Autonomous Underwater Vehicle Navigation and Mapping in Dynamic, Unstructured Environments

by

Clayton Gregory Kunz

B.S. Stanford University (1995), M.S. Stanford University (1997)

Submitted to the Dept. of Electrical Engineering and Computer Science and the
Joint Program in Applied Ocean Science and Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

February 2012

© 2012 Clayton Gregory Kunz. All rights reserved.

The author hereby grants to MIT and to WHOI permission to reproduce and distribute
publicly paper and electronic copies of this thesis document in whole or in part
in any medium now known or hereafter created.

Author.....
Joint Program in Applied Ocean Science and Engineering
November 18, 2011

Certified by.....
Dr. Hanumant Singh
Thesis Supervisor, Associate Scientist, WHOI

Accepted by.....
Dr. James C. Preisig
Chair, Joint Committee for Applied Ocean Science and Engineering, WHOI

Accepted by.....
Professor Leslie A. Kolodziejski
Chair, EECS Committee on Graduate Students, MIT

Autonomous Underwater Vehicle Navigation and Mapping in Dynamic, Unstructured Environments

by

Clayton Gregory Kunz

Submitted to the Department of Electrical Engineering and Computer Science
and the Joint Program in Applied Ocean Science and Engineering
on November 18, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis presents a system for automatically building 3-D optical and bathymetric maps of underwater terrain using autonomous robots. The maps that are built improve the state of the art in resolution by an order of magnitude, while fusing bathymetric information from acoustic ranging sensors with visual texture captured by cameras. As part of the mapping process, several internal relationships between sensors are automatically calibrated, including the roll and pitch offsets of the velocity sensor, the attitude offset of the multibeam acoustic ranging sensor, and the full six-degree of freedom offset of the camera. The system uses pose graph optimization to simultaneously solve for the robot's trajectory, the map, and the camera location in the robot's frame, and takes into account the case where the terrain being mapped is drifting and rotating by estimating the orientation of the terrain at each time step in the robot's trajectory. Relative pose constraints are introduced into the pose graph based on multibeam submap matching using depth image correlation, while landmark-based constraints are used in the graph where visual features are available. The two types of constraints work in concert in a single optimization, fusing information from both types of mapping sensors and yielding a texture-mapped 3-D mesh for visualization. The optimization framework also allows for the straightforward introduction of constraints provided by the particular suite of sensors available, so that the navigation and mapping system presented works under a variety of deployment scenarios, including the potential incorporation of external localization systems such as long-baseline acoustic networks. Results of using the system to map the draft of rotating Antarctic ice floes are presented, as are results fusing optical and range data of a coral reef.

Thesis Supervisor: Hanumant Singh
Title: Associate Scientist

Acknowledgments

This document is only one piece of evidence of a much larger trip that I've been on for the last five years, and this is truly a case where the process of getting there is at least as important as the final destination. The whole experience would not have been possible without the ceaseless support of my family, my friends, my co-workers, and my advisors. It's not really fair, actually, to divide people into groups like that – I think Hanu lands in all four categories. It is not possible for me to express how grateful I am to everyone for their encouragement, support, occasional prodding, and advice. But I can say: thank you.

First of all, thanks to Julie, who pushed me back into graduate school, agreed to move our family across the country, and somehow managed to keep a positive and encouraging attitude while I spent weeks at a time at sea and months at a time pushing toward deadlines that were only replaced by other deadlines once they were reached. Your patience is incomparable, and your encouragement comes from somewhere deeper than an AUV can reach. Thanks to Jasper for forcing me to keep everything in proper perspective, and to Clio, who arrived when I was about halfway into the program, for forcing me to *really* keep everything in proper perspective.

The rest of my family has been a constant source of encouragement from across the country: Dad, July, Jan, Sam, Chet, Donna, and the rest of you who I don't get to visit with as often as I'd like. And mom, who continues inspiring me. It was hard for me and Julie to make the move across the country – thanks for pushing us out here then and for pulling (for) us ever since.

I thank Hanu for many more things than can be listed here. But I'll pick one: thanks for trusting me to only mess up your robot a little bit before throwing it over the side and toward the bottom of the Arctic Ocean the first time I was on board a research vessel. Education comes in many forms. The rest of my committee, Bill, Daniela, and John I thank for their time, attention, and thoughtful advice. And enthusiasm.

For providing venting valves for accumulated steam at work, I thank Chris Murphy, Derya Akkaynak-Yellin, Jeff Kaeli, Jordan Stanway, and Peter Kimball. For providing an inspiring view of what comes next, as well as a lot of help, I thank Chris Roman, Ryan Eustice, Mike Jakuba, James Kinsey, and Oscar Pizarro. And my experience at work would have been a lot less interesting without Dana Yoerger and Judy Fenwick down the hall. John Bailey probably has prevented me from burning down the building

more than once.

Michael Kaess and Hordur Johannsson put together some excellent C++ code implementing iSAM which I relied heavily upon for the work in this thesis. Thanks for being so particular, and thanks for publishing it.

The scientists I've worked with at sea deserve my thanks for reminding me of the need to make something that has a use outside the lab, and for pushing me to keep improving the technology in order to make it even more useful: Roy Armstrong, Liz Clarke, Rob Sohn, Jerry Hoff, Jeremy Wilkinson, and Ted Maksym.

Finally, I could not have lived in Falmouth without a large village of friends supporting me and my family. Andrew and Kelly, Tracy and Andrea, Ben and Tiffany, Josh and Lauren, Steve and Jenny, Dave and Erin, John and Alyssa, Bill, Pete and Honoree, Romain et Julie, you have made it a truly wonderful and fun experience living here. Thanks to Sandipa for understanding me and Julie, and for helping us understand Hanu.

Post-finally, I thank the sources that have funded my tuition and work here. My first year was funded through an MIT Presidential Fellowship. Since then, I've been covered first by the National Science Foundation Censsis ERC under grant number EEC-9986821, and then by the National Oceanic and Atmospheric Administration under grant number NA090AR4320129.

Contents

1 Introduction **8**

1.1 Autonomous underwater mapping 8
1.2 The Seabed AUV 9
1.3 Simultaneous localization and mapping 13
1.4 Navigation and mapping relative to drifting, rotating terrain 15
1.5 Fusion of optical and acoustic data 16
1.6 Assumptions made throughout the thesis 17
1.7 Organization of the thesis 18

2 AUV Navigation **20**

2.1 Underwater localization 20
2.2 Pose graphs 22
2.3 AUV pose graph navigation 25
2.4 External navigation beacons 28
2.5 Calibration of navigation sensors 31

3 Multibeam Navigation and Mapping **35**

3.1 Incrementally building a multibeam map 35
3.2 Determining sensor offset 37
3.3 Constraints induced by multibeam matches 40
3.4 Estimating the information matrix 42
3.5 Results 46

4 Building Visual Maps with Cameras **52**

4.1 Background 52

4.2 Image matching 53
4.3 Constraints induced by matched image pairs 54
4.4 Photomosaicking 57
4.5 Bundle adjustment 61
4.6 Fusing multibeam and vision 64

5 Mapping in drifting, rotating environments **74**
5.1 Motivation 74
5.2 Augmenting the pose graph 76
5.3 Results 78

6 Discussion **82**
6.1 Summary of contributions 82
6.2 Comparison with earlier techniques 83
6.3 Future research 84
6.4 Lessons learned 88
6.5 Conclusion 90

Chapter 1

Introduction

1.1 Autonomous underwater mapping

This thesis addresses the problem of building three dimensional maps of unstructured, dynamic underwater environments using autonomous underwater vehicles (AUVs). It concentrates on the two main methodologies for high resolution imaging, optical and acoustic, seeking to create metric, self-consistent maps while increasing the spatial coverage of optical maps and the spatial resolution of acoustic maps, combining the two modalities when possible. The technology of autonomous underwater vehicles has matured quickly in recent years, and they are now routinely used for research in physical oceanography, geology [55], fisheries management [7], and archaeology [14], as well as in disaster response [5] [22]. Even so, using AUVs to build optical and bathymetric maps remains a difficult problem, not only because the problem of knowing where an AUV is at any point in time (the problem of *localization*) is difficult, but also because the calibration of on-board sensors and the modeling of vehicle motion is hard.

To begin with, an AUV operates underwater, beyond the reach of the electromagnetic waves of GPS satellites, and through six degrees of freedom. When an AUV is making bathymetric maps with acoustic sensors, it must operate typically within 50 meters of the terrain being mapped to ensure reasonable resolution, though this range depends on the acoustic frequency of the instrument being used and the desired map resolution. When using a color camera, an AUV must work within about 4 meters of the area being mapped, because of the rapid wavelength-dependent attenuation of light in water, and because particles floating in the water cause backscatter which occludes the scene of interest. Working this close to the seafloor or any other solid terrain further

complicates localization, because acoustic signals from navigation beacons may reflect strongly off of the terrain, confounding position estimates. It also means that an AUV must move to avoid crashing into the terrain being mapped, preventing the use of any assumption about straight line motion. In addition, most AUV navigation tasks take place in an unstructured environment, free of the man-made objects which in other environments have been used to provide easy-to-identify features (corners, edges, and planes). More recently, AUVs have been tasked to build maps of terrain which may itself be moving through space. In these cases, the desire is to build maps relative to some object that is moving in an unknown, though constrained, fashion, such as a ship hull [62], an ice floe, or an iceberg [31], which further complicates the integration of geo-referenced sensors such as compasses and moored navigation beacons.

The system presented in this thesis is not specific to autonomous underwater vehicles; it can be applied to any underwater system with sufficient on-board sensing, particularly remotely operated vehicles (ROVs), manned submersibles, and towed packages. These vehicles generally have much greater power capacity than AUVs, either because they are being powered through a tether to the surface, or because they are simply larger and able to carry more batteries [15]. The primary benefit of additional power from a mapping perspective is that it is often possible to use flood lights or high-rate strobes, enabling the capture of video-rate imagery for denser coverage of the terrain. Since ROVs and manned submersibles generally carry similar or better attitude and velocity sensors than autonomous vehicles, a mapping system developed specifically for AUVs is generalizable to the case of more capable platforms.

1.2 The Seabed AUV

This thesis proceeds through the steps necessary for the production of acoustic and optical maps in the challenging environments described above. These steps include sensor offset calibration, navigation recovery, and estimation of environmental motion. All of this work is based on real-world field campaigns using the Seabed AUV [52] (see figure 1-1), a hover-capable robot usually used for seafloor imaging, with an acoustic multibeam sensor in addition to one or more color cameras. Two scenarios are examined in detail: building acoustic "bathymetric" maps of the extent below sea level (draft) of drifting, rotating ice floes in Antarctica, and building 3-D maps fusing bathymetry and visual texture of an area of coral reef in Puerto Rico.

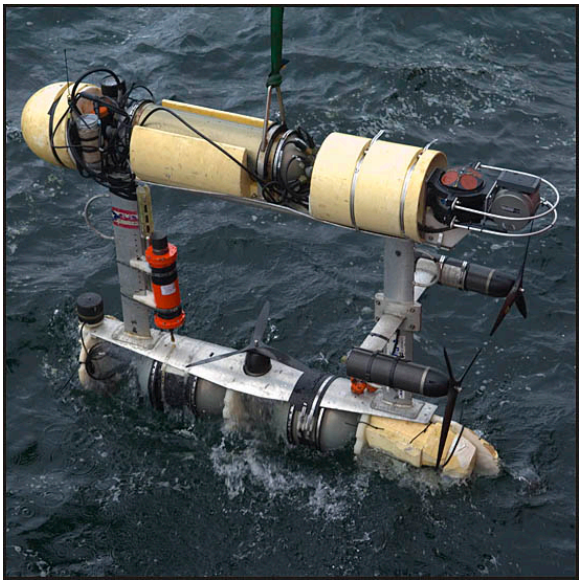

Figure 1-1: The Seabed AUV without its fairings, in under ice configuration. The upper hull contains the control computer, multibeam sonar, doppler velocity log, and camera. The lower hull contains batteries and the fibre optic gyro. Additional science sensors are scattered around the robot. An ultra-short baseline transponder is mounted to the front strut.

There are several important features about the AUV as evidenced by the photograph in figure 1-1. First of all, the AUV is unusual in that it is not torpedo-shaped, and has three thrusters with no other control surfaces. The upper hull contains buoyant syntactic foam, and the lower hull contains batteries and other heavier material, pushing the center of gravity away from the center of buoyancy and thereby increasing passive roll and pitch stability. The three-thruster configuration enables the robot to turn in place using differential steering, and to move vertically, allowing it to work close to the seafloor. It is also clear from the photograph that the sensors used for the navigation system are distributed about the robot, as are the mapping sensors. The primary navigation sensors on the AUV include a depth sensor (lower hull, aft), a north-seeking fibre optic gyro, measuring roll, pitch, true heading, angular velocity, and acceleration (lower hull, midships), an acoustic transducer (lower hull, forward) for determining ranges to a fixed array of beacons, an ultra-short baseline transponder (forward strut) for determining ranges to a beacon on the support ship, and a doppler velocity log (DVL, upper hull, aft), which measures 3-axis velocity relative to local terrain, as well as roll, pitch, and magnetic heading. In addition, there is a conductivity-temperature-depth (CTD)

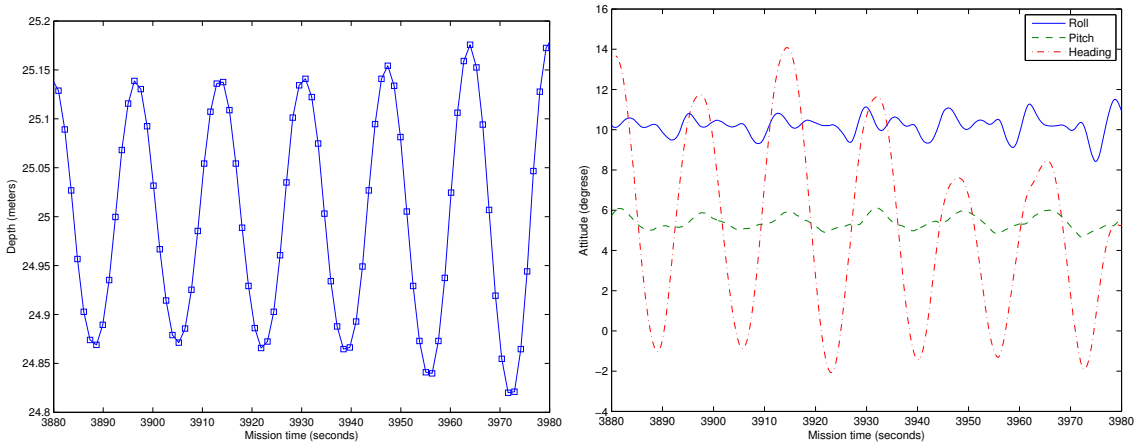

Figure 1-2: An example of imperfect control over 100 seconds of a dive. The graph on the left shows depth swinging through 40 centimeters while the robot is attempting to maintain a constant depth of 25 meters, which is coupled with changes in attitude shown on the right. The heading value on the right has been shifted to make all three variables visible at once. Data are shown at the sampling rates of the sensors: about 1.3 seconds for depth, and 0.1 seconds for attitude.

sensor mounted on the forward strut, which provides an independent depth estimate. All of the sensors operate at different rates, with no hardware synchronization.

The sensors are mounted this way due to mundane considerations of buoyancy, cable routing, and lack of space, and often need to be moved around to make room for new scientific payloads or operating scenarios. Consequently, on any given dive it is impractical to attempt to determine the AUV's center of mass or buoyancy, which sometimes results in the AUV fishtailing or porpoising due to imperfectly tuned control constants, as shown in figure 1-2. This figure shows oscillations that are more characteristic of what one would expect from a towed vehicle, where the depth is dependent on wave action. Under more stable configurations, Seabed's depth and attitude behave as shown in figure 1-3. While this kind of stability is nice from an efficiency standpoint, it is not necessary for the mapping system presented here, as the AUV behaved less than ideally as in figure 1-2 while the under ice data used for much of the thesis were collected.

The multibeam sensor in the under ice configuration is on the upper hull, aft of the DVL. It runs independently of the other sensors, at a rate dependent on range to the terrain, but generally at about ten pings per second. There is minimal acoustic interference between the multibeam and the DVL and acoustic modem, which is filtered out in the initial processing of the multibeam data. When mapping the seafloor,

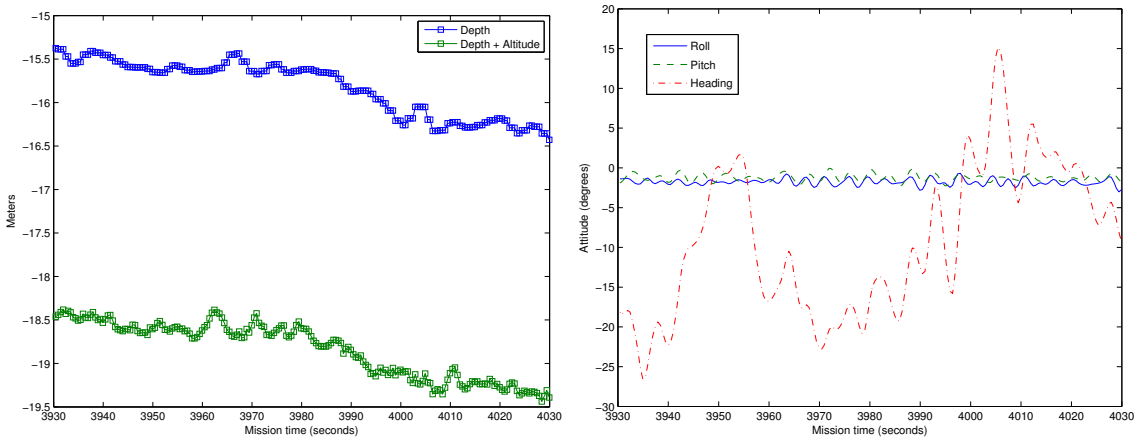

Figure 1-3: Plots of typical AUV behavior, again over 100 seconds of a deployment. During this dive, the robot was attempting to maintain constant altitude over the seafloor, and shows better stability than in the previous figure.

the multibeam is mounted on the lower hull, forward of the battery housing. The AUV can accommodate up to four synchronized cameras and a single strobe, but usually only runs with one or two cameras. Under ice, an upward-facing camera was mounted near the front of the robot in the upper hull, and relied on ambient light. Usually at least one camera is mounted in the front of the lower hull, and a xenon strobe is mounted in the rear of the lower hull.

The internal camera calibration parameters (focal length and lens distortion) may change during the course of a cruise, as the AUV is handled in rough conditions, as shown in figure 1-4. Fortunately the problem of refraction due to the use of a flat water-glass-air interface in the camera housing is largely accommodated for as radial lens distortion – the claims in [59] are valid, but minimal in specific conditions practiced with the Seabed AUV. In spite of the possibility of impacts jarring calibration parameters, we assume in this work that (underwater) camera calibration is available, and in fact use [2], a free implementation based on [68], to calibrate cameras in a salt water test tank.

Building a correct bathymetric map requires knowing all six degrees of freedom of the vehicle or towed sled's pose at the time of each multibeam ping or image acquisition, so on-board and post-processing software must be able to compensate for unstable motion and rough conditions. The location of the mapping sensors must also be calibrated relative to the navigation frame of the vehicle, which in this system is done as

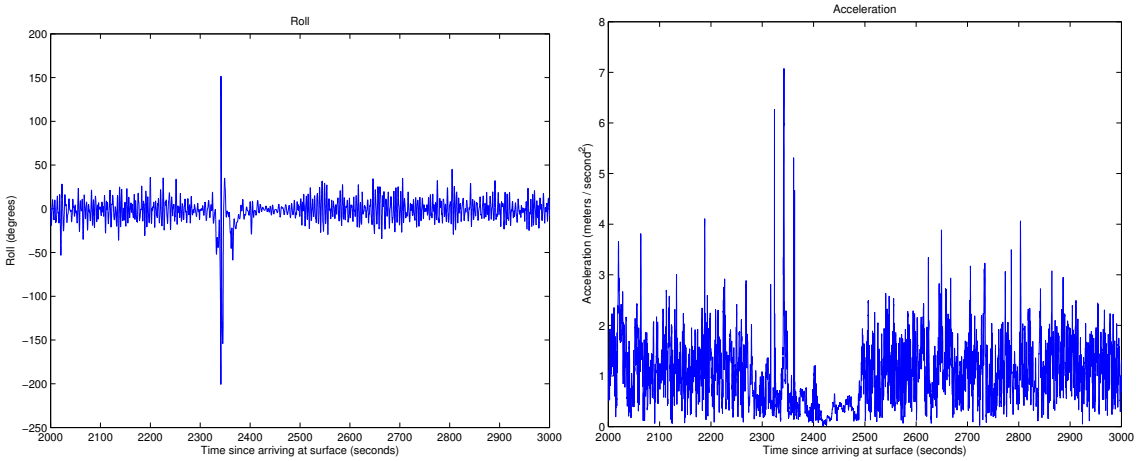

Figure 1-4: Roll and accelerometer readings during a rough recovery in the Bering Sea. The high acceleration rates occurred when the AUV collided with the ship.

part of the mapping processes, described in the following chapters. Finally, it must be noted that in any ocean deployment, the desire for high-quality data is subsumed by operational constraints and the fundamental equation of field robotics, namely

$$N_{recoveries} = N_{deployments}$$

the result of which is often incomplete data. See [34] for a discussion of AUV recovery from under Arctic ice cover, for example.

1.3 Simultaneous localization and mapping

The problems of robot navigation and mapping are coupled by the fact that mapping sensors can be used to provide feedback into navigation estimates. For the last few decades, this coupling has been extensively explored in the robotics literature as Simultaneous Localization and Mapping (SLAM). Problems in SLAM can be thought of as large optimization tasks, in which a set of cost functions which capture the inconsistencies between the robot's measurements of its own motion and of the environment are minimized. A good introduction and overview of the subject is presented in [58]. Measurements are made in a probabilistic framework, allowing the cost functions to be weighted according to the confidence with which a measurement is made or a constraint introduced.

Early work in SLAM addressed the problem of recursively estimating the robot's position and the location of all the landmarks in a map, the canonical example using an extended Kalman filter (EKF) [54]. More recently there has been increased interest in treating the entire pose history of a robot, rather than just the final or most-recent pose, because of conditional independences between variables that arise from such a formulation. One such formulation is the delayed-state information filter, which [10] used for vision-based mapping underwater in a system called Visually Augmented Navigation (VAN). The formulation used in this thesis, which relies more explicitly on the contributions of graph theory in matrix factorization, is square root information Smoothing and Mapping (SAM) [8], which considers the structured nature of the constraints in the navigation and mapping problem through the use of a pose graph. The sparsity of the pose graph is reflected by the sparsity of the Jacobian matrix of the cost function, yielding an efficient optimization which can also be used incrementally [29]. This formulation is described in more detail in the next chapter.

Working with acoustic sensors underwater presents additional challenges to localization and mapping. A multibeam or sidescan sonar can only be used incrementally to build a map, as each "ping" yields on the order of 100 to 500 ranges as a function of angle (in the case of multibeam) or time (in the case of sidescan sonar). Individual pings must be aggregated as the AUV moves and the sensor footprint sweeps over an area in order for distinguishing features of the terrain to become visible. Incorporating these sensors into a SLAM framework thus requires a decision about the scale at which features or landmarks are identified. One possibility, used in [49], is to build small independent maps, treating each as self-consistent, and then to align the maps in a global space. Another is to treat each range measurement as a hypothesis about whether a region of space is vacant or occupied; this approach is used in [12] and [1]. Finally, a bathymetric map can be built "by hand" by manually selecting sensed terrain features that should line up, adjusting the map to force these alignments, and then iterating until the finished product is acceptable. This is the approach taken by [6], which is not a SLAM approach at all, but is in common use in the oceanographic community. The approach followed here is closer to [49] than the others. Small submaps are aligned, each alignment induces a relative constraint between two AUV poses, and then the entire robot trajectory is smoothed over to accommodate these constraints together with the AUV's navigation estimates.

Most of the SLAM methods mentioned above share common underlying theoreti-

cal assumptions that the noise in the system (in measurement and control) is Gaussian, or closely approximated as Gaussian, and that the models of measurement and control are linear, or reasonably linearizable. Under the assumption of Gaussian noise and good linearization, the systems described above are optimal, in the sense that the results they provide are the Bayesian maximum likelihood solutions. The differences between them are largely to do with efficiency, though it is worth noting that the SAM technique iteratively re-linearizes the system at each step in the optimization, rather than relying on (for example) a Taylor series approximation as the EKF does. In addition to these techniques, many nonparametric SLAM frameworks exist which make no assumption of Gaussian noise, most notably those based on the particle filter, (again, see [58] for a good introduction and overview), or Rao-Blackwellized particle filter, which reduces the problem inherent to particle filters of sampling in a high dimensional space by introducing a set of Kalman filters for the motion half of the problem. These include many occupancy grid methods, including [12], [9], and [1]. While particle filters are well-suited for underwater robotics due to the non-Gaussian nature of acoustic ranging measurements, we found the SAM framework to be a natural fit to the AUV mapping problem, in particular because of the ease with which the pose graph can be adjusted to fit the available data and the set of parameters being solved for (see chapter 2). We compensate for the non-Gaussian nature of the problem with aggressive outlier rejection, a theme we will return to throughout the thesis.

1.4 Navigation and mapping relative to drifting, rotating terrain

There has been increased interest recently in the idea of using underwater vehicles to map areas that are not fixed to the seafloor. Those concerned with security of ocean-going vessels would like to use robots to autonomously inspect hulls for sabotage, a challenging problem due to the general lack of distinguishing "terrain" on the side of a ship that could be used as a navigation landmark. Nonetheless, researchers have used SLAM techniques to map ship hulls using either multibeam, as in [62], or using cameras, as in [30] – the former uses a filtering approach, while the latter uses a pose graph optimization approach similar to what is used in this thesis. While neither of these studies explicitly allow for the ship to move, it is clearly a forthcoming step in the re-

search.

More directly relevant to the work here is the recent use of AUVs to map moving icebergs. Icebergs present a very challenging and risky target for mapping, as they can be quite large, fast moving, and have significant draft. A good proof of concept of a solution to this problem is presented in [31], in which the goal is to map the sides of a moving, rotating iceberg with multibeam while circumnavigating it. The approach therein augments the set of variables being estimated with the velocity (in translation and rotation) of the iceberg, so that deformations that might otherwise be introduced into the map to make it self-consistent are instead used to estimate iceberg motion. A significant accomplishment of their work is that the iceberg is not itself instrumented; all iceberg motion estimation must be derived from vehicle odometry and the consistency of the derived map, using only a prior model assuming approximately constant iceberg velocity. By contrast, in this thesis we present a method for mapping the underside of drifting ice floes which makes use of measurements of floe rotation available from the ship moored to the floe. Another significant difference is that the work in this thesis uses a free-swimming AUV; while the work in [31] estimates a full six degrees of freedom for vehicle motion, their use of a ship as an AUV proxy limits the extent of the possible motion of the mapping platform.

1.5 Fusion of optical and acoustic data

As AUV technology has improved in the last decade there has been a strong push in the community to improve the high-resolution "microbathymetric" mapping capability of AUVs, using techniques including stereo imagery, high frequency multibeam sonar, structured light, and combinations of different modalities. There has been a clear progression in the state of the art, starting with 2-D optical maps (photomosaics) [48] [16], and optical maps with sparse 3-D information created using structure from motion [47], to visual SLAM solutions [10] [30]. As dense 3-D maps require either the use of a ranging sensor such as multibeam, or significant image overlap (each point on the terrain must be imaged from at least two locations), there has been increased use of stereo cameras (see, e.g. [43], [23] and [35]) and structured light. Of particular interest here are techniques that combine the best features of optical cameras with the capabilities of acoustic ranging sensors.

The approach in [25] combines multibeam bathymetry with multiple 3-D stereo

models using mesh registration. For simplicity, only sparse stereo models are computed, though the approach should work equally well with a good dense stereo algorithm. Navigation estimates are computed using VAN, and stereo meshes and multibeam maps are registered using the Iterated Closest Point algorithm [67], after which a single mesh is built from the individual pieces to create a large scale visualization. At a higher resolution, [50] replaces multibeam entirely with a laser and camera combination, with very impressive results. This requires precise calibration, however, and is currently only suited to use on a tethered remotely operated vehicle (ROV) due to the need for continuous imaging and relatively high power. On the other hand, their ranging system can be used as a drop-in replacement for multibeam in SLAM systems, and the fact that a camera is already used in the system suggests that building texture-mapped models will be a near-term research goal.

Finally, there has been recent work on directly coupling a single camera with a multibeam sensor on an AUV. The authors in [21] describe a method for calibrating a pair of such sensors on an AUV using a target in a test tank. Given the limitations of working in the field and the lack of a suitably large tank, the approach presented in this thesis does not rely on an explicit relative sensor calibration, but instead derives the relationship as a side-effect of the SLAM algorithm. Further examples of tight integration between cameras and acoustic sensors are presented in [42] and [41], which are concerned with local area imaging, rather than with the creation of large area maps.

1.6 Assumptions made throughout the thesis

Each navigation and mapping technique described in this thesis makes assumptions about the data it uses. These assumptions are described along with the algorithms, but for clarity and explicitness, they are quickly listed here as well. The fundamental assumption underlying any navigation and mapping technique that attempts to maximize map consistency is that there will be sufficient redundancy in the available data to constrain the set of possible solutions. For mapping with AUVs, this means in particular that absolute sensors measuring depth, attitude, and sound speed are available, as are sensors measuring the 3-D velocity vector. Such sensors are ubiquitous on AUVs. In addition, for the specific cases addressed here, we assume that the intrinsic parameters of on-board cameras including lens distortion have been calibrated in water, that there is at least 20% overlap image-to-image and trackline-to-trackline, and that similar side-

to-side overlap is provided on adjacent multibeam transects. For mapping rotating ice floes, we assume rotation rates less yielding less than 15 degrees over the course of a dive. Finally, we assume that sensors provide asynchronous timestamped data with minimal latency, that the mounting positions of cameras and multibeam sonar heads are known relative to the velocity sensor to within about 15 centimeters, and that the vehicle is reasonably stable in roll and pitch. Most notably, we do not assume that any absolute positioning system is available, such as GPS or high-frequency acoustic tracking systems. These assumptions are quite mild, in that underwater vehicles must satisfy all of them in order to be useful as scientific and engineering research platforms – an excessively unstable vehicle will not make it into the water and thus will not be providing data to a navigation and mapping system.

1.7 Organization of the thesis

The contributions of the thesis include: a method for improving navigation by recovering roll and pitch offsets between the attitude sensor and the 3-D odometer, an efficient automatic method for matching 3-D structure from multibeam data to add constraints to the pose graph, the generalization of the navigation task to moving and rotating terrain, and the simultaneous incorporation of visual and acoustic data into a single mapping framework. The upshot of these contributions is the possibility to produce maps including visual texture with resolution five times better than the current state of the art.

The thesis proceeds with the steps necessary to build acoustic, optical, and integrated maps using AUVs. In the next chapter, we examine the basic problem of AUV navigation using odometry and external navigation beacons, introducing the SAM pose graph which is elaborated throughout the thesis. Even this basic navigation problem is a SLAM problem, as the locations of navigation beacons are estimated simultaneously with the trajectory of the robot. The first novel application of the pose graph is to compute attitude offsets between navigation sensors on the AUV, making use of the redundancy between the DVL, the fibre optic gyro, and the depth sensor. We then proceed in chapter 3 to incorporate the first mapping sensor, the multibeam sonar. We describe how local multibeam maps are matched to provide pose constraints between poses in the graph, how match confidence is estimated, and show results mapping fixed terrain. We then move to optical mapping in chapter 4, focusing on dense mapping of

a 2-D area of sea floor. We examine two different kinds of constraints that matched images can provide on the pose graph, and compare the mapping problem using SAM to traditional techniques in photomosaicking. We continue with the simultaneous construction of a 3-D optical map, fusing multibeam sonar with dense imagery, using the pose graph to model constraints induced by both sensors at the same time. In chapter 5 we augment the pose graph to take into account the case where the terrain being mapped is moving and rotating, and closely examine the particular case of mapping the draft of an ice floe. Finally, we discuss results, implications, limitations, and future work in chapter 6.

Chapter 2

AUV Navigation

2.1 Underwater localization

As mentioned in the previous chapter, most underwater vehicles carry a diverse suite of navigation sensors which together provide a redundant and often conflicting set of pose estimates. Underwater, a body can move through six degrees of freedom, and on most vehicles four of these six degrees of freedom are directly measurable: depth, roll, pitch, and heading. This is the case on manned submersibles and tethered remotely operated vehicles as well as on free-swimming AUVs. Onboard pose estimators often simply accept measured values as truth, or minimally filter measured values to smooth out noisy signals. Underwater localization is then reduced to estimating the remaining two degrees of freedom corresponding to horizontal position. There are several approaches commonly in use, described below in order of the amount of additional infrastructure required.

Firstly, an underwater vehicle can be equipped with a velocity sensor. On any vehicle expected to work within a few hundred meters of the seafloor, the typical sensor will be a doppler velocity log (DVL), which measures 3-D velocity relative to the terrain. These sensors only work within range of the terrain, however, which is dependent on their operating frequency. For 1200 kHz systems, for example, the working range is about 40 meters. Lower-frequency units can achieve much greater range, at the expense of larger transducer sizes. When DVLs are out of range of hard terrain, velocity can usually still be measured relative to the water mass, but the AUV will not be able to detect the motion of the water itself as it moves through it. When a DVL is not available, velocity estimates can be derived from thruster speed or electrical current draw, or

by integrating accelerometer measurements, but these are usually not the preferred methods because they are noisy. Regardless, once a 3-D velocity estimate has been determined, a pose estimate can be derived by integrating the velocity estimate using the vehicle's attitude sensor to determine the direction of motion in the geo-referenced frame. The pose estimate will be relative to an arbitrary origin, but can be used for relative navigation. For navigation in an Earth-fixed frame, either velocity estimates must be available on the surface, where GPS can anchor the trajectory, or additional infrastructure is required.

External navigation infrastructure is often provided underwater by a network of acoustic beacons. The particular system used depends on the deployment context, but it is usually a long-baseline (LBL) or ultra-short baseline (USBL) network. In these networks, positions are computed based on the acoustic travel time of signals sent between the vehicle and a set of beacons. The beacons may be moored to the seafloor or attached to a surface buoy or ship, but they generally have a location known to the underwater vehicle. The vehicle localizes by interrogating the network, transmitting an acoustic signal. The beacon responds with its own acoustic signal, and the vehicle determines its range to the beacon by measuring the response time and multiplying by the speed of sound. In traditional LBL navigation, the robot interrogates all of the beacons in the network simultaneously, and if, after outlier rejection, at least two reasonable ranges can be determined, a fix is computed geometrically by intersecting the horizontal plane at the robot's depth with the two spheres induced by the ranges to the LBL beacons [40]. If the ranges are in fact reasonable, then in a network with two beacons there will be two solutions, symmetric about the line between the beacons (called the *baseline*), and the ambiguity is resolved by laying out the network ahead of time to keep the vehicle to one side of the baseline at all times. A simple on-board pose estimator may take these fixes as truth, using high rate velocity-based pose updates as it moves, and resetting the integrated position based on acoustic fixes when they are available. The two methods can also be used together in a complementary filter [64].

There are several variations to LBL that are dictated by context. In the case of USBL, a 2-D bearing measurement (comprising azimuth and elevation) to the beacon can be made, though the use of a transducer array. A vehicle's location can be estimated passively by listening to the interrogations and responses and then measuring the difference in travel times, yielding a hyperbolic solution [24]. Finally, position fixes estimated on the surface (e.g. through acoustic devices too large to carry onboard a

vehicle) can be combined with GPS measurements and then transmitted to an underwater vehicle, either via tether, or using an acoustic modem. Another approach is to compute ship-relative fixes underwater, and then add global context from GPS via an acoustic link [11]. Regardless of the approach, each position estimate using an external sensor can be thought of as a kind of measurement, analogous to the measurements made by the other on-board sensors, and these measurements must be combined in some way to estimate the vehicle's trajectory. We use the abstraction of a pose graph as a tool to perform this estimation.

2.2 Pose graphs

Because they are used as a tool throughout the thesis, we begin with a description of pose graphs as a representation of the navigation and mapping problem, starting with the simple example of integrating velocities to form an estimated robot trajectory. The pose graph consists of two types of nodes, *pose nodes*, and *factors*, which are connected in a bipartite fashion. The pose nodes represent variables to be estimated, while the factors represent constraints on the variables in the pose nodes to which they are connected. In other words, the pose nodes as a group encapsulate the trajectory, map, and other parameters, and the factors as a group encapsulate the robot's measurements and other indirectly measured constraints. The error function captures the difference between what is measured and what is predicted by a given trajectory and map; each factor therefore encapsulates a small part of the total error function, and carries with it a measurement of some kind and a matrix which weighs the contribution of the factor in the overall error function relative to the others. This weight matrix is called an information matrix, because in the linear case if it is set equal to the inverse of the measurement covariance matrix, then the solution with the least overall squared error will be the best linear unbiased estimator of the parameters. The structure of the pose graph directly mirrors the sparsity structure of the error Jacobian function, as changes in variables only affect the error terms of factors to which the pose nodes representing the variables are connected. See [8] for an excellent introduction, including a description of how the sparsity of the problem and the structure of SLAM problems in general lead to efficient solutions. An example pose graph for AUV odometry integration is shown in figure 2-1.

Throughout the thesis, the six degree of freedom AUV pose at time t will be desig-

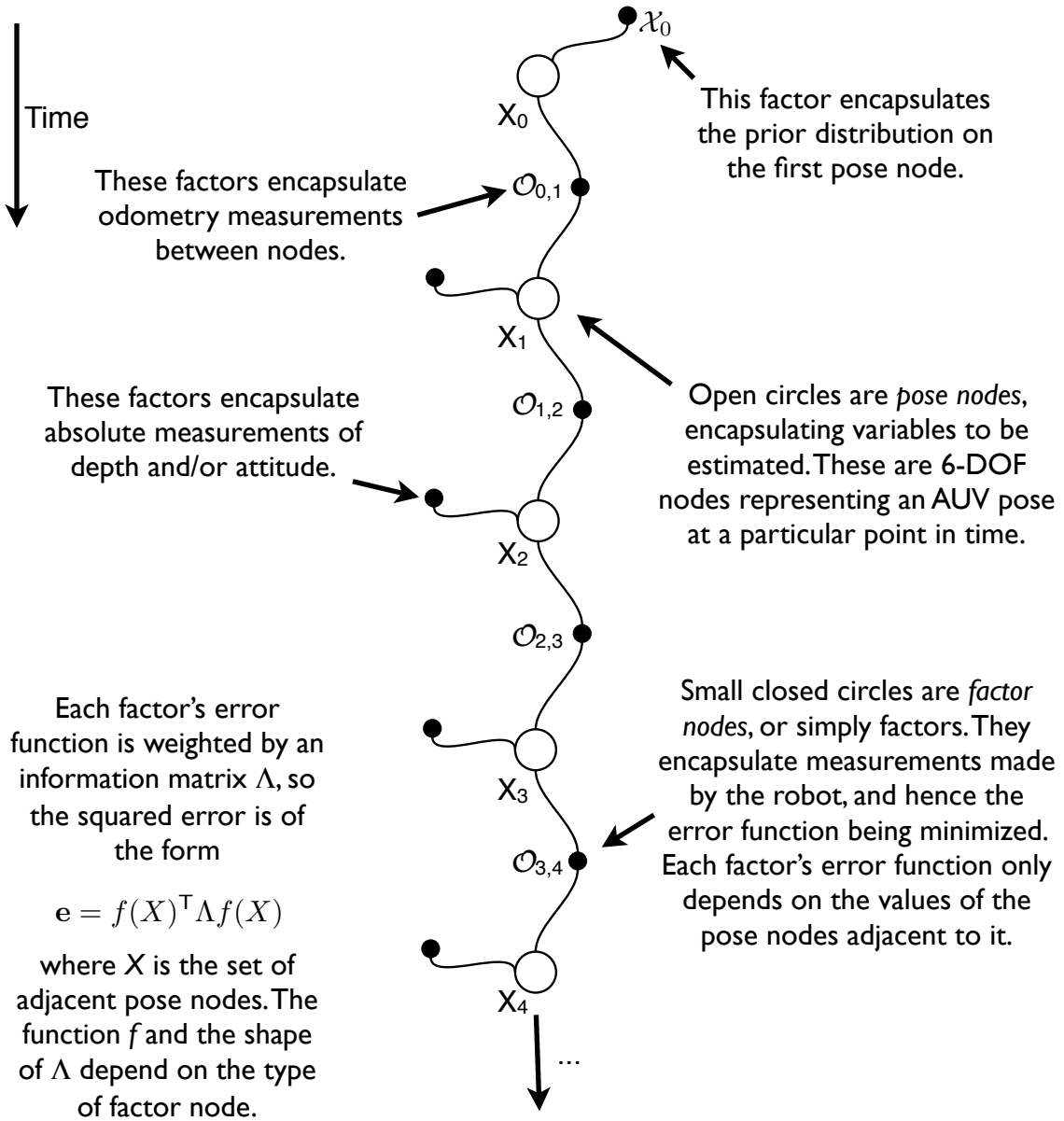

Figure 2-1: The pose graph for integrating velocity measurements and absolute depth and attitude measurements into an AUV trajectory estimate. The trajectory is "anchored" by the first factor node, which encapsulates the prior estimate of the starting AUV position.

nated X_t. Landmarks, with three degrees of freedom, are designated L_i. Other quantities to be estimated (sensor offsets, floe orientations, and so on) will also be designated with capital letters. Pose nodes will be referred to interchangeably with the variables they encapsulate, to make the nomenclature less cluttered. Factors are represented by uppercase script letters, so an odometry factor $\mathcal{O}_{X_t,X_{t+1}}$ will thus relate pose nodes X_t and X_{t+1} rather than something like P_{X_t} and $P_{X_{t+1}}$. An AUV pose X_t comprises six scalars $(x, y, z, \rho, \phi, \theta)$ where (x, y, z) represents meters north, east, and down (toward the center of the Earth) from an arbitrarily defined local origin (at sealevel). The AUV's local frame is defined to have the x axis pointing forward, y axis pointing starboard, and z axis pointing downward, so that (ρ, ϕ, θ) represent roll (rotation about the x axis), pitch (rotation about the y axis), and heading (rotation about the z axis). These frame orientations keep everything right-handed, with positive values for z underwater, and imply that heading angles correspond to what one would expect to read on a true north seeking compass.

In figure 2-1 the odometry factor $\mathcal{O}_{t,t+1}$ linking poses X_t and X_{t+1} carries with it the 6-DOF measurement of relative motion $O_{t,t+1}$, such that the error will be zero if $X_{t+1} = X_t \oplus O_{t,t+1}$, where the $\oplus$ symbol designates the *compounding* operation from [54]. To be explicit, X_t, X_{t+1} and $O_{t,t+1}$ can be equivalently expressed as 4×4 matrices of the form

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{R}_t & \mathbf{t}_t \\ 0_{1\times 3} & 1 \end{bmatrix}$$

with $\mathbf{R}_t$ the 3×3 rotation matrix derived from the roll, pitch, and heading values in X_t, and $\mathbf{t}_t$ the 3×1 translation vector $\begin{bmatrix} x & y & z \end{bmatrix}^\intercal$ in X_t. Then

$$X_{t+1} = X_t \oplus O_{t,t+1} \iff \mathbf{X}_{t+1} = \mathbf{X}_t\mathbf{O}_{t,t+1} = \begin{bmatrix} \mathbf{R}_t\mathbf{R}_{O_{t,t+1}} & \mathbf{R}_t\mathbf{t}_{O_{t,t+1}} + \mathbf{t}_t \\ 0_{1\times 3} & 1 \end{bmatrix}$$

These are strictly rigid body motions, so the last row of each transformation matrix is always $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$. Another way to think about these matrices is as transformations: $\mathbf{X}_t$ transforms points from the AUV's local reference frame at time step t to the world frame, and $\mathbf{O}_{t,t+1}$ transforms points from the AUV's reference frame at time $t+1$ to the AUV's reference frame at time t. Clearly, in the absence of other constraints, a zero error solution can be obtained by "walking down the chain," multiplying the previous pose by the odometry measurement to obtain the next pose.

The error function for an odometry node is then derived from the difference between the measured odometry $O_{t,t+1}$ and the odometry implied by the relative transformation between X_t and X_{t+1} for a given set of values for these pose nodes. The error can be expressed compactly as $(X_{t+1} \ominus X_t) - O_{t,t+1}$, where $A \ominus B$ is defined as $B^{-1} \oplus A$, and the inversion can be derived from the matrix expression above. This error vector has six elements, corresponding to the six degrees of freedom for the relative pose, so three of the terms capture angle differences and must be normalized to lie between $-\pi$ and π. This amounts to comparing 3-D rotations by subtracting Euler angles, which is only valid for small differences; fortunately for odometry factors the angles involved are very small. Other formulations for comparing 3-D rotations include using the axis of a normalized quaternion as in [33], or using the Rodrigues vector (axis of rotation multiplied by rotation magnitude), equivalent to members of the Lie algebra $\mathfrak{se}(3)$ as in [57]. Throughout the thesis, we assume small angles and simply use angular differences, which has been well justified by several data sets collected by the AUV.

The goal of the SAM algorithm is to find values for all the variables in the pose nodes, such that the overall error function

$$\sum \mathbf{e}_{\mathcal{F}}^{\mathsf{T}} \Lambda_{\mathcal{F}} \mathbf{e}_{\mathcal{F}}$$

is minimized, where the sum is over all factors in the graph, and $\Lambda_{\mathcal{F}}$ is the information matrix for factor $\mathcal{F}$. This is a weighted nonlinear least squares problem, because the error term $\mathbf{e}_{\mathcal{F}}$ for each factor is in general the result of a nonlinear function, particularly because of the conversion from rotation matrices to Euler angles. The optimization can be solved with a good initialization point by code publicly available [27] from the authors of [29]. The overall translation ambiguity of the problem is addressed by either forcing the first pose to lie at the origin, or by "anchoring" it with a *prior factor* $\mathcal{X}_0$, as shown in the figure, which has error function $\mathbf{e} = X_0 - \mathcal{X}_0$, so that the error is zero if the pose X_0 matches the "measurement" contained in the factor.

2.3 AUV pose graph navigation

Using a pose graph for underwater navigation requires deciding on a set of pose node and factor types. The first question to address concerns the number of degrees of freedom in the pose nodes representing the AUV's state. The Seabed AUV carries two

independent attitude sensors, an absolute depth sensor, a 3-axis velocity log, and a 3-axis rotation rate sensor. This means that of the six degrees of freedom that the robot can move through, there are absolute measurements available for four (z, roll, pitch, and heading). While it might be tempting for the sake of simplicity to use a pose graph optimizer only to estimate the remaining two degrees of freedom, there is sufficient latency and noise in the measurements to benefit from estimating all six. In addition, the framework makes adding partial measurements of this type straightforward: the pose graph shown in figure 2-1 contains factors representing these kinds of measurements. Additional external absolute measurements of position (e.g. from GPS or long-baseline navigation networks) will be added in the next section.

Each kind of measurement that can be made by the AUV must correspond in the graph to a type of factor node; each type of factor node has its own error function and rules about the kind and number of pose nodes to which it must be connected. The on-board depth and attitude sensors measure four degrees of freedom, and are represented by the factor nodes $\mathcal{X}_t$, with the exception of the prior on the first pose, $\mathcal{X}_0$, which includes a prior on x and y to anchor the trajectory in space. Although the navigation sensors are not synchronized, the update rates are high enough that they can be interpolated to a fixed time base, or to the rate of a reasonably fast sensor; we choose to use the doppler velocity log, which provides an update at about 7 Hz as the "base" sensor. The error function for these 4-DOF measurements $\mathcal{X}_t = \begin{bmatrix} \hat{z}_t & \hat{\rho}_t & \hat{\phi}_t & \hat{\theta}_t \end{bmatrix}$ is simply

$$\mathbf{e}_{\mathcal{X}_t} = \begin{bmatrix} z_t & \rho_t & \phi_t & \theta_t \end{bmatrix}^\mathsf{T} - \begin{bmatrix} \hat{z}_t & \hat{\rho}_t & \hat{\phi}_t & \hat{\theta}_t \end{bmatrix}^\mathsf{T}$$

i.e. the difference between what is predicted by the pose and what is measured by the sensors, again with the angle differences fixed to be between $-\pi$ and π radians. The associated information matrix is diagonal, with fixed values $\begin{bmatrix} 16 & 57 & 57 & 57 \end{bmatrix}$, based on the expected accuracy of the sensors – these correspond to variances of 0.0625 meters in depth, and one degree in each of roll, pitch, and heading. These attitude variances are reasonable because the primary attitude sensor, a fibre-optic gyro, does not rely on the Earth's magnetic field to determine heading; the standard deviation according to the manufacturer is 0.01 degree in roll and pitch, and 0.1 degree times the secant of latitude in heading, but a larger variance is used to allow for latency.

The 6-DOF odometry factors $\mathcal{O}_{t,t+1}$ incorporate relative measurements made by the doppler velocity log (DVL). Each DVL sensor reading includes velocity measure-

ments along the three orthogonal axes in the sensor's reference frame, relative to the sea floor; these measurements are rotated into the vehicle's reference frame as forward, starboard, and downward velocities $\begin{bmatrix} u & v & w \end{bmatrix}^{\mathsf{T}}$. The origin of the AUV's frame is set at the DVL, rotated to make the frame "level,"accounting for roll and pitch biases in the mount. It is important to note that these velocity measurements are terrain-relative, so if the terrain is not flat (relative to gravity) there will be a nonzero w component to the velocity vector even if the AUV is not ascending or descending. Changes in attitude, on the other hand, are provided by the fibre-optic gyro, and are independent of the local terrain. An odometry factor, therefore, contains a measurement capturing the local translation

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} = \delta t \mathbf{R} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

and a measurement capturing the local change in rotation, which is computed by converting the rotation rates into axis-angle form, and then multiplying the angle by δt and converting back to Euler angles. The rotation matrix $\mathbf{R}$ in the above equation captures the permutation from the DVL frame to the vehicle frame, which includes roll and pitch bias terms mentioned above.

Building the pose graph in this way allows both odometry measurements and absolute depth and attitude measurements to contribute to the overall estimation of the trajectory. Since these measurements together overconstrain the system, using all of them should provide a better trajectory estimate than simply using the depth and attitude measurements as truth, which is how the on-board pose estimator on the AUV uses them. In fact, the redundancy provided by these sensors enables us to improve the calibration of the DVL bias, as will be shown in section 2.5.

Since the system is overconstrained it is important to consider the starting estimate for the trajectory, because without a good starting point the iterative optimization process can converge to a local minimum, or even fail to converge. For this case, it is sufficient to initialize each pose node with a modification of the "forward chaining" procedure described above, integrating the velocity at each node to produce a pose estimate, but forcing the local attitude and depth to match interpolated values provided by the depth sensor and fibre-optic gyro. The SAM framework then smooths out the trajectory based on a combination of the odometry and the absolute sensors. Because the disagreement between these sensors is generally small, initializing the graph this

way is sufficient to yield a good solution.

2.4 External navigation beacons

There are two possibilities for incorporating LBL and USBL measurements into the pose graph. To keep the navigation as close to the traditional methodology as possible, the measurement factors could capture Euclidean distance from the 2-D position fixes induced by LBL interrogations – this might be how one would incorporate GPS fixes into a pose graph, for example. A more flexible approach is to treat the problem like a small SLAM scenario, in which each measurement is an individual range to one of the beacons, the locations of which are not fixed. As each beacon is uniquely identifiable by the robot, data association is not an issue. This landmark-based pose graph for SLAM is shown in figure 2-2. The graph includes pose nodes for estimating the location of each beacon in the map, factors capturing the range measurements, and factors capturing the prior estimate of each beacon's location. The pose nodes do not have orientations, because the beacons are omnidirectional, and the ranging factors only have a single error term, so the information matrix is a scalar, i.e. the inverse variance of the range measurement.

A nice effect of using the pose graph framework in this way is that individual ranges can contribute independently to the trajectory estimate, since each measurement constrains part of the whole trajectory, rather than just the robot's pose at a particular point in time. In the extreme case, LBL measurements could be used to constrain the trajectory even if no consistent fix can ever be made at a single point in time – this is analogous to using these ranges to compute "running fixes" in traditional navigation.

The addition of acoustic landmarks into the pose graph exposes some of the underlying assumptions about the SAM framework. Most important is the use of the term *variance* in describing the range measurements to the beacons. If one treats each measurement as a random variable, then describing the measurement with only two terms (the value and its variance) tends to imply that the underlying distribution of the variable is Gaussian. Acoustic signals, however, are notoriously subject to multipath and refraction, and yield ranges which are decidedly non-Gaussian, as can be seen in figure 2-3. One approach to this problem would be to rely on a range model which more accurately reflects the physics, or to switch to a non-parametric estimator such as a particle filter. The approach taken here is to use aggressive outlier rejection, removing ranges

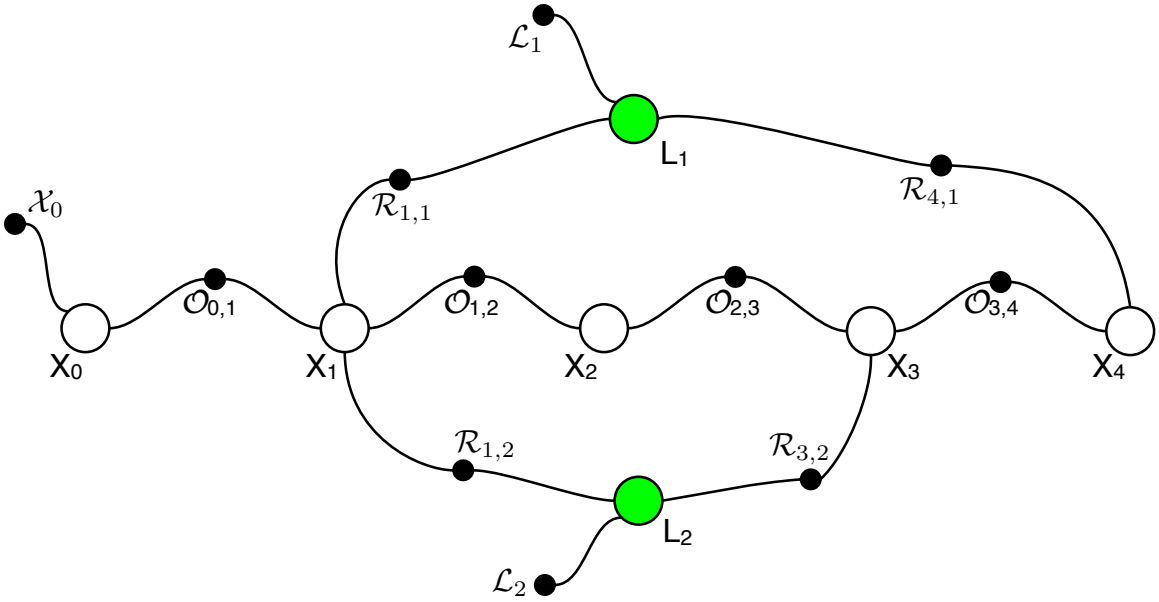

Figure 2-2: The pose graph extended to make use of an external acoustic navigation network. Depth and attitude measurement factors are not shown. The additional factors $\mathcal{R}_{t,i}$ are scalar range measurements from pose X_t to 3-DOF pose nodes representing the location of the acoustic beacons L_i in space. The factors $\mathcal{L}_i$ capture prior estimates of the beacon locations.

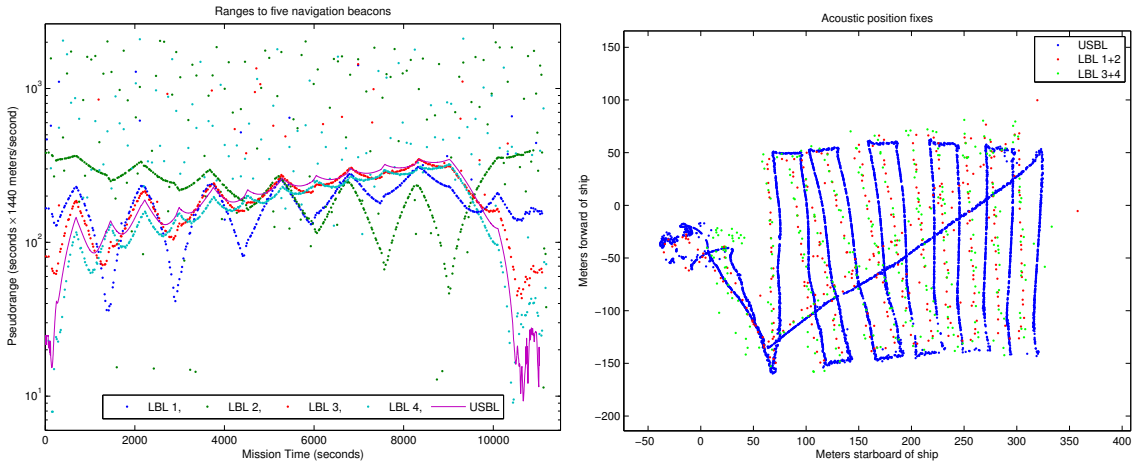

Figure 2-3: Ranges taken to a set of four LBL beacons and one USBL beacon during an under-ice AUV dive. The vertical axis is a log scale to show the range of outliers, and the noise is decidedly non-Gaussian. On the right are position fixes induced by two pairs of LBL beacons and the single USBL fix, both with outliers rejected.

that imply excessive robot speed. It is important to note that while the factor only encapsulates the measurement and its variance, there is no real reliance on the shape of the underlying distribution; the error function is assumed to be nonlinear and is solved iteratively.

Incorporating these acoustic beacons into the SAM framework still requires us to choose an appropriate variance value. As is clear from the right side of figure 2-3, the various position fixes vary widely compared to the desired centimeter-level resolution of the final map. Since each range measurement is a scalar, it is straightforward to adjust the information term at each factor and see how the resulting AUV trajectory changes. Figure 2-4 shows three trajectories, one of which only uses AUV proprioception, and two of which use differing values for the variance of the acoustic range measurement. As expected, varying the weight changes the computed trajectory. Unfortunately, no choice of weight appears to be optimal. Using a variance of 100 meters in the range measurement ($\lambda = 0.1$) might produce a consistent map, but the map will remain biased due to the consistent influence of refraction which varies according to position – this is particularly the case with under-ice data, when the AUV is traveling close to the surface under an acoustically reflective barrier. Decreasing the variance to 4 meters ($\lambda = 0.5$) reveals the instability of the problem, as the best solution for these data places two of the acoustic beacons above sea level, even with a strongly weighted prior on their

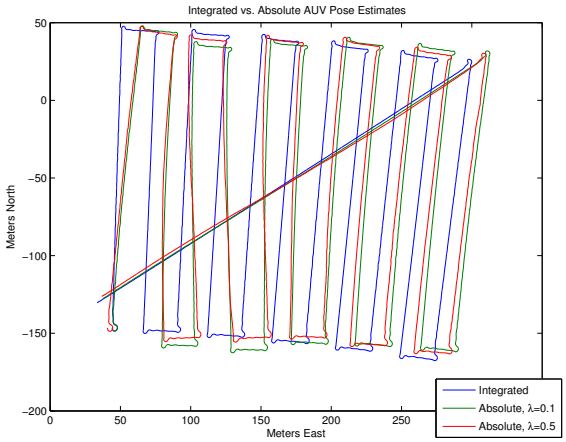

Figure 2-4: Trajectory estimates resulting from different weights applied to the acoustic range measurement factors. The λ values are the square root of the information (inverse variance) used in each factor. The "integrated" trajectory uses the basic pose graph shown in figure 2-1.

locations.

The conclusion to be drawn from adding acoustic beacons to the pose graph is that the variability of acoustic navigation is too high to make it useful for building high-resolution (centimeter scale) bathymetric maps. On the other hand, it is entirely reasonable to use acoustic navigation for AUV tracking and recovery (i.e. operationally) and for building maps on the scale of a few meters, which is in fact what is often done. This conclusion is consistent with what was found in [1], which incorporated both USBL and multibeam measurements into a particle-filter framework, producing maps built from close range (the AUV was 20 meters from the seafloor) but with one meter bin sizes.

2.5 Calibration of navigation sensors

As mentioned above, as an AUV moves through the water, it measures its 3-DOF velocity relative to the terrain, and also measures its absolute depth and possibly its horizontal position using acoustic beacons. The redundancy of these measurements provides the capability to calibrate the attitude offset of the velocity sensor relative to the level frame of the robot. If this offset is not determined, then the position estimate computed by integrating velocity measurements will be incorrect – if there is a pitch bias,

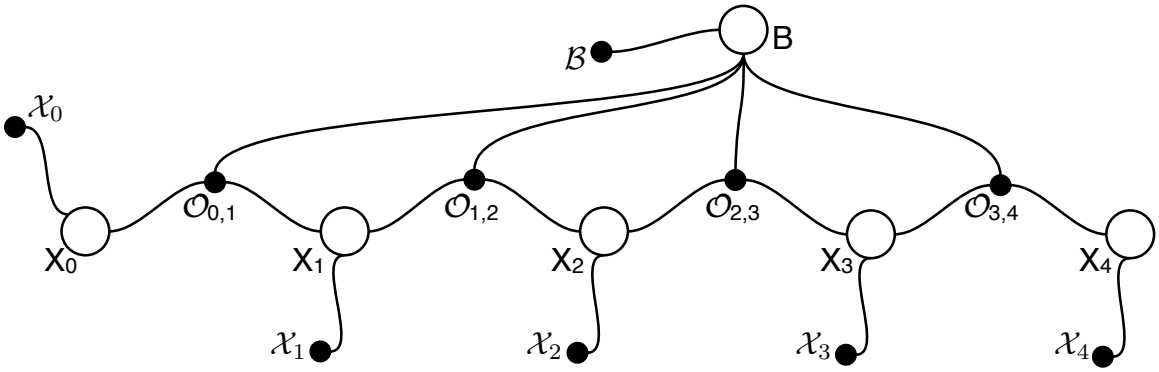

Figure 2-5: The pose graph used to estimate the roll and pitch bias of the DVL. The error function in each odometry node takes into account the roll and pitch bias encapsulated in the 2-DOF pose node B with prior $\mathcal{B}$, so that the integrated vertical velocity better matches the position estimates of the absolute depth sensor.

then some component of forward velocity will instead be attributed to vertical motion, and a roll bias will cause starboard velocity to be similarly confused with vertical motion. A heading bias, on the other hand, will not appear as phantom vertical motion, but will cause incorrect horizontal positions to be estimated, as forward and starboard velocities will be confused with each other.

There have been a few attempts to calibrate this attitude offset using LBL, most notably in [32] and more recently in [56]. Both of these attempts framed the problem in the language of control theory, and drove a bias error term toward zero by feeding LBL position estimates back through a proportional controller. As shown in the previous section, acoustic localization is not currently accurate enough to produce centimeter-level bathymetric maps, so we calibrate the DVL roll and pitch offsets using only proprioceptive sensors. An alternate approach using only 3-axis accelerometers and a gyrocompass (i.e. no absolute depth sensor) is described as preliminary work using only simulated data in [60]. Our calibration takes place as part of the pose graph optimization, using a graph like that shown in figure 2-5. This graph differs from those discussed earlier in that it contains factor nodes which connect to three pose nodes rather than just one or two. This is perfectly acceptable in the framework, and is one reason why factor graphs are easier to visualize than SLAM hypergraphs, in which constraints are captured by edges (which can connect more than two nodes) rather than by factor nodes.

The modified odometry factors in the figure have the same relative attitude error

term as the odometry factors from above, since bias in roll and pitch do not affect the relative attitude change between two poses. The translation is affected by bias, however, so the translation portion of the error term does take the estimated roll bias $\tilde{\rho}$ and pitch bias $\tilde{\phi}$ into account. Recall that the odometry error term from above was $\mathbf{e}_{\mathcal{O}_{i,j}} = (X_j \ominus X_i) - O_{i,j}$. The first term expands to

$$\begin{bmatrix} \mathbf{R}_i^\mathsf{T}\mathbf{R}_j & \mathbf{R}_i^\mathsf{T}\mathbf{t}_j - \mathbf{R}_i^\mathsf{T}\mathbf{t}_i \\ 0_{1\times 3} & 1 \end{bmatrix}$$

The upper-left term, the relative attitude change, is retained, while the upper-right term is changed in this factor to $\mathbf{R}_i^{*\mathsf{T}}\mathbf{t}_j - \mathbf{R}_i^{*\mathsf{T}}\mathbf{t}_i$ where $\mathbf{R}_i^*$ is computed from

$$\begin{bmatrix} \rho_i + \tilde{\rho} \\ \phi_i + \tilde{\phi} \\ \theta \end{bmatrix}$$

As above, the resulting 4×4 matrix is converted back into a 6-DOF vector. Designating this biased transformation as $(X_j \ominus_B X_i)$, the new error function for the odometry factors becomes $(X_j \ominus_B X_i) - O_{i,j}$. The bias pose node can be initialized to the value in the prior factor $\mathcal{B}$ in the graph optimization, which will also constrain the extent of allowable bias based on the value in its information matrix. In the experiments in this thesis, however, we initialized the bias to zero and did not use a prior factor to limit the optimization.

The results of including roll and pitch bias in the pose graph optimization are shown in figure 2-6 for one dive. The depth of the AUV, as computed from integrated vertical velocities, matches measurements from the absolute depth sensor much more closely than before the optimization, though the match is still not perfect. This is because velocity integration is always subject to drift, and because the velocity measured is relative to the local terrain, which is not flat. In later chapters we will also show how including this step in the mapping process changes the estimate of the robot's trajectory, and is therefore useful as a preliminary navigation refinement.

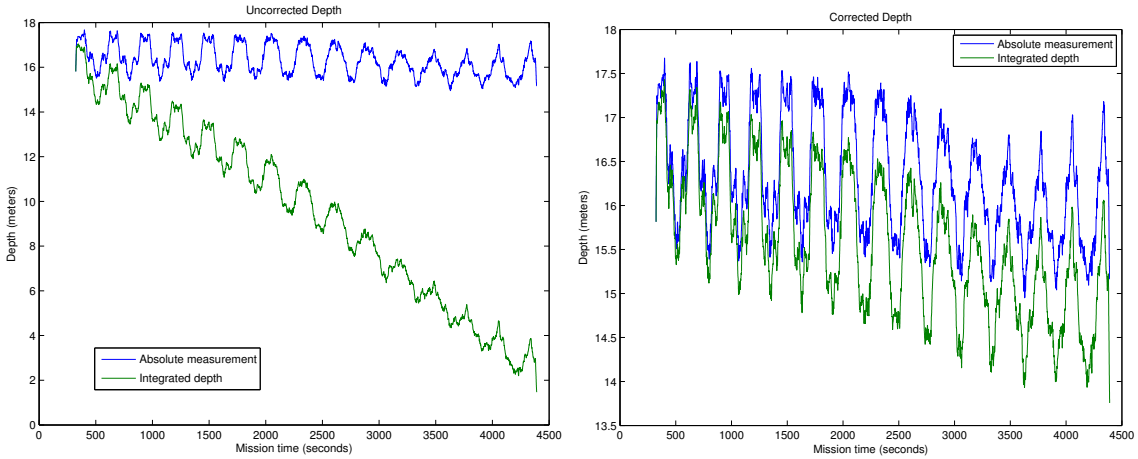

Figure 2-6: A comparison of absolute depth values for a dive with integrated vertical velocity. On the left, the vertical velocity vector used at each time step assumes zero roll and pitch bias. On the right, the velocity has been compensated by roll and pitch biases computed using the pose graph in figure 2-5.

Chapter 3

Multibeam Navigation and Mapping

3.1 Incrementally building a multibeam map

Multibeam echo sounders are the underwater analog to the laser scanners used in terrestrial robots. They produce a set of ranges in a fan-shaped pattern with a field of view of 90 to 120 degrees. For seafloor mapping, the sensors are oriented in "profiling" mode, pointing straight down from the vehicle to which they are attached, so that the field of view is perpendicular to the direction of travel. Because of the finite speed of sound (about 1500 meters per second in sea water), the principles of operation of multibeam sonars are different than those of laser scanners. Laser scanners measure ranges effectively instantaneously, and line scans are effected by a spinning platform in the sensor. Multibeam sonars, on the other hand, measure a whole set of ranges with a single pulse of sound, by "beam forming" the emitted pulse into a fan shape, and then measuring the difference in time of arrival (or the phase differential) of the echoes at several different transducer positions within the sonar head. This allows a ship or an underwater vehicle to create a dense bathymetric map even if it is moving faster than a meter per second through the water. It is important to note that proper beam forming requires good knowledge of the speed of sound, which varies with water density; changes in sound speed affect both the ranges measured by the sensor and their angles from the sensor head. These sound speed measurements are generally provided by a separate CTD (conductivity-temperature-depth) sensor.

Building a map with multibeam data is thus just a question of geometry – if the pose S_t of the multibeam sonar head is known at the time t of each ping, then the range from each beam can be placed into a globally-referenced point cloud, which can eventually

be gridded into a map. The mapping problem is then reduced to determining the head's location at particular points in time. The AUV navigation problem, introduced in the previous chapter, provides a good starting estimate of the AUV's trajectory, so a first cut at multibeam mapping needs the 6-DOF pose offset P between the AUV's frame and the multibeam head, as well as synchronization information so that the best pose estimate can be used at each ping time. To summarize, if X_t is the AUV's pose at time t, P is the multibeam head pose offset, R_j is the rotation describing the beam angle for beam j, and $b_{t,j}$ is the range for beam j at time t, then the corresponding point $p_{t,j}$ is

$$p_{t,j} = X_t \oplus P \oplus R_j \oplus b_{t,j}$$

and the mapping problem requires estimating X_t and P.

One pervasive problem in underwater mapping is lack of available ground truth. With terrestrial robots, it is often possible to check a map against a benchmark, which can be produced using surveying equipment using GPS and fixed laser scanners. Underwater neither GPS nor lasers work, and the sensors available for validation are the same as those used to build maps in the first place. Only in test tanks which can be drained can ground truth be determined. In the ocean, then, the best metric we can use for measuring map "goodness" is self-consistency. The easiest way to measure self-consistency in a multibeam map is to grid, or bin the map down to a fixed resolution, and then to check the variance of the gridding. By doing so, the map is effectively reduced from a 3-D point cloud to a "2.5-D" height map, in which each (x, y) location has a single value for depth, height, or draft (z). The area being mapped is divided into a regular grid, and then the points within each grid cell are grouped. The mean z value becomes the depth value for the cell, and the variance of the z value for the points is an estimate of the consistency of the data. While the consistency of a map can be visually estimated by examining the binning variances in regions seen by multiple tracklines, a quantitative measure of the goodness of the entire map is the average variance over all the cells, or (equivalently, if the number of cells does not change from map to map) by the sum of the variances over all the cells.

3.2 Determining sensor offset

As mentioned above, knowledge of the 6-DOF offset between the AUV frame and the multibeam is a prerequisite for the ability to build a consistent map. While the translation offset is relatively easy to approximate with a tape measure, determining the attitude offset presents more of a problem. In [53], it was shown that the three degrees of freedom in the attitude offset can be treated independently, if one considers roll first, then pitch, and finally heading, assuming that the multibeam is mounted in the typical profiling configuration. Fortunately, the influence of errors in the estimate of each of these degrees of freedom are relatively easy to recognize. The estimation of these parameters requires the AUV to move through a specialized motion, such that the terrain being mapped is ensonified more than once, and from different attitudes. For roll offset estimation, for example, the AUV should either rotate over the terrain being mapped, or move over the same patch of flat terrain from opposing directions. Estimation of the attitude offset is often done manually, using software such as *MB-System* [6], or other commercial solutions. Here we estimate the attitude offset automatically, assuming sufficient overlap in the AUV's trajectory and sufficient generality of the AUV's motion, by examining the total variance of gridded maps produced using each hypothesized attitude offset.

The assumptions of general motion and sufficient overlap have been justified empirically when the AUV moves through typical lawnmower-style survey patterns such as those shown in the previous chapter. For each degree of freedom, the system starts with an approximate initial guess using the nominal mounting angle of zero degrees for mapping the seafloor (the starting roll estimate is 180 degrees for mapping the underside of sea ice), and then steps through a small set of offsets, producing a gridded map for each angle. The error term being minimized is the sum of variances of each grid bin:

$$e(\theta) = \sum_{c \in M} \frac{1}{N_c} \sum_{z \in c} z^2 - \bar{z}^2$$

where the outer sum is over each cell in the gridded map, and the inner sum is over each point in the given grid cell. The error function is well-behaved when the robot's motion in sufficiently general, as shown in figure 3-1. As expected ([53]), the greatest error is contributed by the roll offset, followed by the pitch and heading offsets – the vertical axes on the graphs in the figure are different from each other. When the multibeam is

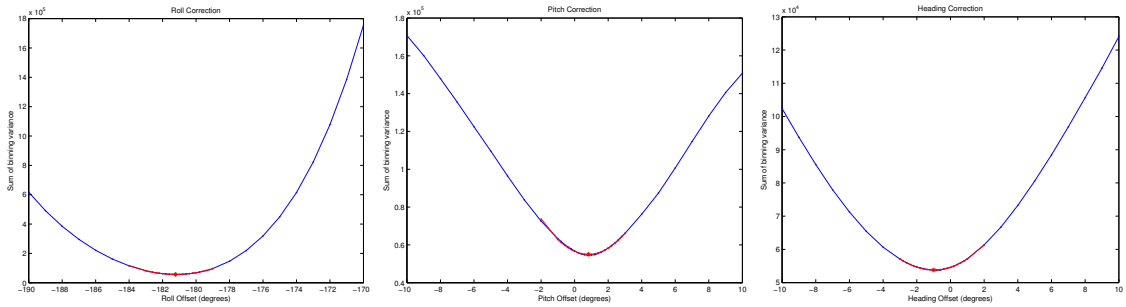

Figure 3-1: The effect of attitude offsets on binned map variances. The vertical axes on the graphs show the sum of variances over each grid cell in the computed map, for the given attitude offset. The left graph shows the error function for roll offset optimization, the middle for pitch offset optimization, and the right for heading offset optimization. The optimization proceeds in order from left to right, and each rotation axis is treated independently given the biases already computed.

mounted far from the DVL, we also compute an estimate for the fore-aft translation offset (the vehicle's x axis) in the same manner, because of the influence of the long lever arm produced when the vehicle pitches.

Once the minima of the functions in figure 3-1 are determined, we perform a quadratic interpolation around the minimum to avoid overfitting. The local quadratic curves are shown in red in the figure, and gridded maps showing the overall improvement after estimating the three degrees of freedom in the attitude offset are shown in figure 3-2.

It is important to remember when considering the estimation of these offsets that the AUV must move through sufficiently general motion over the same patch of terrain, and that the AUV's attitude must be reasonably well known. The estimation is also predicated on the assumption that that offset between the AUV and the multibeam is fixed over the course of the entire dive, or equivalently that any bias in the AUV's attitude estimate is fixed or compensated for over the course of the dive. In some cases, these assumptions are not justified; in figure 1-2 there appear to be biases in the AUV's roll which change from trackline to trackline, and in fact on this particular dive different multibeam roll offsets are produced by the estimator depending on which tracklines are chosen to use in the optimization. More troubling is the fact that if only every other trackline is used to estimate the attitude offset, then the optimization is no longer well-behaved, as the variance can be minimized by moving the multibeam head into a configuration where the grid is reduced to one dimension from two. In this particular exceptional case, the problem can be resolved by bringing in domain

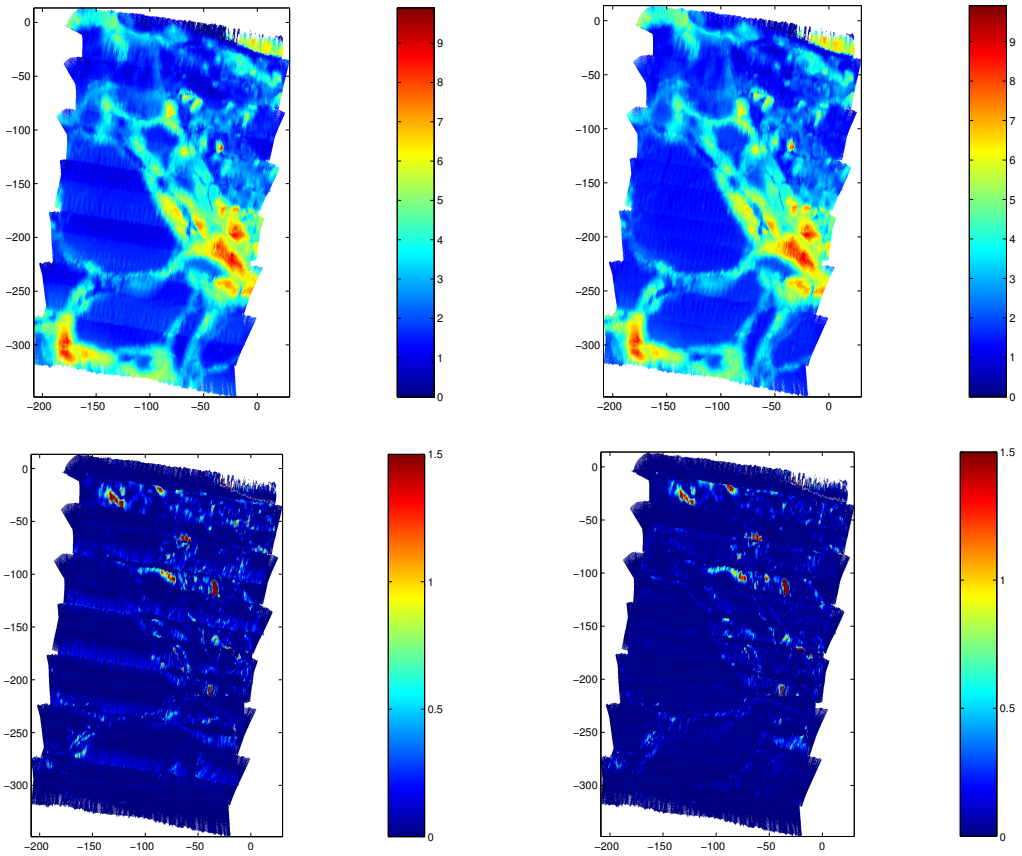

Figure 3-2: Multibeam maps before and after sensor offset calibration. The top row shows the mean depth at each grid cell (units are meters), while the bottom shows the variance (units are meters squared). The left column is before sensor attitude optimization, and the right is after. The largest reduction in variance is produced after the roll offset has been computed.

knowledge (that certain areas of terrain are flat), but it reinforces the conclusion that assumptions made must be justified.

3.3 Constraints induced by multibeam matches

At this point in the mapping process, the reliance of map self-consistency has already been used to improve the map by contributing to the estimation of the offset of the multibeam head. We now return to the pose graph framework, to show how self-consistency can be used to further refine the map by taking into account overlapping tracklines and loop closures. In the smoothing and mapping framework, environmental constraints can be incorporated either through the use of landmarks, or through pose constraints that do not arise from sequential odometry. In early SLAM research, the map was defined as the location of a collection of landmarks, and the corresponding pose graph was similar to that shown in figure 2-2 in the previous chapter, but with potentially large numbers of natural landmarks instead of a handful of localization beacons. Because multibeam sonars can only be used to build maps incrementally, and are of low resolution (relative to laser scanners, for example), distinctive landmarks are difficult to identify and differentiate from one another in single scans. For this reason, multibeam mapping efforts using SLAM such as [51] and [1] do not rely on landmarks, but instead either make use of relative pose constraints (in the former case), or directly use an occupancy grid representation which dispenses with the need for the explicit connection of mapping regions (in the latter case). The approach presented here lies between these two. We build small submaps where the AUV's trajectory crosses over itself and find the transformation that best aligns them, and then determine a relative pose constraint induced by the match. The smoothing and mapping algorithm then treats this constraint like the others in the graph, and optimizes over the whole trajectory. Most notably, the smoothing step does not force the individual submaps to remain static – rather the whole trajectory (and hence the map) is warped to minimize the overall error described by all of the constraints. An example pose graph is shown in figure 3-3.

The submaps themselves are produced using the starting trajectory estimate and multibeam head offset already generated, taking into account the footprint of the sensor on the terrain, as well as the shape of the planned mission. Crossover points are thus planned for, and tracklines are designed to be close enough to each other to allow

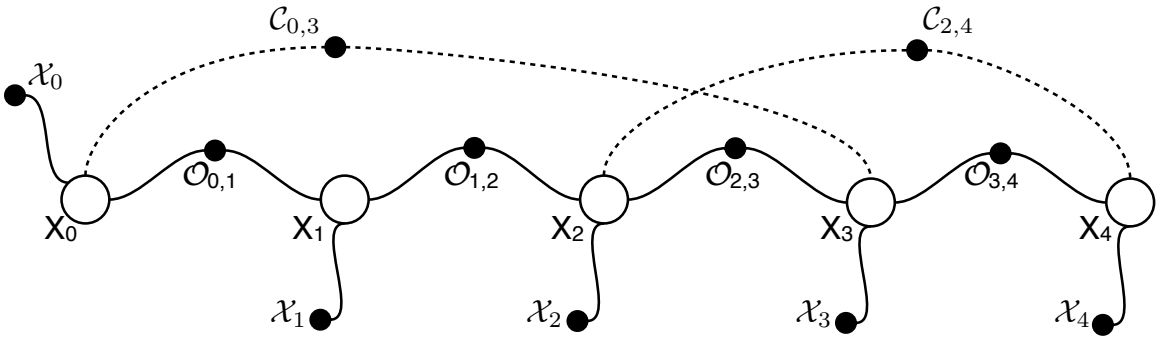

Figure 3-3: An example pose graph showing the types of constraints induced by matching multibeam submaps. Each matched submap induces a single relative pose constraint $\mathcal{C}_{i,j}$, shown here with dashed edges linking the poses to the cross-match factor node.

significant sensor footprint overlap from leg to leg. As will be shown below, both "intersections" in the trajectory and parallel tracklines can be used to produce relative pose constraints. Once two gridded submaps are built, they are aligned by scanning over a horizontal displacement between the two of them, and finding the (x, y) displacement which minimizes the average difference in depths between the two maps, namely

$$\frac{1}{N_{\text{overlap}}} \sum_{(x,y)\in M_1 \wedge (x+\Delta x, y+\Delta y)\in M_2} \left(\bar{z}_{(1,x,y)} - \bar{z}_{(2,x+\Delta x, y+\Delta y)}\right)^2$$

where the sum is over all cells that contain data in both maps, and $\bar{z}_{k,x,y}$ is the mean depth for the cell a location (x, y) in map k. By searching only for a horizontal displacement which aligns the maps, we are assuming that the attitude and depth of the AUV is well-known by this point in the mapping process, which should be the case given the refinement of the trajectory using absolute depth and attitude sensors. An example submap alignment is shown in figure 3-4, which shows gridded maps containing the two submaps both before and after they have been aligned. The corresponding variance images are shown in figure 3-5, from which it is clear that the alignment has improved the local self-consistency of the combined map. The use of depth image alignment differentiates this technique from more general scan matching methods which use a full 6-DOF alignment step, often based on the iterated closest point algorithm [67], which works well for general scan matching, but is better suited for use with point clouds (as opposed to gridded depth maps) produced by dense sensors such as laser scanners. Searching over only two dimensions is simple, and it limits the influence of noise in the

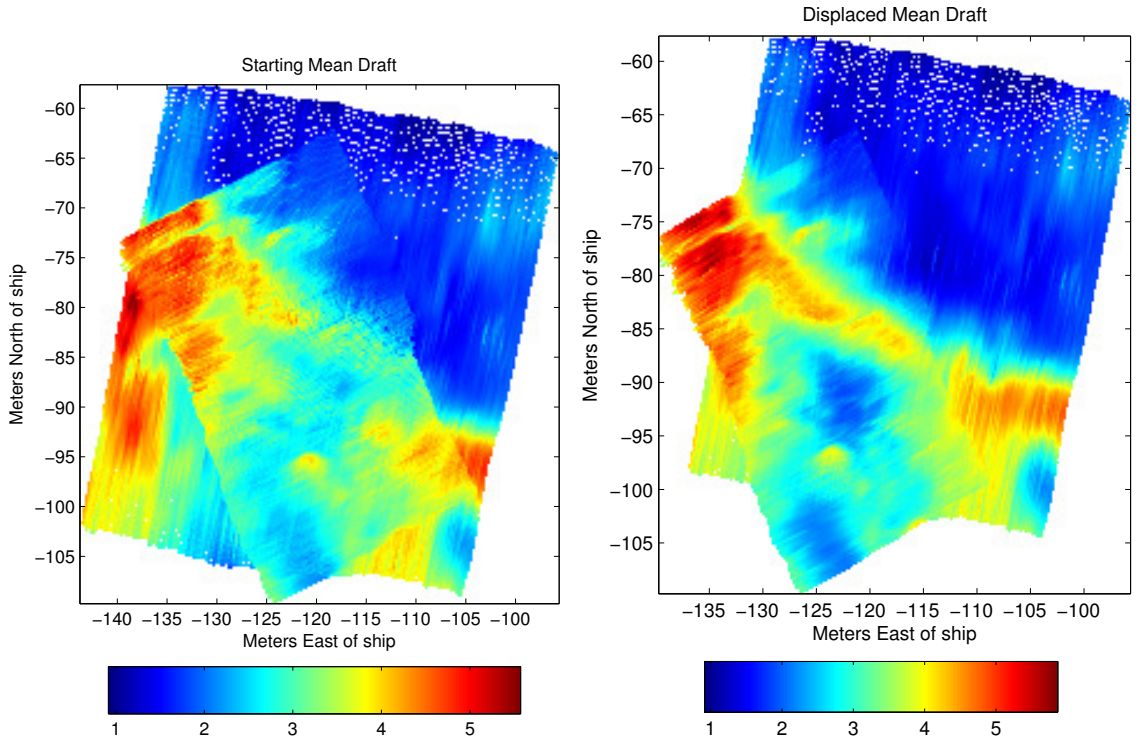

Figure 3-4: Two submaps from different portions of the AUV's trajectory, combined into a single gridded map. These are the mean bin values, representing sea ice draft. The image on the left is before the maps have been shifted to maximize the correlation between them, and the post-shifted map is on the right.

submap binning, which can lead to overfitting when unnecessary degrees of freedom are searched over. Overfitting is also avoided by locally estimating the best match by fitting a 2-D quadratic to the error surface, as discussed in the next section.

3.4 Estimating the information matrix

The smoothing and mapping system requires an information matrix for each factor in the pose graph, which is usually provided by covariance estimates of the velocity sensors on board the AUV. In the case of submap matching, however, the confidence in a match depends entirely on the shapes of the submaps being matched, and the shape of the error surface being minimized. A confidence estimate is provided by fitting a 2-D quadratic locally to the error surface being minimized. The equation in the previous section provides an error term for each (x, y) offset estimate, to which we then fit a

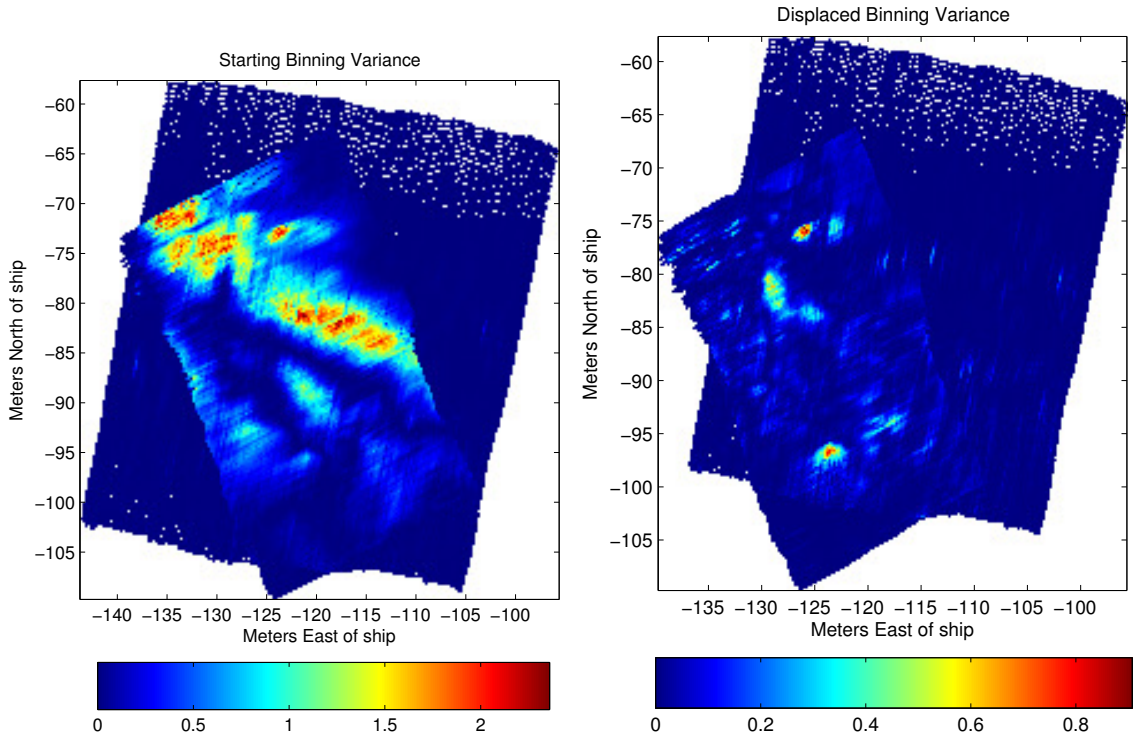

Figure 3-5: The variance within each grid cell for the two maps shown in figure 3-4. Shifting one map relative to the other clearly improves the consistency of the gridding.

quadratic, namely a function of the form

$$e(x, y) = ax^2 + by^2 + cxy + dx + fy + g = \mathbf{p}^\mathsf{T}\mathbf{x}$$

This is done by grouping several error estimates in the neighborhood of the empirical minimum found in the depth error search into a linear system $\mathbf{Ap} = \mathbf{e}$:

$$\begin{bmatrix} \mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \vdots \\ \mathbf{x}_n^\mathsf{T} \end{bmatrix} \mathbf{p} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

and then solving for $\mathbf{p}$ using linear least squares. The extremum of the quadratic is given where the first derivative equations equal zero:

$$\begin{bmatrix} 2a & c \\ c & 2b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = - \begin{bmatrix} d \\ f \end{bmatrix}$$

The shape of the equation around the extremum is governed by the second derivative (Hessian matrix) of the quadratic, which is

$$\mathbf{H} = \begin{bmatrix} 2a & c \\ c & 2b \end{bmatrix}$$

If the analytical minimum from the quadratic is too far from the empirically found minimum or if the determinant of the Hessian is too close to zero (or if it is negative) then the fit is poor, and the match is rejected. Otherwise, the Hessian provides the shape of the local error surface, which translates directly into the information matrix (so $\Lambda = \mathbf{H}$) for the relative pose constraint. Figure 3-6 shows an example error surface, with the empirical minimum shown along with the minimum of the quadratic fit, and the eigenvectors of the Hessian matrix, which give the major and minor axes of the fit paraboloid.

Once the local submap match has been found and the information matrix determined, it must be transformed into a frame of reference appropriate for the pose graph. The pose graph only contains a pose node for each sampling period of the DVL, which is typically running at about 7 Hz and is fixed over the course of a dive. The multi-

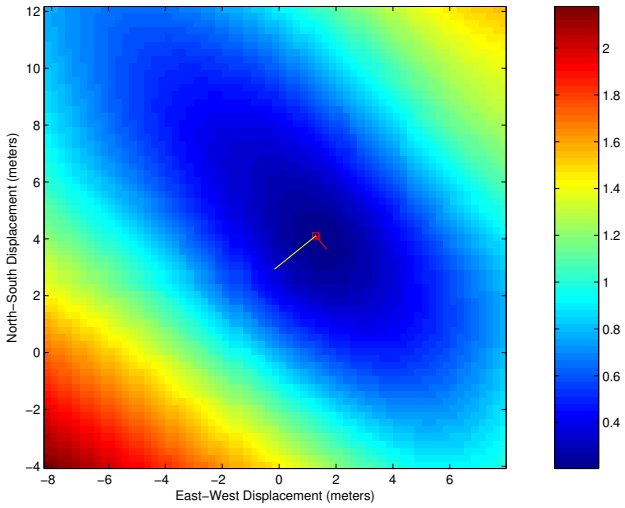

Figure 3-6: The normalized correlation surface. Each pixel in the image corresponds to the average squared difference between the first submap and the second, which has been shifted by the (x, y) coordinates of the pixel. The empirical minimum of the surface is shown by the red $\times$, while the minimum of the fitted quadratic is shown by the red square. The lines from the red square are the scaled eigenvectors of the Hessian of the quadratic.

beam can run at up to 10 Hz, depending on the distance between the head and the terrain being imaged, and according to configuration parameters. A schematic of this relationship is shown in figure 3-7. In the figure, the global (x, y) relationship between the poses s and t, which are not represented in the pose graph, is established by depth image correlation, as described above. The relative constraint between these two poses is $\mathcal{C}_{s,t} = t \ominus s$, and the information matrix Λ must be rotated from the global frame into a local frame relative to pose t. This is done using the local heading of pose s, assuming that the pitch and roll differences between the two frames are minimal. The information matrix rotates as $\Lambda' = \mathbf{R}^\mathsf{T} \Lambda \mathbf{R}$, where $\mathbf{R}$ is the 2×2 rotation matrix corresponding to the (global) heading at pose s.

Once this relative constraint has been determined, a constraint between poses X_i and X_j can be determined by locally interpolating the odometry constraints between nodes X_i and X_{i+1} to determine a local virtual constraint $\mathcal{C}_{i,s} = s \ominus X_i$ (and similarly $\mathcal{C}_{j,t} = t \ominus X_j$). These three local constraints are combined to determine $\mathcal{C}_{i,j}$:

$$\mathcal{C}_{i,j} = \mathcal{C}_{i,s} \oplus \mathcal{C}_{s,t} \oplus \mathcal{C}_{j,t}^{-1}$$

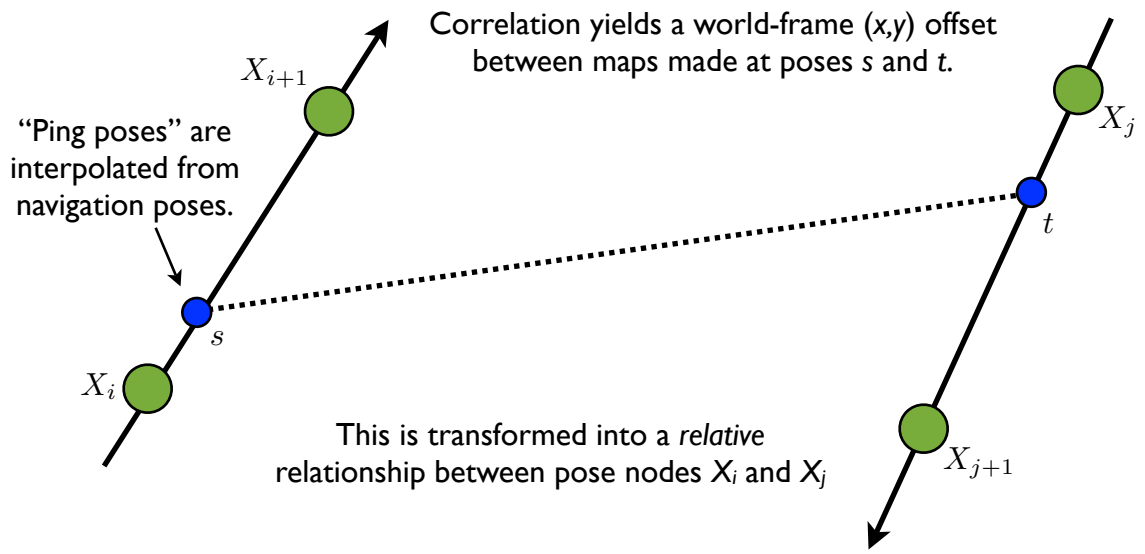

Figure 3-7: The relationship between AUV poses at the time a multibeam ping happens and at the rate of the DVL. Cross-links in the pose graph connect the pose nodes nearest in time to the multibeam ping times.

Once again, the information matrix must be rotated to account for any differences in the headings between nodes X_i and s, though this rotation in general should be quite small, since the time difference between the two poses is much less than one second.

For each matched submap, a single relative pose constraint of the type $\mathcal{C}_{i,j}$ is added to the graph in this way, using the multibeam pings nearest the center of each submap to "anchor" the constraint. The number of factors added to the graph depends on the trajectory of the dive, but as long as the number of these cross-constraints does not increase more than linearly with the size of the graph, the sparsity of the overall system remains intact.

3.5 Results

We applied the AUV navigation and mapping technique described above and in the previous chapter to eight dives undertaken by the Seabed AUV in 2010. Seven of the dives were under sea ice floes near Antarctica, and were attempts to map the shape and draft of the ice above the AUV. In these cases, the multibeam head and the DVL were oriented upward, so that the AUV was navigating relative to ice that in several

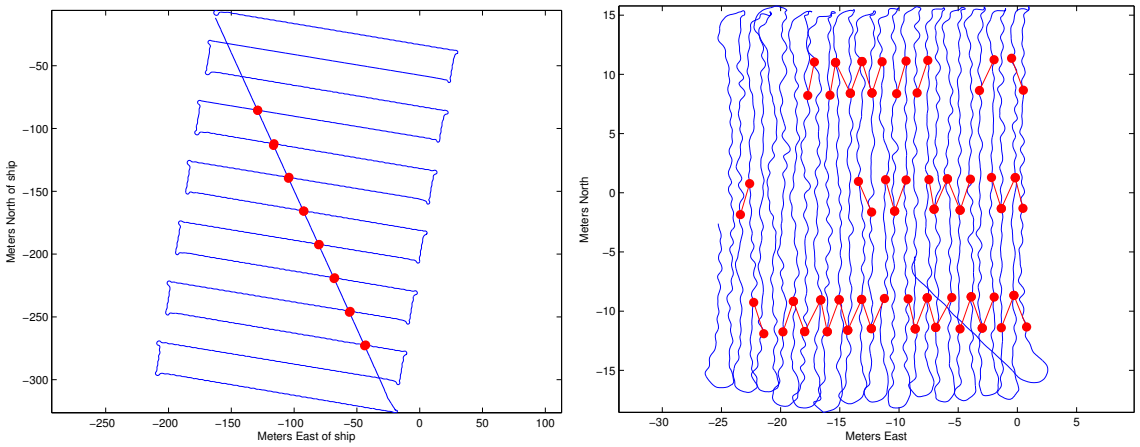

Figure 3-8: Topology of the pose graph for two multibeam mapping dives. The under-ice dive on the left only uses cross links between the diagonal section of the dive and the legs of the lawnmower pattern, while the coral reef dive on the right uses three links per trackline with each adjacent trackline.

cases was drifting, and in one case rotating – we address this additional complication in chapter 5. Under ice, the AUV operated at a constant depth between 15 and 25 meters, and the lawnmower tracklines were spaced to have the same distance apart as the vehicle's depth. The eighth dive took place on a coral reef in Puerto Rico, during which the AUV worked at a constant altitude of three meters, and took pictures every three seconds at the same time that it was acquiring multibeam data. Trajectory graphs with the relative pose constraints induced by matched multibeam submaps for onc under-ice dive (on the left) and the coral reef dive (on the right) are shown figure 3-8. Under ice, each dive included a long diagonal track (sometimes at different depth than the primary grid) which crossed over the lawnmower tracklines, and we used only the intersection points as places to search for submap matches. The coral reef dive did not have a diagonal track. In all cases, there was significant trackline-to-trackline overlap to search for map matches from adjacent lines, but we only did this with the coral reef data, to validate the approach. It is worth noting that the coral reef dive does not use cross-links at every possible point; this is because many potential links are automatically rejected when the Hessian computed from the match indicates a fit which is not concave enough.

The optimized trajectories of the AUV are shown in figure 3-9 for these two dives. In these graphs, the intermediate trajectory marked "attitude corrected" shows the result on the trajectory estimate of optimizing for the roll and pitch bias in the attitude

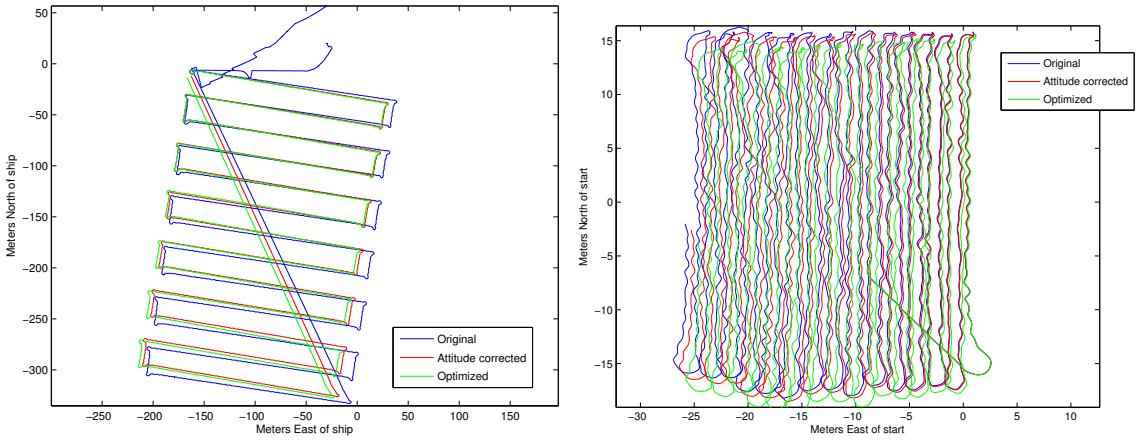

Figure 3-9: The changes induced to the AUV trajectory after adding cross-links to the pose graph, for the two graphs shown in figure 3-8.

sensor, as described in the previous chapter. The graphs show the degree to which the optimization changes the initial trajectory estimate: for the coral reef dive, the change in the final pose after 1034 meters of transit was 1.53 meters, while for the under ice dive, the relative change after 3250 meters of transit was 2.5 meters. In both of these particular cases, and in the six other under-ice dives, the adjustment to the final pose was always less than 0.5% of the distance travelled, which is consistent with the literature ([3], [65]) suggesting that the state of the art in AUV navigation precision (with no feedback from mapping sensors) is 0.5% of the distance travelled.

The error terms induced by the submap links highlighted in figure 3-8 are shown in figure 3-10, again with the under-ice dive on the left and the coral reef dive on the right. For each link, the initial relationship between the poses is shown in blue, and the optimized relationship is shown in red. The links are shown as a small circle at the (x, y) location of one pose, and a line drawn to the (x, y) location of the second pose. The correct location of the second pose relative to the first is drawn with an $\times$, together with an ellipse representing the information matrix induced by the submap matching. The graphs in the second row of the figure show a particular pair of constraints, to illustrate the typical behavior of the smoothing and mapping algorithm: the map is smoothed so that the errors added by the cross-links are minimized together with the errors caused by deviation from the measured odometry.

Overall, the iSAM algorithm is able to minimize the weighted squared error represented in the pose graph because the approach we have taken steers the process suf-

ficiently to avoid local minima or non-convergence. Most importantly, prospective cross-links with low confidence are thrown out altogether, rather than allowed to even weakly contribute to the overall squared error. This is also the case with the noisy acoustic absolute positioning systems examined in the previous chapter: rather than adding very noisy measurements with low information (high variance), we leave the measurements out altogether in the interest of maintaining the stability of the system. For multibeam submap matching, this means rejecting links with Hessian determinants less than 0.001, or with fitted minima that are more than two grid cells away from what was empirically found to be the best match. The resulting pose graphs generally have normalized χ^2 error on the order of 10^{-3}, though this metric is not entirely meaningful as it is not scale-free (i.e. scaling all the information matrices will change the χ^2 error even though the solution remains the same).

Finally, the most direct measurement of the improvement in maps created using this technique comes from the created maps themselves. As mentioned previously, there is no ground truth available, so map self-consistency is the best metric we can use. Figure 3-11 shows the improvements in the variance for binned multibeam maps at several stages of the algorithm. Better maps have lower binning variances, so better maps will have more mass on the left side of the histograms. These graphs show that in addition to minimizing the error in the pose graph (including the consistency of the locally matched multibeam submaps), the algorithm also improves the overall consistency of the multibeam maps, including the majority portions of the dive that were not explicitly matched. Moreover, the large shift in the mass of the multibeam variance histograms indicates an improvement in the effective resolution of the maps being built. While the coral reef data were acquired from a range of only three meters, the fact that most of the binning variance is less than one centimeter is a strong result – gridding the final map to a five centimeter scale is entirely reasonable with this variance. In the next chapter, we will see how this estimate can be further improved through the addition of visual features acquired with a camera.

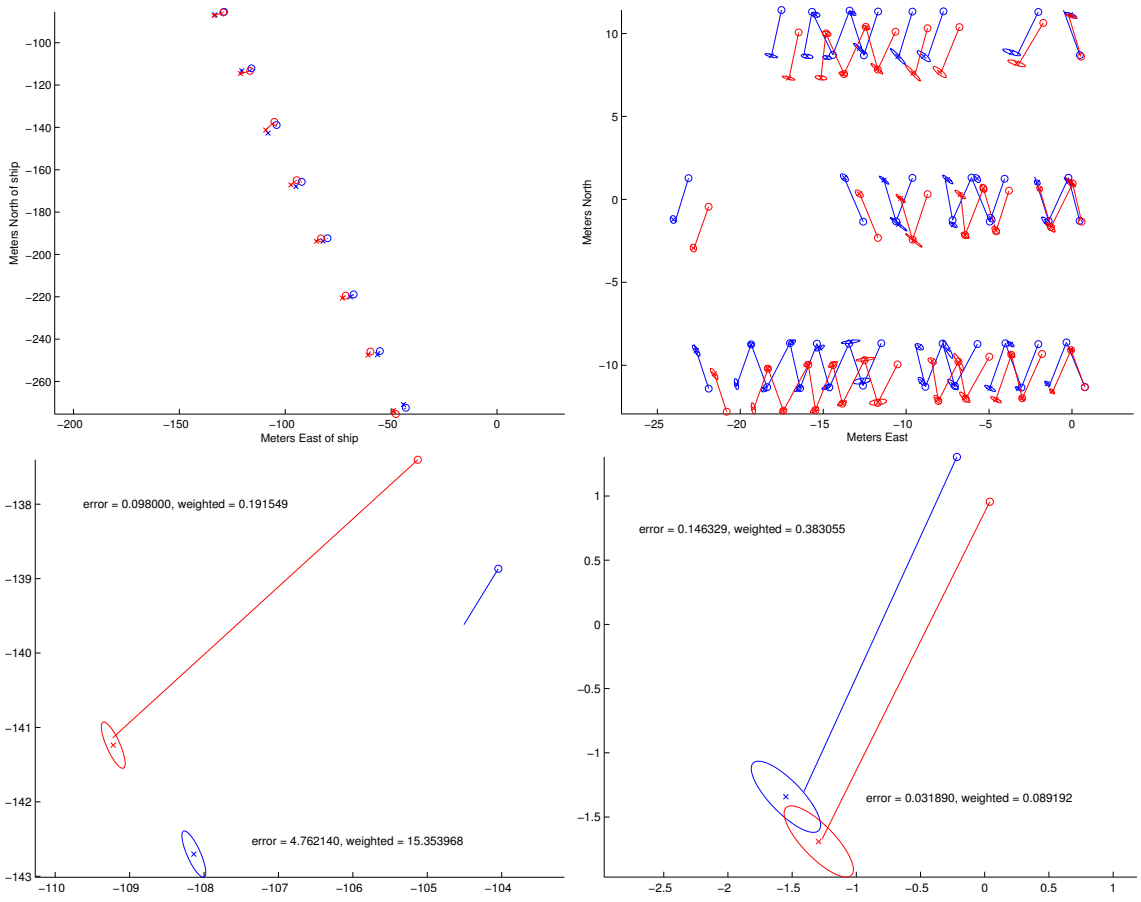

Figure 3-10: The error terms for the cross-link factors shown in the previous figures. In blue are the relative pose relationships before optimization, and in red are the results. The second row shows a zoomed-in constraint from each graph.

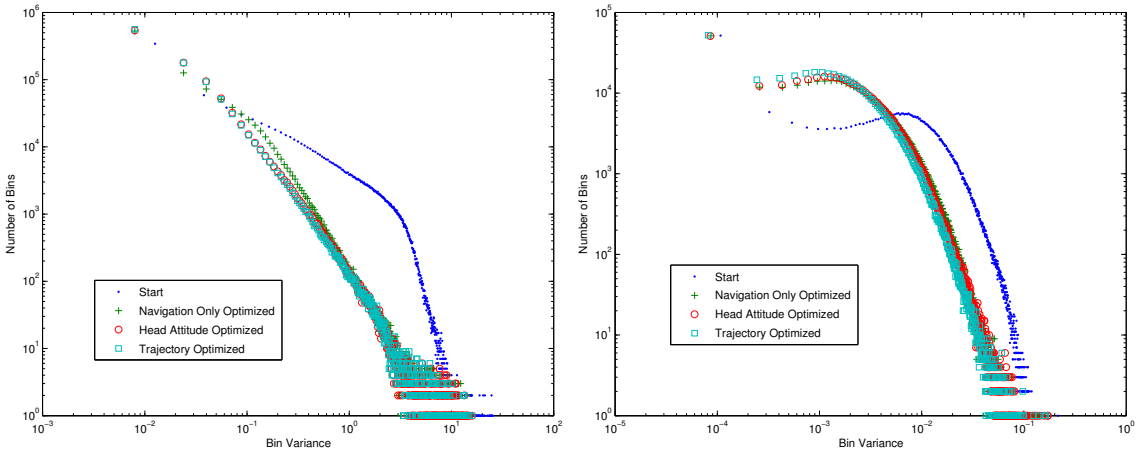

Figure 3-11: Histograms of binning variance at four different steps in the multibeam processing algorithm. The histograms are shown on log-log scales to make them easier to distinguish. Each stage of the algorithm should improve the overall map – as can be seen here, the largest improvement comes from optimizing the roll and pitch offsets of the navigation estimate, though each stage improves the map incrementally.

Chapter 4

Building Visual Maps with Cameras

4.1 Background

Researchers in computer vision, robotics, and photogrammetry have been building maps from camera images for decades. While the approaches have different names (structure from motion, visual SLAM, bundle adjustment) and may start with different information (only visual SLAM assumes odometry information may be available, for example), the underlying mathematics are very similar, and the goals are the same: to recover the camera motion and the 3-D structure of the world from a sequence of images. A good elucidation of the field is given in [17]. This chapter of the thesis quickly reviews the basics, and then delves into the particulars of underwater 3-D scene reconstruction and integration with multibeam data.

The use of cameras for navigation in robotics can be loosely divided into two applications: as odometers, and as a source of landmarks. The former case uses image matches made over small displacements to provide an independent estimate of local robot motion, while the latter case treats images as containing information that will allow the robot to recognize its position at a later point in time. A pair of cameras calibrated together can yield metric 6-DOF relative pose estimates, called stereo visual odometry, a technique widely used in terrestrial robotics [19] [45]. The information can be used in either direction, in fact: stereo visual odometry can inform robot motion, and robot motion can inform camera calibration [35]. Underwater, a DVL generally provides more reliable odometry information over short distances than does a camera, without any calibration requirements [35].

Computer vision has progressed recently to the point that place recognition in some

environments is possible. This means that robots are able to localize after many kilometers of transit, correcting large accumulated drift in odometry and allowing loop closures which enable very large area SLAM. This ability has arisen from the shift in the field from the recognition of individual features to the recognition of feature collections. State of the art implementations use a learned probabilistic model of the kinds of features that are often seen together, and more usefully the unusual scenes that are therefore better suited to identifying particular places [46]. These techniques work particularly well in areas built by people, where unique scenes are more common than in natural environments – underwater place recognition is still an elusive goal.

4.2 Image matching

At the foundation of all of the approaches to using images for navigation must lie a technique for determining when regions of two different images correspond to the same point in the world. Although at video rates dense correspondences can be made using optical flow techniques [20] [18], most modern approaches are based on image features, which are "salient" or uncommon areas that are then encoded to a descriptor vector, whose coefficients are derived from the image content. The most well known of these features is SIFT [37], which selects local maxima in scale space as the salient pixels, and then encodes them using histograms of image gradient orientations around the selected pixel at the found image scale. The key features of effective detectors and descriptors such as SIFT are that they produce high-quality matches, and that they are robust to changes in scale, orientation, and lighting, so that the same point in space can be imaged from different points of view and still be correctly recognized.

The characteristics desired from such detectors of course depend on the underlying application. For underwater mapping, lighting and viewpoint invariance are important, but because the AUV attempts to maintain constant distance from the terrain, we don't expect large changes in scale nor large image changes that cannot be reasonably well-modeled with translation and rotation. Computational complexity is also less of an issue, since the mapping algorithms all take place after the AUV has been recovered from the ocean. With images from the Seabed AUV, each image is preprocessed to account for the illumination pattern of the strobe, features are found using a scaled Harris corner detector, and they are described using coefficients of Zernike polynomials. This combination is effective at finding features in images with low frame-to-frame overlap,

and is robust to image rotation. It is described in detail in [48]. Image preprocessing, and color correction in particular, is an ongoing research problem; see [26] for recent developments, and [61] for a formulation based on adaptive lighting. A typical image pair with matched features is shown in figure 4-1.

Detected features are then matched image to image – each descriptor in one image is compared to all the descriptors in a candidate matching image, using Euclidean distance to compare descriptor vectors (this is not very well grounded, but it is typical practice). Only definitive matches are retained, and outliers in the set of matches between a pair of images are rejected using either RANSAC [13] or least median of squares [39]. For outlier rejection, we use consistency with an image-to-image affine transformation as the model if we expect the scene to be approximately planar, and consistency with a fundamental matrix (i.e. with epipolar geometry) as the model if we expect the scene to have good 3-D structure. While epipolar geometry is valid for any pair of images of the same scene, recovery of the fundamental matrix is only possible if the scene is sufficiently non-planar (see [63]).

4.3 Constraints induced by matched image pairs

We use the common pinhole model of image formation, which captures perspective projection in the matrix equation $\mathbf{p} = \mathbf{KX}$, where $\mathbf{p} = \begin{bmatrix} u & v & w \end{bmatrix}^\mathsf{T}$ is a homogeneous image coordinate, and $\mathbf{X} = \begin{bmatrix} X & Y & Z \end{bmatrix}^\mathsf{T}$ is a point in the world, relative to the coordinate frame defined with the origin at the camera's center of projection, x-axis parallel to the image's horizontal axis, y-axis parallel to the image's vertical axis, and z-axis pointing from the image center out toward the scene. The camera calibration matrix $\mathbf{K}$ contains information about the camera which does not change from image to image: the focal length, pixel aspect ratio, and location of the center of the image. Without knowledge of $\mathbf{K}$ or the camera motion, it is only possible to reconstruct the scene up to a projective ambiguity. Matters are complicated underwater by the fact that light bends as it passes from water into the (air-filled) pressure housing containing the camera, effectively narrowing the camera's field of view and introducing nonlinear distortion – the distortion is reasonably well-captured by typical radial lens distortion models, but see [59] for an analysis of refraction underwater with planar interfaces. For this reason, calibrating the intrinsic properties (the $\mathbf{K}$ matrix and the distortion

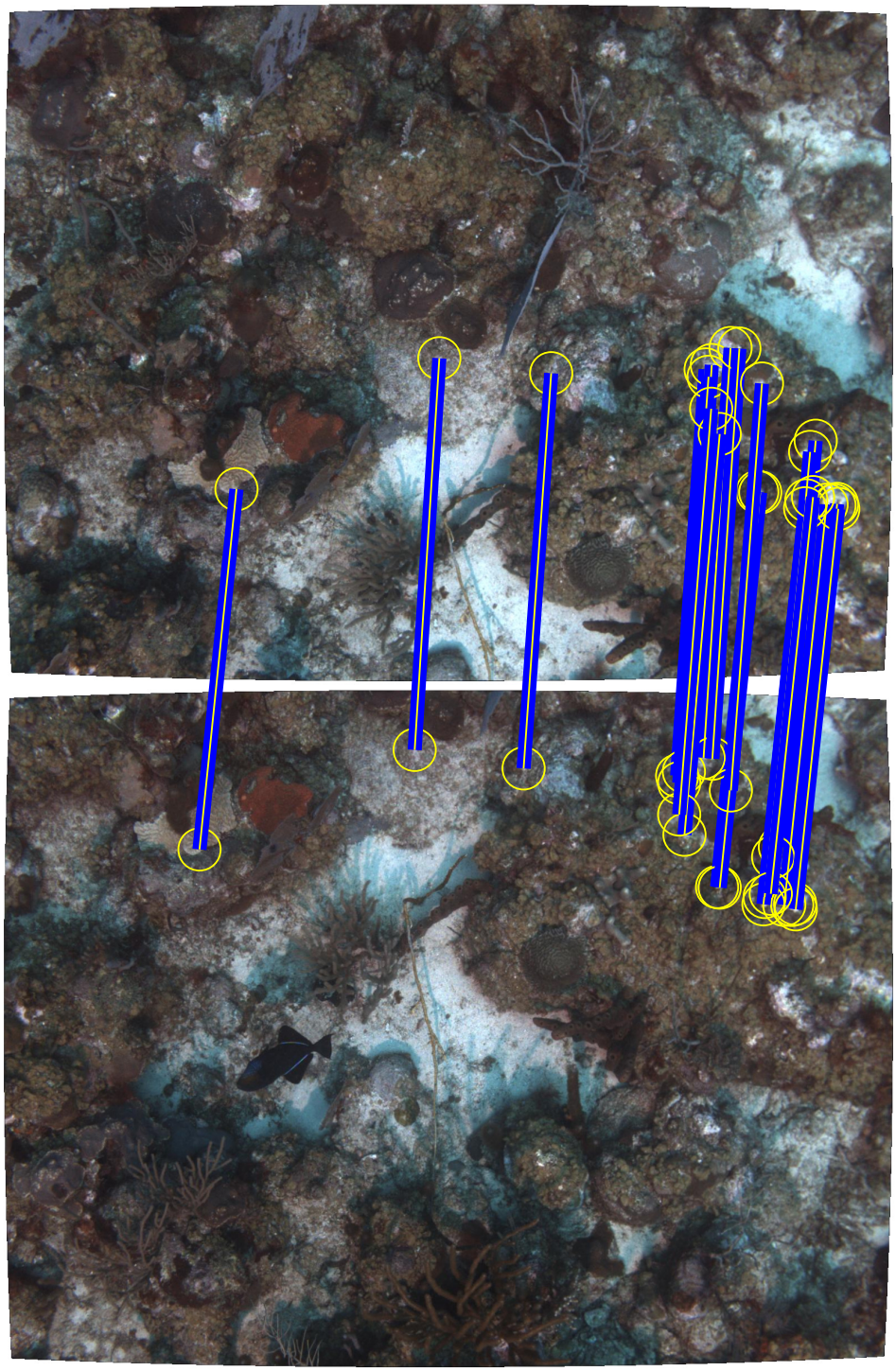

Figure 4-1: Two typical images captured by Seabed over a coral reef, with matched features shown. The images generally have about 50% overlap, so each point on the seafloor is usually seen in only one or two images.

parameters) greatly facilitates mapping.

Given a calibrated camera and a set of matched features over a pair of images, there are two possible ways to constrain the pose graph. The 3×3 calibration matrix $\mathbf{K}$ allows us to use normalized image coordinates, and gives rise to the essential matrix $\mathbf{E}$, which specializes the fundamental matrix to the normalized case. The essential matrix can be estimated from five matched pixels, and it can be decomposed into a 3-D rotation and a translation vector (a fast essential matrix estimator and method for extracting camera rotation and translation direction is given in [44]). This means that even without incorporating any prior estimate of the camera motion from AUV odometry, a pair of image matches in principle can inform five of the six degrees of freedom of the camera motion between two points in time – only the scale of the motion is not recoverable. A 5-DOF relative pose constraint factor can then be introduced into the pose graph, in the same way as a 2-DOF relative pose constraint is used to link multibeam submap matches. The error term for the factor is similar to a 6-DOF odometry factor, but only takes the direction of camera translation into account, ignoring scale. Because the essential matrix captures the motion of the camera, rather than the AUV, an additional pose node captures the 6-DOF extrinsic relationship between the camera's frame and the AUV's frame. This can either be estimated as part of the pose graph, or hard-coded as a constant into the optimization. Each matched image will thus introduce five scalar constraints into the pose graph, and potentially only six additional variables will have to be estimated over the whole trajectory.

An alternative approach, used in this thesis and more common to visual SLAM, is to treat each matched image feature as a proper landmark, with a 3-D location which must be estimated. This formulation adds four scalar constraints to the graph for each matched feature, namely the (x, y) pixel coordinate of the imaged point in each of the two photographs in which it is seen, but also adds three parameters that must be estimated, namely the 3-D location of the imaged point. Each matched feature thus has a net effect of introducing one more constraint into the graph. Since each image pair must have at least five matched features to determine an essential matrix, it makes sense to use this approach rather than the one mentioned above, even though the size of the overall problem will be larger. The resulting pose graph is shown in figure 4-2, which also includes a pose node for the camera's location on the AUV. A prior factor is linked to the camera's extrinsic pose, allowing for an "eyeball" calibration to seed the optimization. We will return to this pose graph momentarily.

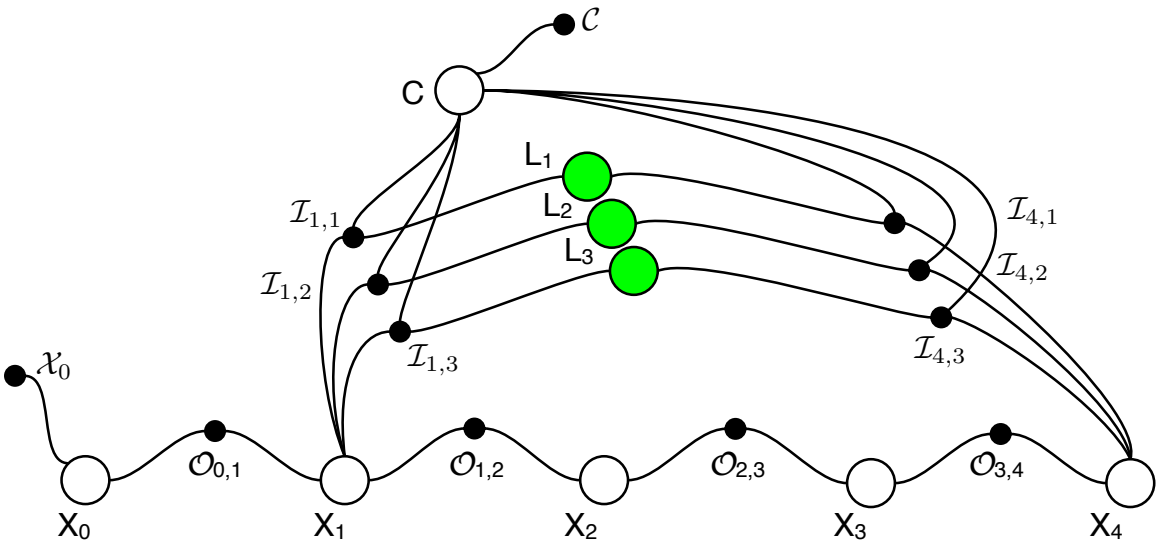

Figure 4-2: The bundle adjustment, or visual SLAM pose graph. Each matched feature is a single 3-D landmark L_s, and the factors $\mathcal{I}_{i,s}$ capture the pixel reprojection error. An additional pose node C captures the 6-DOF relationship between the camera and the DVL, with prior $\mathcal{C}$.

4.4 Photomosaicking

Before proceeding to the full recovery of the AUV motion and the 3-D scene structure, it is worth discussing underwater 2-D photomosaicking. Mosaics are a useful product in and of themselves, because they provide a bird's eye view of a scene which cannot be imaged from far away, similar to an aerial image of the ground. It is often the case that navigation and calibration information is unavailable, especially with legacy data, and it is worthwhile to consider how much can be derived from imagery alone. There is usually not sufficient image-to-image overlap underwater to apply full structure from motion techniques, as many scene points may only be imaged once. Finally, it is reasonable to assume that a good mosaic can only be improved by the addition of navigation data, so a robust mosaicking technique is a good starting point for full 3-D reconstruction.

Starting from the assumption that images are available in the order in which they were acquired, a sequence of linear chains can be built by matching images from consecutive time steps. Features are extracted and described as above, and matches are made along the chain. If a pair of images cannot be matched, the chain is broken and a new

chain started, dividing the graph of image links into connected components. An initial mosaic is built from each connected component by assigning an affine transformation to each image which minimizes the pixel matching error (i.e. each matched feature should be warped to the same pixel in the global mosaic pixel space). Affine transformations are used for image warping rather than full projective warps because it keeps the optimization semi-linear, and because the camera tends to point in approximately the same direction throughout the dive, removing the need for the more oblique views provided by projective transformations.

Using affine warps reduces the mosaicking problem to finding the solution to a homogeneous system (of the form $\mathbf{Ap} = 0$). In order to prevent excessive warping to make features match and to yield solutions which are nontrivial, nonlinear terms are introduced which constrain the degree to which distortion is allowed. As described in [48], these quadratic terms attempt to restrict the affine degrees of freedom of the transformations, and are empirically weighted against the feature matching terms to produce a balanced result. Misalignments are inevitable because of the 3-D nature of the scene, so the final mosaic is rendered using the ubiquitous multiresolution spline technique [4]. Once an initial mosaic is computed, hypotheses about cross links are determined from overlapping images, more images are matched, and the algorithm repeats. After several iterations, a graph with many loops and track-to-track links describes the topology; for the coral reef data the topology is shown in figure 4-3 and a rendering of the corresponding mosaic is shown in figure 4-4.

The photomosaicking problem is quite similar to the pose graph optimization problem: each image corresponds to a pose node, with six degrees of freedom corresponding to the six parameters of an affine transformation. Factors in the graph join nodes which represent matched images, and each image also attaches to a factor restricting the extent of the allowed image warp. The graph is sparse, as it is in the smoothing and mapping problem. The overall error function is largely linear, however, and the nonlinear parts are quadratic, yielding a Jacobian that can be readily computed in closed form. Most importantly, the sparsity of the system means that the size of the problem grows linearly with the number of links in the graph, mirroring the complexity of the AUV mapping problem.

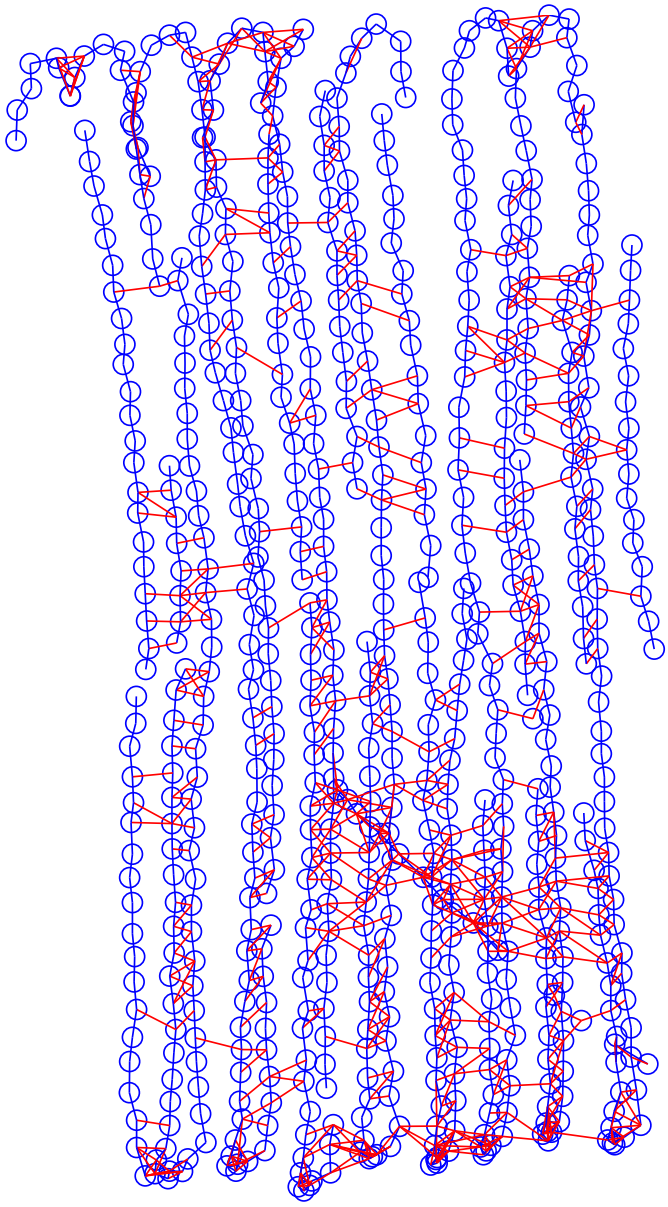

Figure 4-3: Link topology of about half of the coral reef dive. This shows the connectivity of one connected component of the image link graph. Each blue circle represents one image. Blue lines connect matched images that were captured sequentially in time, while red lines connect matched images that were not captured sequentially. Of the 1476 images acquired during the dive, these 744 images were matched automatically into the largest connected component. The location of each image in the graph is estimated from the photomosaic rendering, not from the AUV trajectory as recovered by the pose graph optimization.

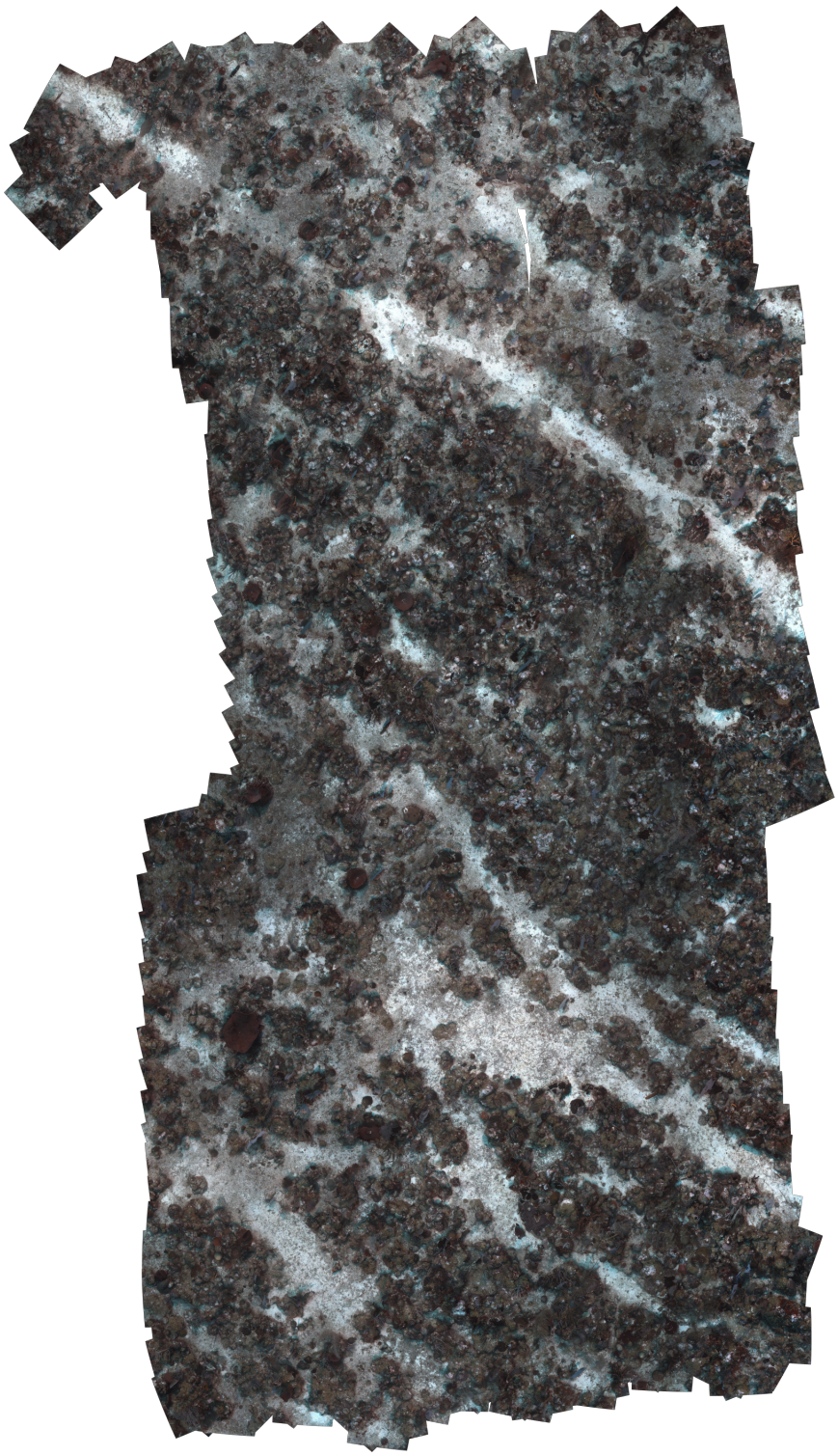

Figure 4-4: The largest mosaic from the coral reef data set, corresponding to the link topology shown in the previous figure.

4.5 Bundle adjustment

Moving from 2-D photomosaicking to 3-D reconstruction, the link topology determined in the mosaicking process is added into the pose graph, with each matched feature contributing one landmark with two observations, as shown above in figure 4-2. Each visual observation landmark in the graph is linked to two factors, one for each observation. Each observation factor is also linked to the pose node X_i corresponding to the AUV's pose at the time the image was acquired, and the node representing the camera's extrinsic location relative to the AUV. Using the equivalent 4×4 matrix $\mathbf{X}_i$ to represent the AUV's pose at time i, and $\mathbf{C}$ to represent the the camera's pose in the AUV's frame, and the 3×1 vector $\mathbf{L}_s$ for a landmark imaged at homogeneous pixel location $\mathbf{p}_{i,s} = \begin{bmatrix} u_s & v_s & 1 \end{bmatrix}^\mathsf{T}$, then the error in the observation factor $\mathcal{I}_{i,s}$ is

$$\mathbf{X}_i\mathbf{C}\mathbf{L}_s - \mathbf{K}^{-1}\mathbf{p}_{i,s}$$

where both vectors in the subtraction have been "regularized" so that their third coordinate is equal to one – the error function measures actual pixel reprojection error, in normalized coordinates. Note that all three factors in the first term of the equation are being solved for in the pose graph optimization. For the 2×2 information matrix, we experimentally chose a matrix with one-fifth the calibrated focal length on the diagonal, corresponding to a variance of five pixels. This means the observed visual features are not individually weighed strongly enough to overwhelm the influence of the AUV odometry, but that as a group the visual features are ultimately able to "pull" tracklines together and close loops. A detail of the mosaic shown in figure 4-4 is shown in figure 4-5 – this figure shows part of the starting conditions for the bundle adjustment pose graph. The bundle adjustment will significantly reduce the reprojection error in the figure by incorporating the vehicle navigation information and the 3-D structure of the scene, as shown below in figure 4-6.

It is important to note that all of the images captured in the dive that have been matched with at least one other image can be used in the pose graph – in other words, the mosaicking process can end with several independent connected components, and each of these connected components containing at least two images can be added to the pose graph. The components must have two images because they must contain matched features (landmarks). Without navigation information provided by the solu-

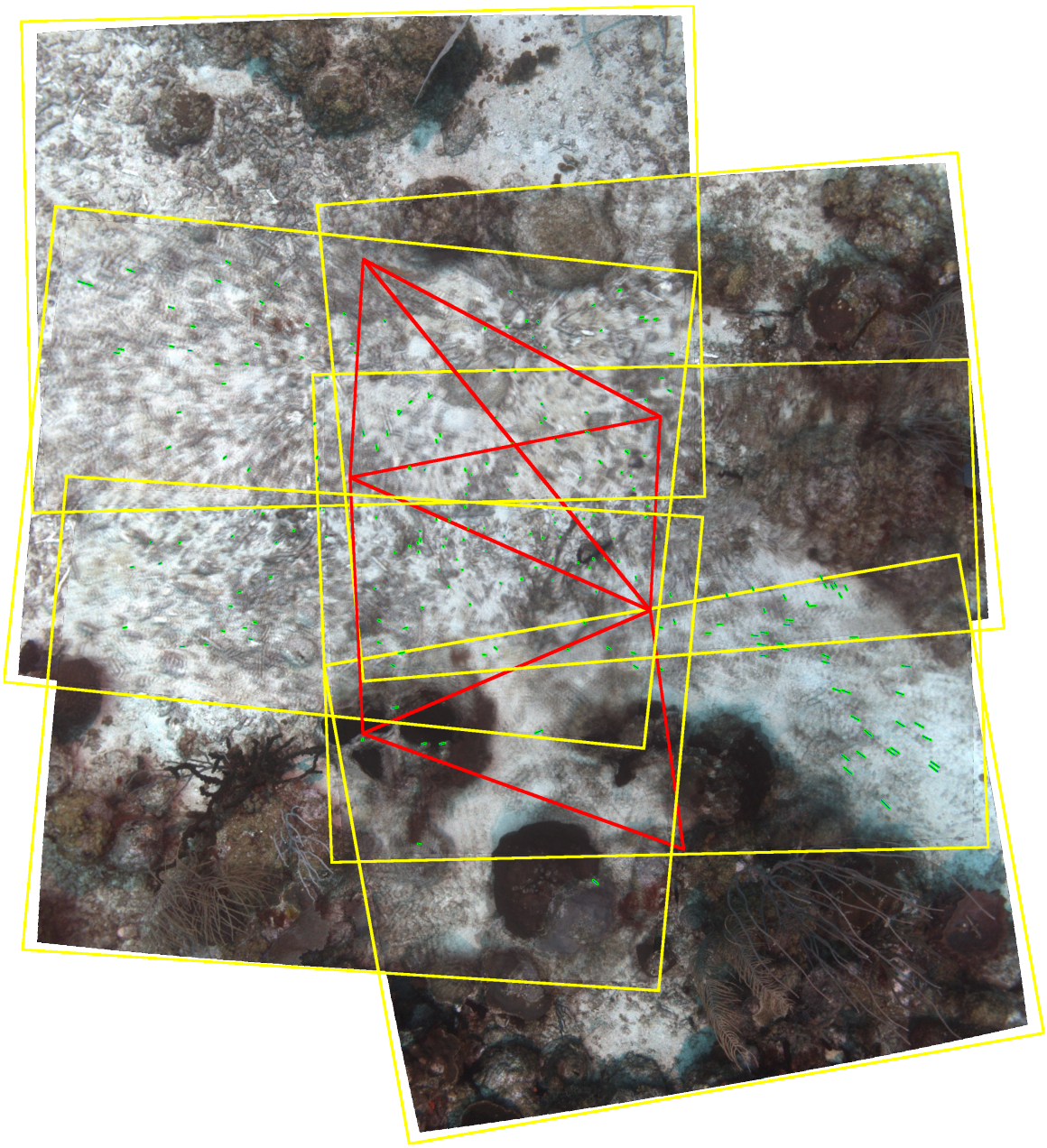

Figure 4-5: A detail of the mosaic in figure 4-4. This shows six images blended by simple averaging. The image boundaries are boxed in yellow, and images with enough matched features to infer an affine transformation linking them are connected by a red line joining their centers. Individual matched features are connected with short green lines; the length of these lines is equal to the mismatching error in the set of affine transformations that defines the mosaic – many are short and thus difficult to see at this scale. These matches are then used as input to the bundle adjustment pose graph, which recovers 3-D structure that reduces this reprojection error.

tion to the pose graph optimization, the complete dive cannot be visualized since each mosaic is independent. We will show below that the optimized trajectory provides enough information to allow the disconnected mosaics to be visualized as a single map.

Each pose node in the graph must be initialized for the optimization. The AUV pose nodes are initialized using odometry, and the extrinsic camera location is initialized using the "eyeball" calibration or nominal mounting parameters – typically the camera is mounted about 1.4 meters forward of the DVL, and is rotated 90 degrees, so that the camera x axis points to vehicle starboard, and the camera y axis points aft. The landmark locations are initialized by triangulation from the AUV pose and camera location estimates, which is a rather noisy guess at the 3-D scene structure, particularly when the AUV has moved a long distance between acquiring matched images. Since it is unlikely that a perfect triangulation will exist for a given feature, the location of a landmark is initialized to the 3-D point closest to the two camera rays under the starting navigation estimate. No weight is given to this initial estimate (there is no prior factor on the landmark locations), but in order to reduce false matches landmarks with ranges differing by more than 20% of the vehicle's altitude as estimated by the DVL are not used.

Even with landmark pruning, there are still more than enough features to improve the map; for the coral reef data set, 19315 features of 35593 were rejected, and 1501 matched image pairs were used involving 1072 images of 1476 total images acquired during the course of the dive. Figure 4-6 shows the estimate of the AUV's trajectory before and after the bundle adjustment step, and the pixel reprojection error in actual (non-normalized) pixels using the start estimates for the camera pose and AUV trajectory, and the improvement in this error after the optimization.

The incorporation of matched visual features into the pose graph not only gives us an improved estimate of the AUV's trajectory and a sparse set of 3-D scene points, but it also provides an improvement on the 2-D photomosaic, by providing a means to link disconnected subgraphs in the link topology together. Without navigation data, any missing link in the starting chain of image matches irrevocably separates a potential global view into disassociated components – rather than assuming the AUV's navigation estimate and the extrinsic camera calibration are perfect and attempting to render a mosaic from pieces, bringing both the image features and the navigation information together in a single framework allows for an improvement of both.

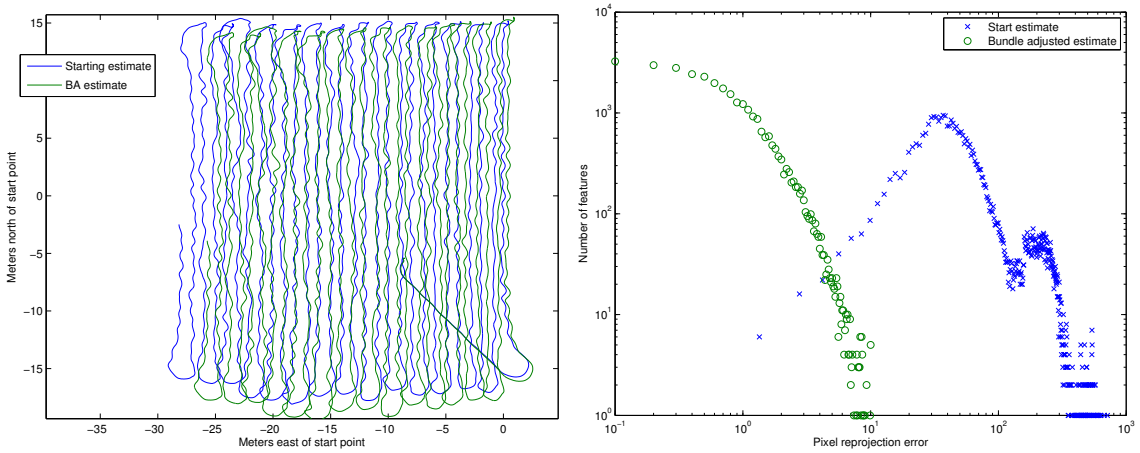

Figure 4-6: The change in the estimated AUV trajectory using only visual landmarks in the pose graph. This is the result of pose-graph visual SLAM, or bundle adjustment with odometry. On the left is the estimated trajectory, and on the right are histograms of the pixel reprojection error before and after the optimization. The DVL roll-pitch optimization described in chapter 2 is applied before bundle adjustment.

4.6 Fusing multibeam and vision

Up to this point, we have described how to incorporate information from two distinct mapping sensors into the estimate of the AUV's trajectory, improving the consistency of the generated maps. The two mapping sensors complement each other well, as each provides good feedback under different environmental circumstances. It makes sense, then, to use information from both sensors when they are available. It should come as no surprise that doing so requires augmenting the pose graph to incorporate both types of mapping constraints. Such a graph is shown in figure 4-7.

This pose graph is built incrementally – before performing the full optimization several intermediate optimizations are undertaken to ensure that the full problem has a good initialization point. As always, we start by computing an estimate of the roll and pitch bias of the DVL. The images must be preprocessed and matched into a set of 2-D photomosaics, as described above. We seed the estimate of the camera's extrinsic location relative to the AUV using a subset of all the matched image features, and then perform the full bundle adjustment. Finally, the multibeam data are incorporated into the graph though the submap matching process described in the previous chapter. The complete pose graph has all of the nodes and factors used in the vision-only graph, plus the relative pose constraints shown as dashed lines in figure 4-7. It is important to note

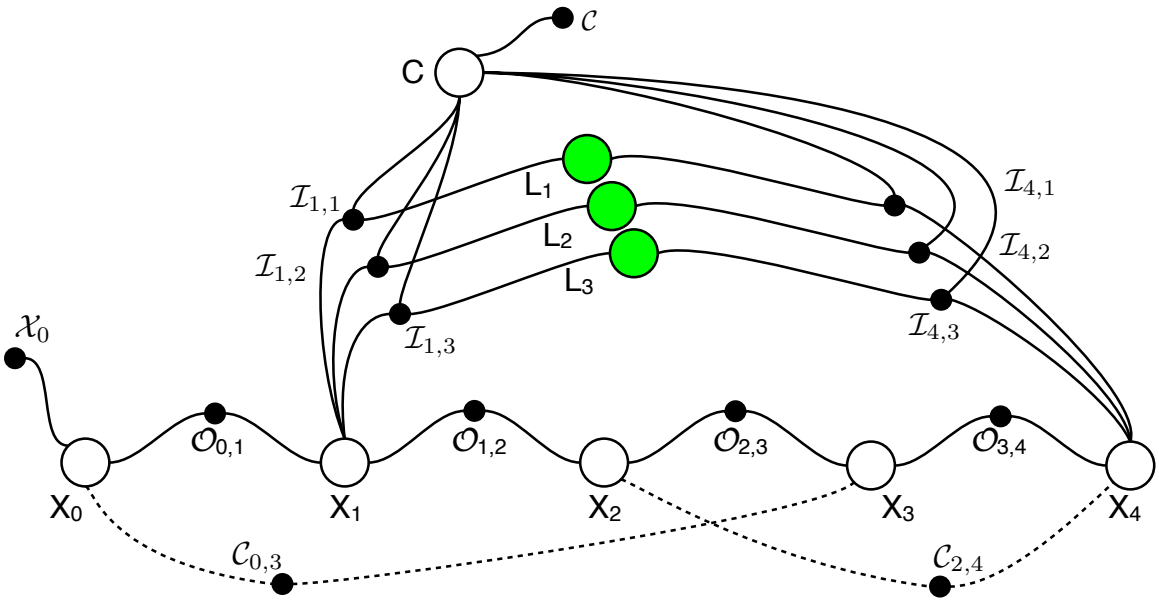

Figure 4-7: Adding multibeam constraints to the bundle adjustment pose graph allows both types of mapping sensors to be incorporated into the problem at the same time. Although they are present in the pose graph, this figure does not show factors representing depth and attitude measurements to reduce clutter.

that while there are several steps taken to seed the initial estimates used for the scene structure and the camera offset, the final optimization incorporates all the constraints available: odometry, depth, and attitude constraints from the AUV's proprioceptive sensors, landmark measurements from visual features, and relative pose constraints from multibeam submap matches. Because each image may contain dozens of features, there are generally many more constraints on the map and trajectory provided by the camera than by the multibeam sensor. The final optimization estimates six values for each AUV pose, three values for each matched feature, and six values for the camera location in the AUV frame. If there are n poses being estimated, k matched visual feature pairs, and l multibeam submap constraints, then the graph will have $n + k + 1$ pose nodes, and about $2n + 2k + l$ factors, depending on how often absolute depth and attitude measurements are taken.

Results

At each step in the process, the influence of the additional information on the quality of the map must be considered. We can consider each sensor modality independently.

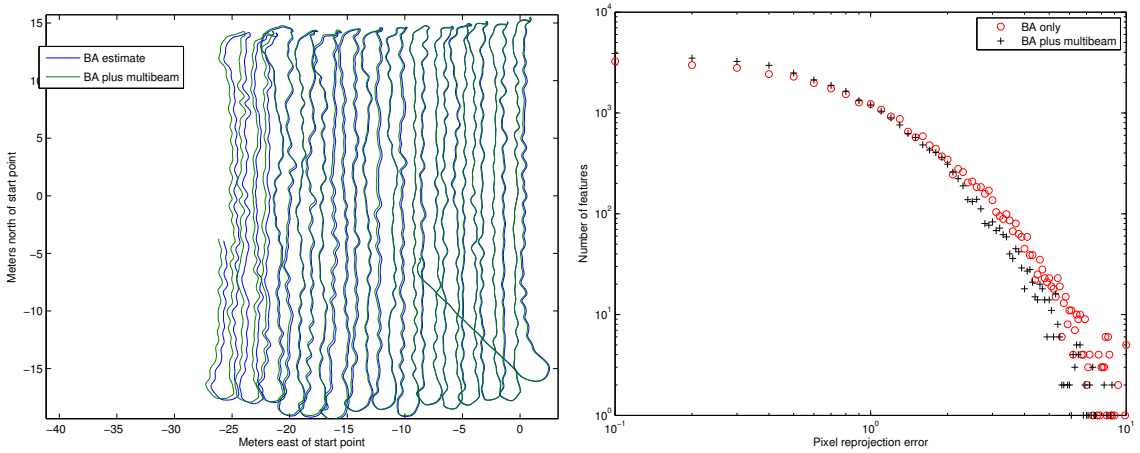

Figure 4-8: After adding multibeam constraints to the bundle adjustment problem, the AUV's trajectory is further refined to the final estimate shown on the left. On the right the pixel reprojection errors are shown, compared to the results in figure 4-6.

Consistency in the multibeam map is measured by the variance in the binning step and reduction in the weighted error of cross links in the pose graph. At the same time, pixel reprojection error is used to check consistency in the use of visual features in the map. The rest of the discussion focuses on the coral reef dive directly, in which 50 multibeam pose constraints were combined with 17023 visual landmarks over 26010 AUV poses. The change in trajectory after incorporating multibeam data into the bundle adjustment result is shown in figure 4-8, together with the change in pixel reprojection error. The minimal change in reprojection error after the multibeam data are incorporated shows that using both mapping sensors does not have a detrimental effect on the quality of the bundle adjustment solution. Unsurprisingly, the portion of the dive with the most image landmarks (the right side of the trajectory shown in figure 4-8) is not altered as much by the addition of multibeam data as is the later portion of the dive, where there were fewer constraints.

The change in multibeam map consistency is shown in figures 4-9 and 4-10. The first of these shows the reduction in the weighted error in the pose graph optimization: this is the error that the smoothing and mapping algorithm is directly trying to minimize, so one would expect an improvement in all cases. The graph on the left shows the starting and ending pose graph error for the multibeam constraints when no visual landmarks are used, while the graph on the right shows the same errors when both multibeam constraints and visual landmarks are used together. The results for this data set are

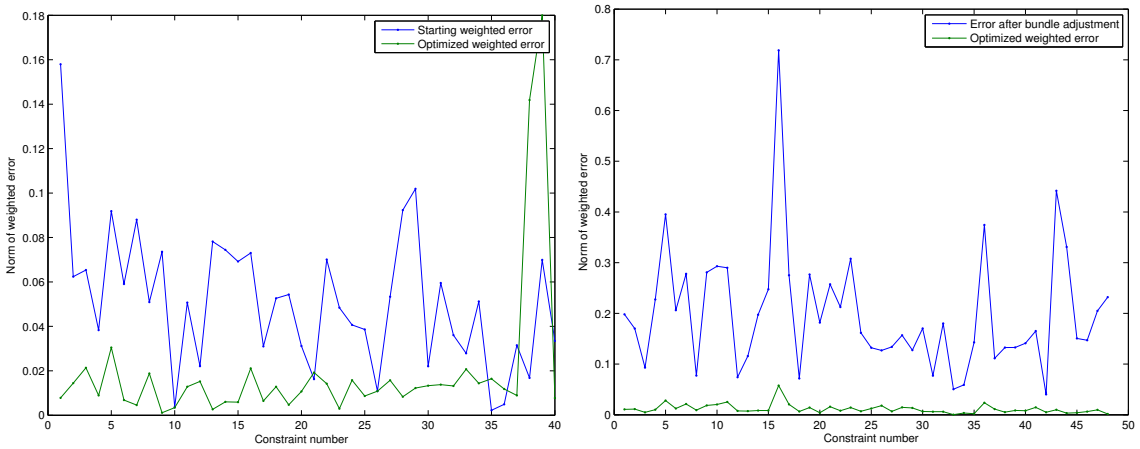

Figure 4-9: The reduction in error in multibeam submap matching (relative pose error). The left graph shows the result of using only multibeam constraints, while the graph on the right shows the starting error from bundle adjusted navigation, and the final result after using the full pose graph from figure 4-7. The final error on the right is comparable to the final error on the left, and the final sum of squared errors on the right is lower than on the left.

somewhat unexpected: initializing the AUV's trajectory using bundle adjustment yields multibeam pose constraints that have weighted error an order of magnitude higher than the case where no visual landmarks are used. The result of the combined optimization, however, is that the relative pose constraints have a lower weighted error than the final result when bundle adjustment was not used. The visual features, in other words, make the AUV's trajectory less consistent for multibeam to begin with, but afford a better overall solution once the optimization is performed.

The changes in binning variance for the built multibeam maps are shown in figure 4-10. These histograms, like those shown in the previous chapter, show the change in consistency of the maps as an effect of the pose graph optimization – the algorithm does not directly attempt to reduce binning variance over the whole map, but only as a side effect of the added pose constraints induced by submap matching. The graph on the left shows that adding visual features to the multibeam mapping process described in the previous chapter results in a multibeam map that has about the same consistency as the best result from multibeam mapping without the addition of visual data. The final binned bathymetry map is shown in figure 4-11.

So if adding multibeam data to the bundle adjusted estimate only marginally improves the vision-only result, and adding visual data to the multibeam estimate only

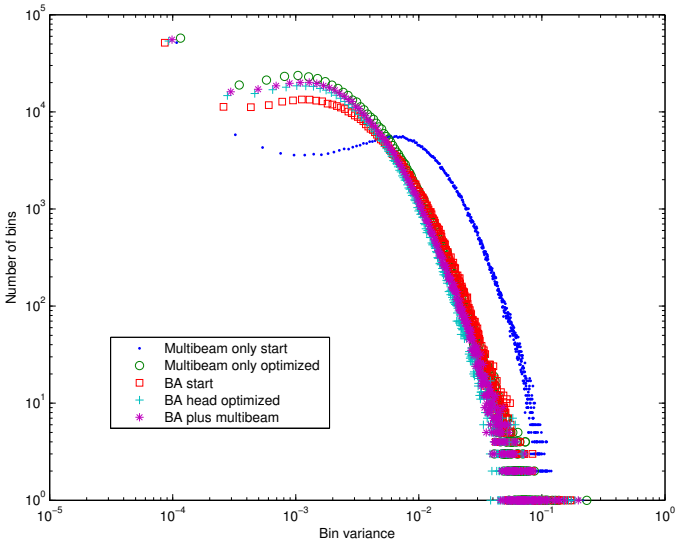

Figure 4-10: Histograms of multibeam binning variances. This graph can be compared to figure 3-11, but the overall variances are much lower because the AUV was operating at close range to the sea floor, and smaller bin sizes were used (5 centimeters as opposed to 25 centimeters).

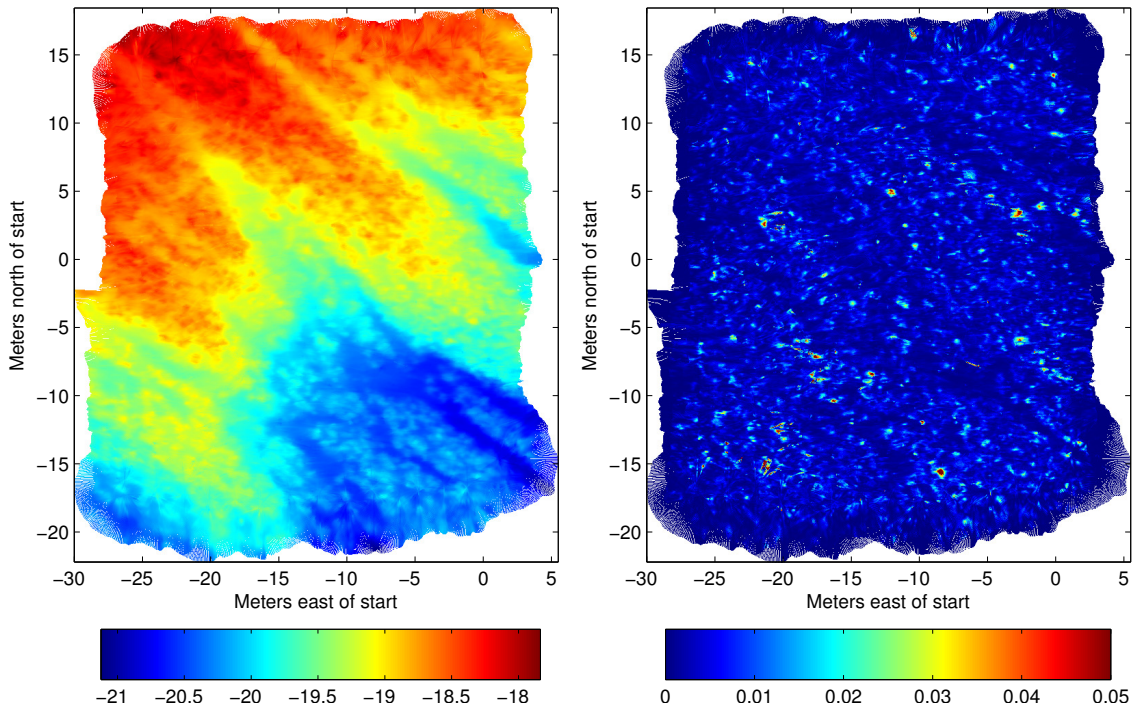

Figure 4-11: The bathymetry map incorporating both multibeam cross links and visual landmarks. On the left is the mean depth in each bin in meters, and on the right is the variance in each bin. Each grid cell is five centimeters on a side.

marginally improves the multibeam-only result, why use both sensors? There are two answers: first of all, using both sensors can be a big help in environments where the data are not as rich as on a coral reef, where there is a large amount of visual texture and 3-D topography (i.e. both sensors perform well in this situation). Secondly, even in situations like in Puerto Rico, fusing the two sensors together gives us a new way in which to visualize the environment: the imagery can be texture-mapped onto the multibeam bathymetry, because we have the 6-DOF AUV pose at the time of every ping and image capture, as well as the 6-DOF offsets to both mapping sensors.

Visualization

While visualization is not the focus of this thesis, a reasonable (if not scalable) visualization can be built as follows: for each image, the ray from the camera center is determined from the AUV's pose, the camera calibration, and the camera's location in the AUV's frame. Then, for each grid cell in the bathymetric map, the image with camera ray passing closest to the grid's center is chosen as the best source of texture information for the grid. The cell center's 3-D point is projected into the camera, and the imaged pixel is chosen as the texture coordinate for that point in space. One can either use a full photomosaic for the texture image, or the collection of images individually. Regardless, there will be seams in the visualization that result from the coarse application of a fine texture to a lower resolution mesh. Still, the visualization is enough to qualitatively verify that the map is a good one, as changes in bathymetry can be seen to correspond to changes in visual texture, as can be seen (with difficulty on paper, unfortunately) in figure 4-12. When the mosaicking process results in several disconnected sets of image links, a single mosaic can be quickly created by using the scene structure recovered from the pose graph optimization to find affine image transformations mapping pixel locations to landmark locations with the z coordinate removed, effectively yielding an ortho-rectified rendering of the scene. Even though finding a set of transformations this way does not directly use the image-to-image matching constraints found in the initial mosaicking process, the resulting rendering is still good enough that it can be used as the texture map for the multibeam bathymetry, or as a useful visualization product in its own right, as shown in figure 4-13. Improving visualization is a distinct thread of research, which has been taken up in earnest elsewhere in the literature – see [25] for a good solution based on stereo imagery, for example.

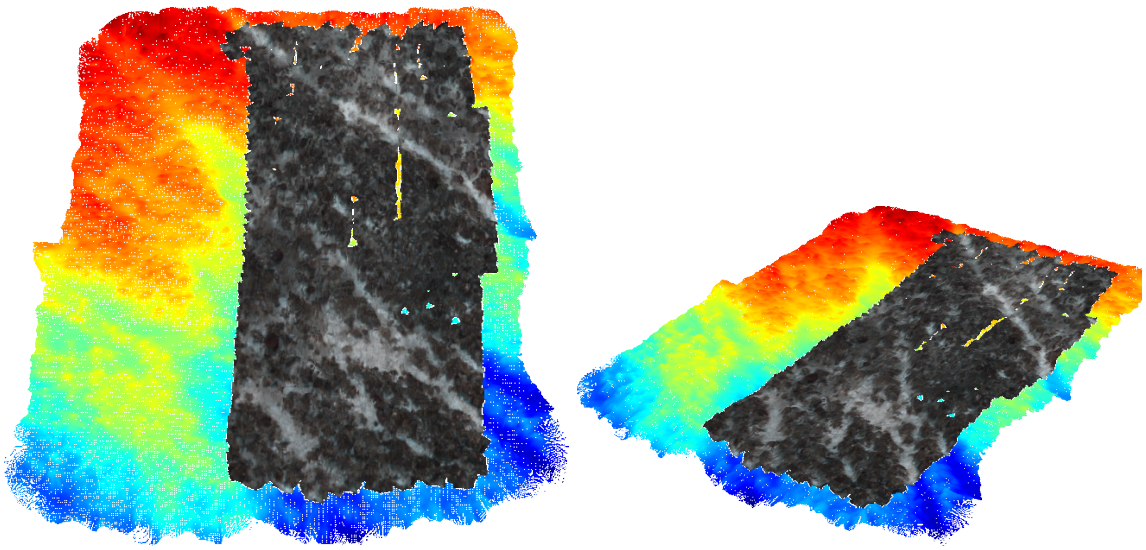

Figure 4-12: Rendering of the multibeam bathymetry, partially texture-mapped using the photomosaic from figure 4-4. Color on the non-texture-mapped portion of the bathymetry corresponds to depth, with red corresponding to 17.75 meters and blue to 21 meters.

Comparing 3-D structure from vision and multibeam

We have focused to this point on the consistency of the sensors individually. It is also worth asking about the consistency in the two different 3-D estimations of scene structure that are built by the mapping sensors. The multibeam gives us relatively dense bathymetry directly, and the bundle adjustment produces a sparser set of 3-D points. How well to they compare? Figure 4-14 shows an outline of two histograms of the vertical distance in meters between visual landmarks and the corresponding grid cell in the bathymetry. Both histograms clearly show a rather wide unimodal distribution – this width is explained by the vastly different resolutions of the mapping sensors, as each pixel images a region in space that is much smaller than the 25 square centimeters captured by a single multibeam grid cell. The two distributions are shifted relative to each other: the blue line, representing the result of the initial combined optimization, shows a bias of about 0.2 meters between the bathymetry as measured by the two sensors. The red line in the figure shows the same comparison after changing the initial camera location in the optimization by moving it vertically by this bias. At first glance, this change might not be surprising: the camera has been moved by a distance approximately equal to the discrepancy between the ranges measured by the two sensors, and

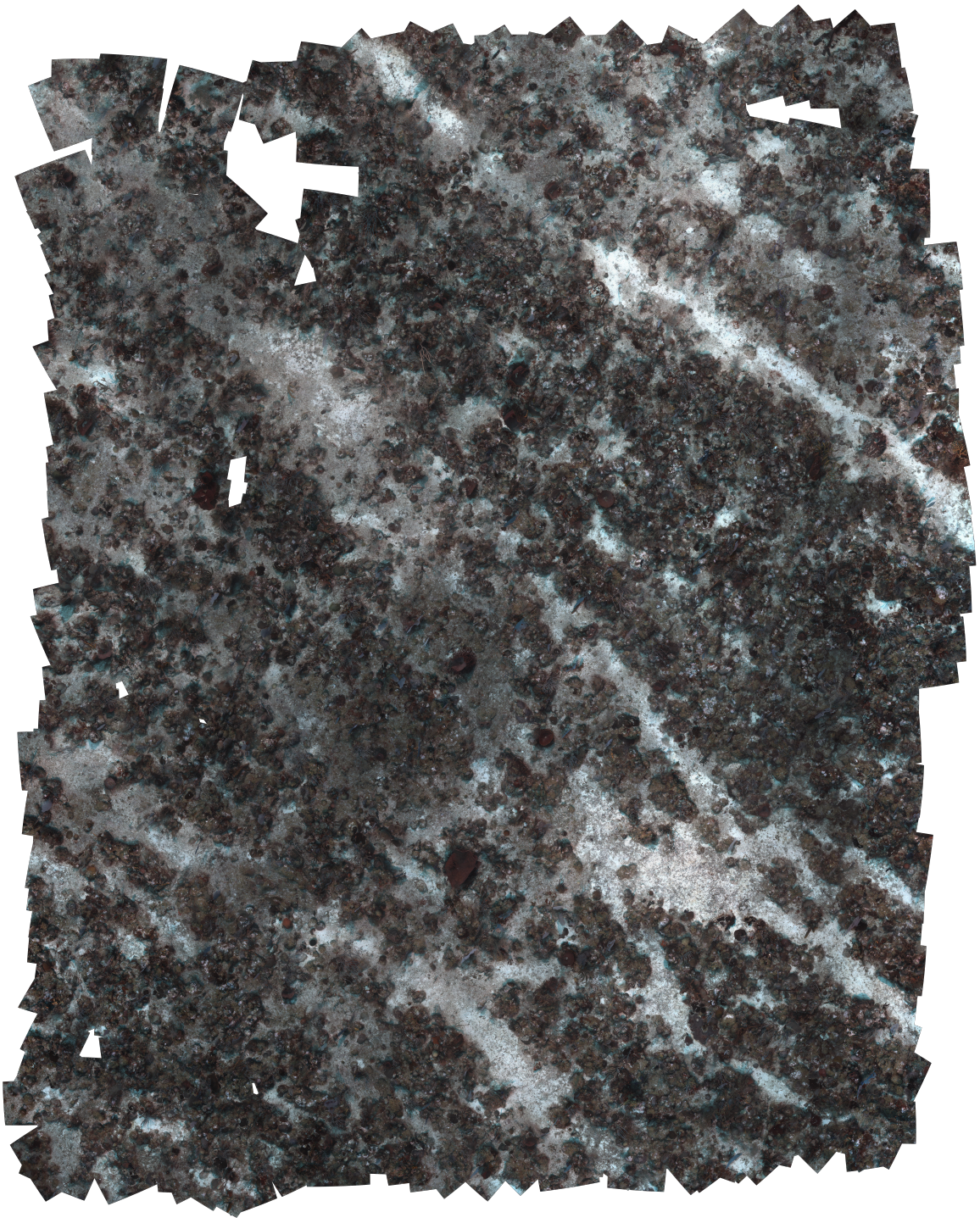

Figure 4-13: The full 2-D mosaic, formed by individually determining affine image transformations mapping feature pixel locations to the scene structure as estimated by the pose graph optimization. This rendering is at 50% of full resolution, and incorporates 1109 images from 37 distinct mosaics.

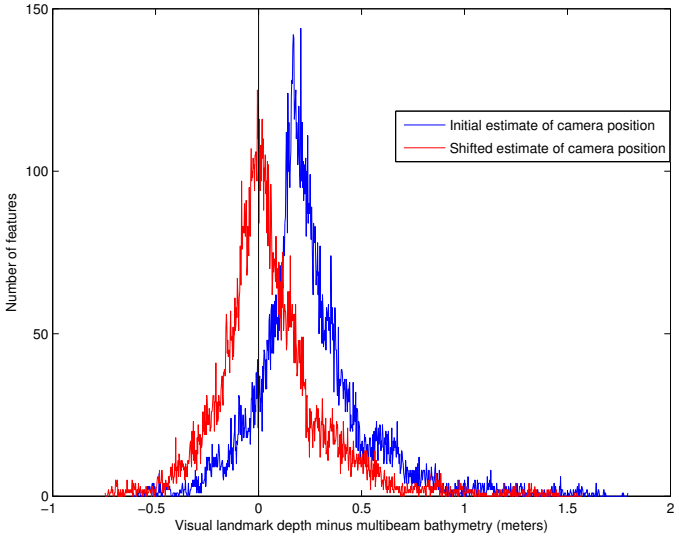

Figure 4-14: The disparity in scene depth computed from multibeam and from camera features. The blue line shows the difference when starting the combined optimization with an "eyeball" estimate of the camera location, while the red line shows the difference after re-starting the optimization with a camera location that has been shifted by the peak in the blue line. The black line is at zero, for reference. The final error in the two optimization runs is approximately equal, as are the pixel reprojection error histograms.

this has seemingly removed the bias. But it is important to realize that the bias has been removed without affecting the final overall optimization error, and the pixel reprojection error histograms have also remained essentially unchanged – in fact the histograms are so similar that it is not useful to show plots of them. This means that the camera can be moved vertically in the AUV's frame at least by this small amount without any penalty in the optimization error.

The fact that such a large shift in the scene structure can take place without significantly affecting the optimization error or the pixel reprojection errors is somewhat disturbing. It indicates that the error surface being minimized over is wide and flat at the bottom, at least with respect to the z position of the camera relative to the AUV. There are two explanations for this. The first is that the AUV is attempting to maintain a constant altitude over the seafloor while capturing images, and the lack of relative vertical motion means that this region of the error space is not being exercised. The second is that the pose graph as it has been defined does not directly compare the 3-D structure of the scene as measured by the two sensors. Only the depths of visual features are computed in the optimization, while the constraints in the map that are

contributed by multibeam are of the form of relative AUV poses. Thus the data fusion is somewhat indirect, in that it takes place through the reconstruction of the AUV's trajectory. While future dives can be planned to allow for increased vertical motion to better constrain the estimate of the camera's position, directly linking the scene structure measured by the two sensors is future work, as will be discussed in chapter 6.

Chapter 5

Mapping in drifting, rotating environments

5.1 Motivation

To this point, we have focused on map making in mostly static environments. While the coral reef environment used in the previous chapter is full of moving animals, and plants that sway in the current, making data association more challenging, the underlying terrain being mapped was fixed to the earth. We now turn our attention to the situation in which the environment being mapped is not anchored, in particular to the case of mapping the 3-D structure of drifting, rotating ice floes. AUVs are especially well-suited to the task of mapping the shape of icebergs and ice floes: it is much more dangerous for people to work under ice than in open water, and tethered robots are more constrained than usual when the support ship cannot maneuver arbitrarily with respect to the robot. Tethers can also become hung up on "stalactites" hanging from ice, complicating operations and recovery, and AUVs can operate over much greater ranges than tethered vehicles when a ship cannot reach the area of interest (under shelf ice, for example).

Moreover, there are compelling scientific reasons to measure the 3-D structure of ice floes, particularly in Antarctica. Because of the influence of snow, the thickness of ice in the southern ocean is not easily remotely measurable [36] [38]. The only reliable way to determine the correlation (if any) between the height of ice above sea level (freeboard) and the thickness of ice is to directly measure it, using surveying equipment or

laser altimeters to determine freeboard, and drilling holes by hand to measure thickness. A multibeam sonar can be used on an AUV to measure the thickness of ice below sea level (draft), so the efficiency of determining the amount of ice in a given floe can be greatly improved by combining draft measurements from AUVs with remote surface altimetry measurements and manual snow depth measurements. This is exactly what the Seabed AUV was used for on the "Icebell" expedition in November 2010, which is the source of the data in this chapter.

Acoustically mapping the underside of an Antarctic ice floe with an AUV might be as simple as reorienting the multibeam and DVL sensors to point upward, figuring out the necessary code changes to handle the new coordinate transformations for the sensors, and using the techniques described in previous chapters to build a map. This is largely the case for fast ice (ice that is grounded to the seafloor so that it does not move), though of course many nontrivial operational changes must be undertaken to ensure the AUV can be recovered after a dive. Under drifting ice, operations quickly become the primary concern, and acoustic navigation aids must be used to track the AUV to prevent vehicle loss (see [24] and [34] for discussions of under ice navigation and AUV operations). For an AUV, navigating with respect to drifting terrain appears much the same as navigating in a current: the vehicle might "crab" to compensate for moving in a direction different from the vehicle's heading. Once the floe starts to rotate, however, the AUVs onboard navigation will deteriorate, because heading relative to the terrain (typically the Earth) is provided by the north-seeking gyrocompass, so the estimated location of goal points will be incorrect from trackline to trackline. While floe rotation rates are often small for the size of surveys carried out on the Icebell expedition, one floe in particular exhibited a rotation rate high enough to severely impact the quality of the resulting map. Figure 5-1 shows this floe's orientation as measured by the ship's compass, which serves as a proxy as the ship was moored to the floe throughout the dive. The floe rate of about 2.8 degrees per hour confounded the AUV's navigation estimate as shown on the right side of the figure. At the same time, the floe was drifting at an average rate of 2 centimeters per second, which was compensated for by the crabbing motion mentioned above.

There has been little work using AUVs to map drifting and rotating ice. The exception is [31], which describes an algorithm to map the shape of icebergs using AUVs which circumnavigate them. That work also uses a SLAM approach, but the mapping is based on particle filters, and does not involve an AUV transiting beneath the ice.

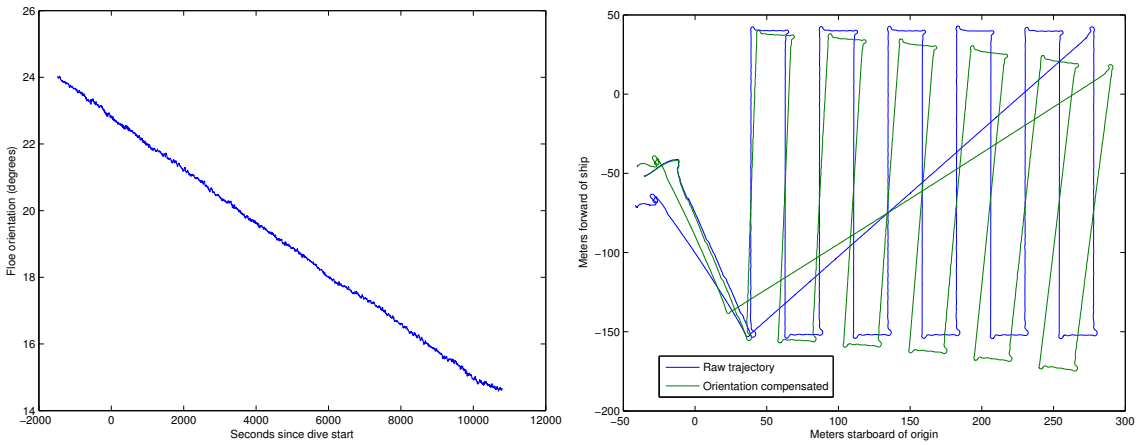

Figure 5-1: On the left, the floe orientation over the course of a 3 hour dive, as measured by the compass of a ship moored to the floe. The confusion in the AUV's initial trajectory estimate is shown on the right.

The thrust of the work, however, is that rotation needs to be accounted for in order to produce anything meaningful, and they provide an algorithm for doing so based on the established literature in terrain-relative navigation. Our approach is similar in that the orientation of the ice is explicitly estimated as part of the SLAM solution, but we do so using the already-described framework of pose graph optimization, so little of the algorithm has to change in order to account for the rotating world frame.

5.2 Augmenting the pose graph

There are a few possible ways to change the pose graph to move from a global reference frame to a terrain-relative frame. One possibility is to add a seventh degree of freedom to each AUV pose node, and then to change the attached factors to account for the transformation from world heading to terrain-relative heading. This is a rather intrusive approach, and unnecessarily reduces the sparsity of the system, as the measurements of AUV attitude are independent of the external measurements of terrain orientation. Instead, we attach pose nodes with a single degree of freedom representing terrain orientation to each AUV attitude measurement factor, as shown in figure 5-2. Since all the factors connecting AUV pose nodes in the graph only measure relative pose (odometry), only the attitude measurement factors constrain the global orientation of the whole trajectory, so it makes sense to inject a terrain-relative orientation

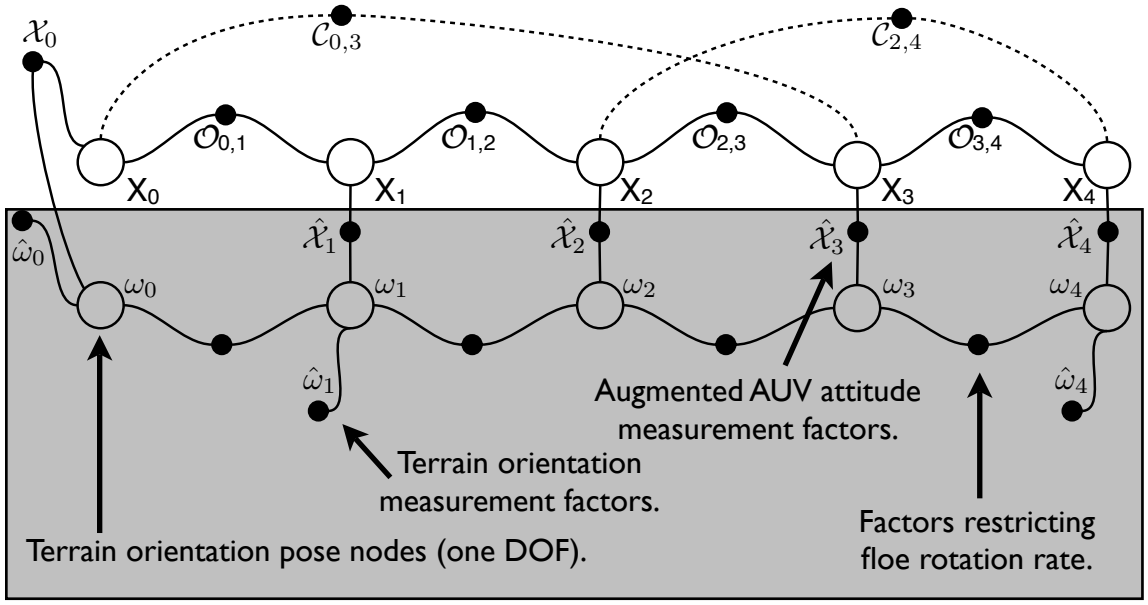

Figure 5-2: The multibeam mapping pose graph, augmented with world heading pose nodes and their factors. Each AUV attitude measurement prior has a corresponding world orientation pose node, which converts the world-relative heading measurement to the terrain-relative reference frame. Changes from the pose graph in chapter 3 are highlighted.

there in the graph. Each attitude measurement factor is modified so that it attaches to a pose node estimating the terrain orientation ω_t as well as an AUV pose node estimating X_t. The factor's error function compares the attitude $\begin{bmatrix} z_t & \rho_t & \phi_t & \theta_t \end{bmatrix}^\mathsf{T}$ in the AUV pose node to the measurement $\begin{bmatrix} \hat{z}_t & \hat{\rho}_t & \hat{\phi}_t & \hat{\theta}_t \end{bmatrix}^\mathsf{T}$ adjusted by the terrain orientation:

$$\mathbf{e}_{\mathcal{X}_t} = \begin{bmatrix} z_t & \rho_t & \phi_t & (\theta_t + \omega_t) \end{bmatrix}^\mathsf{T} - \begin{bmatrix} \hat{z}_t & \hat{\rho}_t & \hat{\phi}_t & \hat{\theta}_t \end{bmatrix}^\mathsf{T}$$

In other words, the attitude measurement factor is trying to enforce a terrain-relative attitude in the AUV pose node, rather than a world-relative attitude.

The augmented pose graph contains additional factors connecting the terrain orientation pose nodes which limit the allowed rotation rate of the terrain. This prevents solutions which might use the additional degree of freedom for each AUV pose to overfit the multibeam submap matches. Finally, measurement factors are added when an external sensor is available to measure the terrain orientation relative to the global reference frame. This might be provided by instrumenting an ice floe with a compass, or

as in the case here, by mooring a GPS-equipped ship to the floe throughout the course of the dive. Because the connection between the ship and the ice is not entirely rigid and GPS is not an ideal heading measurement system, small variations in the measured heading are to be expected, which are allowed by using a variance value of one degree in the terrain orientation measurement factors.

5.3 Results

The pose graph shown in figure 5-2 is a schematic for a real-world dive. In the field, sensor update rates vary significantly, such that there can easily be ten or more AUV pose nodes for each terrain orientation measurement. The number of cross links induced by submap matches also depends on the trajectory of the dive, and the shape of the terrain being mapped, since ambiguous terrain may be rejected outright by the matcher if the determinant of the Hessian is too low. For the ice floe data set, a large section of the mapped terrain was quite flat, resulting in fewer cross links than in other data sets – only five cross links were used in the optimization producing the binned multibeam map and the variance image shown in figure 5-3. This floe is striking in the large area of flat, or "level" ice, and the large protrusion in the lower right corner of the image. The draft of this feature is so deep that it shadows the neighboring terrain in the footprint of the sensor, even through the AUV was moving at least ten meters deeper than the extent of the protrusion.

The variance image in figure 5-3 is unexpectedly large compared to the binning variance in the coral reef data in figure 4-11 of the previous chapter. There are a few reasons for this difference. The most dramatic influence is the terrain itself: the bathymetry of the coral reef data varies from about 18 to 21 meters, while the draft of the ice floe data varies from near zero up to about 18 meters, sometimes very abruptly. Because the grid is effectively a planar projection of depth (or draft), areas of high slope will have high variance, which can be seen in the figure. More disturbing is the roll bias, which is visible in the variance image as bands of high variance in areas where the sensor footprint overlaps on adjacent tracklines. This is caused by time-varying roll bias in the attitude sensor, visible in figure 5-4. While the AUV was operating in a configuration that was less stable in Antarctica than in Puerto Rico, causing it to "fishtail" as it moved, the fibre optic gyro is not a sensor to be lightly dismissed as producing faulty data. A potential workaround is to only use alternating tracklines in the generation of the final

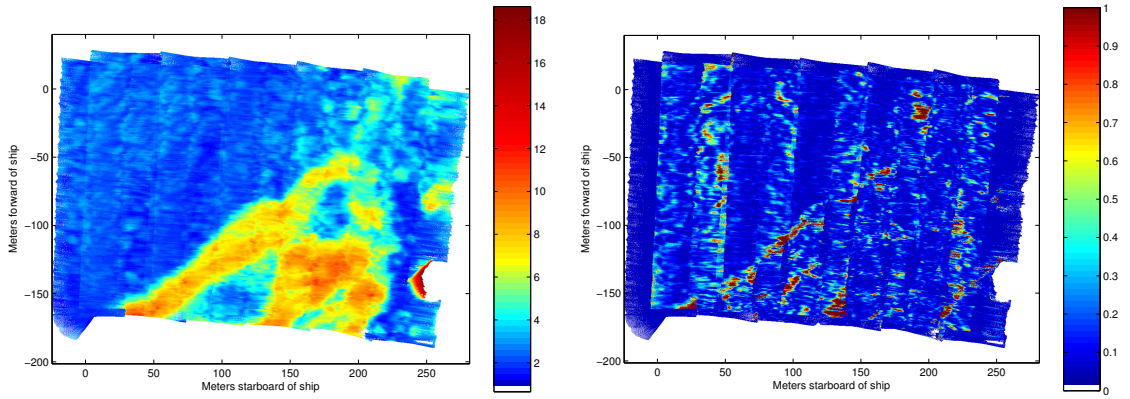

Figure 5-3: Binned multibeam map of the rotating ice floe draft. The mean draft for each bin is shown on the left in meters, and the variance of each bin on the right. The colormap on the right is compressed because of the deep protrusion in the lower-right corner. The variances are generally higher than typical because of varying roll and pitch biases (see text).

map, since the roll offset seems to be consistent along nearly identical headings. This requires a separate treatment of the calibration of the multibeam head attitude offset, however, because the optimization of the offset depends on the AUV moving in opposing directions. The cause and potential solutions to the heading-dependent roll and pitch variation is thus still a subject of ongoing research.

More abstractly, the computed trajectory and changes in multibeam binning variance are shown in figure 5-5. The reduction in variance achieved by adding the cross track constraints is not very large, because there were only five cross links added to the pose graph. This is a result of the small area of the survey, combined with the large region of flat ice which contained several areas where matches were rejected due to inconclusive map matching (i.e. a low Hessian determinant in the quadratic fit of the correlation surface). As expected, modeling the rotation of the terrain yields a marked improvement in the consistency of the generated map, as does optimizing over the attitude of the multibeam head relative to the DVL.

While adding the ability to account for terrain rotation and thus moving from a geo-referenced map to a local map required a large structure to be added to the pose graph, the new structure does not interfere with the relative relationships induced by multibeam map matches or visual landmarks. It is therefore possible to combine multibeam and visual mapping of a drifting, rotating target, by combining the technique in

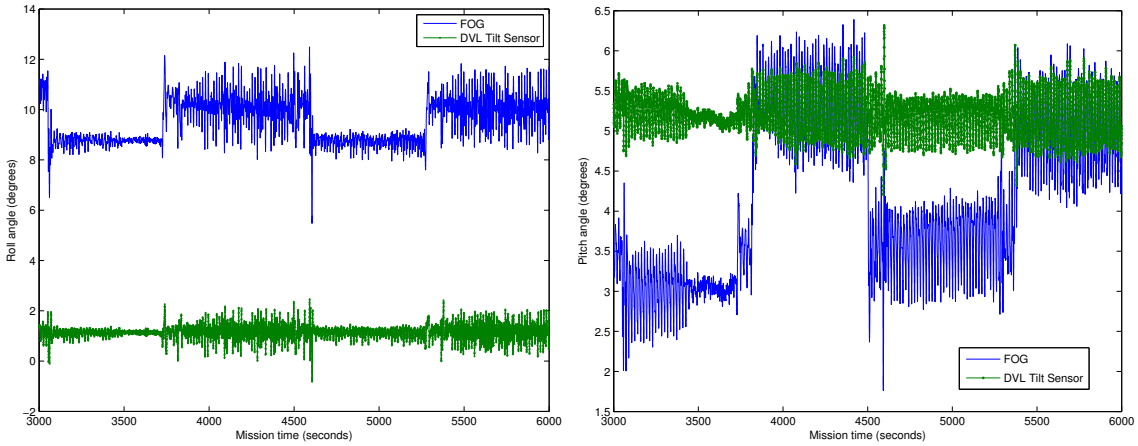

Figure 5-4: Roll (left graph) and pitch (right graph) estimates over a 50-minute period from an under-ice dive. There are two independent sensors which measure attitude, the fibre optic gyro, and tilt sensors in the doppler velocity log. It's important to note here that the relative bias between the two sensors is drifting over the course of the dive, and oscillating from trackline to trackline.

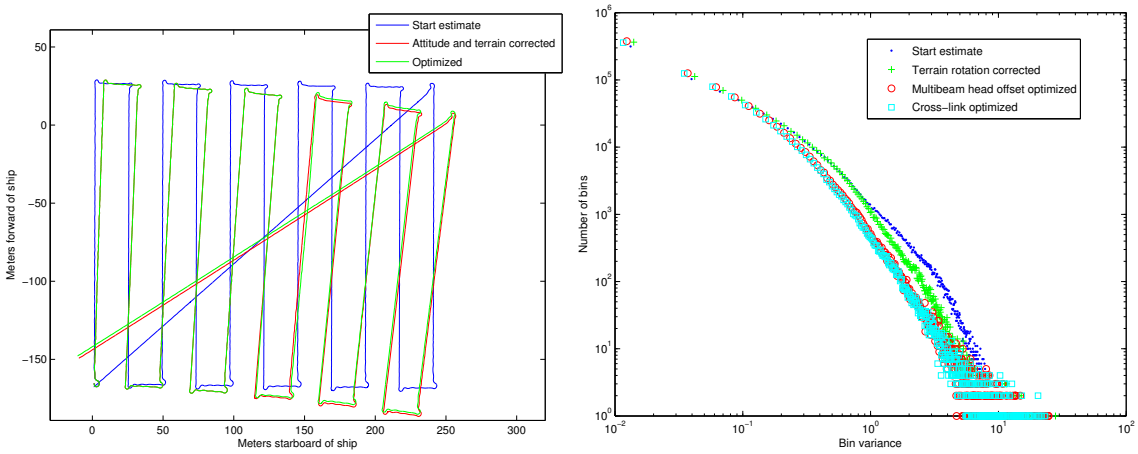

Figure 5-5: On the left, the estimated AUV trajectory before and after optimization is shown. The blue track shows the trajectory without any compensation for floe rotation. The reduction in binning variance at each step of the optimization is shown on the right. While adding the five cross links in the final step does not greatly affect the variance, each previous step is a significant improvement.

this chapter with that of the previous chapter. Due to the scientific requirements of the ice mapping mission prohibiting the use of a strobe, however, we do not have sufficient visual data from under the ice to build a useful map, and such efforts will have to wait for future expeditions.

Chapter 6

Discussion

6.1 Summary of contributions

The essential contribution of this thesis is an end to end *system* for building optical and acoustic maps of underwater terrain using autonomous robots. Making this system function well required solving many sub-problems:

- Auto-calibration of navigation sensor roll and pitch offsets using the vertical velocity estimate of the DVL and the absolute depth sensor to improve the initial navigation estimate. In some cases this first step yielded the most significant improvement in map consistency of the whole system.
- Optimization of the multibeam transducer head offset, automating the "patch test" processing that is often otherwise performed by hand.
- Determining relative pose constraints induced by matched multibeam submaps, using a 2-D correlation-based search with a quadratic fit providing an estimate of the quality of the match (and hence the information matrix used in the pose graph).
- Incorporating acoustic range data and visual features from cameras together in a single mapping framework, and computing the extrinsic camera calibration as part of the SLAM solution.
- Mapping drifting and rotating terrain by augmenting the trajectory being estimated with the orientation of the world frame.

- Producing a visualization of the combined map incorporating 3-D structure from multibeam and texture information from imagery.

We used the pose graph throughout as a convenient representation of the relationship between the variables being estimated and the constraints on those variables induced by the structure of the problems, and because an efficient and easily customizable software package for pose graph optimization (iSAM) was available.

While this document has focused on the analysis of acquired data, much of the work involved was pure oceanographic field robotics: building, operating, maintaining, and recovering underwater vehicles in many different sea and weather conditions. The system has been deployed on several similar, but not identical robots, with a diverse array of available navigation and mapping sensors, to the benefit of biologists, geologists, fisheries scientists, and ice scientists.

6.2 Comparison with earlier techniques

A summary of the methods presented in this thesis and others in the literature is shown in table 6.1. The technique described here stands out in that it includes many of the best features of other published techniques, including scalability, recovery of the full AUV trajectory, calibration of navigation sensors and of the camera location, and the possibility of being implemented in an on-line or incremental fashion. Another unique feature of this method is that it incorporates constraints from visual features, multibeam submap matches, and vehicle odometry simultaneously in a single optimization. Rather than using the two sensor modalities to build two maps which are then fused, each sensor instead has a chance to "correct" the others while a single estimate of the trajectory and map are being determined. In addition, while all multibeam mapping resolution depends to some extent on the operational scenario and the particulars of the hardware being used, we report better experimental binning resolution (0.25 meters resolution from 25 meter range, and 0.05 meters resolution from 3 meter range) than other multibeam mapping methods in the literature. Only the work of Roman et al. exceeds this resolution for dense coverage by using a laser sheet, the bulk of which requires an ROV at this time. While all of the techniques using cameras in this comparison make use of sparse features and therefore recover only sparse structure from cameras, the use of stereo cameras by Williams et al. allows for the computation of

dense structure, but this is not fused with structure from multibeam within the navigation and mapping framework.

6.3 Future research

There are several directions still to be pursued in this work, many of which are large research projects in and of themselves. The first problem is dealing with synchronization and latency in the data. The pose graphs used in this thesis did not allow for timing mistakes in the data, but rather allowed for a limited degree of measurement error through the information matrices encoded in the factors of the graph. In reality, however, there are unmodeled latencies between the sensors, and a limited amount of hardware synchronization available to reduce this latency. In figure 6-1, for example, it is clear that the two depth signals shown are highly correlated, but with a time offset. The lower rate signal is from a Paroscientific depth sensor, which uses the oscillating frequency of a quartz crystal to determine pressure, while the higher rate signal is from a strain gauge that is expected to be noisier. While the strain gauge does produce a noisy signal, it "sees" changes in depth sooner than does the Paroscientific sensor – correlating the signals reveals a latency of about 0.7 seconds, or seven multibeam pings, more than enough to reduce map quality. Determining and accounting for this latency and others in the system – especially between the camera and multibeam sensors themselves – would provide significant improvement.

Another problem for future work addresses the indirect nature of the approach used here for improving the consistency of multibeam maps. The approach used in chapter 3 attempts to improve the overall consistency of the map by adding constraints on individual map subsections. The optimization should improve the local consistency of these regions, but it will not necessarily improve the overall map consistency while doing so. In contrast, the landmark-based approach used to incorporate visual features into the map directly aims to minimize pixel reprojection error for every matched feature, so all of the visual data acquired is used at once in the optimization. To fully address the problem with range data, either the overall map consistency should be used as an error metric in the optimization, or more cross links should be introduced to ensure a better global reduction in binning variance. The latter solution carries with it the risk of reducing the sparsity of the pose graph, but it could still be applied using the framework presented here. Adding a global consistency metric, however, would likely

Table 6.1: A brief comparison of underwater navigation and mapping techniques.

Method	Mapping sensors, coverage	Applications, notes
Eustice [10]	One camera, sparse structure	Wide-area visual surveying. Delayed-state SEIF, can calibrate camera extrinsics, exploits sparsity for scalability.
Roman [49]	Multibeam, one meter gridding	Wide-area medium-resolution surveying. Delayed-state EKF. Relative pose constraints induced from submap matches.
Roman, Inglis, Rutter [50]	Multibeam, one camera, laser sheet, sub-centimeter gridding	Close-up high-resolution mapping of archaeological and geological sites with ROV. Delayed state EKF with relative pose constraints.
Williams, Pizarro, Barkby, Mahon, Johnson-Roberson [66], [1], [25]	Multibeam and stereo cameras, one meter gridding, sparse structure	Particle filter, builds distinct stereo and multibeam maps and fuses them with ICP.
Pizarro, Gracias [47], [16]	One camera, sparse structure	No explicit SLAM, uses odometry to resolve scale ambiguity, can calibrate magnetic compass bias.
Stereo visual odometry, e.g. [43], [35], [45]	Stereo cameras, sparse structure	Augmentation of low-fidelity odometry sensors. No explicit SLAM, uses odometry to calibrate camera rig extrinsics. Subject to unbounded drift.
Fairfield [12]	Multibeam, meter-scale gridding	Mapping complex structures, caves. Particle filter, maps complex 3-D shapes using octree occupancy volume.
Kimball [31]	Multibeam, 7 meter gridding	Mapping uninstrumented rotating icebergs. Particle filter, mapping and navigation in two distinct phases.
This thesis	**Multibeam and one camera, centimeter-scale gridding, mixed structure**	**Wide area surveys of rotating terrain. Square root SAM, exploits sparsity for scalability. Calibrates navigation sensors and camera extrinsics, fuses mapping sensors with trajectory in a single optimization.**

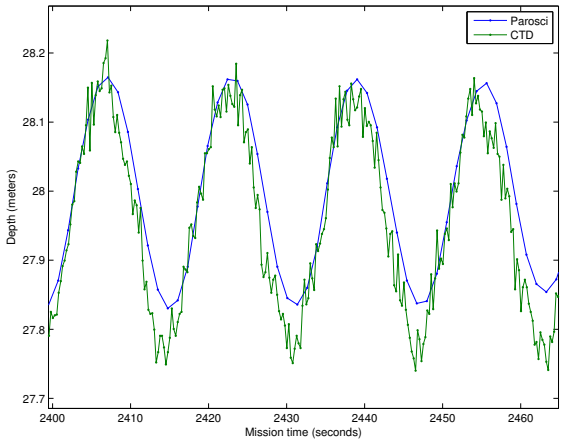

Figure 6-1: The AUV carries two depth sensors with different characteristics. The Paroscientific sensor is expected to have higher accuracy than the CTD, but it has a slower sampling rate, and there is latency between the two signals. The cross-correlation between the two signals interpolated to the faster time base shows a peak at about 0.7 seconds of latency.

require a change in problem representation or at the very least a large reorganization of the pose graph to make use of something like an occupancy grid, so that the binned output is treated directly by the optimizer. The most straightforward approach would be to represent each cell of the grid with a pose node estimating depth, and connecting each such node to AUV pose nodes which measure range to that cell (via a measurement factor). The problem is that as the optimizer runs, the connectivity of the graph would change, as changes in the estimate of the AUV's trajectory would yield measurements of different areas of the terrain, and hence different grid cells. While current graph optimization engines allow for incremental updates to the graph structure to be made [28], these are generally in the form of new landmarks being introduced, new poses being added, or new measurements connecting poses and landmarks being made. We are suggesting a change that would be akin to a wholesale reassignment of several observations from one landmark to another, which would likely seriously impair optimization efficiency, as the sparsity structure of the Jacobian would radically change from iteration to iteration. An efficient implementation capable of such reorganization remains an open question, but it would both allow the direct optimization over the full multibeam map, and provide a means for the bathymetry as measured by the multibeam to be directly compared to the scene structure as measured by the camera, pulling the two sets of depth estimates together and improving the flatness of the error

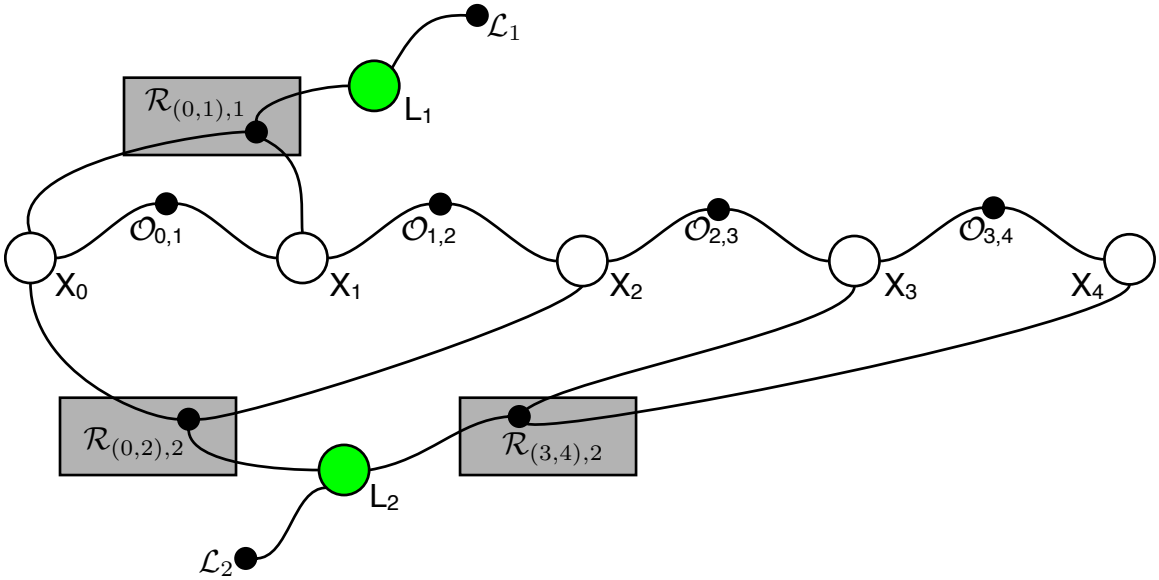

Figure 6-2: An example graph for using acoustic ranges which takes into account the finite travel time of the signal. The three highlighted factors each capture a single measurement of range to one of the two beacons. The two AUV poses attached to a given range factor may differ by meters in typical LBL deployments.

surface relative to the z position of the camera.

In addition to these two directions the research might take in the future, there are a few smaller changes that could be made within the current framework. The fact that range measurements are not effectively instantaneous distinguishes underwater robotics from ground and aerial systems, where lasers are often used for ranging. Underwater, both navigation measurements (USBL, LBL) and mapping measurements (multibeam and other acoustic systems) take place over two points in time, and therefore the mapping system should use the robot's pose at both points in time when making the map. Figure 6-2 illustrates the idea, in which the measurement factor $\mathcal{R}_{(i,j),b}$ for a given range uses the total travel time for an acoustic signal, and compares that against the total two-way range from the AUV at time t_i to the beacon b (or to terrain) and back to the AUV at time t_j. As mentioned above, the current pose graph does not model the terrain directly, so this modification to the graph would only be possible for LBL and USBL ranges. It should be kept in mind however for the future project of more tightly linking the map to the pose graph.

Another line of research concerns dealing with rotating ice floes. In the situation

presented here, the ice floe being mapped was "instrumented by proxy" using the GPS onboard the ship moored to the floe. In many situations (e.g. on icebergs or over long-term deployments) this is not possible and the orientation of the terrain must be induced from the navigation and mapping sensors of the AUV. Another way to think of this is that the terrain-relative heading of the AUV must be determined from the global heading and the map as it is being built. One approach to solving this problem is active: the AUV can decouple its motion from that of the terrain by holding still for a period of time while it observes the motion of the floe, and then uses that observation as a prior as it starts to navigate. This presumes that the terrain motion is observable with the available sensors, and that the AUV is capable of holding station, neither of which should be taken for granted, but it does suggest that the problem is solvable if such behavior is programmed into the robot's missions.

6.4 Lessons learned

The work presented in this thesis is the result of five years of data collection and experimentation in underwater mapping using AUVs. The pose graph optimization algorithms presented here for building multibeam maps were applied to data from seven under-ice dives, each of which covered from 100 to 300 meters square, and the algorithm combining visual features with multibeam bathymetry was applied to data from a single dive in Puerto Rico covering about 30 meters square. These particular dives were chosen because they presented the most interesting cases – only in Antarctica was the goal to map a rotating, drifting object, and of the data collected only the coral reef in Puerto Rico had both rich visual texture and a high degree of variability in the bathymetry. In addition to these cases, there are compelling visual data from an expedition to Alaska involved the use of four synchronized cameras in two stereo pairs. Using calibrated stereo cameras allows for the direct creation of dense texture-mapped range data with each image capture, potentially removing the additional utility added by multibeam. Even though the extrinsic calibration can be automatically computed using AUV motion [35], accurate dense 3-D reconstruction from stereo requires very good intrinsic camera calibration, and of course depends on the degree to which pixel correspondences can be established. In cases where the seafloor is devoid of significant texture (as in a muddy bottom), multibeam is an excellent complement to a stereo pair and should be used whenever available.

More directly relevant to the algorithms presented here are a few steps which can be taken to increase the chances of building a consistent and useful map. First of all, sufficiently redundant information must be provided by the mapping sensors. For the coral reef dive, the planned-for image overlap was about 50% horizontally, with the AUV attempting to maintain constant altitude of three meters with one meter trackline spacing and a lens with field of view of about 35 degrees in water. The vertical image overlap, dictated by strobe recharge times and vehicle velocity, was about 30%. In order to build large mosaics, close loops, and ultimately build a large 3-D model, side-to-side overlap appears to be quite important. The multibeam sensor has an effective field of view of about 90 degrees, so there was more than 50% overlap from trackline to trackline over the coral reef. Under ice, the trackline spacing was set to be equal to the depth, also affording 50% overlap and yielding highly redundant data. The diagonal tracklines shown in the trajectory plots are particularly helpful, and executing these at depths different than those used during the primary survey or using zig-zag patterns instead of straight diagonal transects only serve to increase the usefulness of the data both in mapping and in checking data consistency and validity.

In addition to redundant mapping data, it is clear from a data quality perspective that a highly stable platform is desirable. The Seabed AUV is typically very stable, but its under-ice configuration required a buoyancy redistribution which reduced stability. A well-instrumented platform can compensate for instability, but the increased roll and pitch amplitudes exacerbate the problems induced by unmodeled latency in the sensors, as mentioned above.

Finally, it is common sense that one should take into account the expected terrain when choosing which sensors to deploy on an AUV. Generally speaking, cameras provide better constraints on navigation than do range sensors, because of their high resolution and because modern feature detectors are very good at identifying distinct landmarks. But each sensor has terrain to which it is particularly well-suited. Over a coral reef, with rich visual texture and variable bathymetry, both cameras and multibeam sensors should be used. Over a muddy or sandy bottom, or in turbid water, multibeam is more important. But if power and payload constraints are minimal, it makes sense to use every sensor one has access to, since in oceanography (and in field robotics in general) the unexpected often comes to pass.

6.5 Conclusion

This thesis has presented a complete system for building bathymetric maps of the seafloor incorporating visual texture, as well as 3-D maps of rotating ice floes using an autonomous underwater vehicle. The system improves the state of the art by a factor of five, while automating several calibration steps that were often before performed by hand. Moreover, the ability to measure the draft of ice floes from below with this precision enables scientific inquiry that was not previously possible, which is precisely the purpose of this kind of field robotics. The system has shown itself to be generalizable to several different deployment scenarios, with varying sensor configurations and degrees of control stability – in short the post-processing steps described here are independent of the particular hardware used to collect the data, except for the easy-to-meet assumptions of measurability of depth, attitude and velocity. While this work has revealed several additional lines of research to be pursued, the system as it stands robustly and autonomously creates accurate and useful high-resolution 3-D visual maps of underwater terrain.

Bibliography

[1] Stephen Barkby, Stefan Williams, Oscar Pizarro, and Michael Jakuba. An efficient approach to bathymetric SLAM. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 219 –224, October 2009.

[2] Jean-Yves Bouguet. Camera calibration toolbox for MATLAB [online]. URL: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[3] N.A. Brokloff. Matrix algorithm for doppler sonar navigation. In *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings*, volume 3, pages III/378 –III/383 vol.3, September 1994.

[4] Peter. J. Burt and Edward H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[5] Richard Camilli, Christopher M. Reddy, Dana R. Yoerger, Benjamin A. S. Van Mooy, Michael V. Jakuba, James C. Kinsey, Cameron P. McIntyre, Sean P. Sylva, and James V. Maloney. Tracking hydrocarbon plume transport and biodegradation at deepwater horizon. *Science*, 330(6001):201–204, 2010.

[6] David W. Caress and Dale N. Chayes. Mapping the seafloor: Software for the processing and display of swath sonar data [online]. June 2011. URL: http://www.mbari.org/data/mbsystem/ [cited 06/02/2011].

[7] M. Elizabeth Clarke, Nick Tolimieri, and Hanumant Singh. Using the Seabed AUV to assess populations of groundfish in untrawlable areas. In Richard J. Beamish and Brian J. Rothschild, editors, *The Future of Fisheries Science in North America*, volume 31 of *Fish and Fisheries*, pages 357–372. Springer Science + Business Media B.V., 2009.

[8] Frank Dellaert and Michael Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, 2006.

[9] Austin I. Eliazar and Ronald Parr. DP-SLAM 2.0. In *Proc. IEEE International Conference on Robotics and Automation*, volume 2, pages 1314 – 1320 Vol.2, May 2004.

[10] Ryan M. Eustice, Oscar Pizarro, and Hanumant Singh. Visually augmented navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 33(2):103–122, April 2008.

[11] Ryan M. Eustice, Louis L. Whitcomb, Hanumant Singh, and Matthew Grund. Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. In *Proc. IEEE International Conference on Robotics and Automation*, Rome, Italy, April 2007.

[12] Nathaniel Fairfield, George Kantor, and David Wettergreen. Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, 24(1-2):03–21, 2007.

[13] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[14] Brendan P. Foley, Katerina Dellaporta, Dimitris Sakellariou, Brian S. Bingham, Richard Camilli, Ryan M. Eustice, D. Evagelistis, V. Ferrini, K. Katsaros, D. Kourkoumelis, A. Mallios, P. Micha, D. Mindell, Chris Roman, Hanumant Singh, D. Switzer, and T. Theodoulou. The 2005 chios ancient shipwreck survey: New methods for underwater archaeology. *Hesperia*, 78(2):269–305, April 2009.

[15] D. Gómez-Ibáñez, C.L. Taylor, M.C. Heintz, J.C. Howland, D.R. Yoerger, A.D. Bowen, and L.L. Whitcomb. Energy management for the Nereus hybrid underwater vehicle. In *OCEANS 2010*, pages 1 –9, September 2010.

[16] Nuno Gracias, Sjoerd van der Zwaan, Alexandre Bernardino, and José Santos-Victor. Mosaic based navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 28(4):609–624, October 2003.

[17] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[18] Berthold K. P. Horn and E. J. Weldon Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.

[19] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3946 –3952, September 2008.

[20] Tak-Wai Hui and Ronald Chung. Determining spatial motion directly from normal flow field: A comprehensive treatment. In Reinhard Koch and Fay Huang, editors, *Computer Vision – ACCV 2010 Workshops*, volume 6468 of *Lecture Notes in Computer Science*, pages 23–32. Springer-Verlag, 2011.

[21] N. Hurtós, X. Cufí, and J. Salvi. Calibration of optical camera coupled to acoustic multibeam for underwater 3D scene reconstruction. In *OCEANS 2010 IEEE - Sydney*, pages 1 –7, May 2010.

[22] Wil S. Hylton. What happened to Air France flight 447? *New York Times Sunday Magazine*, page MM38, May 4 2011.

[23] Gabrielle Inglis and Chris Roman. Terrain constrained stereo correspondence. In *Proc. IEEE/MTS Oceans Conference and Exhibition*, 2009.

[24] Michael V. Jakuba, Chris N. Roman, Hanumant Singh, Christopher Murphy, Clayton Kunz, Claire Willis, Taichi Sato, and Robert A. Sohn. Long-baseline acoustic navigation for under-ice autonomous underwater vehicle operations. *Journal of Field Robotics*, 25(11-12):861–879, 2008.

[25] Matthew Johnson-Roberson, Oscar Pizarro, Stefan B. Williams, and Ian Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51, 2010.

[26] Jeffrey W. Kaeli, Hanumant Singh, Chris Murphy, and Clayton Kunz. Improving color correction for underwater image surveys. In *Proc. IEEE/MTS Oceans Conference and Exhibition*, 2011.

[27] Michael Kaess, Hordur Johannsson, and John Leonard. iSAM: Incremental smoothing and mapping [online]. URL: http://people.csail.mit.edu/kaess/isam/ [cited 6/6/2011].

[28] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 2012. To appear.

[29] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365 –1378, December 2008.

[30] Ayoung Kim and Ryan Eustice. Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1559 –1565, October 2009.

[31] Peter Kimball and Stephen Rock. Sonar-based iceberg-relative navigation for autonomous underwater vehicles. *Deep Sea Research Part II: Topical Studies in Oceanography*, 58(11-12):1301 – 1310, 2011. Free-Drifting Icebergs in the Southern Ocean.

[32] James C. Kinsey and Louis L. Whitcomb. Adaptive identification on the group of rigid body rotations and its application to precision underwater robot navigation. *IEEE Transactions on Robotics*, 23(1):124–136, February 2007.

[33] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g^{2}o : A general framework for graph optimization. In *Proc. IEEE International Conference on Robotics and Automation*, Shanghai, May 2011.

[34] Clayton Kunz, Chris Murphy, Hanumant Singh, Claire Pontbriand, Robert A. Sohn, Sandipa Singh, Taichi Sato, Chris Roman, Ko-ichi Nakamura, Michael Jakuba, Ryan Eustice, Richard Camilli, and John Bailey. Toward extraplanetary under-ice exploration: Robotic steps in the arctic. *Journal of Field Robotics*, 26(4):411–429, 2009.

[35] Clayton Kunz and Hanumant Singh. Stereo self-calibration for seafloor mapping using AUVs. In *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, pages 1 –7, September 2010.

[36] Nathan T. Kurtz, Thorsten Markus, Donald J. Cavalieri, Lynn C. Sparling, William B. Krabill, Albin J. Gasiewski, and John G. Sonntag. Estimation of sea ice thickness distributions through the combination of snow depth and satellite laser altimetry data. *J. Geophys. Res.*, 114(C10), 10 2009.

[37] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[38] Ted Maksym and Thorsten Markus. Antarctic sea ice thickness and snow-to-ice conversion from atmospheric reanalysis and passive microwave snow depth. *Journal of Geophysical Research*, 113(C2), 02 2008.

[39] Peter Meer, Doron Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, April 1991.

[40] P.H. Milne. *Underwater acoustic positioning systems.* Gulf Publishing Co., Houston TX, January 1983.

[41] S. Negahdaripour, H. Sekkati, and H. Pirsiavash. Opti-acoustic stereo imaging: On system calibration and 3-D target reconstruction. *IEEE Transactions on Image Processing*, 18(6):1203 –1214, June 2009.

[42] Shahriar Negahdaripour. Epipolar geometry of opti-acoustic stereo imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1776 –1788, October 2007.

[43] Shahriar Negahdaripour and Hossein Madjidi. Stereovision imaging on submersible platforms for 3-D mapping of benthic habitats and sea-floor structures. *IEEE Journal of Oceanic Engineering*, 28(4):625–650, October 2003.

[44] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756 –770, June 2004.

[45] Clark F. Olson, Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, 2003.

[46] Rohan Paul and Paul Newman. FAB-MAP 3D: Topological mapping with spatial and visual appearance. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2649–2656, Anchorage, Alaska, May 2010.

[47] Oscar Pizarro. *Large Scale Structure from Motion for Autonomous Underwater Vehicle Surveys*. PhD thesis, MIT / Woods Hole Oceanographic Institution, 2004.

[48] Oscar Pizarro and Hanumant Singh. Toward large-area mosaicing for underwater scientific applications. *IEEE Journal of Oceanic Engineering*, 28(4), October 2003.

[49] Chris Roman. *Self Consistent Bathymetric Mapping From Robotic Vehicles in the Deep Ocean*. PhD thesis, Massachusetts Institute of Technology, June 2005.

[50] Chris Roman, Gabrielle Inglis, and James Rutter. Application of structured light imaging for high resolution mapping of underwater archaeological sites. In *OCEANS 2010 IEEE - Sydney*, pages 1 –9, May 2010.

[51] Chris Roman and Hanumant Singh. A self-consistent bathymetric mapping algorithm. *Journal of Field Robotics*, 24(1-2):23–50, 2007.

[52] Hanumant Singh, Ali Can, Ryan Eustice, Steve Lerner, Neil McPhee, Oscar Pizarro, and Chris Roman. Seabed AUV offers new platform for high-resolution imaging. *EOS, Transactions of the AGU*, 85(31):289,294–295, August 2004.

[53] Hanumant Singh, Louis Whitcomb, Dana Yoerger, and Oscar Pizarro. Microbathymetric mapping from underwater vehicles in the deep ocean. *Computer Vision and Image Understanding*, 79(1):143–161, 2000.

[54] Randall Smith, Matthew Self, and Peter Cheeseman. *Estimating uncertain spatial relationships in robotics*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[55] Robert A. Sohn, Claire Willis, Susan Humphris, Timothy M. Shank, Hanumant Singh, Henrietta N. Edmonds, Clayton Kunz, Ulf Hedman, Elisabeth Helmke, Michael Jakuba, Bengt Liljebladh, Julia Linder, Christopher Murphy, Ko-ichi

Nakamura, Taichi Sato, Vera Schlindwein, Christian Stranne, Maria Tausenfreund, Lucia Upchurch, Peter Winsor, Martin Jakobsson, and Adam Soule. Explosive volcanism on the ultraslow-spreading gakkel ridge, arctic ocean. *Nature*, 453(7199):1236–1238, June 2008.

[56] M. Jordan Stanway and James C. Kinsey. Sensor alignment using rotors in geometric algebra. In *Proc. IEEE International Conference on Robotics and Automation*, pages 994–999, May 2011.

[57] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Scale drift-aware large scale monocular SLAM. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[58] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents Series. The MIT Press, August 2005.

[59] Tali Treibitz, Yoav Y. Schechner, Clayton Kunz, and Hanumant Singh. Flat refractive geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1, 2011.

[60] Giancarlo Troni and Louis L. Whitcomb. New methods for in-situ calibration of attitude and doppler sensors for underwater vehicle navigation: Preliminary results. In *OCEANS 2010*, pages 1 –8, September 2010.

[61] Iuliu Vasilescu, Carrick Detweiler, and Daniela Rus. Color-accurate underwater imaging using perceptual adaptive illumination. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[62] Matthew Walter, Franz Hover, and John Leonard. SLAM for ship hull inspection using exactly sparse extended information filters. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1463 –1470, May 2008.

[63] Daniel Wedge. The fundamental matrix song [online]. March 2009. URL: http://danielwedge.com/fmatrix/ [cited August 12, 2011].

[64] Louis L. Whitcomb, Dana Yoerger, and Hanumant Singh. Combined doppler/LBL based navgation of underwater vehicles. In *Proc. 11th International Symposium on Unmanned Untethered Submersible Technology*, Durham, New Hampshire, May 1999.

[65] Louis L. Whitcomb, Dana Yoerger, Hanumant Singh, and Jonathan Howland. Advances in underwater robot vehicles for deep ocean exploration: Navigation, control, and survey operations. In John Hollerbach and Dan Koditschek, editors, *Robotics Research: The Ninth International Symposium*. Springer-Verlag, October 1999.

[66] Stefan Williams, Oscar Pizarro, Ian Mahon, and Matthew Johnson-Roberson. Simultaneous localisation and mapping and dense stereoscopic seafloor reconstruction using an auv. In Oussama Khatib, Vijay Kumar, and George Pappas, editors, *Experimental Robotics*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 407–416. Springer Berlin / Heidelberg, 2009.

[67] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[68] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.