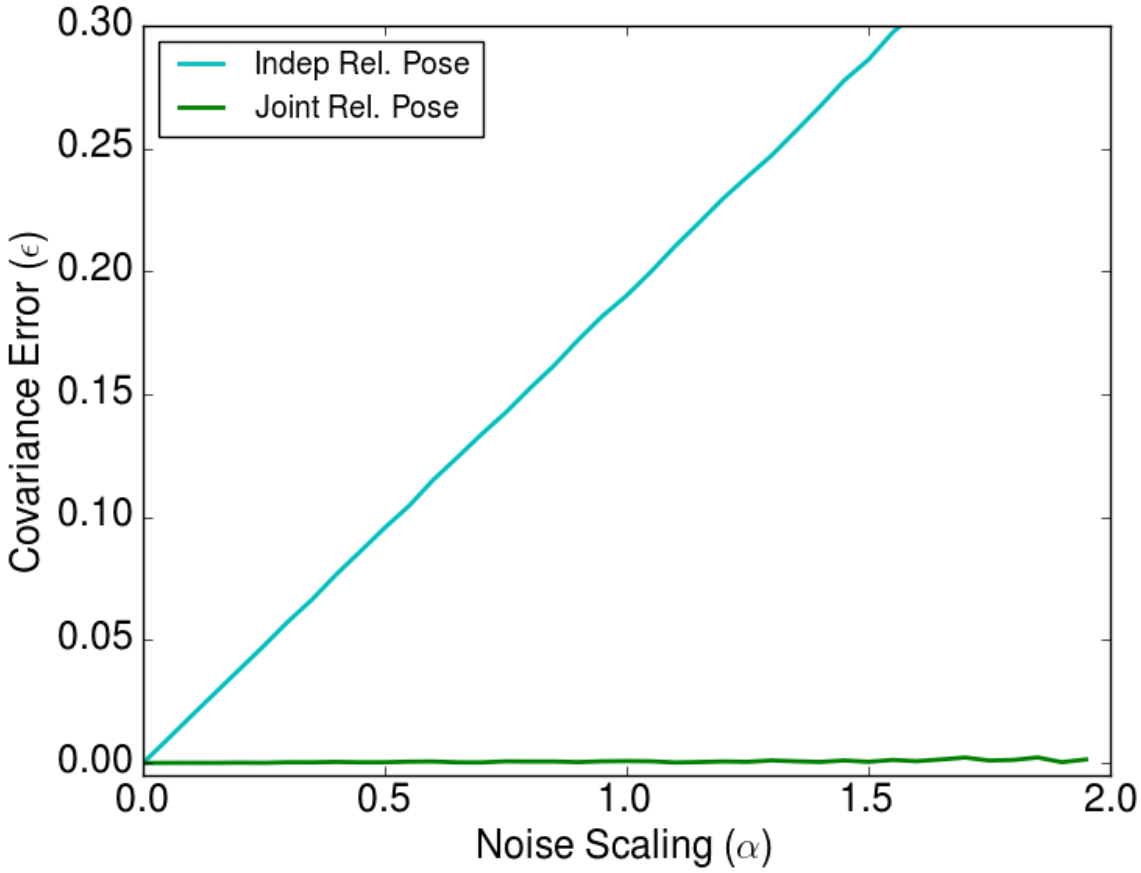


Figure 5.6: Covariance Error Comparison for Relative Pose. Covariance error comparison showing the importance of not ignoring correlation when propagating uncertainty.

5.7.1 Relative Pose Operation Derivation

Given uncertain estimates of two coordinate frames with respect to a common base frame, our aim is estimate the relative transformation between the two poses. For example, given possibly correlated uncertain transformations $\mathbf{T}_{ij}$ and $\mathbf{T}_{ik}$ representing the poses of coordinate frames j and k with respect to frame i, we want to find the mean $\bar{\mathbf{T}}_{jk}$ and covariance $\boldsymbol{\Sigma}_{jk}$ that parameterize the pose uncertainty of frame k with respect to frame j.

We start by assuming that the $\bar{\mathbf{T}}_{ij}$ and $\bar{\mathbf{T}}_{ik}$ are known and that the associated perturbations $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{\xi}_{ik}$ are jointly correlated in the Lie algebra with known covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ij} & \boldsymbol{\Sigma}_{ij,ik} \\ \boldsymbol{\Sigma}_{ij,ik}^{\top} & \boldsymbol{\Sigma}_{ik} \end{bmatrix}.$$

Under the standard group inverse and multiplication actions, the following must hold:

$$\mathbf{T}_{jk} = \mathbf{T}_{ij}^{-1}\mathbf{T}_{ik}. \tag{5.46}$$

Expanding (5.46) using the random variable definition in (5.14) and the derivation in (5.43) results in

$$\begin{aligned}\exp(\boldsymbol{\xi}_{jk}^{\wedge})\bar{\mathbf{T}}_{jk} &= \bar{\mathbf{T}}_{ij}^{-1}\exp(-\boldsymbol{\xi}_{ij}^{\wedge})\exp(\boldsymbol{\xi}_{ik}^{\wedge})\bar{\mathbf{T}}_{ik} \\ &= \exp((-\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}\boldsymbol{\xi}_{ij})^{\wedge})\bar{\mathbf{T}}_{ij}^{-1}\exp(\boldsymbol{\xi}_{ik}^{\wedge})\bar{\mathbf{T}}_{ik} \\ &= \exp(\boldsymbol{\xi}_{ij}'^{\wedge})\exp(\boldsymbol{\xi}_{ik}'^{\wedge})\bar{\mathbf{T}}_{ij}^{-1}\bar{\mathbf{T}}_{ik},\end{aligned} \tag{5.47}$$

where $\boldsymbol{\xi}_{ij}' = -\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}\boldsymbol{\xi}_{ij}$ and $\boldsymbol{\xi}_{ik}' = \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}\boldsymbol{\xi}_{ik}$. Letting

$$\bar{\mathbf{T}}_{jk} \triangleq \bar{\mathbf{T}}_{ij}^{-1}\bar{\mathbf{T}}_{ik}, \tag{5.48}$$

gives us

$$\exp(\boldsymbol{\xi}_{jk}^{\wedge}) = \exp(\boldsymbol{\xi}_{ij}'^{\wedge})\exp(\boldsymbol{\xi}_{ik}'^{\wedge}). \tag{5.49}$$

Expanding the BCH formula in a similar manner to (5.34) and taking the expectation as in (5.35) results in the following up to first order:

$$\begin{aligned}E[\boldsymbol{\xi}_{jk}\boldsymbol{\xi}_{jk}^{\top}] \approx & E[\boldsymbol{\xi}_{ij}'\boldsymbol{\xi}_{ij}'^{\top}] + E[\boldsymbol{\xi}_{ik}'\boldsymbol{\xi}_{ik}'^{\top}] + \\ & + E[\boldsymbol{\xi}_{ij}'\boldsymbol{\xi}_{ik}'^{\top}] + E[\boldsymbol{\xi}_{ik}'\boldsymbol{\xi}_{ij}'^{\top}].\end{aligned} \tag{5.50}$$

Evaluating the expectations in (5.50) results in

$$\begin{aligned}\boldsymbol{\Sigma}_{jk} \approx & \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}\boldsymbol{\Sigma}_{ij}\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}\boldsymbol{\Sigma}_{ik}\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top} - \\ & + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}\boldsymbol{\Sigma}_{ij,ik}\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top} - \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}\boldsymbol{\Sigma}_{ij,ik}^{\top}\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top}.\end{aligned} \tag{5.51}$$

Thus, uncertainty can be propagated through the relative pose function via (5.48) and (5.51). A summary of the uncertainty propagation methods derived in the last three sections is provided in Fig. 5.3.

5.7.2 Ignoring Correlation Leads to Inconsistency

Ignoring correlation of the associated perturbation variables leads to under/over-estimation of uncertainty, depending on if the correlation is positive or negative and if the operation being performed is pose composition or relative pose estimation. Fig. 5.5 and Fig. 5.6 show

an example of this for the case of relative pose with positive correlation.

To create these plots, we generated $M = 10000$ sets of two uncertain poses $\mathbf{T}_{g1}^m = \exp(\boldsymbol{\xi}_{g1}^{m\wedge})\bar{\mathbf{T}}_{g1}$ and $\mathbf{T}_{g2} = \exp(\boldsymbol{\xi}_{g2}^{m\wedge})\bar{\mathbf{T}}_{g2}$, with mean values

$$\bar{\mathbf{T}}_{g1} = \begin{bmatrix} 0.707107 & -0.707107 & 0 & 3 \\ 0.707107 & 0.707107 & 0 & 3 \\ -0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.52}$$

and

$$\bar{\mathbf{T}}_{g2} = \begin{bmatrix} 0.707107 & -0.707107 & 0 & 4.5 \\ 0.707107 & 0.707107 & 0 & 4.5 \\ -0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.53}$$

under the assumption that the perturbation variables $(\boldsymbol{\xi}_{g1}^m)$ and $(\boldsymbol{\xi}_{g2}^m)$ are jointly correlated in the Lie algebra with marginal covariance matrices

$$\begin{aligned} \boldsymbol{\Sigma}_{g1} &= \boldsymbol{\Sigma}_{g2} \\ &= \alpha \cdot \operatorname{diag}([0.005, 0.005, 1e-5, 1e-5, 1e-5, 0.006]), \end{aligned} \tag{5.54}$$

and cross covariance

$$\boldsymbol{\Sigma}_{g1,g2} = \alpha \cdot \operatorname{diag}([0.0005, 0.0005, 0, 0, 0, 0.005]), \tag{5.55}$$

where α is a scaling parameter. We then used the relative pose operation presented in this section to estimate $\mathbf{T}_{12}$ both with and without taking uncertainty into account.

In Fig. 5.6, we evaluate the estimated covariance error for increasing values of α with respect to the Monte-Carlo simulation under the following metric

$$\epsilon \triangleq \sqrt{\operatorname{tr}\left((\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{mc})^\top (\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{mc})\right)}, \tag{5.56}$$

where

$$\boldsymbol{\Sigma}_{mc} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\xi}_m \boldsymbol{\xi}_m^\top, \tag{5.57}$$

with

$$\mathbf{T}_m = (\exp(\boldsymbol{\xi}_{g1}^{m\wedge})\bar{\mathbf{T}}_{g1})^{-1} \exp(\boldsymbol{\xi}_{g2}^{m\wedge})\bar{\mathbf{T}}_{g2} \tag{5.58}$$

and

$$\boldsymbol{\xi}_m = \log(\mathbf{T}_m \bar{\mathbf{T}}_{12}^{-1})^{\vee}. \tag{5.59}$$

Fig. 5.4, Fig. 5.5, and Fig. 5.6 show that ignoring correlation leads to inconsistent estimates of uncertainty.

5.8 Converting to a Lie algebra Based Representation

Although uncertainty characterization in the Lie algebra is more accurate than coordinate based methods, in some cases an existing estimation algorithm may output pose estimates in an alternative parameterization. This section describes how to convert from an existing parameterization such as the SSC multi-variate Gaussian representation (defined in (5.15) and (5.16)) to the proposed representation, as well as how to extract the mean and covariance of a jointly correlated set of poses from a MLE solution such as iSAM (Kaess, Ranganathan, and Dellaert, 2008) or g2o (Kümmerle et al., 2011) directly.

5.8.1 Converting from a Coordinate Based Representation

Assuming we are given a mean parameter vector $\hat{\mathbf{x}}$ and associated covariance $\hat{\boldsymbol{\Sigma}}$ as defined in (5.16), to apply our proposed propagation methods, we must first transform $\hat{\mathbf{x}}$ and $\hat{\boldsymbol{\Sigma}}$ into our proposed representation such that the means of the associated transformations are represented in the group space via $\bar{\mathbf{T}}_{1:n}$ and the perturbations are represented as multivariate Guassian random variables, $\boldsymbol{\xi}_{1:n}$, with covariance $\boldsymbol{\Sigma}_{1:n}$ in the associated Lie algebra as defined in (5.28) and (5.29). Converting from a coordinates based method for uncertainty characterization to the Lie algebra representation is a non-linear operation. As such, the Unscented Transform (UT) (Julier, 2002) is a better choice for converting between representations than a first order linearization method.

The UT, first proposed by Julier and Uhlmann (1997), takes an input Gaussian distribution defined in one space and a non-linear function that maps the input space to an output space and finds a Gaussian approximation for the transformed distribution. This is done by using the input mean and covariance to create a deterministic set $\mathcal{X}_{1:K}$ of $K = 2m+1$ sigma points, where m is the dimension of the input space. These sigma points are then passed through the non-linear function and a weighted mean and sample covariance is used to find a Gaussian approximation to the output distribution.

Assuming we have a function $f : \mathbb{R}^d \mapsto \mathcal{G}$ that maps from a parameter vector, $\mathbf{x}_i$, to an element of the Lie group, $\mathcal{G}$, we can define a function ℓ_i that takes a perturbed parameter

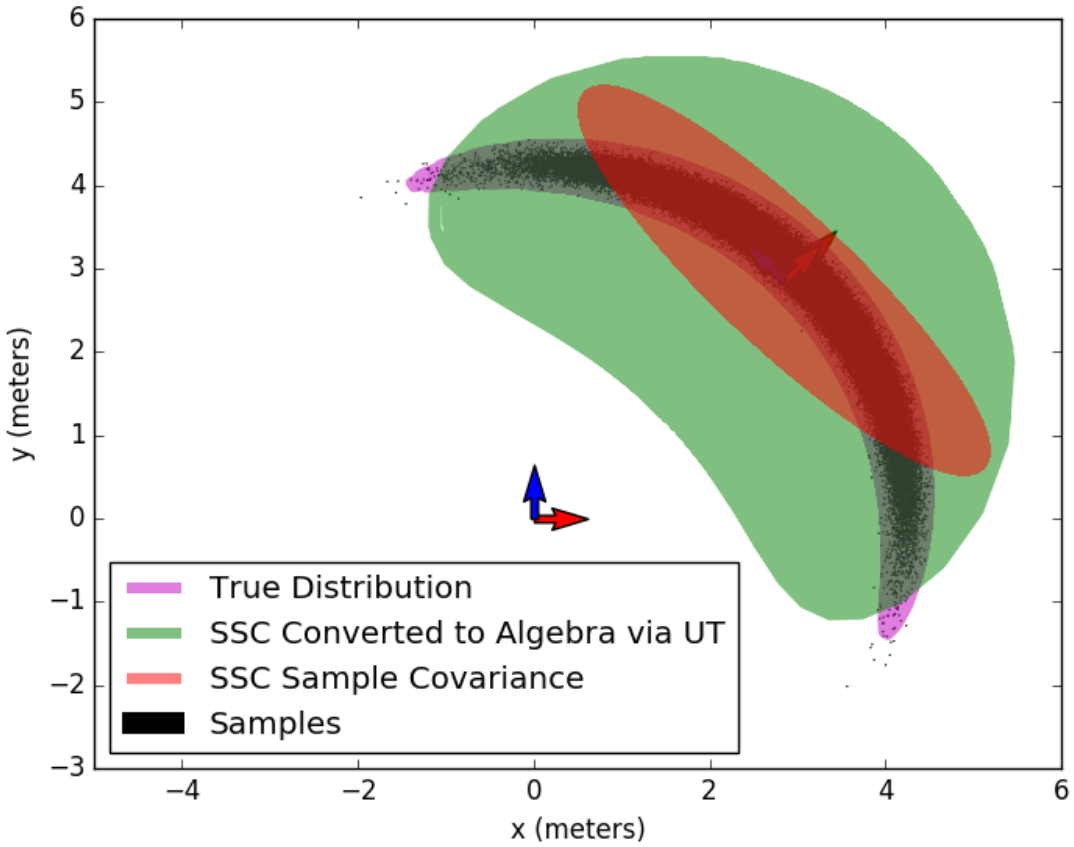


Figure 5.7: Example Conversion from SSC Using the Unscented Transfrom. The Unscented Transform (UT) (Julier, 2002) can be used to transform from an existing, coordinates based representation such as the one used in SSC (5.15) (shown in red) to the Lie algebra based representation (shown in green). However, the use of the SSC representation in the first place results in a loss of information and the resulting representation (shown in green) over approximates the true underlying distribution (shown in pink). All uncertainty bounds shown are 95% likely ellipsoids.

vector $\tilde{\mathbf{x}}_i$, centers it on the identity, and transforms it to the Lie algebra space as follows:

$$\ell_i(\tilde{\mathbf{x}}_i) = \log(f(\tilde{\mathbf{x}}_i) \cdot f(\hat{\mathbf{x}}_i)^{-1}), \tag{5.60}$$

where $\hat{\mathbf{x}}_i$ is the mean parameter vector associated with $\tilde{\mathbf{x}}_i$.

We can then use the standard UT (Julier and Uhlmann, 1997, (12)) to generate a set $\mathcal{X}_{1:K}$ of $2nd + 1$ sigma points from $\hat{\mathbf{x}}$ and $\hat{\mathbf{\Sigma}}$, where n is the number of modeled Lie group elements. After which, the predicted Lie group representation for the jointly correlated poses

can be obtained as follows:

$$\bar{\mathbf{T}}_i = f(\hat{\mathbf{x}}_i) \tag{5.61}$$

$$\mathbf{\Sigma}_{1:n} = \sum_{k=1}^{K} \mathcal{W}_k \ell(\mathcal{X}_k) \ell(\mathcal{X}_k)^\top, \tag{5.62}$$

where ℓ is a vectorized version of ℓ_i and $\mathcal{W}_k$ is the standard weight from (Julier and Uhlmann, 1997, (12)). Fig. 5.7 shows a result of this process.

While, in many cases the use of the UT may be the better than any other transformation, the use of the coordinates based representation (even as an intermediate representation) does lead to some loss of information. This loss of information can be avoided if we can directly represent the uncertainty in the proposed framework when extracting it from the prior estimation solution.

5.8.2 Extracting Pose Uncertainty from a MLE Solution

State-of-the-Art Pose Graph SLAM solvers find a solution by estimating the set of robot poses that maximizes the likelihood of the observed measurements (Dellaert and Kaess, 2006; Kaess, Ranganathan, and Dellaert, 2008; Kaess et al., 2012; Mangelson et al., 2019; Rosen et al., 2016). Traditional, iterative non-linear solvers (Dellaert and Kaess, 2006; Kaess, Ranganathan, and Dellaert, 2008; Kaess et al., 2012; Kümmerle et al., 2011) do this by building up a measurement Jacobian, A, with columns that correspond to elements of the parameter vector $\mathbf{x}$ and with block rows that correspond to weighted, measurement residual error functions that minimize the error between predicted measurements and what was actually measured. At each iteration, this measurement Jacobian is used to form a linear least squares problem of the following form:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|A\mathbf{x} - \mathbf{b}\|^2, \tag{5.63}$$

where $\mathbf{b}$ is the measurement vector not needed for the following derivation. The algorithm alternates between solving this linear least squares optimization problem and relinearizing A around $\hat{\mathbf{x}}$ until convergence (Dellaert and Kaess, 2006; Kaess, Ranganathan, and Dellaert, 2008). Algorithms such as those described in (Rosen et al., 2016) and (Mangelson et al., 2019) that provide a guarantee of global optimality formulate the problem slightly differently, but once a solution has been obtained, a matrix A can still be formed by linearizing the cost around the current solution.

After a solution has been reached, an estimate of the uncertainty of that solution can be found by using A to form the information matrix $\mathcal{I} = A^\top A$ and using the non-zero

Table 5.1: Covariance Error Statistics for Manhattan3500 Dataset - Modeling Correlation. Summary of covariance error statistics for proposed Lie algebra relative pose estimation when correlation is taken into account and ignored for a total of 44425 pose pairs extracted from a solution to the Manhattan3500 dataset (Olson, Leonard, and Teller, 2006), with pose offsets ranging from 5 to 500 nodes.

Method	Covariance Error Mean	Covariance Error Std. Dev.
Proposed Lie algebra Rel. Pose	0.00675104	0.0461455
Proposed (Ignoring Correlation)	2.05667	2.12371

elements of its Cholesky factorization $\mathcal{I} = R^\top R$ to calculate the necessary elements of the marginal covariance $\hat{\boldsymbol{\Sigma}}$ as detailed in (Kaess and Dellaert, 2009). The trick to extracting this covariance with respect to $\boldsymbol{\xi}_{1:n}$ as opposed to $\hat{\mathbf{x}}$ is to make sure that the Jacobian A used to form $\mathcal{I}$ is evaluated with respect to $\boldsymbol{\xi}_{1:n}$ as opposed to $\hat{\mathbf{x}}$. This can be done by numerically evaluating the Jacobian and by perturbing $\boldsymbol{\xi}_{1:n}$ around $\mathbf{0}$ and propagating that perturbation to the linearization point by means of the exponential function as opposed to perturbing the parameters of $\hat{\mathbf{x}}$ directly. Doing this enables the direct extraction of $\boldsymbol{\Sigma}_{1:n}$ and results in increased accuracy, because the distribution we are approximating, in this case, can be accurately represented as a Gaussian distribution.

5.9 Evaluation

This section evaluates the proposed uncertainty characterization method by performing two experiments. The first, performs a parameter sweep over the experiment introduced in Section 5.5.2. The second, extracts covariance information from the result of a Pose Graph SLAM algorithm and compares the predicted relative pose covariance with the sample covariance obtained from Monte Carlo.

5.9.1 Compounding Correlated Odometry

The exact accuracy of our proposed uncertainty characterization method is dependent on a variety of parameters including the number of poses compounded end-to-end and the rotation and translation noise. We explore this dependence by performing a parameter sweep across each of these parameters based on the experiment introduced in Section 5.5.2.

We began by varying the number of poses in the trajectory (N) and held both noise scale parameters fixed at $\sigma_t = \sigma_r = 3$. The results are shown in Fig. 5.8. As the number of steps increases, the accuracy of the final estimate drops off, this is because each of the methods

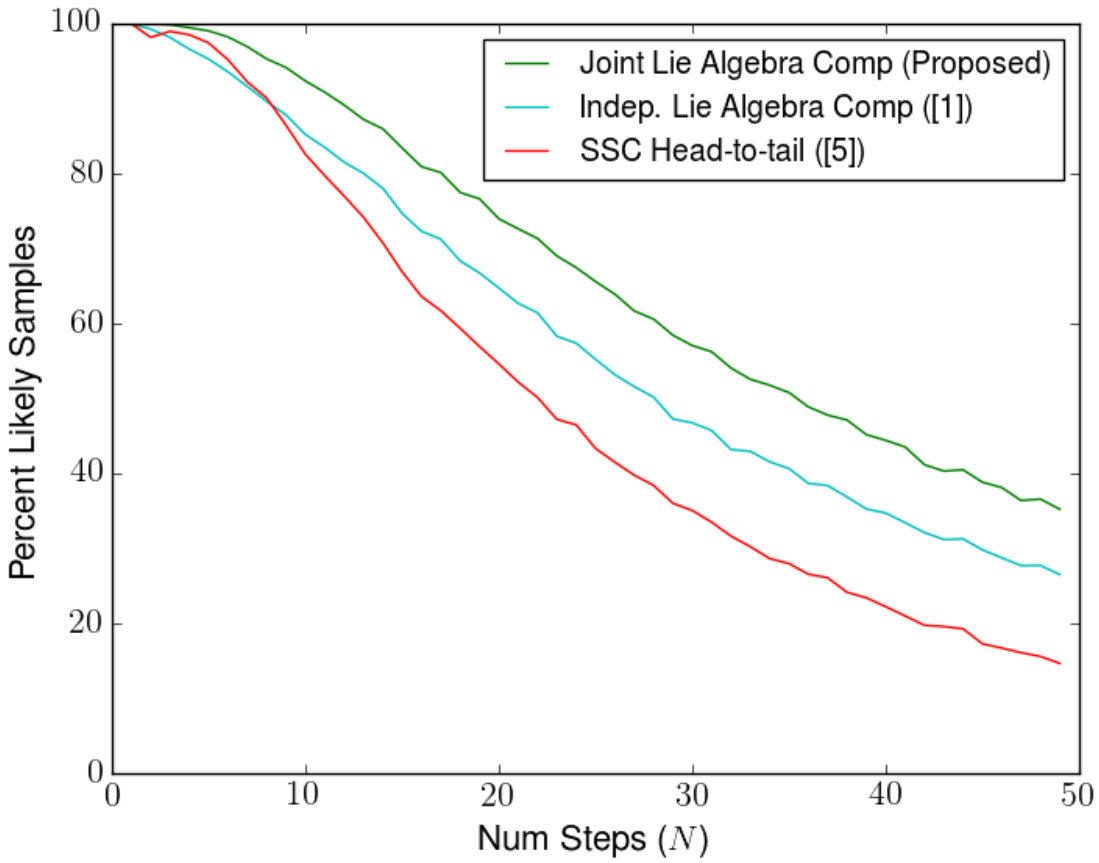


Figure 5.8: Consistency of Pose Composition vs. Trajectory Length. The percentage of final samples that fell within the 99.9% likely covariance ellipsoid as a function of the number of poses in the trajectory sequence. All methods drop off as the number of poses is increased, however our proposed method is consistently most accurate.

are only characterizing uncertainty up-to first order and the lost higher order information builds up as the number of poses increases.

For the noise parameter experiments, we varied the noise scale parameters σ_r and σ_t in turn, while keeping the number of poses fixed at $N = 10$ and the non-varying noise parameter fixed at 3. The results are shown in Fig. 5.9. Rotation noise has a much more significant effect, again this is because of the lost higher-order information (either in terms of the higher order terms of the BCH formula or through linearization for SSC). Both Lie group methods consistently outperform SSC except when rotation noise is very low and translation noise is very high. This is because when rotation error is very low, the non-linearity of the transformation becomes almost negligible and the increased translation error causes the covariance ellipsoid to increase to the point where it includes the final robot position. In

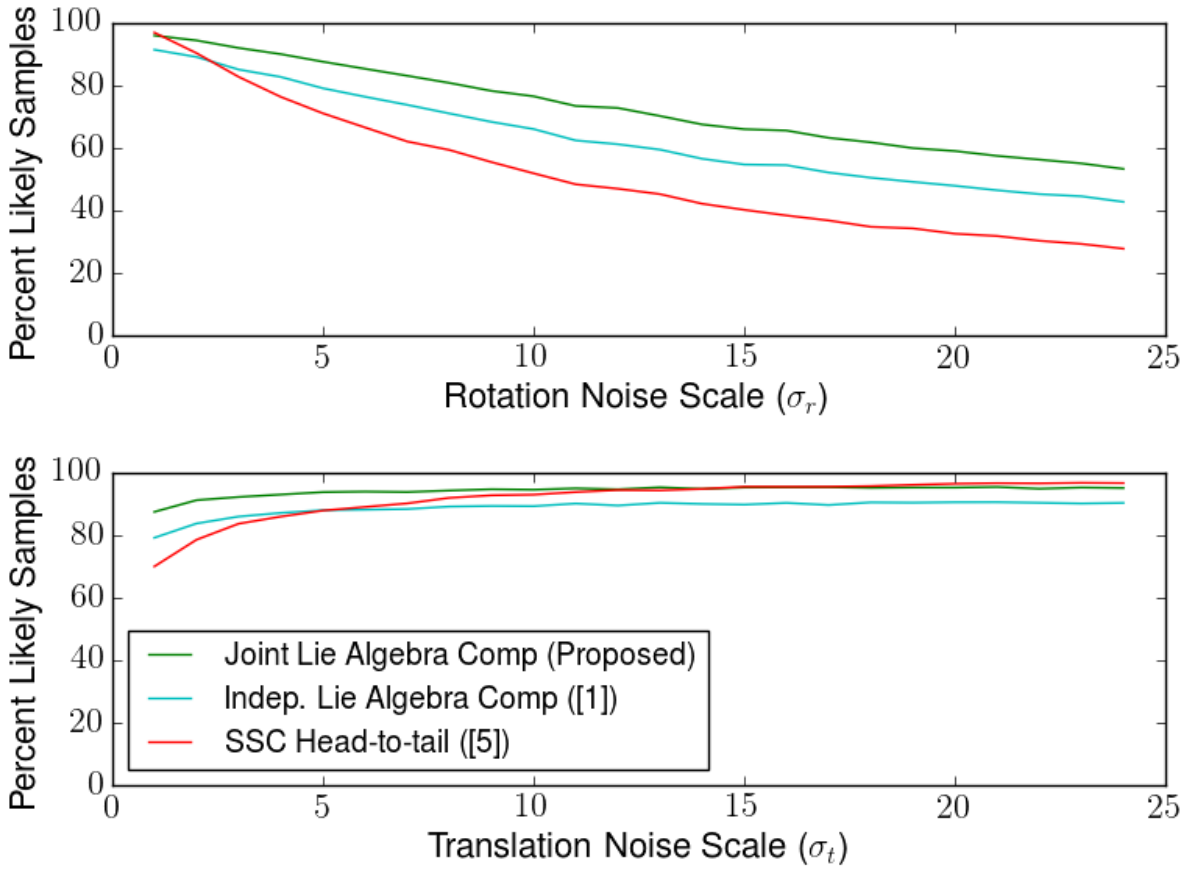
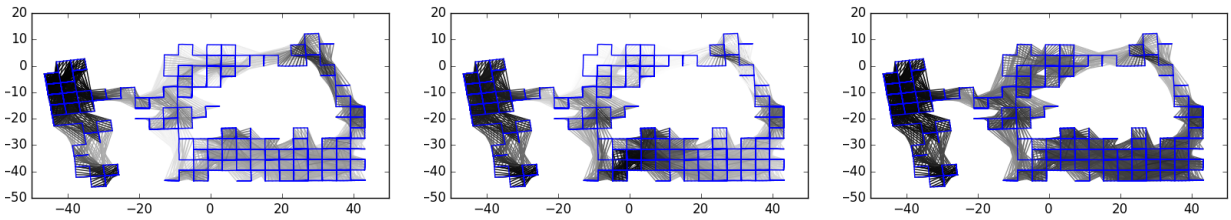


Figure 5.9: Consistency of Pose Composition vs. Noise Level. The percentage of final samples that fell within the 99.9% likely covariance ellipsoid as a function of the rotation and translation noise. For the rotation noise sweep, translation noise was held constant at $\sigma_t = 3$ and for the translation noise sweep, rotation noise was held constant at $\sigma_r = 3$. Note that increases in rotation noise have the largest negative effect.

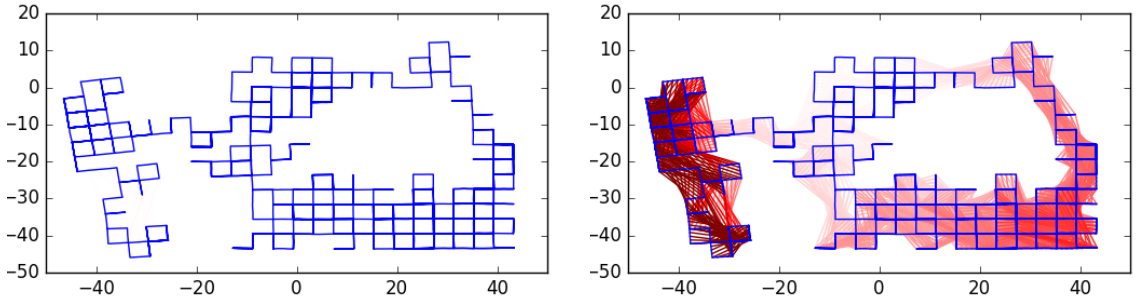
addition, the joint composition method in the Lie algebra is consistently more accurate than when correlation is ignored. For all three parameter sweeps, the induced correlation was held fixed at $\rho = 0.4$.

5.9.2 Extracting Relative Pose From a SLAM Solution

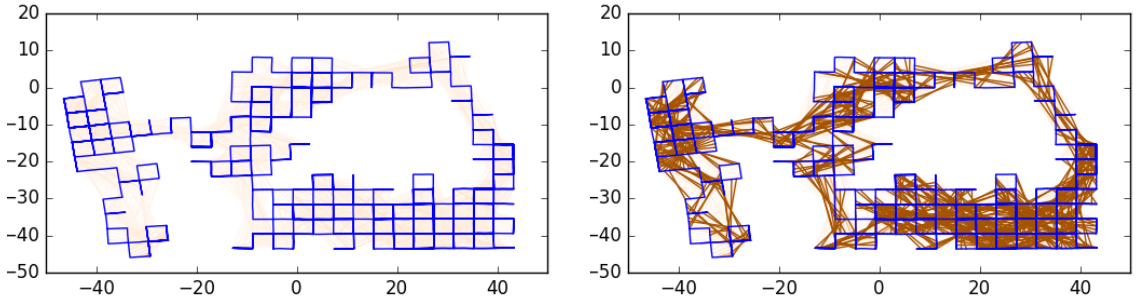
To evaluate relative pose extraction, we used iSAM (Kaess, Ranganathan, and Dellaert, 2008) to find a solution to the Manhattan 3500 dataset (Olson, Leonard, and Teller, 2006). We then extracted joint mean and covariance as described in Section 5.8.2 for pairs of poses at offsets varying from 5 to 50 in increments of 5 as well as for offsets of 100, 200, and 500 nodes. A visualization of the extracted pose pairs and the correlation between them for



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Θ



(d) Proposed - Cov. Error (e) Proposed - Cov. Error (Ignoring Correlation)



(f) Proposed - Normalized Cov. Error (g) SSC - Normalized Cov. Error

Figure 5.10: Correlation and Covariance Error for Manhattan3500 - Relative Pose Offset 50. A visualization of the correlation and covariance error for 3450 pose pairs with an offset of 50 nodes extracted from a solution of the Manhattan3500 dataset (Olson, Leonard, and Teller, 2006). (a), (b), and (c) show the relative poses colored by the correlation coefficient of the x, y, and θ dimensions respectively. White corresponds to a correlation coefficient of 0 and black to a coefficient of 1. (d) and (e) show the covariance error with respect to Monte Carlo for relative pose estimation in the Lie algebra when correlation is and is not taken into account. Dark red corresponds to a covariance error of at least 2 standard deviations above the mean (with respect to when correlation is ignored), while white corresponds to a covariance error of 0. (f) and (g) show the normalized covariance error with respect to Monte Carlo for the proposed method and SSC (Smith, Self, and Cheeseman, 1990). In this case, dark orange corresponds to a covariance error of at least 2 standard deviations above the mean (with respect to SSC relative pose extraction), while white corresponds to a covariance error of 0.

Table 5.2: Covariance Error Statistics for Manhattan3500 Dataset - Proposed vs SSC. Summary of normalized covariance error statistics for proposed and SSC (Smith, Self, and Cheeseman, 1990) relative pose estimation for a total of 44425 pose pairs extracted from a solution to the Manhattan3500 dataset (Olson, Leonard, and Teller, 2006), with pose offsets ranging from 5 to 500 nodes.

Method	Normalized Cov. Error Mean	Normalized Cov. Error Std. Dev.
Proposed Rel. Pose	0.0493121	0.0353072
SSC Tail-to-Tail	0.28778	0.411625

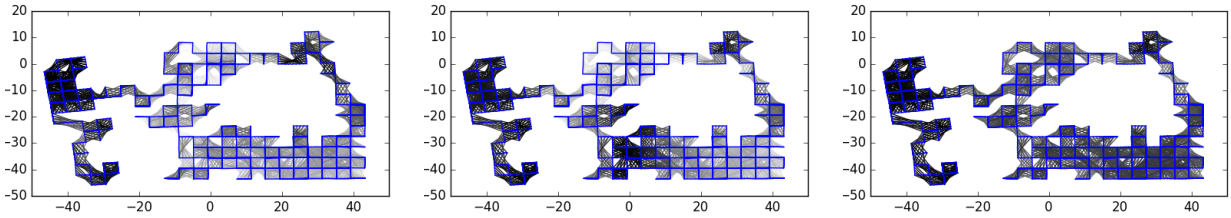
offsets of 50 are shown in Fig. 5.10 (a-c). Equivalent visualizations for offsets of 10, 100, 200, and 500 are shown in Fig. 5.11, Fig. 5.12, Fig. 5.13, and Fig. 5.14.

To investigate the importance of taking into account correlation when estimating relative pose, we performed a Monte Carlo simulation to estimate the true relative pose covariance as described in (5.57) - (5.59), where $\boldsymbol{\xi}_{g1}^{m}$ and $\boldsymbol{\xi}_{g2}^{m}$ are the perturbation variables sampled from the extracted joint covariance and $\bar{\mathbf{T}}_{g1}$ and $\bar{\mathbf{T}}_{g2}$ are the mean values extracted from the iSAM solution. We then used the metric defined in (5.56) to evaluate the covariance error of our method proposed in (5.51) when taking into account correlation (using all four terms in (5.51)) or ignoring correlation (using only the first two terms in (5.51)). Summary statistics are shown in Table 5.1 and visualizations of the error for offsets of 50, 10, 100, 200, and 500 are shown in (d-e) of Fig. 5.10, Fig. 5.11, Fig. 5.12, Fig. 5.13, and Fig. 5.14.

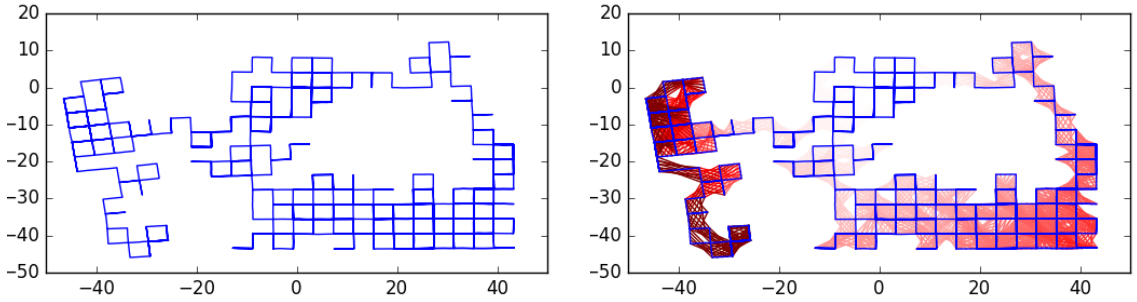
The results in Table 5.1 show that ignoring correlation can lead to a covariance error more than 3 orders of magnitude higher than if correlation is taken into account.

To compare our proposed method to SSC (Smith, Self, and Cheeseman, 1990), we performed a similar experiment except that the Monte Carlo "groundtruth" covariance for SSC was derived by taking the sample relative poses from the previous experiment and converting them to the parameter vector format described in (5.15) and taking the sample covariance. To fairly compare our proposed method and SSC, the covariance matrices were normalized by the Frobenius norm of the Monte Carlo covariance matrix before evaluating the covariance error as defined in (5.56). Summary statistics of the experiment are shown in Table 5.2 and visualizations of the error for offsets of 50, 10, 100, 200, and 500 are shown in (f-g) of Fig. 5.10, Fig. 5.11, Fig. 5.12, Fig. 5.13, and Fig. 5.14.

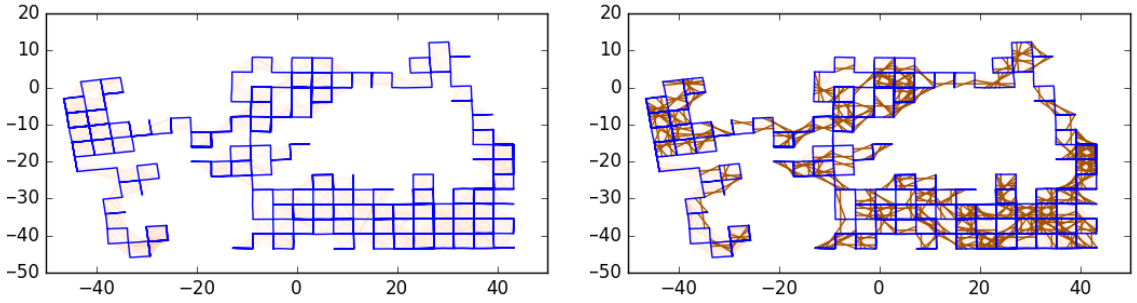
It should be noted that this is not a perfect comparison because the Monte Carlo covariance to which SSC is being compared is a multivariate Gaussian fit to a set of parameter vectors that cannot be accurately modeled as a Gaussian. However, the results do show that even when modeling the true error in the format assumed by the SSC representation, our proposed Lie algebra based method results in an order of magnitude lower covariance error



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Θ



(d) Proposed - Cov. Error (e) Proposed - Cov. Error (Ignoring Correlation)



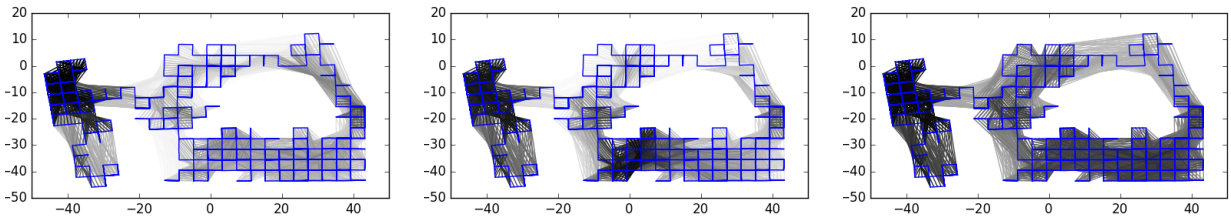
(f) Proposed - Normalized Cov. Error (g) SSC - Normalized Cov. Error

Figure 5.11: Correlation and Covariance Error for Manhattan3500 - Relative Pose Offset 10. A visualization of the correlation and covariance error for 3490 pose pairs with an offset of 10 nodes extracted from a solution of the Manhattan3500 dataset (Olson, Leonard, and Teller, 2006). The color schemes match those of Fig. 5.10.

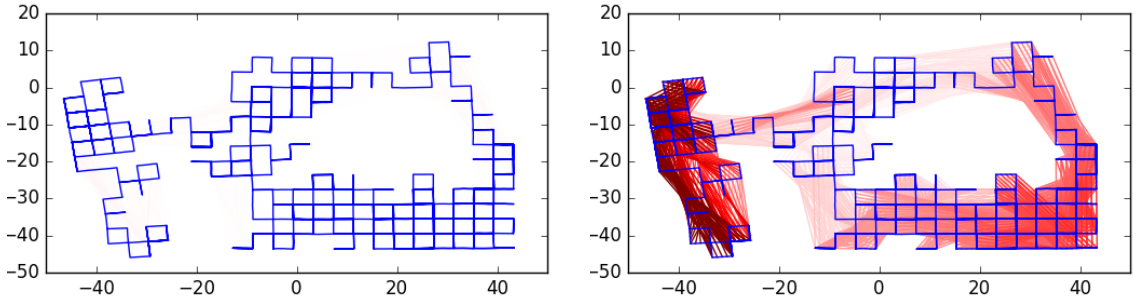
than SSC (see Table 5.2).

5.10 Library Implementation

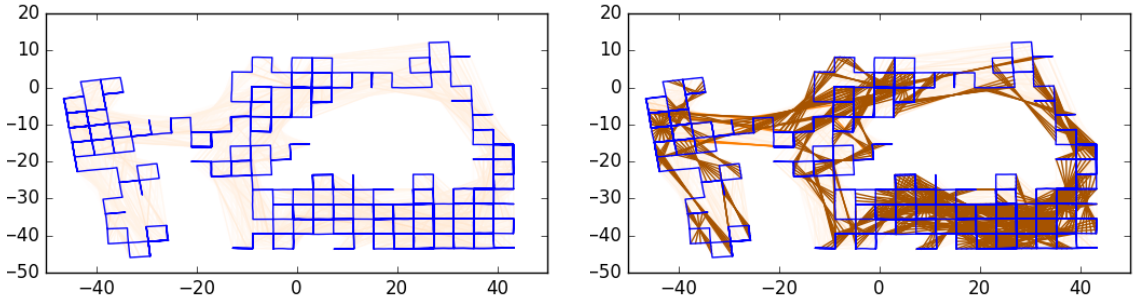
We have released an open source C++ library implementation of our method. It can be downloaded at: `https://bitbucket.org/jmangelson/lie`. It is designed to be simple, intuitive, and easily extendable.



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Θ



(d) Proposed - Cov. Error (e) Proposed - Cov. Error (Ignoring Correlation)



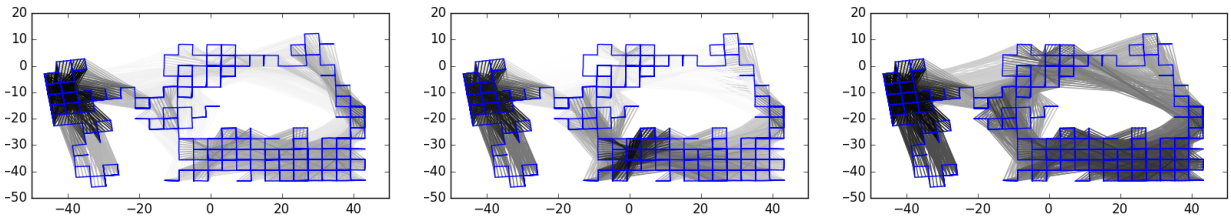
(f) Proposed - Normalized Cov. Error (g) SSC - Normalized Cov. Error

Figure 5.12: Correlation and Covariance Error for Manhattan3500 - Relative Pose Offset 100. A visualization of the correlation and covariance error for 3400 pose pairs with an offset of 100 nodes extracted from a solution of the Manhattan3500 dataset (Olson, Leonard, and Teller, 2006). The color schemes match those of Fig. 5.10.

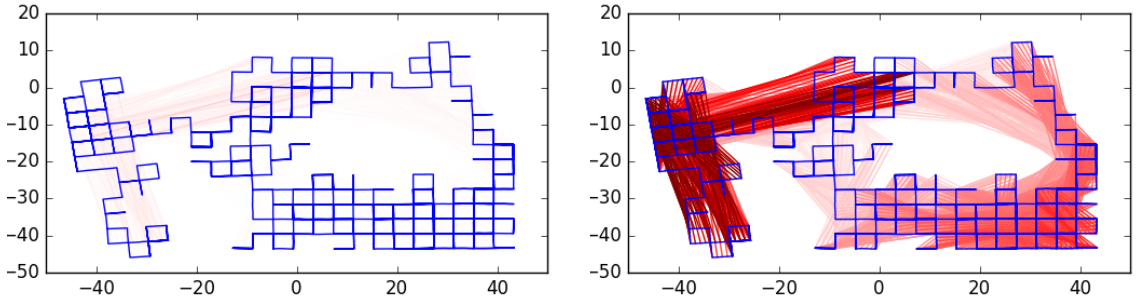
5.10.1 Creating Known and Uncertain SE(3) Objects

Creating both known and uncertain SE(3) objects can be done as follows. After importing the library a variety of constructors can be used to create known SE(3) transformations:

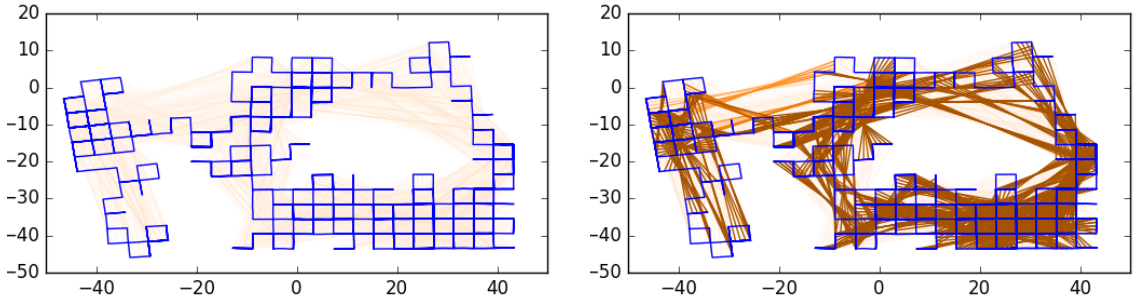
```
#include <lie/se3.hpp>

// Via rotation and translation
Eigen::MatrixXd R(3,3); // 3x3 matrix
```



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Θ



(d) Proposed - Cov. Error (e) Proposed - Cov. Error (Ignoring Correlation)

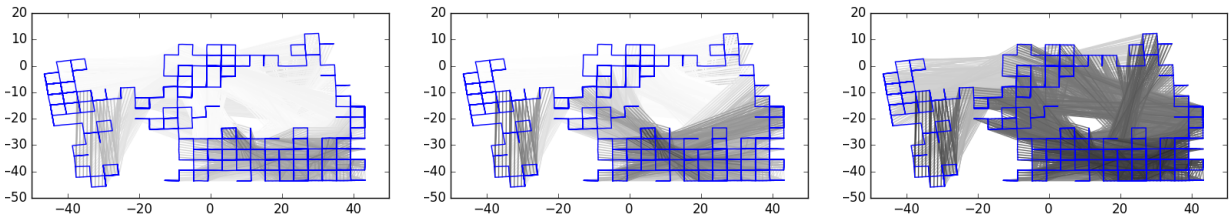


(f) Proposed - Normalized Cov. Error (g) SSC - Normalized Cov. Error

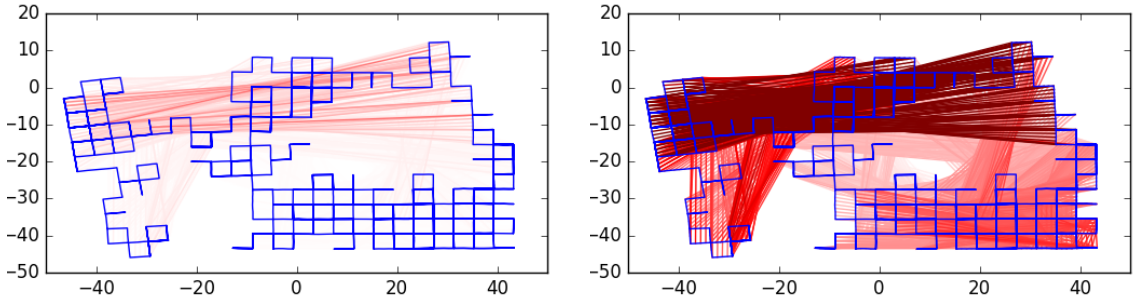
Figure 5.13: Correlation and Covariance Error for Manhattan3500 - Relative Pose Offset 200. A visualization of the correlation and covariance error for 3300 pose pairs with an offset of 200 nodes extracted from a solution of the Manhattan3500 dataset (Olson, Leonard, and Teller, 2006). The color schemes match those of Fig. 5.10.

```
Eigen::VectorXd t(3); // 3x1 vector
Lie::SE3 T_12(R, t);

// Via translation and Euler angle params
Lie::SE3 T_23(x, y, z, theta, phi, psi);
```

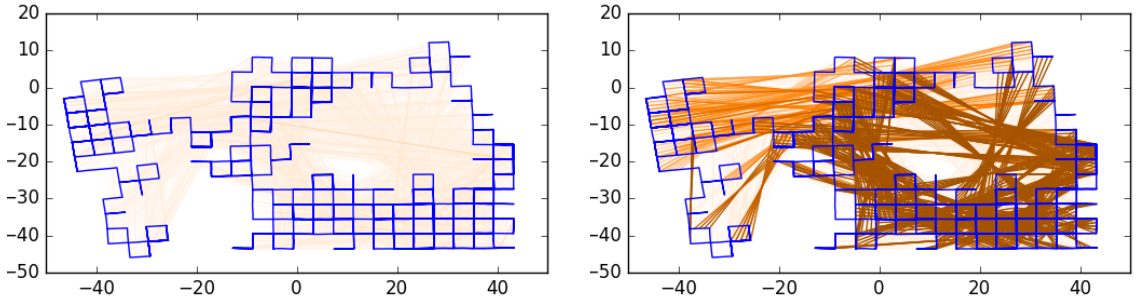
Independent and jointly distributed sets of unknown poses can also be created by passing in a mean value or a tuple of mean values and a covariance matrix.



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Θ



(d) Proposed - Cov. Error (e) Proposed - Cov. Error (Ignoring Correlation)



(f) Proposed - Normalized Cov. Error (g) SSC - Normalized Cov. Error

Figure 5.14: Correlation and Covariance Error for Manhattan3500 - Relative Pose Offset 500. A visualization of the correlation and covariance error for 3000 pose pairs with an offset of 500 nodes extracted from a solution of the Manhattan3500 dataset (Olson, Leonard, and Teller, 2006). The color schemes match those of Fig. 5.10.

```
Lie::SE3 T_ab_mu(R_ab, t_ab);
Eigen::MatrixXd Sigma_ab(6,6);
auto T_ab_uncertain =
  Lie::make_uncertain_state(
    T_ab_mu, Sigma_ab);

Lie::SE3 T_ac_mu(R_ac, t_ac);
Eigen::MatrixXd Sigma_Joint(12,12);
```

```
auto refs =
  Lie::make_uncertain_state(
    std::make_tuple(T_ab_mu, T_ac_mu),
    Sigma_Joint);
auto T_ab_uncertain = std::get<0>(refs);
auto T_ac_uncertain = std::get<1>(refs);
```

The `Lie::make_uncertain_state` function returns either a single SE(3) reference object or a tuple of such objects that can then be used to perform operations.

5.10.2 Performing Operations

The pose composition, inverse, and relative pose operations can be applied interchangeably regardless of whether or not individual SE(3) objects are known or uncertain via the `Lie::compose`, `inverse`, and `Lie::between` functions respectively. The compiler automatically determines whether or not the individual objects are known/unknown or independent/jointly distributed and applies the appropriate formulation.

```
// Composing known poses
auto T_13 = Lie::compose(T_12, T_23);

// Invert both known and uncertain poses
auto T_21 = T_12.inverse();
auto T_ba_uncertain =
  T_ab_uncertain.inverse();

// Calculating relative pose
auto T_bc_uncertain = Lie::between(
  T_ab_uncertain, T_ac_uncertain);
auto T_bc_uncertain2 = Lie::between(
  T_ab_known, T_ac_uncertain);
```

Each function returns a new (from then on assumed independent) SE(3) reference object that can be used for additional operations as necessary.

5.10.3 Converting from Alternative Representations

We have also implemented the methods proposed in Section 5.8. The uncented transform method described in Section 5.8.1 is contained in ‘‘`lie/coord_change.hpp`’’ and released

in the library.

5.10.4 Additional Lie groups

In addition, because the uncertainty propagation method we present simply uses the properties of Lie groups, it can easily be applied to other Lie group types. We have also implemented the `Lie::SO3`, `Lie::SE2`, and `Lie::SO2` classes for the SO(3), SE(2), and SO(2) Lie groups and plan to extend it to other Lie group types as needed.

5.11 Conclusion

Recent interest has shown that pose uncertainty characterization through the use of the Lie algebra leads to more accurate uncertainty propagation. However, recent work assumes that individual poses are independent from one another and have primarily focused on pose composition, ignoring the equally important inverse and relative pose operations.

This paper describes how to represent multiple jointly correlated poses while using the Lie algebra to characterize uncertainty. We also derive the equivalent of the Smith, Self, and Cheeseman (1990) pose composition, pose inverse, and relative pose operations when using the proposed framework and show that they outperform existing methods. Finally, we have released an open source C++ library implementation of our method.

The completed work is in the process of submission and review.

The C++ library is released at: `https://bitbucket.org/jmangelson/lie`.

CHAPTER 6

Conclusion

Teams of autonomous systems have the potential to make a dramatic impact in our society. However, if they are to be widely adopted, reliability needs to be significantly improved. This dissertation presents four novel algorithms/methods that seek to address several of the failure cases impeding robust localization and mapping. Although these methods can be applied regardless of domain, our primary motivation is the development of multi-agent teams of autonomous underwater vehicles.

6.1 Contributions

The contributions of this thesis include:

- In Chapter 2, we proposed a method for selecting a set of consistent measurements that can be used to merge two pose graphs into a single consistent map. Unlike other work in this area, we formulated the problem as a combinatorial problem that does not require an initial estimate and guarantees the selected set of measurements are pairwise consistent with one another. We then showed how existing algorithms for the maximum clique problem can be applied to find a solution efficiently. We evaluate on both simulated and real-world data and show that we significantly outperform existing state-of-the-art algorithms.

- In Chapter 3, we proposed an algorithm for localizing a query trajectory to a reference trajectory without any initial estimate of alignment and without performing data association. We used an approximation of the $\ell 2$ norm and linear relaxation techniques to derive a method for approximate optimization over SE(2) via linear programing. Our method outperforms existing convex optimization methods because they fail to enforce that the estimated transformation be approximately rigid.

- In Chapter 4, we formulated the planar Pose Graph simultaneous localization and mapping (SLAM) and Landmark SLAM problems as polynomial optimization programs and showed that they can be solved via the sparse bounded degree sum-of-squares (Sparse-BSOS) hierarchy of semidefinite program (SDP)s. We then proved that the globally optimal solution is always reached on the first step of the hierarchy, signifying that the planar Pose Graph and Landmark SLAM algorithms can be framed as convex optimization problems.

- In Chapter 5, we proposed a framework for modeling the uncertainty of jointly correlated poses in the Lie algebra of the Special Euclidean group. We then derived first order uncertainty propagation formulas for the pose composition, pose inverse, and relative pose operations. We evaluated our methods and showed that our method results in significantly increased uncertainty estimate consistency when compared to existing methods that either do not model correlation or use coordinates based representations. We released a C++ library implementation of our proposed framework and method.

6.2 Future Work

While this thesis has made several contributions towards robust localization and mapping, there are still significant problems that need to be resolved to enable the regular reliable operation of autonomous systems in unstructured environments. In this section, we describe some of those additional areas for future work.

6.2.1 Generalization to Low-Rank Measurement Types

Chapter 2 proposed an algorithm called pairwise consistency maximization (PCM) that can be used for selecting a consistent set of measurements for reliably merging maps generated by multiple vehicles. In addition, Chapter 4 proposed a method called sparse bounded degree sum-of-squares SLAM (SBSOS-SLAM) that can be used to find a globally optimal solution to the planar SLAM problem regardless of initialization. However, as currently formulated, the SBSOS-SLAM algorithm is restricted to full degree of freedom (DOF) relative pose constraints, and the pairwise consistency check used in PCM is most effective when the measurements are full-rank.

However, many of the sensors available to underwater vehicles are only partial-rank, such as five DOF monocular camera constraints or single DOF range only constraints. Generalization of the SBSOS-SLAM algorithm to arbitrary measurement types would dramatically

increase its usefulness and the generalization of PCM to group-k consistency maximization (GkCM) would increase its accuracy.

6.2.2 Scalability and Real-time Operation

An additional avenue for future work would be the development of methods to improve the scalability and efficiency of the SBSOS-SLAM framework. To increase the reliability of real-time navigation in a priori unknown environments, the algorithm will need to run efficiently on board a mobile platform.

There is potential for this to be done by further exploiting the sparsity inherent to the SLAM problem itself. It may be possible to exploit the structure of the problem to extract a more optimal variable ordering and grouping to decrease the solve time of the SDP steps in the Sparse-BSOS hierarchy. Doing so could enable both increased efficiency and scalability.

6.2.3 Robust Planning

The work presented in this thesis has focused on increasing the robustness of the localization and mapping task; however, this is only one of many sub-tasks that an autonomous system must perform to operate in a real-world environment. Similar or alternative techniques need to be developed to quantify and increase the robustness of the various planning tasks performed by autonomous agents. Many of these tasks can also be framed as convex or polynomial optimization problems. These problems also present interesting directions for future research.

6.2.4 Other Elements of Reliable Autonomy

Finally, as described in Section 1.2, there are additional elements of reliable autonomy outside of algorithmic robustness that need to be solved to enable truly reliable autonomous systems. The problems of constraint management, system cognizance, and system interoperability all present important avenues for future work.

APPENDICES

APPENDIX A

Useful Definitions from Graph Theory

This section presents some useful definitions from graph theory that are used in Chapter 2 (Bollobás, 1965; Diestel, 2000).

Definition A.1. *A **graph** is a pair $G = \{V, \mathcal{E}\}$, where $\mathcal{E}$ is a collection of 2-element subsets of V. We refer to an element of V as a **vertex** (**vertices** plural) and an element of $\mathcal{E}$ as an **edge** (**edges**).*

Definition A.2. *A graph is called **complete** if there is an edge connecting every pair of vertices.*

Definition A.3. *We call a graph $G' = \{V' \subseteq V, \mathcal{E}' \subseteq \mathcal{E}\}$ a **subgraph** of G and denote it by $G' \subseteq G$.*

Definition A.4. *A complete subgraph is called a **clique**.*

APPENDIX B

A Review of Convex Optimization

Convex optimization is important because it defines a set of optimization problems where every local minimum is a global minimum. This property enables you to guarantee that the solution you have found is the true optimum as opposed to only a locally optimal result.

In addition, because convex optimization problems have been well studied, a variety of specialized algorithms and libraries have been developed to solve these problems. Thus, formulating a problem as a convex optimization problem enables us to take advantage of a large body of work focused on solving those problems efficiently (Boyd and Vandenberghe, 2004).

In the following section, we will run through some important definitions that are useful in understanding convex optimization. We will then present some examples of convex optimization problems that are used in Chapter 3 and Chapter 4.

B.1 The Definitions

These definitions were primarily taken from the book *Convex Optimization* by Stephen Boyd Boyd and Vandenberghe (2004).

B.1.1 Convex Sets and Functions

Definition B.1.1. *A set C is a **convex set** if the line segment between any two points in C lies within C, i.e. if for any $x_1, x_2 \in C$ and any θ with $0 \leq \theta \leq 1$, we have*

$$\theta x_1 + (1 - \theta)x_2 \in C.$$

Definition B.1.2. *A function $f : \mathbf{R}^n \to \mathbf{R}$ is a **convex function** if* $\operatorname{dom} f$ *is a convex set*

and if for all $x, y \in \operatorname{dom} f$, *and* θ *with* $0 \leq \theta \leq 1$, *we have*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y).$$

Geometrically, this inequality means that the line segment between $(x, f(x))$ and $(y, f(y))$, which is the *chord* from x to y, lies above the graph of f. A function is **strictly convex** if strict inequality holds wherever $x \neq y$ and $0 < \theta < 1$.

Definition B.1.3. *The* ***graph*** *of a function* $f : \mathbf{R}^n \to \mathbf{R}$ *is defined as*

$$\{(x, f(x)) | x \in \boldsymbol{dom} f\},$$

which is a subset of $\mathbf{R}^{n+1}$.

Definition B.1.4. *The* ***epigraph*** *of a function* $f : \mathbf{R}^n \to \mathbf{R}$ *is defined as*

$$\boldsymbol{epi} f = \{(x, t) | x \in \boldsymbol{dom} f, f(x) \leq t\},$$

which is a subset of $\mathbf{R}^{n+1}$.

Less formally, an epigraph is the portion of the space above the graph. A function is convex if and only if its epigraph is a convex set.

B.1.2 Convex Optimization

An optimization problem in standard form is written as

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) \leq 0, \ i = 1, ..., m \\ & h_i(\mathbf{x}) = 0, \ i = 1, ..., p \end{aligned} \tag{B.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is called the **optimization variable** and the function $f_0 : \mathbb{R}^n \to \mathbb{R}$ the **objective function** or **cost function**.

Definition B.1.5. *A* ***convex optimization problem*** *is one of the form*

$$\begin{aligned} \underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad & f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(\mathbf{x}) \leq 0, \ i = 1, ..., m \\ & \mathbf{a}_i^\top \mathbf{x} = \mathbf{b}_i, \ i = 1, ..., p \end{aligned} \tag{B.2}$$

where $f_0, ..., f_m$ are convex functions.

Here we notice that because the objective function and inequality constraints must be convex and because the equality constraint functions are affine (convex), the feasible set of the problem is convex and the cost function is convex. Thus for convex optimization functions, we minimize a convex objective over a convex set Boyd and Vandenberghe (2004).

B.2 Examples of Convex Optimizataion Problems

B.2.1 Linear Programs

A general linear program (LP) has the form

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{c}^\top \mathbf{x} + d \\ \text{s.t.} \quad & \mathbf{G}\mathbf{x} \preceq \mathbf{h} \\ & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned} \tag{B.3}$$

LPs are the simplest example of convex optimization problems. In a LP, the cost function is affine while both equality and inequality constraints are linear.

Current methods can solve linear programs efficiently for large numbers of variables and constraints. Typical problems may include hundreds of thousands or even millions of variables or constraints.

B.2.2 Semi-Definite Programs

Another example of a convex optimization program is the semi-definite program (SDP).

An SDP has the form

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & x_1\mathbf{F}_1 + \cdots + x_n\mathbf{F}_n + \mathbf{G} \succeq 0 \\ & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned} \tag{B.4}$$

where $\mathbf{G}, \mathbf{F}_1, \cdots, \mathbf{F}_n \in \mathbf{S}^k$, $\mathbf{A} \in \mathbb{R}^{p\times n}$, and $\mathbf{S}^k$ is the space of semidefinite $k \times k$ matrices.

In comparison with LPs, SDPs can handle much more complicated constraints, while still being convex and thus providing guarantees of global optimality. However, the nature of SDPs makes them more difficult to solve. Taking into account the structure of the underlying problem is essential in order to solve large SDPs efficiently.

BIBLIOGRAPHY

BIBLIOGRAPHY

P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard. Robust map optimization using dynamic covariance scaling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 62–69, Karlsruhe, Germany, May 2013.

A. A. Ahmadi and P. A. Parrilo. Sum of squares and polynomial convexity. In *ProceISRReedings of the IEEE Conference on Decision and Control*, Shanghai, China, Dec 2009.

L. A. Andersson and J. Nygards. C-sam: Multi-robot slam using square root information smoothing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2798–2805, Pasadena, CA, USA, May 2008.

M. ApS. *The MOSEK C optimizer API manual Version 7.1 (Revision 54).*, 2015. URL http://docs.mosek.com/7.0/capi/.

M. ApS. *The MOSEK Fusion API for C++ 8.1.0.38*, 2018. URL https://docs.mosek.com/8.1/cxxfusion/index.html.

ASCE. Infrastructure report card: A comprehensive assessment of america's infrastructure. Technical report, American Society of Civil Engineers, 2017.

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

A. Baker. *Matrix groups: An introduction to Lie group theory*. Springer Science & Business Media, 2012.

C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.

T. D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

M. Barni, F. Buti, F. Bartolini, and V. Cappellini. A quasi-euclidean norm to speed up vector median filtering. *IEEE Transactions on Image Processing*, 9(10):1704–1709, 2000.

B. Bollobás. On generalized graphs. *Acta Mathematica Hungarica*, 16(3-4):447–452, 1965.

T. M. Bonanni, B. Della Corte, and G. Grisetti. 3-d map merging on pose graphs. *IEEE Robotics and Autonomous Letters*, 2(2):1031–1038, 2017.

N. Boumal. A Riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints. *ArXiv e-prints*, June 2015.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

J. Briales and J. Gonzalez-Jimenez. Cartan-Sync: Fast and global SE(d)-synchronization. *IEEE Robotics and Autonomous Letters*, 2, 2017.

C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. University of Michigan North Campus long-term vision and lidar dataset. *International Journal of Robotics Research*, 35(9):1023–1035, 2015.

L. Carlone, M. K. Ng, J. Du, B. Bona, and M. Indri. Rao-blackwellized particle filters multi robot SLAM with unknown initial correspondences and limited communication. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 243–249, Anchorage, Alaska, USA, May 2010.

L. Carlone, A. Censi, and F. Dellaert. Selecting good measurements via 1 relaxation: A convex approach for robust estimation over graphs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2667–2674, Chicago, Illinois, USA, September 2014.

L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 125–132, Hamburg, Germany, September 2015.

L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert. Planar pose graph optimization: Duality, optimal solutions, and verification. *IEEE Transactions on Robotics*, 32(3):545–565, 2016.

S. Carreno, P. Wilson, P. Ridao, and Y. Petillot. A survey on terrain based navigation for AUVs. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–7, Sydney, Australia, May 2010.

M. E. Celebi, F. Celiker, and H. A. Kingravi. On euclidean norm approximations. *Pattern Recognition*, 44(2):278–283, 2011.

S. M. Chaves, A. Kim, E. Galceran, and R. M. Eustice. Opportunistic sampling-based active SLAM for underwater visual inspection. *Autonomous Robots*, 40(7):1245–1265, 2016.

G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011.

A. Cunningham, M. Paluri, and F. Dellaert. DDF-SAM: Fully distributed SLAM using constrained factor graphs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3025–3030, Taipei, Taiwan, October 2010.

A. Cunningham, K. M. Wurm, W. Burgard, and F. Dellaert. Fully distributed scalable smoothing and mapping with robust multi-robot data association. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1093–1100, St. Paul, Minnesota, USA, May 2012.

A. Cunningham, V. Indelman, and F. Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5220–5227, Karlsruhe, Germany, May 2013.

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12): 1181–1203, 2006.

R. Diestel. *Graph Theory*. Springer-Verlag, Berlin, New York, 2000.

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

J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert. Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5807–5814, Seattle, Washington, USA, May 2015.

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

E. Eade. Lie groups for 2D and 3D transformations, 2017. URL http://ethaneade.com/lie.pdf. Accessed: 2018-12-05.

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*, Edmonton, AB, Canada, Aug 2005.

R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based slam. *IEEE Transactions on Robotics*, 22(6):1100–1114, 2006.

U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost np-complete. In *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science*, pages 2–12, Kazimierz Dolny, Poland, Sept. 1991.

C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

U. Frese. A proof for the approximate sparsity of slam information matrices. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.

U. Frese. Treemap: An o (log n) algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21(2):103–122, 2006.

R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

R. Hartley, J. G. Mangelson, L. Gan, M. G. Jadidi, J. M. Walls, R. M. Eustice, and J. W. Grizzle. Legged robot state-estimation through combined forward kinematic and preintegrated contact factors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–8, Brisbane, Australia, May 2018.

G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme. Active planning for underwater inspection and the benefit of adaptivity. *International Journal of Robotics Research*, 32(1):3–18, 2013.

F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard. Advanced perception, navigation and planning for autonomous in-water ship hull inspection. *International Journal of Robotics Research*, 31(12):1445–1464, 2012.

A. Howard. Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, 25(12):1243–1256, 2006.

A. Howard and N. Roy. The robotics data set repository (radish), 2003. URL http://radish.sourceforge.net/.

A. Howard, M. J. Matark, and G. S. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 434–439, Lausanne, Switzerland, Oct 2002. IEEE.

G. Hu, K. Khosoussi, and S. Huang. Towards a reliable slam back-end. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 37–43, Tokyo, Japan, Nov 2013.

S. Huang and G. Dissanayake. Convergence and consistency analysis for extended kalman filter based SLAM. *IEEE Transactions on Robotics*, 23(5):1036–1049, 2007.

V. Indelman, E. Nelson, J. Dong, N. Michael, and F. Dellaert. Incremental distributed inference from arbitrary poses and unknown data association: Using collaborating robots to establish a common reference. *IEEE Control Systems Magazine*, 36(2):41–74, 2016.

S. J. Julier. The scaled unscented transformation. In *Proceedings of the American Control Conference*, volume 6, pages 4555–4559, Anchorage, Alaska, USA, May 2002.

S. J. Julier and J. K. Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics, 1997.

M. Kaess and F. Dellaert. Covariance recovery from a square root information matrix for data association. *Journal of Robotics and Autonomous Systems, RAS*, 57(12):1198–1210, December 2009.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.

M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *International Journal of Robotics Research*, 31(2):216–235, 2012.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, pages 82:35–45, 1960.

A. Kim and R. M. Eustice. Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Transactions on Robotics*, 29(3):719–733, 2013.

B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller. Multiple relative pose graphs for robust cooperative mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3185–3192, Anchorage, Alaska, USA, May 2010.

Y. Kim and A. Kim. On the uncertainty propagation: Why uncertainty on lie groups preserves monotonicity? In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3425–3432, Vancouver, BC, Canada, Sept. 2017.

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3607–3613, Shanghai, China, May 2011.

J. B. Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.

J. B. Lasserre, K.-C. Toh, and S. Yang. A bounded degree SOS hierarchy for polynomial optimization. *EURO Journal on Computational Optimization*, 5(1-2):87–117, 2017.

Y. Latif, C. Cadena, and N. José. Robust loop closing over time. In *Proceedings of the Robotics: Science & Systems Conference*, Sydney, Australia, July 2012.

J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics*, 7(3):376–382, 1991.

J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: systems and algorithms. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Kongresshuas Baden-Baden, Germany, Jun 2011.

H. Li, X. Huang, J. Huang, and S. Zhang. Feature matching with affine-function transformation models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12): 2407–2422, 2014.

J. Li, P. Ozog, J. Abernethy, R. Eustice, and M. Johnson-Roberson. Utilizing high-dimensional features for real-time robotic applications: Reducing the curse of dimensionality for recursive bayesian estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–8, Deajeon, Korea, Oct 2016.

M. Liu, S. Huang, G. Dissanayake, and H. Wang. A convex optimization based approach for pose slam problems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1898–1903, Vilmoura, Portugal, Oct 2012.

A. W. Long, K. C. Wolfe, M. J. Mashner, and G. S. Chirikjian. The banana distribution is gaussian: A localization study with exponential coordinates. *Proceedings of the Robotics: Science & Systems Conference*, 265, July 2012.

F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

J. G. Mangelson, R. W. Wolcott, P. Ozog, and R. M. Eustice. Robust visual fiducials for skin-to-skin relative ship pose estimation. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–8, Monterey, CA, USA, September 2016.

J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–8, Brisbane, Australia, May 2018.

J. G. Mangelson, R. Vasudevan, and R. M. Eustice. Communication constrained trajectory alignment for multi-agent inspection via linear programming. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–7, Charleston, SC, USA, October 2018.

J. G. Mangelson, J. Liu, R. M. Eustice, and R. Vasudevan. Guaranteed globally optimal planar pose graph and landmark slam via sparse-bounded sum-of-squares programming. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–8, Montreal, Canada, May 2019.

D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society of Industrial and Applied Mathematics*, 11(2):431–441, 1963.

P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, Boston, Massachusetts, USA, Jun 1995.

J. Melo and A. Matos. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Engineering*, 139:250–264, 2017.

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*, Edmonton, Alberta, Canada, Jul 2002.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: an improved particle filtering algorithm for simul- taneous localization and mapping that provably converges. In *Proceedings of the International Joint Conference on Artifical Intelligence*, pages 1151–1156, 2003.

P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *Proceedings of the International Symposium on Robotics Research*, Tokyo, Japan, Aug 1989a.

P. Moutarlier and R. Chatila. An experimental system for incremental environment modelling by an autonomous mobile robot. In *Proceedings of the International Symposium on Experimental Robotics*, pages 327–346, Montreal, Quebec, Canada, Jun 1989b.

R. M. Murray, Z. Li, and S. Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics*, 17(6):890–897, 2001.

E. Olson. Recognizing places using spectrally clustered local matches. *Robotics and Automation Systems*, 57(12):1157–1172, 2009.

E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *International Journal of Robotics Research*, 32(7):826–840, 2013.

E. Olson, M. R. Walter, S. J. Teller, and J. J. Leonard. Single-cluster spectral graph partitioning for robotics applications. In *Proceedings of the Robotics: Science & Systems Conference*, Cambridge, Massachusetts, USA, June 2005.

E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2262–2269, Orlando, Florida, May 2006.

P. Ozog, N. Carlevaris-Bianco, A. Kim, and R. M. Eustice. Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle. *Journal of Field Robotics*, 33(3):265–289, 2016.

B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W. keng Liao, and A. Choudhary. Fast algorithms for the maximum clique problem on massive graphs with applications to overlapping community detection. *Internet Mathematics*, 11(4-5):421–448, 2015. doi: 10.1080/15427951.2014.986778. URL http://dx.doi.org/10.1080/15427951.2014.986778.

L. Paull, S. Saeedi, M. Seto, and H. Li. Auv navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149, 2014.

M. Pfingsthorn and A. Birk. Generalized graph SLAM: Solving local and global ambiguities through multimodal and hyperedge constraints. *International Journal of Robotics Research*, 35(6):601–630, 2016.

G. M. Reis, M. Fitzpatrick, J. Anderson, L. Bobadilla, and R. N. Smith. Augmented terrain-based navigation to enable persistent autonomy for underwater vehicles. In *Proceedings of the IEEE International Conference on Robotic Computing*, pages 292–298, Taichung, Taiwan, Apr 2017.

M. L. Rodríguez-Arévalo, J. Neira, and J. A. Castellanos. On the importance of uncertainty representation in active slam. *IEEE Transactions on Robotics*, 2018.

D. Rosen, L. Carlone, A. Bandeira, and J. Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the Special Euclidean group. In *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics*, 2016.

D. M. Rosen, C. DuHadway, and J. J. Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5822–5829, Seattle, Washington, USA, May 2015.

S. Saeedi, M. Trentini, M. Seto, and H. Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

J. Saunderson, P. A. Parrilo, and A. S. Willsky. Semidefinite relaxations for optimization problems over rotation matrices. In *ProceISRRedings of the IEEE Conference on Decision and Control*, pages 160–166, Los Angeles, California, Dec 2014.

A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Proceedings of the Robotics: Science & Systems Conference*, Seattle, Washington, USA, Jun 2009.

L. Smail. Junction trees constructions in bayesian networks. In *Journal of Physics: Conference Series*, volume 893, page 012056. IOP Publishing, 2017.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robots*, pages 167–193, 1990.

R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*. Wiley New York, 2005.

N. Sünderhauf and P. Protzel. Towards a robust back-end for pose graph SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1254–1261, St. Paul, Minnesota, USA, May 2012.

K. Tapp. *Matrix groups for undergraduates*, volume 79. American Mathematical Soc., 2016.

S. Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.

S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

S. Thrun and Y. Liu. Multi-robot slam with sparse extended information filers. In *Proceedings of the International Symposium on Robotics Research*, San Francisco, CA, USA, Oct 2005. Springer.

S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Y. Ng. Simultaneous mapping and localization with sparse extended information filters: Theory and initial results. *International Journal of Robotics Research*, 23:693–716, 2004.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

J. M. Walls, A. G. Cunningham, and R. M. Eustice. Cooperative localization by factor composition over a faulty low-bandwidth communication channel. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 401–408, Seattle, WA, USA, May 2015.

M. Walter, R. Eustice, and J. Leonard. A provably consistent method for imposing sparsity in feature-based SLAM information filters. In *Proceedings of the International Symposium on Robotics Research*, pages 214–234, San Francisco, CA, USA, Oct 2005.

Y. Wang and G. S. Chirikjian. Nonparametric second-order theory of error propagation on motion groups. *The International journal of robotics research*, 27(11-12):1258–1273, 2008.

S. E. Webster, J. M. Walls, L. L. Whitcomb, and R. M. Eustice. Decentralized extended information filter for single-beacon cooperative acoustic navigation: Theory and experiments. *IEEE Transactions on Robotics*, 29(4):957–974, 2013.

S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. *Journal of Field Robotics*, 28(6):854–874, 2011.

T. Weisser, J. B. Lasserre, and K.-C. Toh. Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity. *Mathematical Programming Computation*, 10(1):1–32, 2018.

D. O. Wheeler, D. P. Koch, J. S. Jackson, T. W. McLain, and R. W. Beard. Relative navigation: A keyframe-based approach for observable gps-degraded navigation. *IEEE Control Systems Magazine*, 38(4):30–48, 2018.

S. B. Williams, G. Dissanayake, and H. Durrant-Whyte. Towards multi-vehicle simultaneous localisation and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2743–2748, Washington D. C., USA, May 2002.

R. W. Wolcott and R. M. Eustice. Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving. *International Journal of Robotics Research*, 36: 292–319, 3 2017.

Q. Wu and J.-K. Hao. A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3):693–709, 2015.

X. S. Zhou and S. I. Roumeliotis. Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct 2006. IEEE.

D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the ACM Annual Symposium on the Theory of Computing*, pages 681–690, Seattle, Washington, USA, May 2006.