Underwater Localization and Mapping with Imaging Sonar

Eric Westman

CMU-RI-TR-19-77

The Robotics Insitute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Michael Kaess, Chair
Martial Hebert
George Kantor
John Leonard, MIT

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Robotics.*

Keywords: Simultaneous Localization and Mapping; Underwater Navigation; 3-D Reconstruction; Sonar

In memory of my father

Abstract

Acoustic imaging sonars have been used for a variety of tasks intended to increase the autonomous capabilities of underwater vehicles. Among the most critical tasks of any autonomous vehicle are localization and mapping, which are the focus of this work. The difficulties presented by the imaging sonar sensor have led many previous attempts at localization and mapping with the imaging sonar to make restrictive assumptions, such as a planar seafloor environment or planar sensor motion. Lifting such restrictions is an important step toward achieving general-purpose autonomous localization and mapping in real-world environments.

In this dissertation, I take inspiration from related problems in the field of computer vision and demonstrate that imaging sonar localization and mapping may be modeled and solved using similar methods. To achieve accurate large-scale localization, I present degeneracy-aware acoustic bundle adjustment, a feature-based algorithm inspired by optical bundle adjustment. To achieve 3-D mapping of underwater surfaces, I propose several distinct algorithms. First, I present a method akin to shape-from-shading that uses a generative sensor model to infer a dense pointcloud from each image and fuses multiple such observations into a global model. Second, I describe a volumetric albedo framework for general-purpose sonar reconstruction, which derives from the related problem of non-line-of-sight reconstruction. This method performs inference over the elevation aperture and generates best results with a rich variety of viewpoints. Lastly, I present the theory of Fermat paths for sonar reconstruction, which utilizes the 2-D Fermat flow equation to reconstruct a particular set of object surface points with short baseline motion.

Acknowledgments

I would first like to express my deepest gratitude to my advisor Michael Kaess for giving me this opportunity and making all of the work in this dissertation possible. His patient guidance, insight, and wisdom have helped direct me countless times upon this journey and have provided me with an exceptional model to follow in developing my own research skills. None of this work would have been possible without Michael's influence in my studies. I would also like to thank Martial Hebert, George Kantor, and John J. Leonard for supporting me throughout the completion of this dissertation. They have all provided excellent insights and feedback that have improved the quality of this work and my own quality as a researcher.

My collaborators, colleagues, and peers have made great impacts on this work that cannot be understated. I would like to thank Ioannis Gkioulekas for his collaboration – much of this work would not have been possible without the many insights he has provided. My peers who have worked by my side on underwater robotics, and particularly the HAUV, deserve much credit for helping me expand the horizons of my own knowledge, carry out experiments, collect data, appreciate the challenges of waterproofing electronics, and develop a taste for San Diego sushi: Pedro, Josh, Jie, Paloma, Akshay, Suddhu, and Bing. I have thoroughly enjoyed walking through every step of this process with Ming Hsiao, and deeply appreciate all of the enlightening discussions we have shared over the years, and the many ways that he has helped to improve my work and my capabilities. All of my other labmates and colleagues throughout my time at the Robot Perception Lab have made a wide variety of contributions to the development of my work and myself, and made my experience at RPL one to remember: Tiffany, Garrett, Puneet, Jerry, Guofeng, Rafael, Lipu, Monty, Allie, Wei, Zimo, Jack, Eric, Josh, and Chen.

I would like to thank my family for supporting me not just throughout my time at CMU, but through all the years that have led up to it. I would not have come this far or been capable of this work without the constant love and support of my family and particularly my parents. And lastly, I have utmost appreciation for my wife Hannah, who has loved and encouraged me unfailingly, every step of the way.

Soli Deo gloria.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Why underwater?

Over the past several decades, the development of mobile robots has resulted in great advancements in the areas of exploration, inspection, and search and rescue. Using remotely operated or autonomous robots to carry out such tasks can remove humans from risky or dangerous scenarios as well as improve the effectiveness and efficiency of these tasks. The most well-known and highly publicized robotic platforms are typically ground and aerial vehicles. These operate in the environments that we are most familiar with as humans: on the roads, inside buildings and man-made structures, in the open air in outdoor environments, etc. The development of robots that operate in such environments is obviously crucial, yet it is also important to develop robots that are capable of venturing in environments that we do not encounter on a regular basis. The best example of such an unfamiliar and inhospitable environment for humans is underwater.

While over 70% of the earth's surface is covered by water, the seas have long been a difficult place to explore. Recent methods using satellite sensing have generated maps of all of the earth's ocean floors, but at a resolution of about 5 km [97]. Only a small percentage of the world's oceans have been mapped in detail with modern underwater sensing equipment. High resolution, small scale maps of contained underwater environments are often useful or even necessary for some applications. These types of environments include, but are not limited to, archaeological sites (e.g. shipwrecks or plane wrecks), ship hulls, bridge and pier pilings, oil and gas rigs, underwater pipelines, and coral reefs. Moving towards fully autonomous mapping, exploration, inspection, and maintenance of these types of small-scale underwater environments is the ultimate goal of the work conducted in this thesis.

1.2 Why imaging sonar?

Remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) have utilized a wide variety of sensors. The attenuation of electromagnetic signals underwater prohibits the use of GPS and radio-based systems for localization except when operating at the surface. For truly underwater missions, localization may be aided by inertial measurement units (IMUs) or inertial navigation systems (INSs), which are interoceptive sensors that measure accelerations and rotational velocities using accelerometers and gyroscopes. The velocities and accelerations may be integrated over time to provide the odometry estimates. Acoustic sensors such as Doppler velocity logs (DVLs) are often used to directly measure the vehicle's translational velocity, which may be fused with inertial measurements to provide a more accurate state estimate based on dead reckoning. But the pose estimate from any dead reckoning-based localization system will drift from the true pose over a sufficiently long period of time. In order to achieve a drift-free state estimate, exteroceptive sensors must be utilized.

For open-air robotic platforms, the most commonly used exteroceptive sensors are cameras and laser-based lidars. These sensors provide rich, high-bandwidth measurements of the surrounding environments, usually using the infrared and visible portions of the electromagnetic spectrum. Cameras and specially-designed underwater lidars (e.g. [1]) have been used by underwater vehicles for tasks such as localization, mapping, and object recognition. However, due to the attenuation of visible and infrared light underwater as well as the high turbidity often encountered in real-world underwater environments, these sensors often have a very limited functional range.

Acoustic sonars (originally an acronym for SOund Navigation And Ranging) have been used for decades by mounting them on ships. Some classic modes of sonar sensing include side scan sonar and synthetic aperture sonar (SAS). Side scan sonar and SAS are normally used as fairly long-range sensing modalities. These are often mounted on surface vessels or underwater vehicles to map large areas of the seafloor, usually require specific vehicle trajectories, and are capable of generating maps over areas of thousands of meters with resolution on the order of magnitude of several meters. While localization, mapping, detection and tracking have been performed with these sensing modalities on underwater vehicles [6, 22, 33, 83, 91, 95, 101], in this work I am interested in localization and mapping in smaller-scale environments with resolution on the centimeter level.

In recent years, high-frequency imaging sonars such as the Soundmetrics ARIS and DIDSON [103, 104] and the Tritech Gemini [68] have been developed which have many qualities that are nicely suited to our goal of unmanned exploration, inspection, and mapping of small-scale underwater environments. These acoustic sensors are unaffected by high turbidity and may be used with ranges of anywhere from 1-2 meters to over 50 meters. They can provide images at up to 10-15 frames per second with no restrictions on the vehicle motion or trajectory. The shorter range may also allow for sub-centimeter resolution depending on the configuration. Lastly, imaging sonars compile acoustic returns from a swath of volume with a considerable elevation angle aperture which provides rich, informative images. These sensors have already proven useful for image registration, mosaicing, localization, and 3-D mapping with underwater vehicles, which are all discussed thoroughly in Section 2. For these reasons, I choose to utilize imaging sonar as the primary exteroceptive sensing modality.

1.3 Why localization and mapping?

Localization and mapping have long been considered fundamental components of any mobile or autonomous robotic platform. In this work, mapping is partially an end-goal itself - we would like to generate accurate, high-resolution 3-D maps or surface reconstructions of underwater scenes. These types of maps aid the task of inspection, which is another end goal of this work. Some kind of map is also necessary for exploration, and more specifically for path planning, which is the process of an autonomous robot deciding how to navigate an environment. An accurate state estimate of the vehicle with respect to the sensed environment is crucial both for constructing accurate maps as well as for path planning and collision avoidance.

While there has been some work and discussion in recent years of pushing robots towards “end-to-end learning,” wherein machine learning algorithms are used to map directly from input sensor data to output robot actions, this effort is hotly debated. For the moment, my estimation is that the fundamental algorithms of localization and mapping will be crucial to future generations of mobile and autonomous robots, and choose to take this more traditional approach rather than the pure machine learning approach.

1.4 Imaging sonar sensor

In this section, we will examine the imaging sonar sensor, the standard sensor model, and the challenges associated with using this type of device. The imaging sonar utilizes an array of transducers, which emit relatively high frequency sound waves (hundreds of KHz to several MHz) and detect the sound that is reflected back toward the sensor. The output is an acoustic image of the underwater scene.

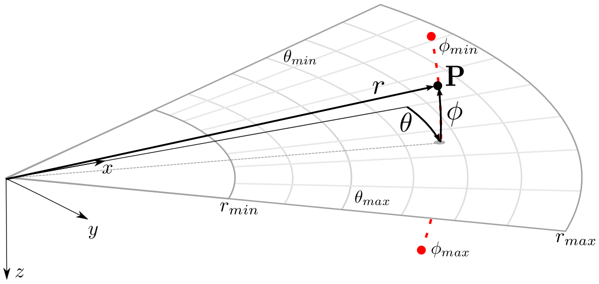


Figure 1.1: The basic imaging sonar sensor model of a point feature. Each pixel provides direct measurements of the bearing / azimuth (θ) and range (r), but the elevation angle (ϕ) is lost in the projection onto the image plane - analogous to the loss of the range in the perspective projection of an optical camera. The imaged volume, called the frustum, is defined by the sensors limits in azimuth $[\theta_{min}, \theta_{max}]$, range $[r_{min}, r_{max}]$, and elevation $[\phi_{min}, \phi_{max}]$.

Imaging sonars are quite analogous to optical cameras in that they provide a 2D image observation of a 3-D environment. I will use a slightly different coordinate frame convention when describing the imaging sonar sensor model as compared to the classic pinhole projective camera model. The x axis points forward from the acoustic center of the sensor, with the y axis pointed to the right and z axis pointed downward, as shown in Figure 1.1. Note that the pinhole camera model typically aligns the z axis in the forward facing direction. The change in coordinates here is largely due to the image formation. Rather than projecting the scene from the viewable frustum into a forward-facing image plane, as in the pinhole camera model, the scene is projected into the zero elevation plane, which is the xy plane in the sonar frame.

Following the notation in [50], consider a point $\mathbf{P}$ with spherical coordinates (r, θ, ϕ) - range, azimuth, and elevation, with respect to the sonar sensor. The corresponding Cartesian coordinates are then

$$\mathbf{P} = \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = \begin{bmatrix} r\cos\theta\cos\phi \\ r\sin\theta\cos\phi \\ r\sin\theta \end{bmatrix}. \tag{1.1}$$

In the projective camera model, the azimuth and elevation angle are measured by the pixel location, but the range is lost in the projection. Every point in 3-D space along the ray corresponding to a given pixel would project to the same pixel location in the image. Surfaces along this ray optically occlude other surfaces that lie along ray at a greater range from the camera, unless the occluding surfaces are transparent. In the imaging sonar sensor model, a 3-D point is projected into the xy image plane as

$$\mathbf{p} = \begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix} = \frac{1}{\cos\phi}\begin{bmatrix} X_s \\ Y_s \end{bmatrix}. \tag{1.2}$$

Note that the elevation angle is entirely lost in the projection, much like the loss of range in the projective camera model. However, instead of each pixel corresponding to a ray that passes through the sensor origin, a pixel corresponds to a finite elevation arc in 3-D space. In Figure 1.1, this is shown by the dotted red line. This nonlinear projection adds significant complexity to the tasks of localization and mapping which is not present in the case of the projective camera. Most imaging sonar sensors have a relatively narrow elevation field of view, anywhere from $1° - 28°$. A wider elevation angle opening provides a richer, more informative view of a larger volume, but makes it more difficult to utilize simplifying assumptions to handle the elevation ambiguity. In fact, concentrator lenses are often used with these sensors to reduce the elevation field of view to just $1°$, in which case it may be reasonable to assume all viewed points lie on the xy imaging plane (zero elevation).

The *intensity* of the imaging sonar pixel measurement is the most complex component of the sonar image formation. For the projective camera, each pixel corresponds to a single surface patch, and the intensity of the pixel is essentially a measurement of the light reflected toward the sensor from that surface patch (this is ignoring rare cases involving translucent surfaces, refractive media, mirrors, etc.). However, for the imaging sonar, each pixel need not correspond to a single surface patch. In fact, any surfaces that happen to lie along the elevation arc may reflect sound emitted by the sensor. Therefore a single pixel in a sonar image may correspond to multiple surfaces. This creates significant complexity when attempting to perform localization or mapping with an imaging sonar.

While the main advantage of using imaging sonar underwater is the ability to penetrate turbid waters, which optical cameras fail to do, this comes with several drawbacks. Here I discuss these drawbacks and difficulties, which are also laid out in great detail in [50, 75]:

- Low resolution: While inexpensive consumer cameras provide images with several millions of pixels, imaging sonars offer a much lower resolution. The SoundMetrics DIDSON [104] that is used in my experiments offers a resolution of 96 azimuthal beams by 512 range bins, for a total of fewer than 50,000 pixels, at a frame rate around 10 fps.
- Inhomogeneous resolution: As shown in Figure 1.1, the footprint of a pixel in the imaging plane increases with its range, effectively decreasing the resolution of the sensor at longer ranges.
- Low signal to noise ratio - High amounts of speckle noise at present throughout the images, like most acoustic sensors. The relatively low operating frequencies of imaging sonars (compared to optical cameras), and the fact that they must emit their own sound rather than simply measure ambient sound, means that a lower SNR is inevitable.
- Acoustic artifacts - Cross-talk between the transducers may induce artificial measurements

in the images. Multi-path artifacts occur when sound is reflected off of multiple surfaces before returning to the transducer, thereby creating a false measurement of a surface at an incorrect range and possibly an incorrect bearing angle as well. On the other hand, surfaces that are at an oblique angle to the emitted sound may not reflect sufficient sound directly back to the sensor, creating the illusion of empty space when there is actually a surface present. Smooth surfaces such as sheet metal or flat plywood tend to reflect sound very specularly, exacerbating this problem.

- Variance across viewpoints - In optical cameras, the intensity of a pixel is determined by the light frequencies and intensities that a surface patch reflects. If the lighting source does not change drastically, these characteristics tend to be rather constant across very different viewpoints of the surface patch. This is the main reason why optical multi-view geometry and structure-from-motion have been such fruitful fields of research. In the sonar case, the pixel intensity is dependent on the surface patch normal angle relative to the sensor, so the contribution of a surface patch to the pixel intensity may vary significantly as the sensor moves to different viewpoints.

Despite these difficulties, many researchers have found it worthwhile to use imaging sonars for localization and mapping, and I seek to further advance these methods in this work.

1.5 Thesis statement and contributions

My thesis statement in this dissertation is as follows:

Despite the physical differences between optical cameras and acoustic sonars as sensing modalities, computer vision approaches to optical localization and mapping may be adapted for imaging sonar localization and mapping.

The contributions of this dissertation to the domain of sonar localization may be summarized as follows:

- A fully degeneracy-aware solution to the problem of acoustic bundle adjustment, or feature-based acoustic SLAM.
- A method for long-term, drift-free localization in a confined underwater environment using a pose graph formulation and incorporating loop closures based on my acoustic bundle adjustment algorithm.
- A joint-compatibility based framework for associating feature cloud matching between two sonar images.

The contributions of this dissertation to the domain of sonar mapping may be summarized as follows:

- A method for generating a pointcloud surface reconstruction from a single sonar image given an initial object edge, in the vein of optical shape from shading, and a framework for fusing multiple surface estimates taken from different viewpoints into a coherent global map.
- Connecting the problem of imaging sonar reconstruction to non-line-of-sight (NLOS) reconstruction for the first time in the literature, which yields the two following novel algorithms for imaging sonar reconstruction.
- A general-purpose imaging sonar scene reconstruction framework based on a volumetric albedo scene model.
- A theory of Fermat paths for imaging sonar reconstruction. This includes a derivation of the 2-D Fermat flow equation and a method of estimating the 3-D locations of particular surface points from short baseline motion.

1.6 Outline of thesis

In **Chapter** 2 I conduct a thorough review of the academic literature that is relevant to the topics in this thesis. First, I discuss algorithms that are related to localization, such as image registration and mosaicing, but which ultimately do not attempt to estimate a 6-DOF sonar pose. Next, I discuss a variety of works best classified as feature-based localization, which use corresponded interest points from different images to solve for a relative transformation, from 3-DOF up to 6-DOF. I also examine works focused on imaging sonar mapping, which are categorized into two groups: (1) reconstruction of sparse feature points and (2) dense 3-D reconstruction. Lastly, I present the non-line-of-sight reconstruction problem and relate it to the problem of imaging sonar reconstruction. This connection is exploited in the algorithms we present in Chapters 6 and 7.

In **Chapter** 3 I describe some of the fundamental models and mathematics that underlie the contributions of this thesis. I present the maximum a posteriori (MAP) estimation framework and the common nonlinear least squares problem which is the backbone of my sparse feature-based localization and mapping framework. I also describe the AUV platform that is used in all of my real-world experiments throughout the remainder of the dissertation.

In **Chapter** 4 I describe the problem of acoustic bundle adjustment, or sparse feature-based SLAM, which may be used as a core component of AUV localization. First, I present my degeneracy-aware algorithm, which takes care to only consider well-constrained information in the solution. Then, I provide a method for incorporating loop closure constraints resulting from the acoustic bundle adjustment optimizations into a pose graph framework for long-term, drift-free localization. These contributions are evaluated rigorously in simulation and on real-world datasets.

In **Chapter** 5 I describe a generative model-based imaging sonar reconstruction algorithm. A method of reconstructing full 3-D pointcloud surface estimates from a single sonar image is described, and a framework for fusing them into a coherent global model is presented. The algorithm is applied to piling inspection in both test tank and real-world environments.

In **Chapter** 6 I present a volumetric albedo framework for imaging sonar mapping. Modest simplifying assumptions allow the problem of imaging sonar reconstruction to be reduced to a linear inverse problem, which may be augmented with a variety of priors and regularization terms. I show how to solve the resulting regularized problems using the alternating direction method of multipliers (ADMM) algorithm. I demonstrate the effectiveness of the proposed approach in simulation and on real-world datasets.

In **Chapter** 7 I present a theory of Fermat paths for imaging sonar reconstruction. I derive the 2-D Fermat flow equation, and prove that it applies to object boundary points and points of specular reflection with in the 2-D cross-sectional plane defined by each image column. A method of reconstructing the full 3-D location of the points from short baseline motion is presented and evaluated on simulated and real-world datasets taken in a test tank environment.

In **Chapter** 8 I discuss the significance of the work presented in this dissertation and how it may be applied. I consider the future of the field of underwater imaging sonar sensing and offer concluding remarks.

Chapter 2

Related works

2.1 Localization

Localization is a broad topic in the robotics community, with many different but overlapping strategies. I attempt to bring some clarity and structure to my review of these methods by distinguishing between: (1) image registration and mosaicing algorithms and (2) algorithms which aim to localize using features, rather than aligning images. I further breakdown the feature-based localization algorithms based on whether or not they assume a mostly planar environment.

2.1.1 Image registration and mosaicing

Image registration is the process of transforming images into a common frame or finding an optimal transformation to overlay the images on top of each other. This is usually practiced on a pair-wise basis, registering a target image to a reference image. As a result of, or during, this process, some type of sensor transformation may also be computed (i.e. translation, rotation, or some combination of the two). Mosaicing refers to the process of creating an large image by superimposing the registered images from a video or image sequence. While these methods do not address the problem of general-purpose 6-DOF imaging sonar localization, they provide valuable insight and partial localization solutions.

Many of the early methods in the domain of sonar image registration extract features using Harris corners [58, 78] or more recently, SIFT [77] or Gabor [123] features. These methods use correspondences of the features to estimate rotation and translation in the imaging plane (xy plane), and typically assume no change in the z translation, roll angle, or pitch angle of the sensor. In [53], the planar scene assumption is also used to solve for a 3-DOF sensor motion between two sonar frames, while simultaneously registering the images. Extracted feature points are assumed

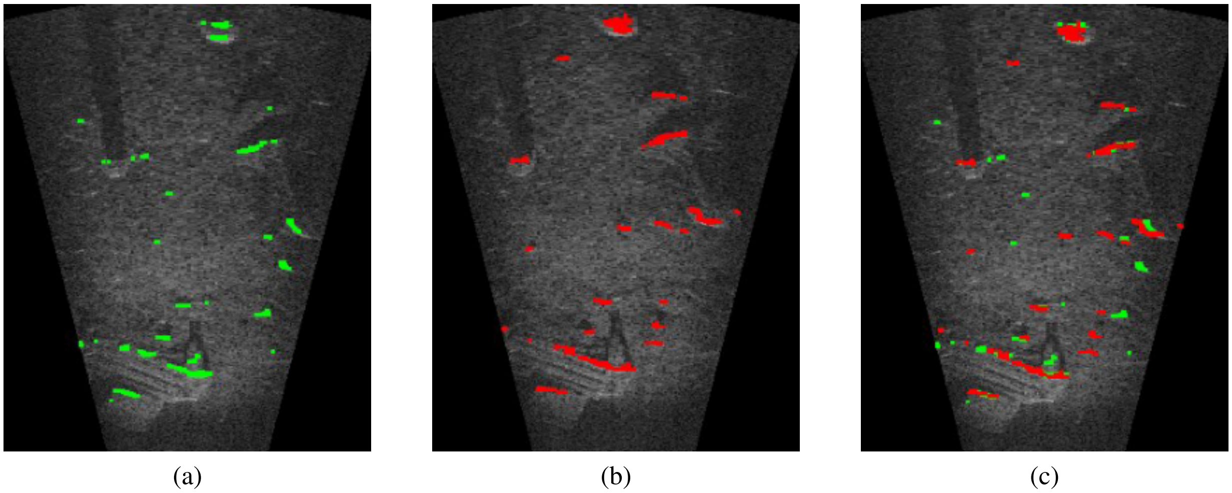


Figure 2.1: Sample images from a sequence using the DIDSON sonar for localization against a flat seafloor. The method in [53] clusters high-gradient pixels in the target image (a) and reference image (b) to be registered to each other. (c) The normal distance transform (NDT) is used to align the images without explicitly associating the features between the two images. Images courtesy of [53].

to lie on a plane, whose normal direction is either assumed to be exactly parallel to the z-axis or estimated by other sensors. Clusters of high-gradient pixels are used with the normal distribution transform (NDT) to iteratively align the two images, as shown in Figure 2.1. While this method provides great flexibility for the frontend as explicit feature correspondences are not required, the relative transformation estimate is only 3-DOF, just like the previously described algorithms.

In a break-away from the feature-based approaches, [50, 51] take registration into the frequency domain, using the Fast Fourier Transform to offer more robust performance in the presence of noise and acoustic artifacts. This allows the construction of mosaics with more detail and higher resolution than individual sonar frames. The authors compared this approach to other state-of-the-art methods and demonstrated the increased accuracy and robustness that the Fourier-based method has over feature-based algorithms [49].

More recently, a novel approach to image registration and planar motion estimation attempts to estimate the optical flow between consecutive frames [43, 44]. This work utilizes a two-stage, coarse to fine alignment pipeline to estimate a pixel displacement map (DM). The fine DM estimation stage is based on adaptive filtering on a Hilbert space-filling curve and is capable of making subpixel refinement to the DM estimate. Once the final DM is estimated, a sensor motion estimate is computed using statistics from the DM and some pre-defined “expected motion.” This method seems to set the standard for state-of-the-art performance for imaging sonar registration and mosaicing, as validated by extensive experimental results. However, all of these image registration and mosaicing methods are targeted for scenarios where the motion is entirely or dominantly planar, and where the imaging target is largely planar. While these criteria hold very well for cases of imaging seafloors or flat portions of ship-hulls, they do not generalize to non-planar motions or environments.

2.1.2 Feature-based localization - planar case

There is certainly quite a bit of overlap between algorithms in the domains of image registration and general sonar localization. In the previous subsection, I described algorithms that either do not consider the elevation ambiguity of the sonar sensor model, or take a more pure image processing approach to the problem of image alignment. In this subsection, I consider works that aim to estimate the 3-D motion of the sensor, but still utilize the planar scene assumption.

One of the early works in this direction was [98], which considered two-view acoustic homography. This work is primarily concerned with the "backend" – the estimation of sensor motion based on detected and corresponded points that are taken as a given by some "frontend" module. The points are assumed to lie on a plane, whose normal vector is jointly optimized with the relative rotation and translation between the two sensor poses. While this iterative, nonlinear-least squares solution is similar to the bundle adjustment / ASFM framework that my proposed method is built upon, it suffers from several sources of error. First, the planar approximation introduces error as the detected features rarely lie on a truly flat surface in real-world applications. Second, the proposed framework optimizes only over the pose and surface normal parameters – it does not explicitly include the 3-D landmark positions in the state of the optimization. Additionally, it does not consider the degeneracies of the optimization, instead relying on Levenberg-Marquardt to converge to a solution despite the possibility that the system may be poorly constrained.

Further improvements in sonar pose estimation using the planar assumption were made [8, 9]. These works correspond features detected on objects themselves and associate them with the points at which the objects shadows are cast upon the seafloor. This correspondence provides additional constraints with which to estimate the sensor motion and disambiguate the elevation angle of the object features. A novel Gaussian cluster map is also proposed for the frontend feature extraction and association, which demonstrates improvement over the NDT-based map representation. In [75], a thorough mathematical analysis is provided on the image flow of 3-D objects and the shadows they cast on a flat seafloor. Various 3-D motion estimation methods are proposed, including a 4-DOF motion estimation algorithm, a full 6-DOF motion estimation algorithm, and simpler solutions for the special cases of pure rotation and translation. However, these algorithms all rely on the detection of object-shadow pairs detected in a seafloor mapping scenario or otherwise planar environment.

Various methods of acoustic motion estimation and were surveyed in [5] including Acoustic Stereo Imaging (ASI). The author describes ASI as a system in which two multibeam imaging sonars are fixed in an axis-aligned configuration with an offset in the z-axis (i.e. one mounted some known vertical distance from the second). This method is limited to utilizing features that are within the viewable frustum of both sensors and is not applicable to scenarios where the use of two sensors is impractical.

To the best of my knowledge, the term “bundle adjustment" was first applied in the context of imaging sonar SLAM in [100]. This work presented two contributions to the realm of feature-based imaging sonar SLAM. The first contribution was to apply A-KAZE [4] for the task of feature detection. A-KAZE utilizes a nonlinear scale space for anisotropic diffusion. In contrast to Gaussian diffusion, which is often used to eliminate high frequency noise, the anisotropic diffusion spreads out intensity with inverse proportion to the image gradient. This preserves boundaries in the image that are blurred by Gaussian diffusion. This is particularly helpful for sonar images, which are subject to high amounts of speckle noise. In this work, the A-KAZE features were matched using a RANSAC-based method for robustness. The other main contribution of this work was a two-view bundle adjustment optimization. The authors target this method for seafloor mapping, and assume (1) the features are detected on the seafloor plane and (2) the elevation angle of the detected features is zero. Using these assumptions, they perform a non-linear least squares optimization, in which they optimize the relative transformation between the two views and the seafloor plane’s normal angle using the reprojection error of the feature points. This simplified model is then used in a pose-graph framework to incorporate loop closure constraints for long-term localization. In contrast to this simplified bundle adjustment framework, my proposed solution does not assume that features lie on a planar surface, takes care to address the degeneracy of the optimization, and only incorporates information from the well-constrained directions into the pose graph.

2.1.3 Feature-based localization - general case

Finally, I address approaches to imaging sonar localization that do not use the planar scene assumption or place constraints on the sensor motion.

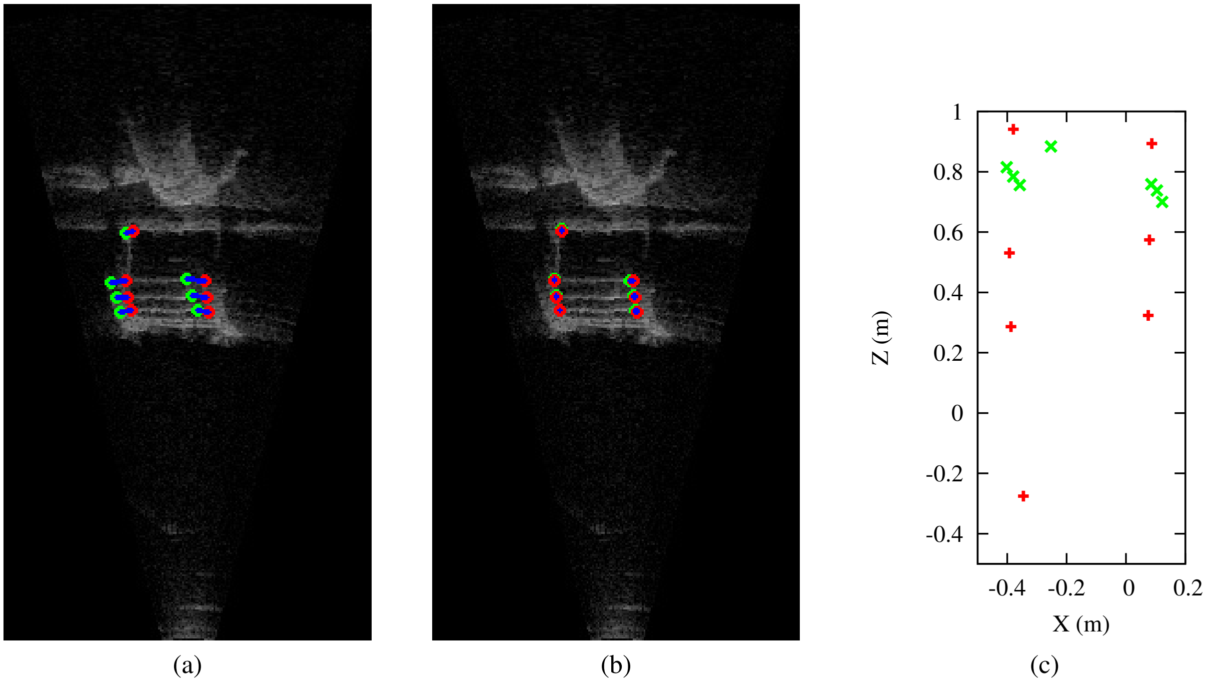
The acoustic structure from motion (ASFM) algorithm [46, 48] introduced the bundle adjustment framework common in the visual SLAM literature to the problem of underwater acoustic SLAM. Similar to the case of visual SLAM, ASFM optimizes a nonlinear least squares objective function based on the reprojection error of feature points in 2D acoustic images. The proposed formulation optimizes over both the sensor poses and the observed 3-D landmark locations. This backend optimizer may be used with any frontend module that extracts and corresponds features across different sonar frames. The original work was focused on the reconstruction of sparse feature points and not sensor localization. In this work, I build upon this ASFM framework with a focus on sensor localization, analyzing in depth the structure of the acoustic bundle adjustment problem, its inherent degeneracies, and provide a more accurate solution.

ASFM has recently been applied to ship hull mapping, in which common features observed over a group of three frames (a "clique") are used to perform a local ASFM optimization [67]. The result of the optimization is used to generate pose-to-pose constraints that are fused with vehicle odometry in a pose-graph framework. This work uses saliency-aware, high-dimensional, learned features [66] for detecting potential high information gain loop-closure cliques to optimize using ASFM. The purpose of this framework is two-fold: (1) to generate loop closure cliques that provide well-constrained systems for the ASFM optimization and (2) to only include sonar-based loop closure constraints that add significant information to the overall SLAM problem. While this work provides a robust frontend for detecting sonar loop closure candidates, it does not explicitly consider the inherent degeneracies of the ASFM optimization, and in doing so discards loop closure cases which may be able to provide valuable constraints to the overall SLAM solution. Additionally, it double counts the vehicle odometry measurements, as they are used in the overall pose graph as well as in the local ASFM optimizations.

Similar to the ASFM algorithm, [120] proposed an acoustic localization algorithm that fuses constraints from feature measurements with measurements from an inertial sensor in an extended Kalman filter (EKF) framework using stochastic cloning. This work also provides a linear triangulation method for initializing the 3-D positions of point landmarks from multiple viewpoints. The observability analysis of these triangulation equations provides insight into the degenerate directions of sensor motion, but the degeneracy is not explicitly handled in the proposed framework.

2.2 Mapping

In discussing previous imaging sonar mapping algorithms, we will distinguish between methods aimed at mapping sparse feature points and methods aimed at densely reconstructing surfaces of entire objects or scenes.



(a)

(b)

(c)

Figure 2.2: The ladder dataset used to demonstrate the 3-D mapping capabilities of ASFM. (a) The red circles show the feature observations in one of the five frames that comprise the dataset. The initial estimates of the 3-D landmarks are projected into the image and shown in green. (b) After optimization, the estimated 3-D landmarks results in much lower reprojection error. (c) A frontal view of the reconstructed landmarks, where one is taken from each side of one of three steps on the ladder. Images courtesy of [48].

2.2.1 Sparse mapping

An early foray into the area of general purpose (non-planar) sparse mapping proposed detecting corner points that lie on contours in sonar images [16] and solving for the elevation angle and relative pose transformation via an evolutionary algorithm [15]. In contrast to my proposed nonlinear least squares optimization, the evolutionary optimization is sampling based, and therefore may converge to a result with a lower objective function cost, especially if the initial estimate for my gradient-based approach is inaccurate. However, this is also a fallback of the evolutionary method, in that the sampling-based approach does not account for the inherent degeneracies of the sonar sensor model and is more likely to find an incorrect solution that overfits the solution to noise in the data. Moreover, since I seek to deploy my sonar sensor on underwater vehicles that are usually equipped with navigation equipment, it is very often the case that a good initial pose estimate will be available, increasing the appeal of my gradient-based method.

Another method by Folkesson et al. [35] sought to track and map features from imaging sonar sequences to improve large-scale navigation, as our proposed degeneracy-aware bundle adjustment algorithm. The authors acknowledge that the Gaussian parameterization is a poor approximation for the elevation angle of a feature point, which ought to be modeled as a uniform distribution. To approximate the uniform distribution, they introduce a change of variables, modeling a cubic function of the elevation angle rather than the elevation itself. This allows for maintaining a Gaussian parameterization that is suitable for their filtering-based navigation framework, while flattening the distribution of the elevation angle to better approximate the uniform distribution.

To the best of my knowledge, the first work to utilize an acoustic bundle adjustment framework to reconstruct sparse 3-D points from multiple imaging sonar frames was the ASFM algorithm [46, 48], which I first introduced in Section 2.1. In contrast to the plane-based bundle adjustment proposed in [100], ASFM directly optimizes over the full 3-D positions of the point features. The effectiveness of the ASFM algorithm in recovering the 3-D positions of landmarks was demonstrated with both simulated data and real-world sonar images using manually-selected feature points. This original work did note that there are quite a few types of sensor motion that are *degenerate*, in that the geometry of the multiple measurements of a feature point do not provide good constraints on its 3-D location. However, this was not explored in more depth in the original work. Nevertheless, the original work demonstrated the ability of ASFM to recover the 3-D positions of landmarks with both simulated data and real-world sonar images using manually-selected feature points. Figure 2.2 shows a sample image from the sequence, manually selected and associated features which lie on the rungs of an underwater ladder, the locations of the reprojected landmarks before and after optimization, and a view of the 3-D reconstructed points.

Several improvements to the original ASFM algorithm were made in subsequent years. A somewhat different line of work used the same reprojection error of features from multiple viewpoints to optimize for their 3-D positions, but advanced the process by implementing an automatic feature extraction and association method [52, 61]. However, these methods were highly tailored for detecting corner features on a particular object of interest that was mapped, and are not entirely suitable for general purpose underwater scenes. A follow-up work to ASFM implemented a brute-force search, automatic data association algorithm based on chi-squared confidence intervals on the reprojection error of features in the acoustic images [47]. This method still relied on manually extracted features, though. Both of these data association algorithm are essentially obsoleted by my proposed joint-compatibility data association algorithm, which uses the same reprojection-error based chi-squared tests, but is more robust than an individual association matching algorithm (as in [52]) and faster than a full brute-force search (as in [47]).

The same type of optimization based on reprojection of sparse feature points across multiple images was presented in the framework of an Extended Kalman Filter (EKF) in [70]. This is in contrast to the batch, offline optimization proposed by the original ASFM algorithm. This method also relies on manually selected and corresponded feature points, does not account for degeneracy in the EKF updates, and assumes some kind of odometry measurements or pose priors are available. The authors then extended this approach to consider the interesting case of reconstructing line segment features using an EKF framework [71]. This may be most useful when operating in man-made environments where precisely straight lines features may be present. This algorithm performs automatic extraction and association of line segment features between images, assuming a pose prior is provided. While I do not explore line features in this work, it is certainly an interesting line of research for future investigation.

In this work, I focus on improving the accuracy of both the localization and mapping result of the ASFM formulation of acoustic bundle adjustment. This has been previously published in two independent papers investigating a non-parametric treatment of the detected point landmarks to handle degeneracy of the elevation angle [117] and a degeneracy-aware nonlinear least squares optimization to handle the degeneracy of the relative pose transformation in the bundle adjustment [118]. My fully degeneracy-aware solution to acoustic bundle adjustment offers improvements over previous state-of-the-art methods for full 6-DOF localization using imaging sonar, although the lack of open source implementations of most works prohibits direct comparison.

2.2.2 Dense mapping and surface reconstruction

While various algorithms have been developed for monocular optical cameras which can simultaneously reconstruct a dense scene model while tracking the camera motion without the aid of other sensors ([81, 92]), this type of algorithm seems quite far off, if not entirely infeasible, for underwater imaging sonar mapping due to the sensor's relatively low information content, low SNR, and complex image formation process. Considering these challenges, most of the research on dense 3-D mapping and surface reconstruction using imaging sonar has focused on the simpler case of *mapping with known poses*, wherein the sensors poses are assumed to be precisely known, either due to the sonar being operated in a highly controlled laboratory environment or by utilizing other high-precision sensors for localization. Additionally, many methods that have been developed are targeted for more narrowly defined operating conditions or utilize simplifying assumptions. In this section, I will try to bring some more clarity and structure to the discussion by classifying the relevant works into several categories: (1) seafloor mapping (2) space carving (3) linear approximations and (4) generative image formation models.

Seafloor mapping

Some of the earliest works to venture into the area of dense surface reconstruction with known poses began more than two decades ago. A two stage approach to reconstructing an object on the seafloor was proposed in [121]. First, an height map (along the z-axis) was estimated by relating shadow contours with their corresponding occluding edges. Then a reflection map was computed using computerized tomography (CT) by scanning the object of interest in a circular trajectory. The height map and reflection map are then combined to generate the 3-D object model. While this method has been superseded by newer reconstruction algorithms, it was one of the first to introduce the idea of utilizing cast shadows to help disambiguate the missing elevation angle in the sonar images, which remains a dominant theme in modern techniques.

A similar method of estimating the height or elevation angle for seafloor mapping was recently implemented on a mobile AUV platform [21]. Using the standard seafloor mapping configuration with the sonar pointed downward at a slight angle, the vehicle uses a forward moving lawnmower style trajectory. The authors use several image processing steps to extract the so-called "highlight extension," the high-intensity portion of the image corresponding to returns from the object, which is directly in front of the acoustic shadow cast by the object on the seafloor. By assuming the top of the objects causes the acoustic backscatter of the highlight extension, the height of the object at the leading edge of the highlight extension may be easily computed. Thus, each individual sonar image results in a single 3-D height-map line scan. The height maps from different images are combined in a single model using the vehicle's on-board navigation system, which in this work relied on a DVL. The main drawbacks of this method are that it generates very little 3-D information from each sonar image, restricts the vehicle motion to mostly forward-moving, and assumes a relatively simple object geometry by estimating a 2.5-D height map, rather than a full 3-D surface reconstruction.

Space carving

Within the past several years, several novel dense 3-D mapping algorithms were presented that proposed different implementations of the same general framework for 3-D reconstruction: space carving. Intuitively, these methods assume that all space is initially "occupied," and successively carve away "free space" volume based on the "negative information" or low-intensity regions of the sonar images. The remaining volume is taken to be a conservative estimate of the volume occupied by objects.

The first method that I will discuss focused creating a binary image from each input sonar image, called the feasible object region mask (or FORM image) [7]. In each column of a polar coordinate sonar image (corresponding to a single azimuth beam), the leading edge of the FORM starts with the first pixel with an intensity above some threshold. This and all the pixels with a greater range and the same azimuth angle are set to 1, which indicates that all of the volume corresponding to those pixels *may or may not* be occupied by an object. The rest of the image, which is all of the pixels from the minimum range up to the leading edge is set to 0, which indicates that all of the imaged volume corresponding to those pixels *must* be free space. In some of the results presented, the leading edges of the FORM region were manually selected or edited. The FORM is projected along the entire elevation arc into a voxel grid, which stores the binary occupied / free space value for each voxel. The authors propose utilizing alpha shapes to define the 3-D contours of the object of interest. When alpha shape constraints are generated from a sufficient variety of viewpoints, they define an approximate surface mesh model of the object or scene of interest, which theoretically ought to contain the entire object of interest.

Impressive results reconstructing relatively small objects underwater were presented using this alpha shape space carving method in [7, 11]. While these results qualified as the state of the art at the time of publication, the proposed method has significant limitations. The biggest drawback of this algorithm is its non-probabilistic nature. Each sonar image defines 3-D volume that is permanently labeled "free space." If there were any errors in detecting the leading object edge, or inaccuracies in the sensor pose, then free space may be incorrectly carved from the model. Repeated or redundant measurements are unable to correct erroneous carvings. For this reason, the FORM images computed from real data are generally manually generated, corrected, or examined. Additionally, relying on the binary FORM image discards most of the information provided in the sonar image. This ensures that a large number of images from a large variety of viewpoints are required in order to generate a reasonable model. Furthermore, the assumption that dark pixels with low intensity correspond to free space may not hold true. My own experiments demonstrate that when an object's surface normal is at a particularly large angle to the incidence of the acoustic waves from the sonar, and the object's surface is sufficiently smooth, then the reflection will be mostly *specular* rather than *diffuse*, and the object will not register in the sonar image. This may be seen in Figure 4.7, where a plate of aluminum sheet metal is imaged from the side. Although the sheet metal lies within the viewable frustum of the sensor, only the magnets that are mounted on the sheet register on the sensor.

The second space carving method offers a different implementation, but is grounded on the same principles. The authors propose utilizing a voxel grid, implemented as an octree, to model the occupied space of the object(s), rather than surface-based alpha shapes [39, 40]. In each sonar image, every pixel is associated with the set of voxels which intersect with the pixel's elevation arc. Each voxel simply records the *minimum sonar image intensity* that is has ever been associated with. The voxel grid may simply be thresholded to determine the set of voxels that are occupied. After the entire sequence, an occlusion resolution step is performed to remove interior points, ideally leaving just the voxels that correspond to the exterior surface of the object. Like the first space carving method, this algorithm is based on the assumption that low intensity or dark pixels correspond to free space, but it is somewhat more flexible in that there exists an adjustable threshold that controls the sensitivity of the method to the pixel intensity. While the voxel grid requires large amounts of memory for high-resolution or large-scale maps, it allows for a straight-forward implementation of space carving. Overall, the results are quite comparable to [7, 11], although direct quantitative comparison has not been performed to the best of my knowledge.

The space carving approach to 3-D reconstruction will generally yield a conservative estimate of the occupied regions of a scene or object. That is, the actual object ought to be entirely contained by the estimated occupied volume. This is well suited to convex-shaped objects but makes it difficult to accurately reconstruct non-convex shapes. If most of the volume corresponding to an elevation arc is free space, but just a small portion is occupied and results in a high intensity pixel measurement, the entire volume corresponding to the elevation arc is not carved. Therefore, if the geometry of the object does not allow entire elevation arcs to fit into hollow spaces, space carving will not be able to accurately reconstruct these concave sections of objects. This is a fundamental limitation of the space carving approach, which we seek to move beyond in this work.

Occupancy grid mapping

A line of work that utilizes volumetric representations reframes the classical occupancy grid mapping framework [31] for the imaging sonar sensor model. An inverse sensor model is used to update all voxels that project into an image pixel with some occupancy measure that is a function of the pixel intensity [63, 65, 112]. By accumulating measurements from many redundant images, particularly by rolling the sensor along its viewing axis, a 3-D occupancy model of the object is constructed. This general approach will be less effective at carving out free space, since it is akin to averaging the intensity value over multiple measurements rather than taking the minimum over all observations. However, it is less sensitive to geometrical errors, such as inaccuracies in the sensor pose, which may be further refined by aligning submaps built from different viewpoints [113].

Generative models

Perhaps the most fundamental question about any sensor is: how, exactly, are the measurements generated? Answering this is essentially the main objective of an imaging sonar simulator - to generate an accurate, realistic sonar image given the sensor position and a model of its environment. Although an image formation model may be developed for purposes other than simulating images, I will use the terms "image formation model" and "sonar simulator" interchangeably.

Various imaging sonar simulators have been proposed in recent years [19, 29, 38, 62, 72, 96]. Most rely on some form of ray-tracing to associate image pixels with the scene surfaces that they image. These simulators vary quite a bit in how they model the generated pixel intensity based on the imaged surfaces, including different aspects of the physical models underlying the propagation of acoustic waves underwater as well as various models of surface reflectance. Some even model the cross-talk that occurs between beams and multipath effects that occur due to echoing [38]. Unfortunately, most of these simulators are quite slow, requiring several minutes to generate a single simulated image of a scene. In contrast, a recent simulator utilizes precomputed shading information and GPU acceleration to generate realistic images in real-time [20, 19].

For underwater scenes imaged by sonars, surfaces are sometimes assumed to reflect sound perfectly diffusely, and may therefore be modeled using Lambertian reflectance. One line of work that is of particular interest to us considers various models of diffuse Lambertian reflection, including one in which the image pixel intensity is

$$I_s = k \cos^m (\alpha) \tag{2.1}$$

where k is a normalization constant, α is the angle of incidence between the surface normal and direction of propagation of the acoustic beam, and $1 \leq m \leq 2$ [10]. The authors derive a slightly different formulation for the pixel intensity based on reflection from a single surface patch:

$$I_s = k \cos \alpha \, \cos \phi \, \delta \phi \tag{2.2}$$

where $\delta \phi$ is the elevation angle width that covers the differential surface patch δA. Since the authors' experiments are performed with a DIDSON sensor that has a $14°$ elevation field of view ($\pm 7°$), they approximate $\cos \phi \approx 1$ and use the simplified model

$$I_s = k \cos \alpha \, \delta \phi \tag{2.3}$$

although this becomes less suitable for wider-aperture configurations. This model is extended to account for the effect of the finite length of the acoustic pulses transmitted by the sonar, which becomes considerable when short window lengths are used for high-resolution imaging. Since the small objects of interest are imaged while resting on the test tank floor, they also account for several types of multipath trajectories in their model that occur due to reflections between the object and seafloor. In evaluating their model, the authors approximate the ensemble average of pixel measurements by averaging the values of entire nearby azimuthal beams in one or multiple images. This is done to account for the speckle noise, unknown phase shifts, and other sources of error that make individual sonar images quite noisy. The various proposed models are evaluated against the ensemble averages, with each additional modification to their proposed method (accounting for finite pulse length and multipath trajectories) reducing the error further [12]. The results are some of the most impressive in terms of accurately modeling the shading of sonar images, although the presented experiments were limited to imaging simple spherical and cylindrical shapes.

The same line of work proposed an algorithm for reconstructing a 3-D object from a single image (or using ensemble averages over adjacent azimuth beams or images) using any of the above pixel intensity models [10]. The algorithm requires initialization of 3-D points at the leading and occluding edges of the object, which were also spheres, cylinders, and concave semi-cylinders. These were obtained either through the shadows cast on the test tank floor [10], or ground-truth models. Then, the interior points on the object surface are iteratively solved for using the pixel shading model. This provides a shape-from-shading (SfS) type of approach to single-view sonar reconstruction, although there remain ambiguities that must be resolved by some other cues (such as acoustic shadows). Optical SfS can generate highly accurate 3-D reconstructions up to scale, due to the loss of range in perspective projection. For the acoustic case, the elevation ambiguity and symmetry of the image formation model means that the vertical orientation of an object will be subject to ambiguity without any prior constraints or information. This general SfS-style approach to inferring a 3-D model from a single image will be explored more in Chapter 5.

A recent work attempts to use this generative image shading model to refine an estimated 3-D object model [79]. The authors propose using the object surface model resulting from their space carving method [7, 11] as an initial estimate for a subsequent optimization that attempts to enforce consistency between the sonar image formation model and the actual measured data. They propose a simple formulation that optimizes over the 3-D point locations P_i of surface patches and an intensity scaling parameter k in a nonlinear least squares framework:

$$k^*, \{P_i^*\} = \underset{k,\{P_i\}}{\operatorname{argmin}} \sum_i \left(I_s\left(k, P_i\right) - \hat{I} \right) + \lambda_o \left\| \bar{I}_s\left(k, P_i\right) - \bar{\hat{I}} \right\| \tag{2.4}$$

where the second term is a regularization term with weighting λ_o that enforces consistency between the average predicted and measured intensities. They propose and evaluate this cost function as well as modified cost functions that take penalize sharp changes in surface normals between neighboring surface patches. The simulated results demonstrate some improvements in the overall surface reconstruction as a result of applying this optimization, although it seems to be quite sensitive to inaccuracies in the initial estimate and liable to get stuck in local minima.

Linear approximations

In [40], the authors utilize the Lambertian reflection model to characterize the image intensity I_s at pixel coordinates (r, θ) by the integral of acoustic intensities reflected by surfaces back toward the sensor over the elevation aperture $\phi_1 < \phi < \phi_2$:

$$I_s\left(r, \theta\right) = \int_{\phi_1}^{\phi_2} \beta\left(\phi\right) V_s\left(r, \theta, \phi\right) \frac{\mathbf{v} \cdot \mathbf{n}_{r\theta\phi}}{\|\mathbf{v}\| \, \|\mathbf{n}_{r\theta\phi}\|} d\phi. \tag{2.5}$$

Here $\beta\left(\phi\right)$ is the beam pattern which describes the angular distribution of energy emitted by the sensor, $V_s\left(r, \theta, \phi\right)$ is the albedo of the surface at $\left(r, \theta, \phi\right)$, and the last term in the integral is the cosine of the angle between the surface normal $\mathbf{n}_{r\theta\phi}$ and direction of propagation of the acoustic beam $\mathbf{v}$. This formulation is very similar to the Lambertian model presented in 2.1 with $m = 1$.

The authors approximate the integral over the elevation arc as an integral over the u-axis, which is defined as parallel to the z-axis but centered on the pixel of interest (r, θ). Furthermore, they assume the sensor trajectory consists solely of translation along the z-axis. Then, the image formation model as a function of the z position of the sensor may be formulated as a convolution

$$I_{r\theta}\left(z\right)=\int_{z-\Delta z_r}^{z+\Delta z_r}\beta_r\left(u-z\right)V_{r\theta}\left(u\right)\rho\left(z,u,\mathbf{n}_{r\theta u}\right)du \tag{2.6}$$

where $\Delta z_r \approx r\tan\frac{\phi_2-\phi_1}{2}$, $\beta_r$is the warped beam pattern at range r, ρ is a general reflectance function that replaces the cosine term, and $V_{r\theta}\left(u\right)$ is an indicator function that represents the existence of a surface at the corresponding point. Since the reflectance function requires knowledge of the surface geometry, solving for $V_{r\theta}\left(u\right)$ is a blind deconvolution with a spatially-varying kernel. The authors set the reflectance function to unity for all surfaces and discretize the u-axis in order to formulate the convolution as a sparse linear system, which may be solved using a non-negative least squares solver. This method was evaluated in simulation as well as in test tank experiments, using a plotter head to control the exact position of the sensor, and field experiments, where the sensor was mounted on an autonomous underwater inspection vehicle with sensor poses provided by a suite of navigation sensors, including a DVL, gyroscope, and compass. The reconstruction results on comparable with recent space carving methods.

While this approach may be considered one of the state-of-the-art 3-D reconstruction techniques for imaging sonar, it suffers several limitations and sources of error. Most notably, the sensor trajectory is limited to pure z-translation, which is difficult to achieve with an underwater vehicle without introducing motion in the other five degrees of freedom, which detracts from the accuracy of the reconstruction. Using the u-axis to approximate the elevation arc also introduces errors that increase significantly with the size of the elevation aperture, making this method more appropriate for narrow aperture imaging sonars.

2.2.3 Non-line-of-sight reconstruction

This dissertation, along with my previous work [115, 116] mark the first time that imaging sonar reconstruction is connected with non-line-of-sight (NLOS) reconstruction. In this section, we introduce the problem of NLOS reconstruction and discuss its relation to imaging sonar mapping.

NLOS reconstruction refers to the problem of imaging and reconstructing a scene by active but indirect illumination. This may take the form of transmissive observation (looking through a diffuser) or reflective observation (looking around a corner). We will consider the case of reflective observation in this work, although both problems may be formulated equivalently. Similarly, while a variety of illumination and sensing techniques may be utilized, we will consider a setup that uses a laser light emitter and a single-photo avalanche diode (SPAD) detector to measure reflected light.

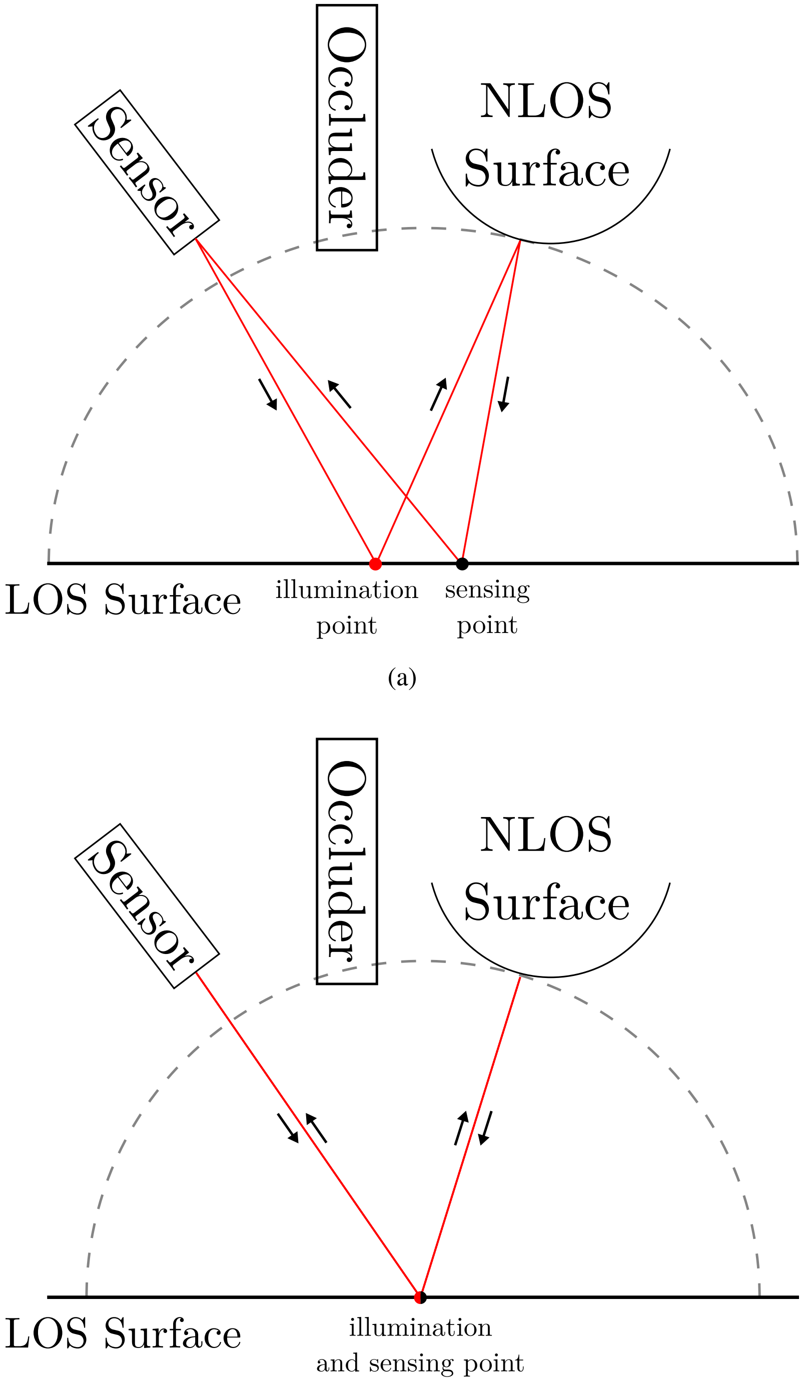
Figure 2.3a provides a visualization of a typical set-up for generic NLOS imaging. The main components of the setup are the light emitter and sensor (represented as a single transceiver here), the line-of-sight (LOS) surface, the non-line-of-sight surface, and some occluding barrier that prevents direct line-of-sight from the sensor to the NLOS surface. Planar walls, which reflect light rather diffusely, are typically used as the LOS surface in experiments, and gives rise to the phrase "looking around corners" with regards to reflective observation. All surfaces are assumed to reflect and scatter light in all directions.

To generate a measurement, a pulse of light is emitted, hitting a particular point of illumination on the LOS surface. The light is reflected off the LOS surface and scattered. We follow one branch of the scattered light which then interacts with the NLOS surface at a particular point, being reflected and scattered again. Finally, this light hits the LOS surface at a variety of points and scatters for the third time. However, only light reflected off of the particular sensing point back towards the sensor is measured. Due to the three scattering bounces, this is a very small amount of light and requires very sensitive sensors, such as a SPAD. The resulting measurement is called the light transient, which is a measure of the intensity of light detected from the sensing point as a function of time. Varying the 2-D illumination and sensing points gives rise to the so-called 5-D light transient.

The point of active illumination and the sensing point at which returning light is measured are generally different points, as shown in Figure 2.3a. Due to the known speed of light and positions of the sensor and LOS surface, the time measurements may be transformed to range measurements and the distances between the sensor and LOS illumination and sensing points removed from the transient. Then, the problem of reconstructing the 3-D NLOS scene may be viewed as one of ellipsoidal tomography. This is due to the fact that each range measurement is constrained to lie on the ellipsoid induced by the modified range measurement and the illumination and sensing points as foci. Confocal imaging refers to the scenario when the illumination and sensing points are identical, as shown in Figure 2.3b. This results in a 3-D transient rather than 5-D, and reduces the problem to one of spherical tomography. Additional details on the NLOS problem may be found in [3, 17, 108, 110, 119].

Connection to imaging sonar reconstruction

Confocal NLOS imaging may be thought of as a using virtual range-only sensor at the illumination and sensing point on the LOS surface, which both emits and detects light reflected off the NLOS surface. The intensity of detected light is obviously lower in confocal NLOS imaging due to the reflections off the LOS surface, but the properly compensated range measurements are



(a)

(b)

Figure 2.3: (a) The general NLOS scenario that measures three bounce reflections. (b) The confocal NLOS scenario, in which the illumination and sensing points on the LOS surface are the same.

the same. This means that confocal NLOS reconstruction is equivalent to reconstructing a 3-D model using a 1-D, active, range-only sensor. Similarly, imaging sonar reconstruction uses a 2-D, active, range and azimuth sensor to reconstruct a 3-D model. In contrast to NLOS reconstruction, which attempts to recover the missing azimuth and elevation information, imaging sonar reconstruction need only recover the missing elevation angle, as the 1-D array of transducers already disambiguates the azimuth angle. This connection between these problems is exploited in the last two algorithms proposed in this dissertation for imaging sonar reconstruction: volumetric albedo and Fermat paths, which are presented in Chapters 6 and 7, respectively.

Chapter 3

Preliminaries

3.1 Notation

Throughout the rest of this document, I generally adhere to the following notation in my mathematical equations. In general, matrices shall be denoted with bold, capital letters (e.g. $\mathbf{A}$), vectors shall be denoted with bold, lower-case letters (e.g. $\mathbf{v}$), and scalars with non-bold lower-case letters (e.g. c). Sets of variables may be denoted with a non-bold capital letter (e.g. X) or calligraphic capital letter (e.g. $\mathcal{X}$). One major exception to this rule the 6-DOF pose variable, which will generally be denoted as x_i, with an appropriate subscript.

The notation in each Chapter will generally be distinct and will not refer to the notation used in a previous Chapter, except when explicitly stated otherwise.

3.2 Maximum a posteriori estimation

In this section I describe the maximum a posteriori (MAP) estimation framework that underlies my formulation of acoustic bundle adjustment and factor graph SLAM. I also derive the nonlinear least squares (NLS) optimization that is used to solve the MAP estimation problem.

MAP estimation attempts to find the most likely state X of the modeled system given a set of measurements $Z = \{\mathbf{z}_1, \dots \mathbf{z}_N\}$. I represent this type of optimization graphically using *factor graphs*, as in Figure 3.1. Using this representation, large, clear circles are *nodes*, which represent the *state variables* X to be optimized. Small, colored circles are *factors*, which represent the *measurements* Z which constrain the variables to which they are connected. In this derivation, I assume that all measurements states are vector-valued. Section 3.3 gives an exposition on how I treat 6-DOF poses in this framework.

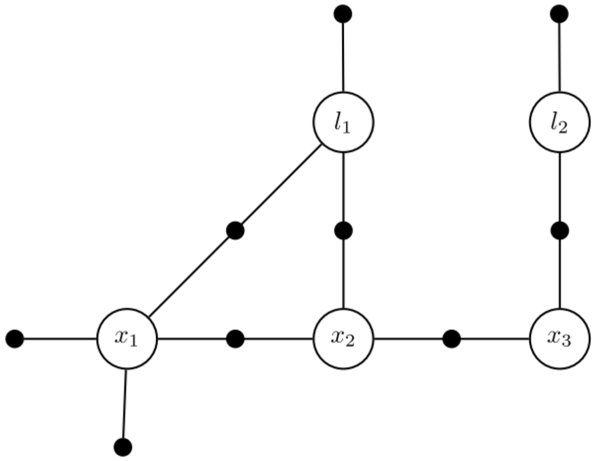


Figure 3.1: Factor graph representation of a toy SLAM problem. Here the state X is comprised of poses x_1, x_2, and x_3 as well as landmarks l_1 and l_2. The measurements are represented by the smaller black vertices and are not labeled for simplicity. Figure courtesy of [26].

I start the derivation by maximizing the posterior of the state:

$$\begin{aligned} X^* &= \underset{X}{\operatorname{argmax}}\; p\left(X|Z\right) && (3.1)\\ &= \underset{X}{\operatorname{argmax}}\; p\left(X\right)p\left(Z|X\right) && (3.2)\\ &= \underset{X}{\operatorname{argmax}}\; p\left(X\right)l\left(X;Z\right) && (3.3)\\ &= \underset{X}{\operatorname{argmax}}\; p\left(X\right)\prod_{i=1}^{N} l\left(X;\mathbf{z}_i\right) && (3.4)\end{aligned}$$

where $l\left(X;Z\right) \propto P\left(Z|X\right)$ is the likelihood of the state X given the measurements Z, and likewise for an individual measurement $\mathbf{z}_i$. Here I assume conditional independence of measurements, which is encoded in the connectivity of the factor graph. Note that although I use the notation $p\left(z_i|X\right)$, the measurement $\mathbf{z}_i$ is only conditioned on the subset of variables from the state X to which it is connected in the factor graph. If there is no prior knowledge of the state, which I will assume here for simplicity, $p\left(X\right)$ may be dropped. As is standard in the SLAM literature, I assume additive Gaussian noise in the measurement model:

$$l(X;\mathbf{z}_i) \;\propto\; \exp\left\{-\frac{1}{2}\left\|h_i(X)-\mathbf{z}_i\right\|^2_{\mathbf{\Sigma}_i}\right\} \qquad (3.5)$$

Here $h_i\left(X\right)$ is the *prediction function*, which predicts a value $\hat{\mathbf{z}}_i$ of the measurement $\mathbf{z}_i$ based on the state estimate X. The covariance matrix $\mathbf{\Sigma}_i$ represents the uncertainty of the measurement

$\mathbf{z}_i$ and may be derived from sensor specifications or found empirically. In principle, these are the three components that the corresponding factor define: (1) the measurement itself (2) the prediction function and (3) the noise model (taking the form of a covariance matrix or, as I see later, square-root information matrix).

The monotonic logarithm function and Gaussian noise model allow us to simplify the optimization into a nonlinear least squares problem:

$$X^* = \underset{X}{\operatorname{argmin}} - \log \prod_{i=1}^{N} l\left(X; \mathbf{z}_i\right) \tag{3.6}$$

$$= \underset{X}{\operatorname{argmin}} \sum_{i=1}^{N} \left\|h_i\left(X\right) - \mathbf{z}_i\right\|_{\mathbf{\Sigma}_i}^2 \tag{3.7}$$

where I use the notation $\|v\|_{\mathbf{\Sigma}}^2 = v^T \mathbf{\Sigma}^{-1} v$ to denote Mahalanobis distance.

The Gauss-Newton algorithm (GN) is commonly used to solve the nonlinear least squares problem in Eq. 3.7 by iteratively solving linear approximations of the nonlinear system. Given some initial state estimate X^0, the prediction function is linearized as

$$h_i\left(X\right) = h_i\left(X^0 + \mathbf{\Delta}\right) \approx h_i\left(X^0\right) + \mathbf{H}_i \mathbf{\Delta} \tag{3.8}$$

$$\mathbf{H}_i = \left.\frac{\partial h_i\left(X\right)}{\partial X}\right|_{X^0} \tag{3.9}$$

where $\mathbf{\Delta} = X - X^0$ is the state update vector and $\mathbf{H}_i$ is the *Jacobian* matrix of the prediction function $h_i\left(X\right)$. Substituting this linearized approximation into Eq. 3.7 yields

$$\mathbf{\Delta}^* = \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\|h_i\left(X^0\right) + \mathbf{H}_i\mathbf{\Delta} - \mathbf{z}_i\right\|_{\mathbf{\Sigma}_i}^2 \tag{3.10}$$

$$= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\|\mathbf{H}_i\mathbf{\Delta} - \left(\mathbf{z}_i - h_i\left(X^0\right)\right)\right\|_{\mathbf{\Sigma}_i}^2 \tag{3.11}$$

$$= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\|\mathbf{A}_i\mathbf{\Delta} - \mathbf{b}_i\right\|^2 \tag{3.12}$$

$$= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \left\|\mathbf{A}\mathbf{\Delta} - \mathbf{b}\right\|^2 \tag{3.13}$$

where $\mathbf{A}_i = \mathbf{\Sigma}_i^{-1/2}\mathbf{H}_i$ and $\mathbf{b}_i = \mathbf{\Sigma}_i^{-1/2}\left(\mathbf{z}_i - h_i\left(X^0\right)\right)$ are the *whitened* Jacobian matrix and error vector. $\mathbf{A}$ and $\mathbf{b}$ are obtained by simply stacking all the terms $\mathbf{A}_i$ and $\mathbf{b}_i$ into a single matrix and vector, respectively. Upon making this linear approximation, I may clearly define the three components that the ith factor must define:

1. The error vector $\boldsymbol{z}_i - h_i\left(X^0\right)$, which is based on the prediction function.
2. The Jacobian matrix $\mathbf{H}_i$ of the prediction function.
3. The square-root information matrix $\mathbf{\Sigma}_i^{-1/2}$.

Setting the derivative of Eq. 3.13 to zero results in the so-called normal equations:

$$\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{\Delta}^* = \mathbf{A}^T\mathbf{b} \tag{3.14}$$

which may be solved for the current iteration's update $\mathbf{\Delta}^*$ directly by means of the pseudo-inverse

$$\mathbf{\Delta}^* = \mathbf{A}^\dagger\mathbf{b} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{3.15}$$

or by using the Cholesky or QR decomposition. The update $\mathbf{\Delta}^*$ is applied to compute the updated state estimate, which is used as the linearization point for the next iteration in the GN solver. The solver terminates after the magnitude of the update vector falls below a threshold or after a maximum number of allowed iterations.

The Levenberg-Marquardt algorithm (LM) is often used as an alternative to GN, particularly for systems that may be poorly conditioned. LM solves a "damped" version of the normal equations $\left(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}\right)\mathbf{\Delta}^* = \mathbf{A}^T\mathbf{b}$, where λ is an adaptively selected scalar. If the computed update $\mathbf{\Delta}^*$ increases the overall residual, then the step is not taken, λ is increased, and the system is resolved. Increasing λ steers the solution away from the GN update direction and towards the steepest descent update direction. Additional details of the factor graph representation, nonlinear least squares SLAM formulation, and GN and LM algorithms are discussed thoroughly in [26].

This nonlinear least squares optimization for MAP estimation is utilized in three methods I present in this paper:

1. Two-view acoustic bundle adjustment, presented in Chapter 4.1
2. Pose graph SLAM framework with sonar constraints for long-term localization, presented in Chapter 4.2
3. Factor graph SLAM and calibration framework, presented in Appendix A

These optimizations are distinct processes, and I will use the general notation introduced in this section in discussing each framework individually.

3.3 Treatment of 6-DOF poses

Localizing a vehicle or sensor in the 3-D world means estimating a 6-DOF pose, which is comprised of a 3-DOF rotation and 3-DOF translation. Since poses are not represented as vectors in my state, I need to make a few modifications to the nonlinear least squares framework previously derived. Here I make use of the mechanics of Lie group theory. A Lie group is essentially a group that is also a differentiable manifold. I represent the full 6-DOF pose as an element of the special Euclidean matrix Lie group $SE\left(3\right)$, which is the semi-direct product of the special orthogonal group $SO\left(3\right)$ and $\mathbb{R}^3$. The tangent space defined at the group's identity element is the Lie algebra $\mathfrak{se}\left(3\right)$. The logarithm map $\log\left(\cdot\right)$ maps an element from the group manifold to its corresponding element in the Lie algebra. The exponential map $\exp\left(\cdot\right)$ is the inverse function, which takes an element from the Lie algebra to the Lie group manifold.

A pose $x \in SE\left(3\right)$ is represented by a 4×4 homogeneous transformation matrix

$$\mathbf{T}_x = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \,|\, \mathbf{R} \in SO\left(3\right), \mathbf{t} \in \mathbb{R}^3. \tag{3.16}$$

There exist six *generators* of the Lie algebra $\mathfrak{se}\left(3\right)$, which I omit in their explicit form for brevity here. These are 4×4 matrices which may be linearly combined to generate an element $\hat{\boldsymbol{\xi}} \in \mathfrak{se}\left(3\right)$. This takes the form

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{\omega}_\times & \mathbf{u} \\ 0 & 0 \end{bmatrix} \tag{3.17}$$

where

$$\boldsymbol{\omega}_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{3.18}$$

and $\boldsymbol{\omega}, \mathbf{u} \in \mathbb{R}^3$. The "vee" operator extracts a vector comprising the six parameters of the element of the Lie algebra: $\boldsymbol{\xi} \triangleq \hat{\boldsymbol{\xi}}^\vee = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{u} \end{bmatrix}$. For convenience, I will refer to this minimal 6-vector as an element of the Lie algebra $\boldsymbol{\xi} \in \mathfrak{se}\left(3\right)$ with the understanding that it is actually the vector of coefficients which specifies the linear combination of the generators which create $\hat{\boldsymbol{\xi}}$.

The exponential map that converts from $\mathfrak{se}\left(3\right)$ to $SE\left(3\right)$ is the matrix exponential of $\hat{\boldsymbol{\xi}}$, but I will denote as $\exp\left(\boldsymbol{\xi}\right)$,

$$\exp\left(\boldsymbol{\xi}\right) = \mathbf{I} + \begin{bmatrix} \boldsymbol{\omega}_\times & v \\ 0 & 0 \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} \boldsymbol{\omega}_\times^2 & \boldsymbol{\omega}_\times v \\ 0 & 0 \end{bmatrix} + \frac{1}{3!}\begin{bmatrix} \boldsymbol{\omega}_\times^3 & \boldsymbol{\omega}_\times^2 v \\ 0 & 0 \end{bmatrix} + \ldots \tag{3.19}$$

which actually has the closed form expression

$$\begin{aligned}
\exp\left(\boldsymbol{\xi}\right) &= \begin{bmatrix} \mathbf{R} & \mathbf{V}\mathbf{u} \\ 0 & 1 \end{bmatrix} && (3.20)\\
\mathbf{R} &= \mathbf{I} + \frac{\sin\theta}{\theta}\boldsymbol{\omega}_{\times} + \frac{1-\cos\theta}{\theta^{2}}\boldsymbol{\omega}_{\times}^{2} && (3.21)\\
\mathbf{V} &= \mathbf{I} + \frac{1-\cos\theta}{\theta^{2}}\boldsymbol{\omega}_{\times} + \frac{\theta-\sin\theta}{\theta^{3}}\boldsymbol{\omega}_{\times}^{2} && (3.22)\\
\theta &= \left\|\boldsymbol{\omega}\right\|_{2} && (3.23)
\end{aligned}$$

where Taylor series expansions should be used for the appropriate terms in practical implementations when θ is small. The logarithm map may be computed as

$$\begin{aligned}
\boldsymbol{\omega}_{\times} &= \frac{\theta}{2\sin\theta}\left(\mathbf{R}-\mathbf{R}^{T}\right) && (3.24)\\
\mathbf{u} &= \mathbf{V}^{-1}\mathbf{t} && (3.25)\\
\theta &= \arccos\left(\frac{\mathtt{tr}\left(\mathbf{R}\right)-1}{2}\right). && (3.26)
\end{aligned}$$

Note that while there are two different formulas for θ, the first is used to compute the angle given the Lie algebra element, and the second is to compute it given the Lie group element.

Equipped with this framework, I now introduce modified error functions for measurements which define an absolute 6-DOF pose prior or a relative 6-DOF pose measurement. Let us consider just a single error term from Eq. 3.11. For a measurement z_i which specifies a 6-DOF pose prior on x_i with covariance $\boldsymbol{\Sigma}_i$, the error is expressed as

$$\left\|\mathbf{H}_i\boldsymbol{\xi}_i - \log\left(\mathbf{T}_{x_i}^{-1}\mathbf{T}_{z_i}\right)\right\|_{\boldsymbol{\Sigma}_i}^{2} \qquad (3.27)$$

where $\mathbf{H}_i$ is the Jacobian of the exponential map and $\boldsymbol{\xi}_i$ is the minimal 6-vector representation of the update to the pose in the Lie algebra frame. For a measurement $z_{i,j}$ which specifies a relative 6-DOF transformation from x_i to x_j with covariance $\boldsymbol{\Sigma}_{i,j}$, the error is expressed as

$$\left\|\mathbf{H}_i\boldsymbol{\xi}_i + \mathbf{H}_j\boldsymbol{\xi}_j - \log\left(\mathbf{T}_{x_j}^{-1}\mathbf{T}_{x_i}\mathbf{T}_{z_{i,j}}\right)\right\|_{\boldsymbol{\Sigma}_{i,j}}^{2}. \qquad (3.28)$$

These error functions and allow us to express the optimization in terms of the *local coordinates* in the frame of the Lie algebra for all 6-DOF pose variables in my optimization that lie on the $SE\left(3\right)$ manifold. This treatment of the 6-DOF poses is implemented in the GTSAM library [28]. My presentation of the Lie group theory is based on [2, 30], which go into greater detail

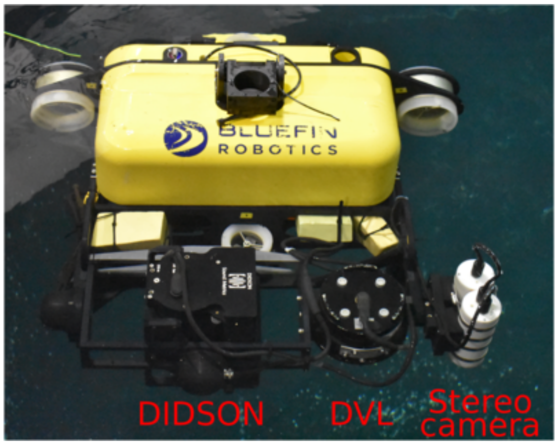


Figure 3.2: The Bluefin HAUV which is used in my real-world experiments. The DVL and stereo camera are mounted in front of the vehicle, and fixed facing downward at the bottom of the tank. The DIDSON imaging sonar is mounted to a tilt actuator that provides $90°$ of motion, allowing the sonar to point directly downwards (as in this image), directly to the side of the vehicle, or anywhere in between.

on this subject matter. A more thorough discussion on the representation of 6-DOF poses in the nonlinear least squares framework may be found in [26].

3.4 Test vehicle configuration

3.4.1 Sensors

While the proposed frameworks may be used with a variety of underwater robots, I perform my experiments with the Bluefin Hovering Autonomous Underwater Vehicle (HAUV) [36], as shown in Figure 3.2. This vehicle is equipped with several sensors for onboard navigation: a 1.2MHz Teledyne/RDI Workhorse Navigator Doppler velocity log (DVL), an attitude and heading reference system (AHRS), and a Paroscientific Digiquartz depth sensor. The AHRS utilizes a Honeywell HG1700 IMU to measure acceleration and rotational velocities. The DVL is an acoustic sensor that measures translational velocity with respect to the water column or a surface, such as the seafloor, test tank floor, or ship hull.

A stereo pair of two Prosilica GC1380 cameras are also mounted on the vehicle, on the same roll cage as the DVL sensor. The camera intrinsics are calibrated underwater using the pinhole camera model, after correcting the images for radial and tangential distortion, as described in [109]. Since I do not have an underwater motion capture system to provide ground-truth localization, I utilize images from one of the cameras to perform visual SLAM, which is used to evaluate my sonar localization method. This work was previously published in [114] and is described in Appendix A.

3.4.2 On-board odometry

A proprietary navigation algorithm fuses measurements from the AHRS, DVL, and depth sensor to provide odometry estimates for the vehicle, in the coordinate frame of the DVL sensor. It is important to note that the depth sensor gives direct measurements of the vehicle's depth, or Z position. The AHRS is also capable of producing very accurate, drift-free estimates of the vehicle's pitch and roll angles by observing the direction of gravity. The X and Y translation and yaw rotation are not directly observable, and are therefore estimated by dead reckoning of the DVL and IMU odometry measurements. The pose estimate will inevitably drift in these directions over long-term operation. This naturally leads to a formulation that treats the vehicle's odometry measurements as two separate types of constraints: a relative pose-to-pose constraint on XYH motion (**X** and **Y** translation and **H**eading / yaw) and a unary ZPR constraint (**Z** translation, **P**itch and **R**oll rotation) [107].

For these XYH and ZPR measurement factors, I make use of the Euler angle representation of 3-D rotations to represent a pose x_i as the 6-vector $\left[\psi_{x_i}, \theta_{x_i}, \phi_{x_i}, t^x_{x_i}, t^y_{x_i}, t^z_{x_i}\right]^\top$, where ψ_{x_i}, θ_{x_i},ϕ_{x_i} and are the roll, pitch, and yaw (heading) angles, respectively. All Euler angles are normalized to the range $[-\pi, \pi)$ radians. The likelihood functions for these factors are defined as

$$l(x_{i-1}, x_i; u_i) \quad \propto \quad \exp\left\{-\frac{1}{2}\left\|f_i(x_{i-1}, x_i) - \mathbf{u}_i\right\|^2_{\mathbf{\Gamma}_i}\right\} \tag{3.29}$$

$$l(x_i; v_i) \quad \propto \quad \exp\left\{-\frac{1}{2}\left\|g_i(x_i) - \mathbf{v}_i\right\|^2_{\mathbf{\Lambda}_i}\right\} \tag{3.30}$$

where $f_i(x_{i-1}, x_i)$ and $g_i\left(x_i\right)$ are the XYH and ZPR measurement functions and $\mathbf{\Gamma}_i$ and $\mathbf{\Lambda}_i$ their corresponding covariance matrices. The measurements $\mathbf{u}_i$ and $\mathbf{v}_i$ are simply 3-vectors specifying the change in the XYH directions and the absolute ZPR measurement, respectively.

The prediction functions $f\left(x_{i-1}, x_i\right)$ and $g\left(x_i\right)$ are comprised of the corresponding components of $x_{i-1,i}$ (the relative transformation from x_{i-1} to x_i) and x_i, respectively:

$$f\left(x_{i-1}, x_i\right) = \left[\begin{array}{ccc} t^x_{x_{i-1,i}} & t^y_{x_{i-1,i}} & \phi_{x_{i-1,i}} \end{array}\right]^T \tag{3.31}$$

$$g(x_i) = \left[\begin{array}{ccc} t^z_{x_i} & \theta_{x_i} & \psi_{x_i} \end{array}\right]^T. \tag{3.32}$$

I assume independent normally distributed noise in each direction for these two measurement

Noise models

	$\sigma_{i,x}$	$\sigma_{i,y}$	$\sigma_{i,\phi}$	$\sigma_{i,z}$	$\sigma_{i,\theta}$	$\sigma_{i,\psi}$
Value	$0.005 + 0.002\Delta t$	$0.005 + 0.002\Delta t$	$0.005 + 0.002\Delta t$	0.02	0.005	0.005
Units	m	m	rad	m	rad	rad

Table 3.1: Values of noise model parameters for the XYH and ZPR odometry factors and AprilTag measurement factors. Δt denotes the difference in time between the two poses involved in the corresponding XYH odometry factors. The uncertainty of the ZPR measurements is constant due to direct observability, and the uncertainty of the XYH factors increases linearly over time.

types:

$$\mathbf{\Gamma}_i = \operatorname{diag}\left(\left[\begin{array}{ccc} \sigma_{i,x}^2 & \sigma_{i,y}^2 & \sigma_{i,\phi}^2 \end{array}\right]\right) \tag{3.33}$$

$$\mathbf{\Lambda}_i = \operatorname{diag}\left(\left[\begin{array}{ccc} \sigma_{i,z}^2 & \sigma_{i,\theta}^2 & \sigma_{i,\psi}^2 \end{array}\right]\right). \tag{3.34}$$

However, since the XYH measurements are relative and prone to drift over time, I scale $\mathbf{\Gamma}_i$ with time. The specific values I use in all of my experiments with the HAUV are shown in Table 3.1.

Chapter 4

Localization

4.1 Two-view acoustic bundle adjustment

4.1.1 Setup

The goal of my degeneracy-aware bundle adjustment algorithm is to generate a 6-DOF pose-to-pose constraint using only corresponding bearing-range measurements from two sonar viewpoints. Note that this contrasts with the previously proposed ASFM methods [46, 48, 117], which include odometry measurements in the optimization. I choose this approach so that the sonar-based pose-to-pose constraints may be fused with the odometry constraints in a computationally efficient pose-graph framework without double-counting any measurements.

The state of a bundle adjustment optimization is normally comprised of all involved poses and landmarks. Figure 4.1 shows the factor graph representation of a two-view bundle adjustment optimization, consisting of two poses x_A and x_B as well as N 3-D point landmarks $\mathbf{l}_1, \ldots, \mathbf{l}_N$. A standard method is to place a prior measurement on the first pose, as represented by the dotted portion of the graph. However, since we seek to solve for a single 6-DOF of freedom constraint between the two poses, x_A is removed from the state. This may be thought of as equivalent to taking the limit as the uncertainty on the pose prior approaches zero. The solid portion of the factor graph shows the true representation of my optimization: only pose x_B and the landmarks are explicitly represented as variables in the state. The sensor pose x_A is taken as constant in the corresponding bearing-range measurements.

Therefore the state we wish to optimize is $x = \{x_B, \mathbf{l}_0, \ldots, \mathbf{l}_N\}$. I follow [48] and parameterize the landmarks using spherical coordinates $\mathbf{l}_i = \begin{bmatrix} \theta_i & r_i & \phi_i \end{bmatrix}$ (bearing, range and elevation) relative to x_A, which marks the reference coordinate system. I denote the set of bearing-range measurements used in the optimization as $\mathbf{z} = \left\{\mathbf{z}_1^A, \ldots, \mathbf{z}_N^A, \mathbf{z}_1^B, \ldots \mathbf{z}_N^B\right\}$, where $\mathbf{z}_i^P$ is the

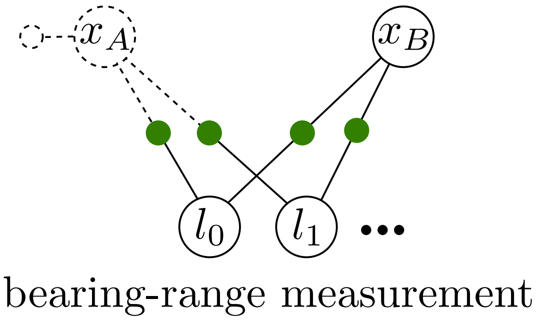


Figure 4.1: Factor graph representation of the two-view sonar optimization. In my two-view configuration, I optimize the relative 6-DOF transformation between the two views and the positions of all observed landmarks. x_A is dotted to signify that it is treated as a constant and is therefore not explicitly modeled in the bundle-adjustment optimization.

bearing-range measurement corresponding to $\mathbf{l}_i$ taken from pose P. I assume feature correspondences are provided by some frontend module. A novel algorithm for determining robust feature correspondences is presented in Section 4.2.3.

Recall the three components that must be defined by a factor: (1) the measurement (2) the prediction function and (3) the noise model. The measurement $\mathbf{z}_i = \begin{bmatrix} z_{\theta,i} & z_{r,i} \end{bmatrix}^T$ is simply a bearing-range observation of a feature point, and the covariance matrix representing the Gaussian noise model assumes independent noise in the bearing and range components:

$$\boldsymbol{\Sigma_i} = \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}. \tag{4.1}$$

The last component to define is the prediction function. Here I define two separate prediction functions for the measurements taken from x_A and x_B: $\hat{\mathbf{z}}_i^A = \mathbf{h}_i^A(\mathbf{l}_i)$ and $\hat{\mathbf{z}}_i^B = \mathbf{h}_i^B(x_B, \mathbf{l}_i)$. Note that rather than writing the predictions as functions of the entire state (as in $\mathbf{h}_i(x)$), I specify the particular components of the state that are used in the prediction. The predicted measurement of landmark $\mathbf{l}_i$ from pose x_A is simply

$$\mathbf{h}_i^A(\mathbf{l}_i) = \begin{bmatrix} \theta_i \\ r_i \end{bmatrix} \tag{4.2}$$

is $\mathbf{l}_i$ is represented by spherical coordinates relative to x_A. The measurement function $\mathbf{h}_i^B(x_B, \mathbf{l}_i)$

is:

$$\begin{aligned} \mathbf{h}_i^B\left(x_B, \mathbf{l}_i\right) &= \boldsymbol{\pi}\left(\mathbf{q}_i\right) = \begin{bmatrix} \operatorname{atan2}\left(q_{i,y}, q_{i,x}\right) \\ \sqrt{q_{i,x}^2 + q_{i,y}^2 + q_{i,z}^2} \end{bmatrix} && (4.3) \\ \mathbf{q}_i &= T_{x_B}\left(\mathbf{p}_i\right) = \mathbf{R}_{x_B}^T\left(\mathbf{p}_i - \boldsymbol{t}_{x_B}\right) && (4.4) \\ \mathbf{p}_i &= C\left(\mathbf{l}_i\right) = \begin{bmatrix} r_i \cos\theta_i \cos\phi_i \\ r_i \sin\theta_i \cos\phi_i \\ r_i \sin\phi_i \end{bmatrix} && (4.5) \end{aligned}$$

where $\mathbf{p}_i$ is the landmark in Cartesian coordinates relative to x_A and $\mathbf{q}_i = \begin{bmatrix} q_{i,x} & q_{i,y} & q_{i,z} \end{bmatrix}^T$ is the Cartesian representation of the point in the frame of x_B. Finally, I must be able to evaluate the Jacobian matrices $\mathbf{H}_i^A$ and $\mathbf{H}_i^B$ of these two measurement functions. While these may be computed numerically, it is usually more computationally efficient to use analytical Jacobians, which I derive here.

The partial derivatives of the measurement functions $\mathbf{h}_i^A\left(\mathbf{l}_i\right)$ and $\mathbf{h}_i^B\left(x_B, \mathbf{l}_i\right)$ are zero with respect to all landmarks other than $\mathbf{l}_i$. Therefore, I will only examine the block components of $\mathbf{H}_i^A$ and $\mathbf{H}_i^B$ corresponding to the partial derivatives with respect to x_B and $\mathbf{l}_i$. Since pose A is my reference coordinate frame, the Jacobians of $\mathbf{h}_i^A\left(\mathbf{l}_i\right)$ are trivial:

$$\begin{aligned} \frac{\partial \mathbf{h}_i^A\left(\mathbf{l}_i\right)}{\partial x_B} &= \mathbf{0} && (4.6) \\ \frac{\partial \mathbf{h}_i^A\left(\mathbf{l}_i\right)}{\partial \mathbf{l}_i} &= \mathbf{I}_{2\times 3}. && (4.7) \end{aligned}$$

The Jacobians of $\mathbf{h}_i^B\left(x_B, \mathbf{l}_i\right)$ may be computed using the chain rule:

$$\begin{aligned} \frac{\partial \mathbf{h}_i^B\left(x_B, \mathbf{l}_i\right)}{\partial x_B} &= \frac{\partial \hat{\mathbf{z}}_i^B}{\partial \mathbf{q}_i}\frac{\partial \mathbf{q}_i}{\partial x_B} && (4.8) \\ \frac{\partial \mathbf{h}_i^B\left(x_B, \mathbf{l}_i\right)}{\partial \mathbf{l}_i} &= \frac{\partial \hat{\mathbf{z}}_i^B}{\partial \mathbf{q}_i}\frac{\partial \mathbf{q}_i}{\partial \mathbf{p}_i}\frac{\partial \mathbf{p}_i}{\partial \mathbf{l}_i} && (4.9) \end{aligned}$$

where

$$\frac{\partial \hat{\mathbf{z}}_i^B}{\partial \mathbf{q}_i} = \begin{bmatrix} \frac{-q_{i,y}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & \frac{q_{i,x}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & 0 \\ \frac{q_{i,x}}{\sqrt{q_{i,x}^2+q_{i,y}^2+q_{i,z}^2}} & \frac{q_{i,y}}{\sqrt{q_{i,x}^2+q_{i,y}^2+q_{i,z}^2}} & \frac{q_{i,z}}{\sqrt{q_{i,x}^2+q_{i,y}^2+q_{i,z}^2}} \end{bmatrix} \tag{4.10}$$

$$\frac{\partial \mathbf{q}_i}{\partial x_B} = \begin{bmatrix} [\mathbf{q}_i]_\times & -\mathbf{I}_{3\times3} \end{bmatrix} \tag{4.11}$$

$$\frac{\partial \mathbf{q}_i}{\partial \mathbf{p}_i} = \begin{bmatrix} \mathbf{R}_{x_B}^T \end{bmatrix} \tag{4.12}$$

$$\frac{\partial \mathbf{p}_i}{\partial \mathbf{l}_i} = \begin{bmatrix} -r_i \sin\theta_i \cos\phi_i & \cos\theta_i \cos\phi_i & -r_i \cos\theta_i \sin\phi_i \\ r_i \cos\theta_i \cos\phi_i & \sin\theta_i \cos\phi_i & -r_i \sin\theta_i \sin\phi_i \\ 0 & \sin\phi_i & r_i \cos\phi_i \end{bmatrix}. \tag{4.13}$$

Here $[\cdot]_\times$ denotes the 3×3 skew-symmetric cross-product matrix of a 3-vector.

The most straight-forward way to solve this optimization would be to use GN or LM. The previous formulations of ASFM use LM to solve this type of optimization, largely because the optimization is often poorly conditioned. In this two-view optimization framework, there are $6 + 3N$ variables and $4N$ constraint equations (two from each bearing-range measurement in each sonar image), so at least six bearing-range measurements from each image are needed for the optimization to be fully constrained. However, even with six or more measurements, the optimization may be poorly constrained, depending on the geometry of the sensor motion and the initial estimate provided to the optimization.

4.1.2 Landmark elevation degeneracy

The most glaring degeneracy in the two-view acoustic bundle adjustment optimization is that of a point landmark's elevation angle. My previous work in [117] proposed a modification to the standard optimization to handle this degeneracy. This method calls for removing the elevation angle of each landmark from the state vector, so that a landmark only constitutes two variables in the state. I replace each 3-vector landmark in the state with a 2-vector parameterization: $\mathbf{m}_i = \begin{bmatrix} \theta_i & r_i \end{bmatrix}^T$. This decouples the elevation angle from the Gaussian parameterization, and allows us to treat the elevation angle in a non-parametric fashion. I also introduce slightly modified notation for the measurement functions: $\mathbf{h}_i^A(\mathbf{m}_i)$ and $\mathbf{h}_i^B(x_B, \mathbf{m}_i)$.

Practically, there is no difference in the measurement function corresponding to pose x_A: $\mathbf{h}_i^A(\mathbf{m}_i) = \begin{bmatrix} \theta_i & r_i \end{bmatrix}^T$. However, the elevation angle ϕ_i is no longer explicitly modeled by $\mathbf{m}_i$, yet some estimate of the elevation angle is still needed to compute a projection of the landmark in the frame of x_B, using the new measurement function $\mathbf{h}_i^B(x_B, \mathbf{m}_i)$. I address this performing a direct search over the valid range of ϕ_i to find the elevation angle with the lowest reprojection error:

$$\mathbf{h}_i^B\left(x_B, \mathbf{m}_i\right) \;=\; \boldsymbol{\pi}\left(T_{x_B}\left(c_{i,\phi^*}\right)\right) \tag{4.14}$$

$$\phi_i^* = \underset{\phi \in \Phi}{\operatorname{argmin}} \left\| \boldsymbol{\pi}\left(T_{x_B}\left(c_{i,\phi}\right)\right) - \mathbf{z}_i^B \right\|_{\boldsymbol{\Sigma_i}}^2 \tag{4.15}$$

where $\Phi = \{\phi_{min}, \phi_{min} + \Delta\phi, \ldots, \phi_{max} - \Delta\phi, \phi_{max}\}$, $\Delta\phi$ is selected such that n_{elv} uniformly spaced angles are sampled from the valid range, and $c_{i,\phi}$ denotes the Cartesian coordinates corresponding to the spherical coordinates $\begin{bmatrix} \mathbf{m}_i^T & \phi \end{bmatrix}^T$. This direct search lets us treat the belief of the elevation angle as a uniform distribution over the valid range, which is a much more appropriate treatment than a unimodal Gaussian representation that may result in the optimization getting stuck in local minima. Additionally, as the search is over a bounded one-dimensional space, it is computationally efficient for small systems such as the considered two-view scenario.

Knowing the Jacobians of $\mathbf{h}_i^A\left(\mathbf{l}_i\right)$ and $\mathbf{h}_i^B\left(x_B, \mathbf{l}_i\right)$, the Jacobians of $\mathbf{h}_i^A\left(\mathbf{m}_i\right)$ and $\mathbf{h}_i^B\left(x_B, \mathbf{m}_i\right)$ are trivial, as the last column of the corresponding Jacobians are simply removed, as ϕ_i is no longer part of the state:

$$\frac{\partial \mathbf{h}_i^A\left(\mathbf{m}_i\right)}{\partial x_B} \;=\; \mathbf{0} \tag{4.16}$$

$$\frac{\partial \mathbf{h}_i^A\left(\mathbf{m}_i\right)}{\partial \mathbf{m}_i} \;=\; \mathbf{I}_{2\times 2} \tag{4.17}$$

$$\frac{\partial \mathbf{h}_i^B\left(x_B, \mathbf{m}_i\right)}{\partial x_B} \;=\; \frac{\partial \hat{\mathbf{z}}_i^B}{\partial \mathbf{q}_i}\frac{\partial \mathbf{q}_i}{\partial x_B} \tag{4.18}$$

$$\frac{\partial \mathbf{h}_i^B\left(x_B, \mathbf{m}_i\right)}{\partial \mathbf{m}_i} \;=\; \frac{\partial \hat{\mathbf{z}}_i^B}{\partial \mathbf{q}_i}\frac{\partial \mathbf{q}_i}{\partial \mathbf{p}_i}\frac{\partial \mathbf{p}_i}{\partial \mathbf{m}_i} \tag{4.19}$$

where

$$\frac{\partial \mathbf{p}_i}{\partial \mathbf{m}_i} = \begin{bmatrix} -r_i \sin\theta_i \cos\phi_i^* & \cos\theta_i \cos\phi_i^* \\ r_i \cos\theta_i \cos\phi_i^* & \sin\theta_i \cos\phi_i^* \\ 0 & \sin\phi_i^* \end{bmatrix}. \tag{4.20}$$

4.1.3 Sensor pose degeneracy

In contrast to a landmark's elevation angle, the relative pose between the two viewpoints may often be under-constrained in multiple degrees of freedom. Considering the multivariate space of potentially valid sensor poses, and the fact that no inequality constraints exist on the sensor pose parameters as in the case of the elevation angle, a search over the parameter space is not a suitable solution to this type of degeneracy.

I adopt the general approach of *solution remapping* in nonlinear optimization, as presented in [122]. This technique operates on the linear approximation of the nonlinear system at each step in the optimization. Therefore, I follow the same formulation of the two-view optimization presented in Section 3.2, up until the linear approximation in Eq. 3.13. I make use of the singular-value decomposition (SVD) of the unmodified $m \times n$ measurement Jacobian matrix: $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, where $\mathbf{U}$ is an orthogonal $m \times m$ matrix, $\mathbf{S}$ is a diagonal $m \times n$ matrix of singular values $\sigma_1 \leq \cdots \leq \sigma_n$, and $\mathbf{V}$ is an orthogonal $n \times n$ matrix. The pseudoinverse of $\mathbf{A}$ may be computed as $\mathbf{A}^\dagger = \mathbf{V}\mathbf{S}^\dagger\mathbf{U}^T$, where $\mathbf{S}^\dagger$ is an $n \times m$ diagonal matrix with diagonal $\left[\begin{array}{ccc} 1/\sigma_1 & \ldots & 1/\sigma_n \end{array}\right]$. Using this decomposition, the linearized least squares problem in Eq. 3.13 may be solved as $\mathbf{\Delta}^* = \mathbf{V}\mathbf{S}^\dagger\mathbf{U}^T\mathbf{b}$, which yields the same update vector $\mathbf{\Delta}^*$ as by solving with Cholesky or QR decomposition, or by directly computing $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$.

However, the SVD provides valuable information that the Cholesky and QR do not: singular values. A small singular value σ_i denotes a poorly constrained direction in the state space specified by the corresponding column of $\mathbf{V}$, $\boldsymbol{v}_i$. The idea of *solution remapping* is to only update the state in the directions that are well-constrained. This is achieved by setting a threshold σ_{min} below which a singular value and its corresponding update direction will not be added to $\mathbf{\Delta}^*$. In this formulation, I solve the linear least squares problem using a modified pseudoinverse

$$\mathbf{\Delta}^* = \mathbf{A}_D^\dagger\mathbf{b} = \mathbf{V}\mathbf{S}_D^\dagger\mathbf{U}^T\mathbf{b} \tag{4.21}$$

where $\mathbf{S}_D^\dagger$ is an $n \times m$ diagonal matrix with diagonal $\left[\begin{array}{ccccc} 0 & \ldots & 1/\sigma_s & \ldots & 1/\sigma_n \end{array}\right]$ and σ_s is the smallest singular value greater than the threshold σ_{min}. This procedure generates an update vector $\mathbf{\Delta}^*$ only using the well-constrained directions of the state [122]. Under this framework, there is no need to dampen the system heuristically as in LM. These degeneracy-aware updates are applied successively using GN until the magnitude of the updates falls below a threshold, or until a maximum number of iterations are performed.

4.1.4 Importance of initial estimate

The initial estimate provided to a nonlinear optimizer often severely impacts the final state estimate obtained. It is typically assumed that the initial estimate is close enough to the global minimum that the successive linearizations made by the GN-style optimizer result in convergence to the true global minimum of the state. Here I examine the significance of the initial estimate in my two-view bundle adjustment optimization.

Using the non-parametric representation of the landmarks' elevation angle yields the state

$x = [x_B, \mathbf{m}_1, \ldots, \mathbf{m}_N]$. The initial estimate for $\mathbf{m}_i$ is simply set to the bearing-range measurement of the landmark from x_A. The initial estimate of x_B would ideally come from a state estimate provided by another sensor or sensors onboard the underwater vehicle, such as an inertial measurement unit (IMU) or Doppler velocity log (DVL). In the absence of such sensors, a motion-model may be utilized to predict the state. Consider the case where no prediction is available from sensors or a motion model, and the initial pose estimate is taken to be zero motion (i.e. $x_A = x_B$). Then the reprojection error of any landmark is exactly the same for any selected elevation angle and the optimal elevation angle chosen would be arbitrary. Systematically selecting a positive or negative elevation angle would bias the system towards an unlikely solution. If on the other hand, I assume an elevation angle of zero for each landmark, then the Jacobian of the reprojection in the frame of $x_B$would be

$$\frac{\partial \mathbf{h}_i^B\left(x_B, \mathbf{m}_i\right)}{\partial x_B} = \frac{\partial \hat{\mathbf{z}}_i^B}{\partial \mathbf{q}_i} \frac{\partial \mathbf{q}_i}{\partial x_B} \tag{4.22}$$

$$\begin{aligned} \frac{\partial \hat{\mathbf{z}}_i^B}{\partial \mathbf{q}_i} \frac{\partial \mathbf{q}_i}{\partial x_B} &= \begin{bmatrix} \frac{-q_{i,y}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & \frac{q_{i,x}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & 0 \\ \frac{q_{i,x}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & \frac{q_{i,y}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & q_{i,y} & -1 & 0 & 0 \\ 0 & 0 & -q_{i,x} & 0 & -1 & 0 \\ -q_{i,y} & q_{i,x} & 0 & 0 & 0 & -1 \end{bmatrix} \quad (4.23) \\ &= \begin{bmatrix} 0 & 0 & -\sqrt{q_{i,x}^2 + q_{i,y}^2} & \frac{-q_{i,y}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & \frac{-q_{i,x}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & 0 \\ 0 & 0 & 0 & \frac{-q_{i,x}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & \frac{-q_{i,y}}{\sqrt{q_{i,x}^2+q_{i,y}^2}} & 0 \end{bmatrix} \quad (4.24) \end{aligned}$$

since $q_z = 0$. This assumption of zero elevation projects all landmarks onto the xy plane and results in an estimated motion only in the plane (x and y translation and yaw rotation), as the columns of $\mathbf{A}$ corresponding to z translation and roll and pitch rotation are all zero. This harkens back to previous formulations of sonar image registration and localization, where the motion estimation was strictly limited to planar motion [53] or planar motion and z translation [9].

Thus, we see that in this acoustic bundle adjustment framework, the initial pose estimate has a large impact on the final solution of the optimization. Specifically, if no pose estimate is available from other sensors or a motion model, then the only motion within the xy plane can be recovered. Due to this limitation, the acoustic bundle adjustment framework is best applied in conjunction with other sensors that provide a state estimate prediction, rather than as a standalone localization solution in the absence of any other sensors or ego-motion estimation.

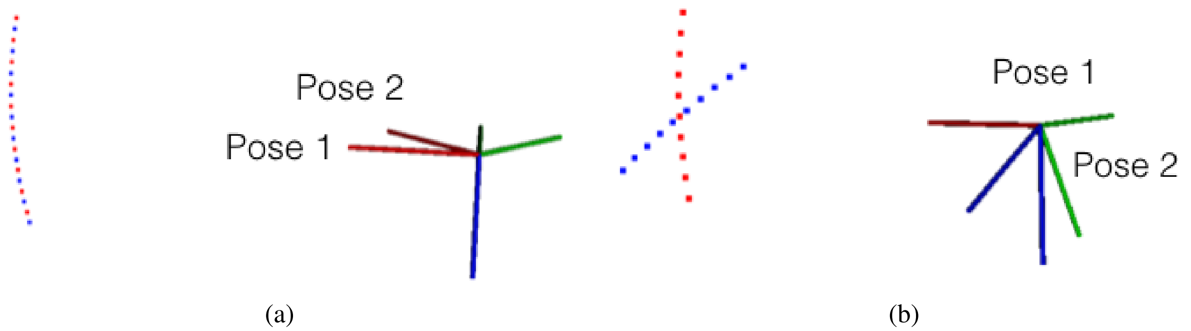


Figure 4.2: (a) The elevation arcs corresponding to measurements of the same point from pose 1 (red points) and pose 2 (blue points) separated by pure yaw motion are exactly aligned and the point's elevation angle is entirely ambiguous. (b) The elevation arcs have minimal overlap when the poses are separated by pure roll rotation—the observed point's elevation angle is well-constrained.

4.1.5 Triangulating well-constrained landmarks

While the primary purpose of this degeneracy aware acoustic bundle adjustment framework is to use feature matches to accurate estimate the sensor pose, we may also utilize it to identify well-constrained landmarks and accurately reconstruct their 3-D positions. As described in detail in [117], we may form a "structure-only" bundle adjustment optimization, in which the poses are treated as constant and the landmarks are modeled as $\mathbf{l}_i = \begin{bmatrix} \theta_i & r_i & \phi_i \end{bmatrix}$. The same SVD analysis of the optimization presented in Section 4.1.3 may then be used to analyze the conditioning of the elevation angle of each landmark. Any landmark with a sufficiently constrained landmark may then be modeled using the fully parametric representation $\mathbf{l}_i$.

Figure 4.2 shows a comparison of the constraints on a landmark under two different types of motion: pure yaw rotation and pure roll rotation. Under pure yaw rotation, repeated measurements correspond to exactly the same elevation arc, so the elevation angle of the landmark cannot be "triangulated" by an acoustic bundle adjustment algorithm. This is also the case for pure pitch rotation. For the case of pure roll rotation, as the magnitude of the rotation increases the elevation angle becomes more well-constrained and can actually be "triangulated" by acoustic bundle adjustment. See for further discussion on this topic [48, 117].

4.2 Pose graph SLAM framework

A pose graph is a type of factor graph in which the only variables are poses. Rather than explicitly modeling landmarks detected in sonar images and maintaining the bearing-range mea-

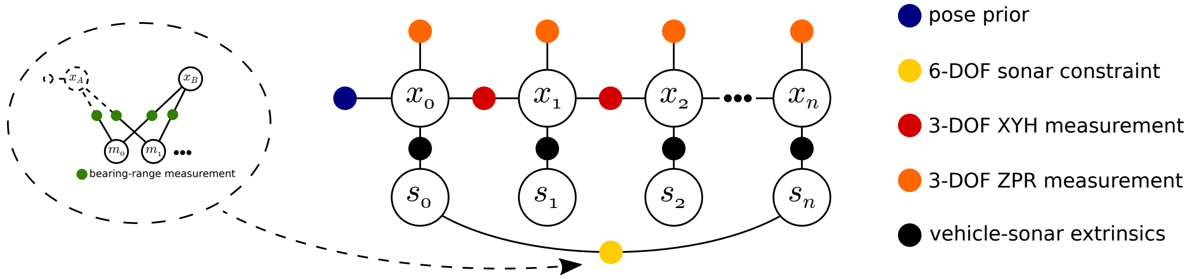


Figure 4.3: The pose graph framework I propose for long-term navigation. A two-view bundle adjustment problem is optimized, and the resulting pose-to-pose constraint is added to the pose graph along with the XYH and ZPR odometry measurements obtained from the onboard navigation.

surements in the overall factor graph, the landmarks are marginalized out locally in my two-view bundle adjustment optimization. While this is sub-optimal from an information-theoretic point of view, the sparsity of this formulation allows for much more efficient optimization than a full SLAM representation such as in [46, 117]. A visual representation of such a pose-graph with a large-scale loop closure is shown in Figure 4.3.

Similarly to the two-view bundle adjustment problem, the pose graph is framed as a MAP estimation problem and solved by means of nonlinear least squares, as presented in Section 3.2. The state consists only of vehicle and sonar poses $x = \{x_0, \ldots, x_n, s_0, \ldots, s_n\}$ and no landmarks are explicitly modeled. I utilize three primary types of measurements in this pose graph framework for localization – two to model the vehicle odometry and one for pairwise sonar constraints derived from the two-view bundle adjustment optimization. Two additional factor types are trivial: the prior on the first vehicle pose to tie the trajectory down to the global frame and the vehicle-sonar extrinsics. The vehicle-sonar extrinsics are modeled using a constant transformation and very low, constant uncertainty since the sonar is kept at a fixed pose relative to the vehicle frame throughout my datasets.

Many different types of odometry measurement constraints may be utilized in conjunction with my pose-to-pose sonar constraints. Here, I follow [107, 114] in using the XYH and ZPR odometry factors, which I describe in more detail in Chapter 3.4.2. These factors accurately model the dynamics of the hovering vehicle, in that the on-board dead reckoning state estimate may drift in the XYH directions, but will not drift in the ZPR directions.

4.2.1 Sonar constraint - measurement

The mean of the pose-to-pose constraint generated by the two-view optimization is simply the optimized relative pose: $z_{ij} \triangleq x_B^*$. The relative pose measurement is formulated as an element of the $SE(3)$ manifold, as described in Chapter 3.3. Note that the pose graph optimization also requires the square root information matrix $\mathbf{\Sigma}_{ij}^{-1/2}$ to whiten the system and create a least-squares framework. Recovering the square-root information matrix is more involved, and is detailed in the following subsection.

4.2.2 Sonar constraint - information

To compute the square-root factor corresponding to x_B^*, we utilize the information matrix of the overall degeneracy-aware linearized system at the final iteration of the two-view optimization as described in Section 4.1.3:

$$
\begin{aligned}
\mathbf{\Gamma} &= \mathbf{A}_D^T\mathbf{A}_D && (4.25)\\
&= \mathbf{V}\mathbf{S}_D^T\mathbf{U}^T\mathbf{U}\mathbf{S}_D\mathbf{V}^T && (4.26)\\
&= \mathbf{V}\mathbf{S}_D^T\mathbf{S}_D\mathbf{V}^T && (4.27)\\
&= \begin{bmatrix} \mathbf{\Gamma}_{11} & \mathbf{\Gamma}_{12} \\ \mathbf{\Gamma}_{21} & \mathbf{\Gamma}_{22} \end{bmatrix}. && (4.28)
\end{aligned}
$$

Here we use $\mathbf{\Gamma}_{11}$ to denote the top left 6×6 block of the information matrix corresponding to the pose x_B^*, $\mathbf{\Gamma}_{22}$ to denote the bottom right block that corresponds to the landmark terms, and $\mathbf{\Gamma}_{12}$ and $\mathbf{\Gamma}_{21}$ to denote the cross-correlation terms. In order to condense the information from the entire system into a single information matrix on x_B^*, we will marginalize out the landmark variables. We can perform this easily with the Schur complement, which is equivalent to marginalization in the case of normal distributions and has been widely used in probabilistic robot localization and SLAM in a variety of applications, including underwater, ground, and planetary vehicles [32, 82, 102]. This is written simply as:

$$\mathbf{\Lambda} = \mathbf{\Gamma}_{11} - \mathbf{\Gamma}_{12}\mathbf{\Gamma}_{22}^{-1}\mathbf{\Gamma}_{21}. \tag{4.29}$$

The block $\mathbf{\Gamma}_{22}$ is always invertible due to my 2-vector parameterization of the landmarks - the bearing and range of every landmark are directly observed and are well-constrained. The resulting information matrix $\mathbf{\Lambda}$ may very likely be singular and not positive definite, due to the use of the degeneracy-aware $\mathbf{A}_D = \mathbf{U}\mathbf{S}_D\mathbf{V}^T$. In the case that any singular values were zeroed out, $\mathbf{A}_D^T\mathbf{A}_D$ will be a singular matrix. The only directions of the state that may be in the null-space of $\mathbf{A}_D$ would be in the space of the transformation x_B (since all of the landmarks are well-constrained by construction). Therefore, if $\mathbf{A}_D^T\mathbf{A}_D$ is singular, $\mathbf{\Gamma}_{11}$ and $\mathbf{\Lambda}$ will also be singular and not positive definite. In this case, the standard method of computing the square root information matrix by Cholesky decomposition of $\mathbf{\Lambda} = \mathbf{R}^T\mathbf{R}$ will fail. Instead, we can utilize the LDL decomposition of $\mathbf{\Lambda}$ to obtain:

$$\begin{aligned}
\mathbf{\Lambda} &= \mathbf{PLDL}^T\mathbf{P}^T && (4.30)\\
&= \left(\mathbf{PLD}^{1/2}\right)\left(\mathbf{D}^{T/2}\mathbf{L}^T\mathbf{P}^T\right) && (4.31)\\
&= \mathbf{R}^T\mathbf{R} && (4.32)
\end{aligned}$$

where $\mathbf{P}$ is a permutation matrix, $\mathbf{L}$ is a lower triangular matrix, $\mathbf{D}$ is a diagonal matrix, and $\mathbf{R}$ is the square root factor of $\mathbf{\Lambda}$. $\mathbf{P}$ is necessary for numerical stability when decomposing a non positive-definite matrix. Therefore, $\mathbf{R}$ has the unusual property of not being an upper-triangular matrix, as it normally is for an invertible information matrix. However, this non-triangular square root information matrix is entirely compatible with the nonlinear least squares optimization and may be used to "whiten" the Jacobian matrices and error vectors, as in Eq. 3.12.

With the square-root information matrix and the measured relative pose transformation x_B^*, we can easily incorporate the two-view sonar constraint between poses x_i and x_j into the pose graph framework as described in Section 4.2.1, using $\mathbf{z}_{ij} = x_B^*$ and $\mathbf{\Sigma}_{ij}^{-1/2} = \mathbf{R} = \mathbf{D}^{T/2}\mathbf{L}^T\mathbf{P}^T$. The pose graph may be solved efficiently using the state-of-the-art iSAM2 algorithm [56] for real-time localization. The only criterion that must be met in order to be able to solve the pose graph is that the overall measurement Jacobian matrix $\mathbf{A}$, as defined in Eq. 3.13, must not be singular. A particular square root factor $\mathbf{R}$ corresponding to a two-view sonar constraint may be singular and provide no constraints in some directions as long as the other measurements (odometry in this case) do provide constraints in those directions. Therefore, it is important to utilize these two-view sonar constraints in conjunction with complementary measurements that provide some information in at least the directions that are not constrained by the two-view sonar measurements. My proposed framework always meets this criterion, as the combination of the XYH and ZPR odometry measurements fully constrain each pose.

4.2.3 Frontend: feature detection and association

All of the work discussed thus far has dealt with the *backend* of my proposed feature-based bundle adjustment algorithm: the optimization of sensor poses and landmarks given measurements and correspondences. The *frontend* of such a system is the component responsible for the detection and association of features. While the frontend feature detection and association is not the focus of this work, I propose a novel implementation for associating point features between two sonar frames.

Joint compatibility branch and bound (JCBB) [80] has often been considered the gold standard algorithm for probabilistic association of landmarks in a SLAM context. Several more recent works have essentially improved the computational efficiency of the original JCBB algorithm for the special case of feature cloud matching [86, 99]. Assuming that features are independently measured from two different poses, these algorithms use joint compatibility tests to evaluate the error of potential data association hypotheses. This is more robust to noisy measurements than data association algorithms based on individual compatibility because it evaluates the compatibility of the entire set of feature matches, rather than separately considering pair-wise compatibility for each feature matching.

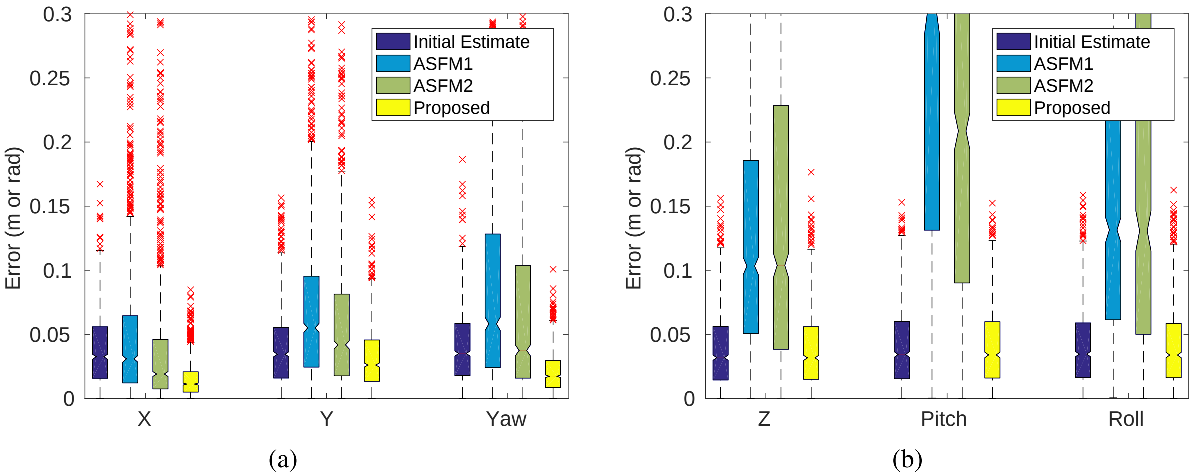
For the real-world experiments described in Section 4.3.2, I use the joint compatibility framework described in [99] for efficient data association between the two sonar frames in my two-view bundle adjustment problem. In Section 4.1, my notation assumed $\mathbf{z}_i^A$ and $\mathbf{z}_i^B$ correspond to the same landmark. Here we will use $\mathbf{z}_{j_i}^B$ to denote the measurement from pose x_B that is considered as a possible match to $\mathbf{z}_i^A$. The entire framework is built on the measurement function, which evaluates the error between $\mathbf{z}_i^A$ and its proposed matched feature $\mathbf{z}_{j_i}^B$:

$$f_{ij_i}\left(x_B, \mathbf{z}_i^A, \mathbf{z}_{j_i}^B\right) = \mathbf{h}_{ij_i}\left(x_B, \mathbf{z}_i^A\right) - \mathbf{z}_{j_i}^B. \tag{4.33}$$

Here $\mathbf{h}_{ij_i}\left(x_B, \mathbf{z}_i^A\right)$ projects measurement $\mathbf{z}_i^A$ into the coordinate frame of pose x_B using the optimal elevation angle as found by direct search, as in the two-view optimization:

$$\begin{aligned}
\mathbf{h}_{ij_i}\left(x_B, \mathbf{z}_i^A\right) &= \boldsymbol{\pi}\left(T_{x_B}\left(C\left(\begin{bmatrix} z_{\theta,i}^A & z_{r,i}^A & \phi_i^* \end{bmatrix}^T\right)\right)\right) && (4.34)\\
\phi_i^* &= \underset{\phi\in\Phi}{\operatorname{argmin}} \left\| \boldsymbol{\pi}\left(T_{x_B}\left(C\left(\begin{bmatrix} z_{\theta,i}^A & z_{r,i}^A & \phi \end{bmatrix}^T\right)\right)\right) - \mathbf{z}_{j_i}^B \right\|_{\boldsymbol{\Sigma}_{j_i}}^2 . && (4.35)
\end{aligned}$$

I implement the joint compatibility framework described in [99] using this measurement function, assuming that the features are independently measured at both poses. The only other required input is a relative pose estimate and pose uncertainty, which may be estimated from the pose graph and by propagating the uncertainty of odometry measurements. For the numbers of features used in my experiments, the algorithm is very quick and finds a robust correspondence between the feature clouds in real-time.



(a)

(b)

Figure 4.4: Absolute value of pose error of the various estimation methods over 1000 Monte Carlo simulations. The error in the well-constrained directions is shown in (a) and the error in the poorly constrained directions is shown in (b). The notch denotes the mean and the colored boxes indicates the 25th and 75th percentile. The whiskers extend to the most extreme data considered inliers, and outliers are marked in red.

4.3 Results and discussion

The two-view sonar bundle adjustment is implemented in C++ using the Eigen library [37] for efficient matrix operations and decompositions. I implement the pose graph framework using the GTSAM library [28], which includes an implementation of iSAM2 that is used for optimization. To evaluate the proposed algorithms I conduct both simulated experiments and real-world experiments in a test tank environment. All computation is performed on a consumer laptop computer with a 4-core 2.50GHz Intel Core i7 CPU and 8 GB RAM.

4.3.1 Simulation: two-view

I conduct various Monte Carlo simulations of my two-view sonar bundle adjustment algorithm. In all simulations, I assume ground-truth feature correspondences are known between the two sonar frames and add Gaussian noise to the bearing-range feature measurements ($\sigma_\theta = 0.01$ rad, $\sigma_r = 0.01$ m). I simulated the characteristics of the DIDSON imaging sonar used in my test tank experiments, by using the same azimuthal field of view (28.8°) and elevation field of view (28° using the spreader lens) and a range of $1 - 3$ m. In each simulation random 3-D point landmarks are generated, with a minimum of 6 points and an average of 12 viewed per two-view optimization. I use a constant threshold of $\sigma_{min} = 50$ throughout all simulations.

The first quantity that I sample in my simulations is the ground-truth relative pose transforma-

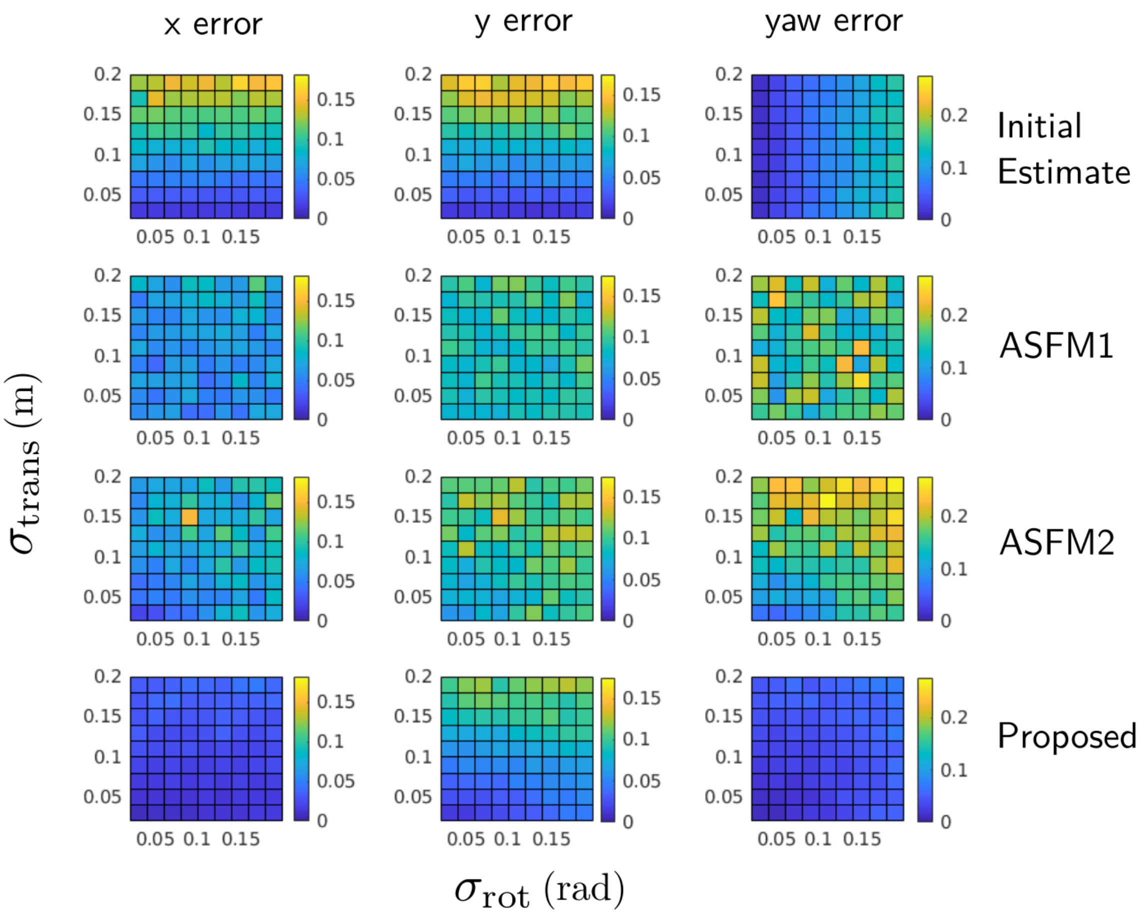


Figure 4.5: Average error over 100 Monte Carlo simulations for various levels of noise in the initial pose estimate. All plots in a row show the errors for a single pose estimation method and each column shows the error in one particular degree of freedom. The x-axis for each plot shows the value of the standard deviation σ_{rot} corresponding to the distribution of the noise $\mathcal{N}(0, \sigma_{rot})$ that is added to all rotation degrees of freedom, and likewise for the y-axis and σ_{trans}.

tion. I sample random “small” transformations with Euler angles drawn from $\mathcal{U}\left(-0.3 \text{ rad},\ 0.3 \text{ rad}\right)$ and translation components drawn from $\mathcal{U}\left(-0.3 \text{ m},\ 0.3 \text{ m}\right)$. Small transformations generally result in a two-view optimization that is poorly constrained - mostly in the ZPR directions. This allows us to demonstrate the advantage of my proposed degeneracy-aware algorithm over two previous approaches. The evaluated approaches are:

- **ASFM1** - The formulation presented in Section 4.1.1 and [46, 48], which solves the optimization via LM.
- **ASFM2** - The formulation presented in Section 4.1.2 and [117] which models the elevation angle non-parametrically and solves the optimization via LM.
- **Proposed** - My formulation presented in Section 4.1.3, which utilizes the non-parametric elevation angle formulation as well as the degeneracy-aware GN algorithm for optimization.

The initial estimate of the transformation is corrupted with Gaussian noise: $\mathcal{N}\left(0, 0.05 \text{ rad}\right)$ in the three Euler angles and $\mathcal{N}\left(0, 0.05 \text{ m}\right)$ in the three translation directions. The box and whisker plots in Figure 4.4 show the errors in each of the six degrees of freedom for the three pose estimation methods as well as the error of the initial estimate, over 1000 Monte Carlo simulations. The plots are separated into the well-constrained DOF in Figure 4.4a and the poorly constrained DOF in Figure 4.4b. In the well-constrained XYH directions, the proposed method significantly decrease the error compared to the initial estimate and the previous methods ASFM1 and ASFM2. In the poorly constrained ZPR directions, my proposed method hardly makes any updates to the initial estimate at all, while the previous methods actually significantly increase the error. While these previous formulations are quick to reach incorrect local minima and “overfit” the solution to noise in the measurements, my method cautiously provides updates to the state estimate in only the directions that are well-constrained by the underlying geometry.

I repeat the previous simulations, but vary the noise levels of the initial pose estimate in rotation and translation over a wide range. Figure 4.5 shows the average error in the well-constrained XYH directions of each evaluated approach for each of these noise levels. These simulations rigorously demonstrate that my proposed method outperforms the previously proposed algorithms in terms of the mean and variance of the pose estimation error.

4.3.2 Experimental: pose graph

I validate my proposed pose graph formulation by conducting real-world robotic experiments in a controlled test tank environment. The test tank is cylindrical - 7m in diameter and 3m in height. The robotic platform I use is the hovering autonomous underwater vehicle (HAUV) [36],

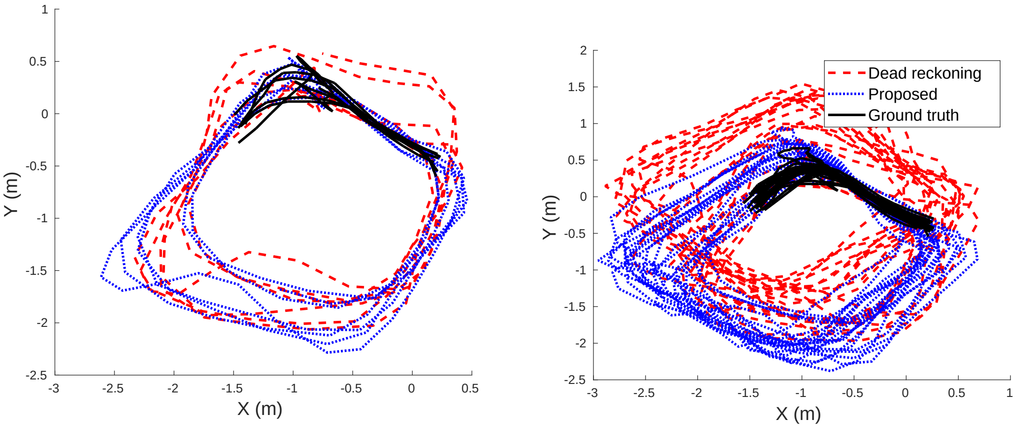


Figure 4.6: Top-down view of the trajectories of the dead reckoning and proposed pose graph SLAM solution as compared to the ground truth for Short dataset (left) and Long dataset (right). The pose-graph significantly reduces drift throughout the sequence despite only making loop closures at one corner of the rectangular trajectory.

as shown in Figure 3.2. This vehicle is equipped with several sensors for onboard navigation: a 1.2MHz Teledyne/RDI Workhorse Navigator Doppler velocity log (DVL), an attitude and heading reference system (AHRS), and a Paroscientific Digiquartz depth sensor. The AHRS utilizes a Honeywell HG1700 IMU to measure acceleration and rotational velocities. The DVL is an acoustic sensor that measures translational velocity with respect to the water column or a surface, such as the seafloor, ship hull, or in my case, the test tank floor. The vehicle also has a SoundMetrics DIDSON 300 sonar [104] mounted on the front of the vehicle, with $90°$ range of tilt motion controlled by an actuator.

For ground-truth sonar localization, I use the fiducial-based visual SLAM algorithm presented in [114]. This work combines vehicle odometry measurements with camera observations of AprilTag fiducials which are placed on the floor of the test tank. It uses the familiar factor graph SLAM formulation to optimize for the vehicle poses, the fiducial poses, as well as for the vehicle-camera extrinsics. The vehicle odometry consists of a proprietary algorithm that fuses information from the DVL, AHRS, and depth sensor to calculate a state estimate in the frame of the DVL. In order to produce a good estimate of the vehicle-camera extrinsics, the visual SLAM system was used to calibrate extrinsics before collecting the datasets I use in these experiments. The extrinsics are then modeled as constant when collecting ground-truth data for my experiments. For these experiments, I compare the trajectories of each localization method in the vehicle frame, as both the sonar and visual SLAM based solutions explicitly model and estimate the vehicle poses.

In the DVL frame, the x axis points directly forward from the vehicle, with the y axis directed

Test tank experiments - absolute trajectory error (ATE) (meters)

	Dead reckoning	Li modified	ASFM2	Proposed
Short trajectory	0.230	0.290	0.558	**0.074**
Long trajectory	0.5194	0.252	0.769	**0.159**

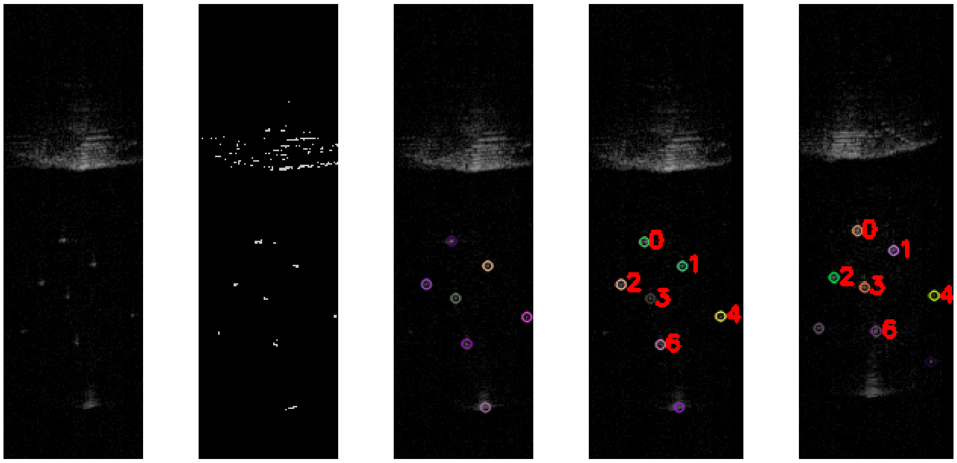
Table 4.1: Localization error of the evaluated methods on the test tank datasets. The short and long datasets used 37 and 66 loop closures, respectively, for all evaluated algorithms since the frontend feature detection and association is distinct from the backend optimization. My proposed method produces a significantly more accurate localization result than the other evaluated methods for both datasets.

toward the right and z axis. I use a measured, fixed transformation to model the extrinsics of the sonar sensor relative to the vehicle frame (DVL frame). Due to the configuration of the vehicle, the sonar's xy plane is parallel to the vehicle's xy plane, but the z axis points up rather than down. This configuration is ideal for correcting drift in the vehicle localization: the directions in which the dead reckoning state estimate drifts (XYH in the DVL frame) are precisely aligned with the directions that are best constrained by sonar loop closures (XYH in the sonar frame).

In these experiments I evaluate four different localization methods. In all of these methods, I add zero-mean, time-scaled Gaussian noise in the XYH directions of the vehicle odometry measurements to simulate a state estimate from a vehicle with a consumer grade IMU and no DVL: $\mathcal{N}\left(0, 0.02 \text{ m/s}\right)$ in the XY directions and $\mathcal{N}\left(0, 0.02 \text{ rad/s}\right)$ in the yaw / heading direction. The four localization methods are as follows:

- **Dead reckoning** - Using the noisy odometry measurements.
- **Li modified** - I consider a modified version of the method proposed by Li et al. [67]. In the original work, Li et al. perform an optimization using the original ASFM formulation [46] using cliques of 3 imaging sonar frames. This framework also includes the odometry measurements in the ASFM optimization, thereby double counting the odometry information. I use this same framework on pairs of sonar frames but utilize the non-parametric landmark representation, which is necessary to prevent the optimization from becoming too degenerate to solve.
- **ASFM2** - The same as Li modified but without the odometry measurements in the optimization, which eliminates the double-counting of the vehicle odometry information.
- **Proposed** - My novel, fully degeneracy aware method detailed in Section 4.2.

As my test tank environment consists of very smooth surfaces and lacks any naturally occurring features, I added features artificially. I placed an aluminum frame with small stacks of magnets attached to the bottom in the water near the surface. The magnets protrude from the bottom of



(a) (b) (c) (d) (e)

Figure 4.7: (a) Sample polar coordinate sonar image. The magnet features may be seen clearly by the eye. The section of high intensity at the top of the image is the test tank wall. (b) Adaptive thresholding creates a binary image. (c) Using various criteria on the size, shape, and distribution of the blobs, blob detection is able to identify most of the magnets as features of interest without falsely detecting features on the tank wall. (d) Features from this frame are matched with features from a previous keyframe (e) using the joint compatibility data association algorithm described in Section 4.2.3.

the frame and are visible to the sonar when the sensor is approximately level with the frame. The features are detected in the sonar images by using adaptive thresholding and blob detection, which is shown in Figure 4.7.

I recorded two datasets, 6 and 18 minutes in duration, in which the vehicle repeats a rectangular trajectory in the xy plane. The features are visible only when the vehicle is near one particular corner of the rectangle. The AprilTag fiducials are also visible only near this corner of the trajectory. Vehicle and sonar poses are added to the pose graph at least every 2 seconds, and loop closures are added between sonar poses when a positive association is made between sonar frames with at least 5 matched features. The oldest compatible sonar pose is preferred when making a loop closure, to provide longer time scale loop closures. To prevent adding unnecessary loop closures, a minimum time difference of 1 second is required in order to add a loop closure constraint to the pose graph. Figure 4.6 shows a top-down view of the trajectories of the dead reckoning and proposed pose graph solution. While the dead reckoning state estimate drift from the ground-truth, the pose graph solution corrects drift by adding loop closures at one corner of the rectangular trajectory. Note that I only consider poses in which at least one AprilTag is observed for the ground-truth trajectory, so that it is not affected by drift. Therefore, only poses near the top-left corner of the trajectory are shown.

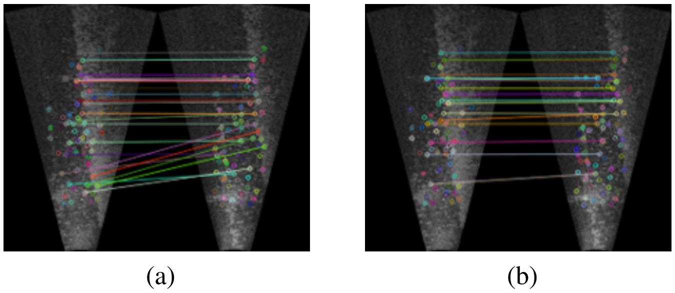

(a) (b)

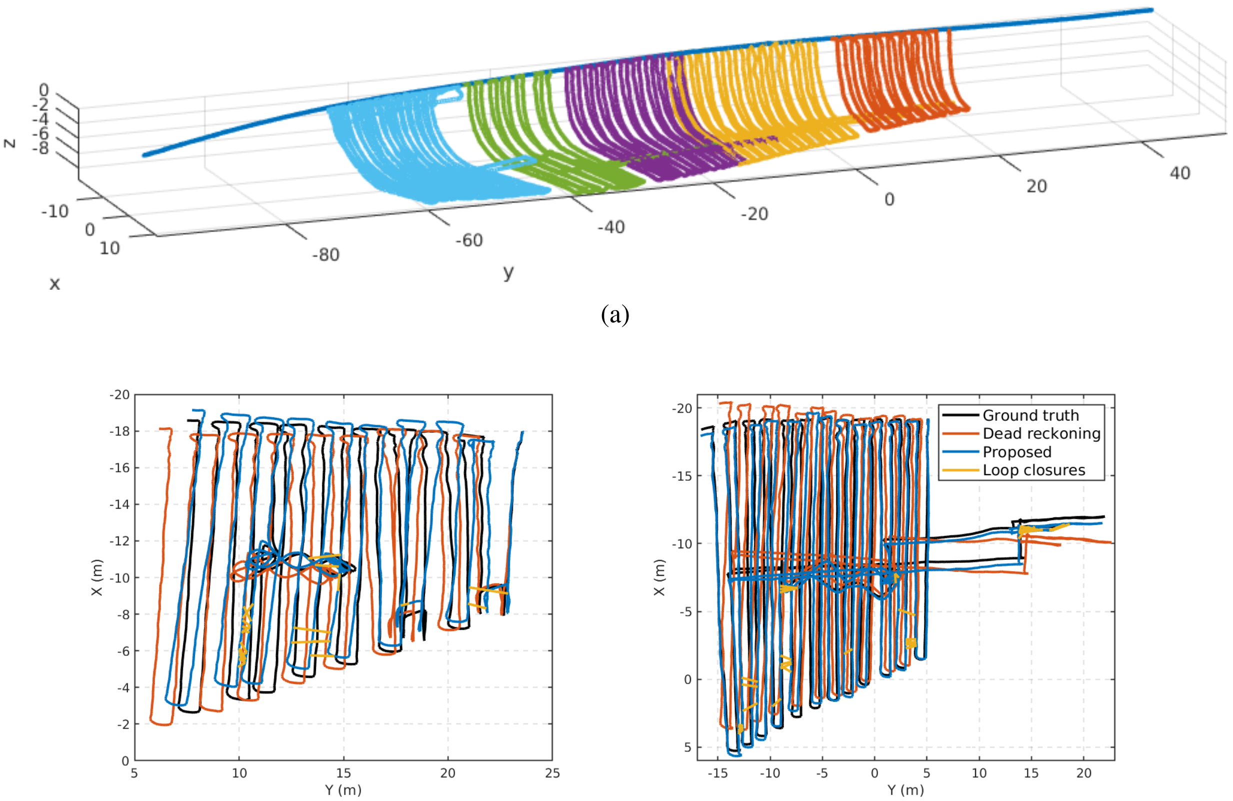
Figure 4.8: (a) Feature matches between two frames in a loop closure proposal clique, as determined by Li et al [67]. (b) My proposed joint compatibility-based feature association method rejects several incorrect feature matches, resulting in a significantly more reliable set of feature correspondences.

Table 4.1 shows the absolute trajectory error (ATE) [105] of the four evaluated methods. The Li modified method may actually increase overall localization error, due to the degeneracy of the ASFM optimizations and double-counting the noisy odometry measurements. However, ASFM2 degrades localization accuracy compared to dead reckoning due to the degeneracy of the pose transformation in some of local optimizations. Finally, my proposed method decreases the localization error compared to all other methods as I have taken proper care to solve the bundle adjustment problem in a degeneracy aware fashion and only provide constraints in the well-constrained directions, without double-counting odometry measurements.

4.3.3 Experimental: ship hull

To demonstrate its applicability to real-world applications, I test my pose graph localization algorithm on the ship hull inspection datasets presented by Li et al. in [67]. I compare my pose graph optimization method to both dead reckoning localization as well as Li et al.'s proposed approach. In [67], the authors generated a dead reckoning trajectory by sequentially adding noise in all degrees of freedom to the ground truth odometry measurements, causing the state estimate to drift in all directions. To more accurately model the HAUV's dead reckoning-based state estimate, I add relative noise between poses in the X, Y, and yaw directions and noise to global observations in the Z, pitch, and roll directions [114].

I utilize several components of the frontend presented in [67] to allow for a direct comparison of my proposed work's contributions. First, I only consider sonar images that are deemed sufficiently salient for potential loop closures. I also utilize the same A-KAZE features that are detected by Li's method. While Li's method utilizes an information-gain approach to sampling poses for potential loop closures, I simply uniformly sample poses that are close to the current



(a)

(b)

(c)

Figure 4.9: (a) Isometric view of the six ship-hull datasets, each plotted with a distinct color. (b) and (c) show sample ground truth, dead reckoning, and SLAM trajectories for datasets 2 and 3, respectively.

Ship hull localization error

Mission		1	2	3	4	5	6
Error in X [m]	DR	0.587	0.354	0.662	**0.234**	0.357	0.783
	Li	0.606	**0.270**	1.595	1.367	0.547	1.500
	Prop.	**0.579**	0.603	**0.389**	0.427	**0.356**	**0.457**
Error in Y [m]	DR	0.352	0.687	0.565	0.406	0.414	0.496
	Li	0.350	0.771	0.615	0.489	0.530	1.625
	Prop.	**0.344**	**0.291**	**0.370**	**0.288**	**0.285**	**0.484**
Error in yaw [degrees]	DR	0.383	**0.579**	2.842	1.803	3.177	**1.918**
	Li	0.392	0.852	3.392	2.941	2.792	4.149
	Prop.	**0.381**	1.21	**2.029**	**1.479**	**1.647**	1.998

Table 4.2: Localization errors of dead reckoning (DR), the method of Li et al. (Li) and my proposed method (Prop.) on the six ship hull datasets presented in [67]. Each error metric denotes the mean error of all poses in the trajectory over 10 trials of each dataset. My proposed method achieves the lowest error on 14 out of 18 metrics.

pose for potential loop closures. I use the feature matches resulting from Li's method, which utilizes descriptor and geometric information, as input to my joint compatibility feature association algorithm to further refine the matches. This helps eliminate outlier matches that are accepted by Li's method, as depicted in Figure 4.8. Finally, Li et al. propose generating a loop closure by using a clique of three sonar images in a local ASFM optimization. This is done in order to decrease the degeneracy of the optimization, making it more likely to converge to a stable solution. While my degeneracy-aware solution makes this clique formulation unnecessary, I still consider cliques of three sonar images for loop closures, but I treat each clique as two pairs (1-2 and 1-3). If at least seven features are matched between both pairs of images, I perform two separate acoustic bundle adjustment optimizations and add both resulting constraints into the overall pose graph.

Table 4.2 shows the localization error metrics used to evaluate (1) dead reckoning localization (2) the method of Li et al. and (3) my proposed method on the six ship hull datasets. I consider the error of each pose in the trajectory in the global X, Y, and yaw directions, as these are the directions that drift with dead reckoning. My method significantly decreases the localization error compared to dead reckoning and the method of Li et al. in almost all cases. The method of Li et al. often increases the error due to the degeneracy of the ASFM optimizations, despite taking multiple steps to alleviate this, including using the clique-based formulation. Additionally, my method achieves lower error despite making significantly fewer loop closures in comparison to Li et al. (fewer than 100 compared to over 200 on average). The reduction in the number of loop closures is attributed to my joint compatibility feature association framework, which rejects a large number of potential loop closures due to poor feature matching.

Figure 4.9 visualizes top-down views of several trajectories resulting from dead reckoning and my proposed method compared with the ground truth. Nearly all of the loop closures are made between consecutive passes in the lawnmower pattern of the trajectory, which limits the amount of drift that can be corrected. Additionally, the ship hull generally lacks distinctive acoustic features, which makes it difficult to establish sufficient correspondences to perform a loop closure. While the ship hull setting may not be the ideal test case for my acoustic bundle adjustment algorithm, these results demonstrate the advantages of my formulation of acoustic bundle adjustment over previous attempts in real-world setting.

4.4 Conclusion

In this Chapter I have presented a novel solution to the feature-based imaging sonar bundle adjustment problem that emphasizes accurate pose estimation. I focus on analyzing the case of pairwise bundle adjustment, but my framework is easily applicable to systems with three or more sonar viewpoints. I also propose a pose graph framework that efficiently combines odometry measurements with pose-to-pose constraints derived from my two-view sonar bundle adjustment algorithm. The pose-to-pose constraints may be added for local or large–scale loop closures to correct drift in the trajectory that inevitably accumulates with a dead reckoning based localization algorithm. My two-view bundle adjustment algorithm is evaluated extensively in simulation and is proven to outperform previous algorithms [67, 117]. I use real-world test tank and field trial experiments to demonstrate the effectiveness of my pose graph algorithm in correcting drift that accumulates from the vehicle odometry.

It is clear that the main limiting factor of this work is achieving accurate and robust feature detection and correspondence from multiple viewpoints. This is fundamentally a more challenging problem for acoustic sonar sensors than optical cameras due to the image formation process and the poor signal to noise ratio. This should still be considered an open research topic, and further advancements may significantly improve the performance of my acoustic bundle adjustment algorithm in environments where distinctive point features are present.

The n-view acoustic bundle adjustment algorithm described in this chapter is technically an algorithm for SLAM, although I have presented it particularly for the purpose of localization. While the solution does generate a sparse map of 3-D point landmarks, this sparse map is generally not useful for other tasks that an AUV may need to carry out. On the other hand, dense 3-D maps are often critical for general robot autonomy, as they can enable real-time collision-free planning and thorough inspection. The remainder of this dissertation is oriented towards generating such maps with the imaging sonar sensor.

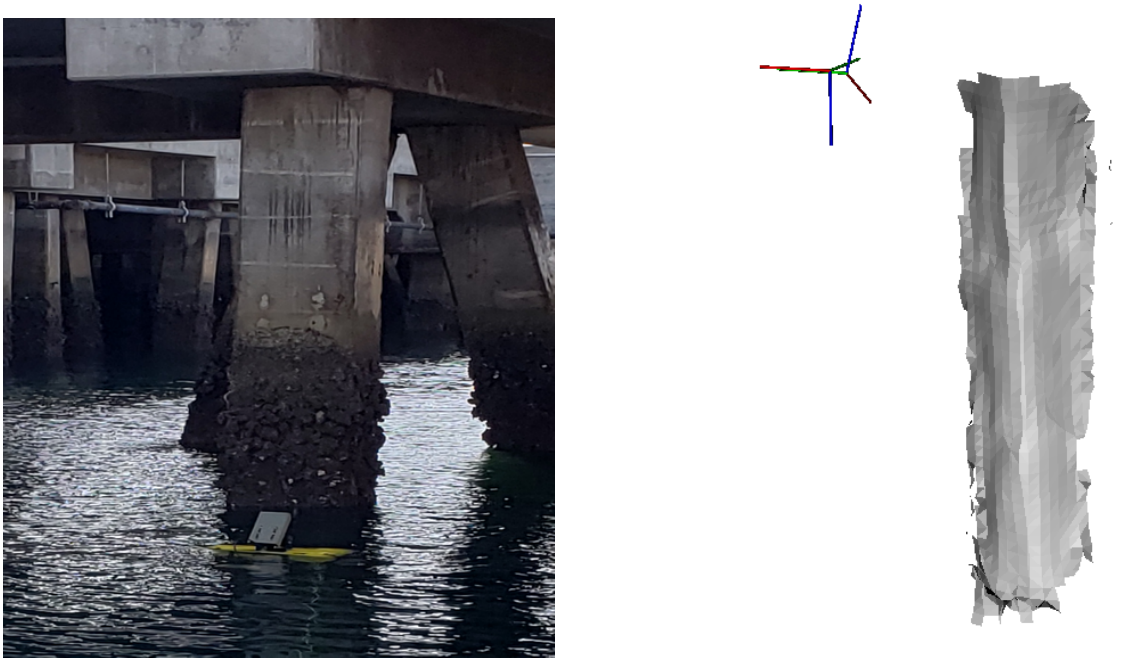
Chapter 5

Generative models

5.1 Introduction

Most of the prior works on dense sonar mapping covered in Chapter 2.2.2 attempt to create 3-D models of objects using multiple images taken from known poses. Space carving [7, 11, 39, 40] takes a cautious approach, using the so-called "negative information" (low intensity pixels) to carve out free space between the sensor and the first detected surface. On the other hand, occupancy grid mapping [112, 113] uses the "positive information" (high intensity pixels presumably generated by reflections from surfaces) to infer which portions of the volumetric scene are occupied by an object. Both of these frameworks require many redundant viewpoints in order to generate a reasonable model of the surfaces or objects of interest. In the field, there are often restrictions on how an underwater vehicle can move and orient its sensors. Due to this motion restriction, it is beneficial to develop methods for 3-D sonar mapping that can infer shape directly from a single image, or from multiple images taken from a relatively limited variety of viewpoints.

A body of work by Aykin et al. [10, 12, 39, 40], to the best of my knowledge, marks the first attempt to directly estimate the elevation angle of each pixel in a sonar image. This method is constrained to the scenario of objects lying on the seafloor. Upon detecting the seafloor plane and manually extracting object and shadow edges, the bottom and top 3-D contours of the object of interest may be easily triangulated. With these edges bounding the object's surface, the interior of the object is iteratively "filled-in" based on the generative sensor model and the measured sonar image intensities. With a stationary sensor and planar seafloor, robustness to noise may be increased by fusing multiple measurements and multipath artifacts may even be accounted for. This method has laid the groundwork for my proposed algorithm, in which I seek to apply generative model-based surface reconstruction to arbitrary scenes, not just seafloor mapping.



(a)

(b)

Figure 5.1: (a) The above-water portion of a pier piling, with the HAUV preparing to scan. (b) A mesh model of the 3-D reconstruction generated by my proposed algorithm. The coordinate axes represent the vehicle and sonar poses.

This chapter takes steps towards the ultimate goal of autonomous underwater mapping of arbitrary structures using imaging sonar. Specifically, I present:

- a general framework for reconstructing 3-D objects from a single sonar scan, by explicitly estimating the missing elevation angle using a generative sensor model
- a method of fusing surface measurements from multiple viewpoints to create a cohesive object model
- experimental results demonstrating the ability of my proposed algorithm to accurately reconstruct underwater piling structures using a robotic AUV platform.

5.2 Problem statement and background

The problem I wish to solve in this work is as follows. Given a set of polar coordinate sonar images, the poses from which they were taken, and a generative sensor model, produce a three-dimensional reconstruction of the imaged surfaces. While Chapter 1.4 describes *where* a imaged surface point or patch projects in a sonar image, of particular interest in this work is how the imaged surface contributes to the *intensity* of the corresponding pixel measurement. In this chapter I denote a sonar image pixel as $I(r, \theta)$, where r and θ are the range and azimuth measurements that correspond to the discrete image row and column, respectively.

Ideally, the pixel intensity is influenced only by the interaction of the sound waves with the imaged surfaces, although in reality there are multiple confounding factors. Assuming isotropic sound emission by the sensor, this ideal model can be expressed generally as

$$I(r, \theta) = \int_{\phi=\phi_{\min}}^{\phi=\phi_{\max}} \mathbf{1}(r, \theta, \phi)\, \Omega(r, \theta, \phi)\, d\phi \tag{5.1}$$

where $\mathbf{1}(r, \theta, \phi)$ is an indicator function denoting the existence of an imaged surface at the 3-D location and $\Omega(r, \theta, \phi)$ encodes how the sound is reflected by the surface and propagated back to the sonar [40]. Note that this model disregards multipath returns, in which the sound reflects off of multiple surfaces before returning to the sensor.

While a variety of reflection models have been used that consider specular and / or diffuse scattering, the specular component often may appear to be negligible due to the generally rough surfaces of underwater objects and the grazing incident angles often used with sonar sensors [12, 64, 72]. In this work I adopt a simple diffuse reflection model for all imaged surfaces, assuming each pixel images a single surface patch:

$$I(r, \theta) = k \cos^{m}(\alpha) \tag{5.2}$$

where k is a normalization constant, $1 \leq m \leq 2$, and α is the angle of incidence between the incoming acoustic beam and the surface normal of the patch. I assume that a time / range varying gain (TVG / RVG) has been applied to the raw image to correct for the spatial spreading of the sound waves. It is important to note that my proposed algorithm may utilize *any* reflection model, not just the particular one I have selected for my experiments.

5.3 Frontend - image processing

The frontend of the proposed method operates on each input sonar image individually. The two goals of this module are: (1) to denoise the sonar image and (2) to identify the pixels that correspond to surface measurements. Upon completing these steps, the denoised sonar image and the binary image mask corresponding to object surfaces may be passed to the backend for surface reconstruction.

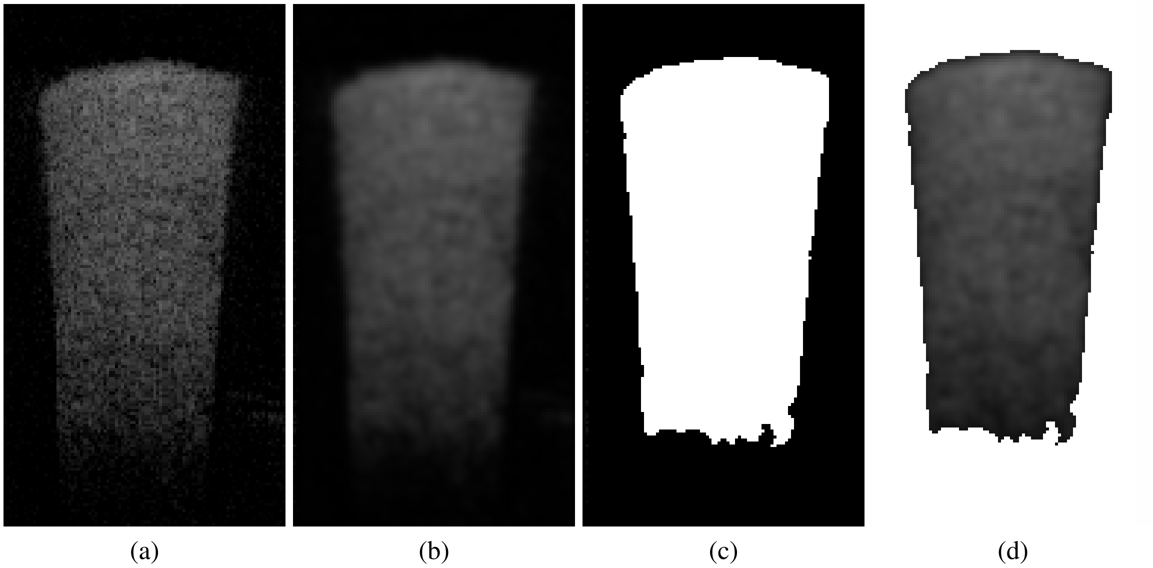


Figure 5.2: The stages of my frontend image processing pipeline, demonstrated on my test tank piling dataset. (a) The raw polar coordinate sonar image, (b) denoising using anisotropic diffusion, (c) the surface segmentation using MSER and (d) the binary surface mask applied to the denoised image.

5.3.1 Denoising

Sonar images suffer from significantly higher speckle noise than optical images. Previous attempts to denoise sonar images include averaging multiple images taken from the same viewpoint [78]. Since my algorithm is targeted for robotic underwater mapping, in which the vehicle and sensor poses may not be precisely known or controlled, I seek a method of denoising each image individually. To this end, I adopt the procedure of anisotropic diffusion [90]. This step blurs the image as in a standard Gaussian blurring process, but preserves distinct edges in the image by scaling the diffusion in proportion to the inverse of the image gradient. This has been previously used with success as a denoising step before detecting point features for sparse SLAM systems [100, 117]. An example of the denoising process applied to a sonar image of a mock-up piling (a rectangular prism shape) is shown in Figures 5.2a and 5.2b.

5.3.2 Surface segmentation

Convolutional neural networks (CNNs) have rapidly become the de facto approach to image segmentation in the field of computer vision [69]. Their emergence has been made possible in part due to very large amounts of training data available. Recent years have seen CNNs successfully applied to sonar images for various tasks, including crosstalk removal [106], object detection [57], and global context perception [24, 25]. However, collecting a sufficient quantity of sonar images for training is a significant challenge to the application of these methods for underwater sonar perception. While I perceive the future of surface segmentation to lie in the field of machine learning, I leave this approach to future work.

I take a simpler approach to surface segmentation by finding maximally stable extremal regions (MSER) [74] on the denoised image. This is a blob detection algorithm that I use to find large connected components with gradual changes in pixel intensity. Each segmented component corresponds to distinct, continuous surface imaged by the sensor. An example of the MSER algorithm applied to a denoised sonar image is shown in Figure 5.2b - 5.2d. I denote the resulting binary image surface mask as $M(r, \theta)$, where $M(r_i, \theta_j) = 1$ denotes a detected surface.

5.4 Backend - surface reconstruction

In order to generate a 3-D reconstruction from a sonar image and corresponding surface mask, several assumptions must be made. I assume that the sonar and scene geometry are configured such that each pixel (elevation arc) images a single surface patch. For simply shaped objects, this assumption holds true when the sonar is positioned at a grazing angle. I also assume that for a continuous surface, the elevation angle along a particular bearing angle either increases or decreases monotonically as the range increases. A violation of this assumption would cause a self-occlusion, and the corresponding pixels would presumably not be classified as surface pixels by the frontend of my algorithm.

My approach is inspired by [10], which uses the 3-D locations of the leading and trailing object edges as initialization to iteratively refine the 3-D object reconstruction and update the generative model normalization parameter. However, if the full generative model is known a priori, a 3-D reconstruction can be obtained using just one object edge as initialization.

5.4.1 Edge initialization

In this work, I focus my experiments on reconstructing underwater piling structures, which are long columns that support structures such as bridges or piers. I take advantage of the fact that a piling spans the entire elevation field of view of the sonar sensor, which is depicted in Figure 5.3. As long as the sonar is tilted at least $\phi_{\max}$ degrees from perpendicular to the piling, each pixel's elevation arc will image only one surface patch. Furthermore, the closest detected surface patch in each image column (discrete bearing angle bin), may be easily determined to lie at elevation ϕ_{min}. The same principle may be applied to determine that the 3-D position of

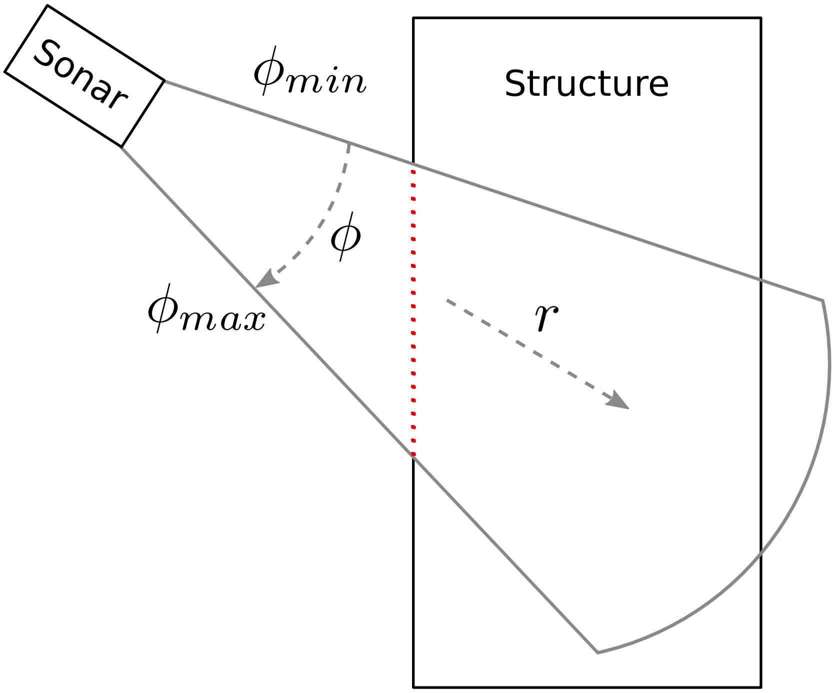


Figure 5.3: A side view of a sonar imaging a piling. A 2D cross-section of the viewable frustum is depicted, which corresponds to a single bearing angle. The imaged area of the piling is shown in dotted red. This depicts how the elevation angle of the imaged surface increases or decreases monotonically with increasing range.

the trailing edge of the surface is at $\phi_{\max}$. However, for larger tilt angles, the structure may not span the bottom edge of the elevation frustum. For the purposes of this work, I utilize the less restrictive single edge initialization, which may be applied to a variety of settings apart from piling inspection.

5.4.2 Dense 3-D reconstruction

With the leading object edge located at ϕ_{min}, the remainder of the surface points in the image can be filled in using constraints from the generative model. I follow the general procedure described by Aykin et al. [10] but can reconstruct the points in a single pass through the image, without iteratively updating the normalization parameter.

The single pass through the image $I\left(r,\theta\right)$ is performed row-by-row, beginning with the first row r_0. I use the shorthand $I_{i,j} := I\left(r_i,\theta_j\right)$ and $M_{i,j} := M\left(r_i,\theta_j\right)$, and use $\mathbf{p}_{i,j}$ to denote the 3-D point in the sensor frame corresponding to the pixel at $I_{i,j}$. Pixels in row r_i are stepped through column-by-column. If $M_{i,j}$, $M_{i+1,j}$, and either of $M_{i,j-1}$ or $M_{i,j+1}$ are identified as surface pixels, then the constraints from the generative sensor model can be utilized to compute the elevation angle of point $I_{i+1,j}$.

Assuming the elevation angle (and therefore 3-D location) of $\mathbf{p}_{i+1,j}$ is known, an approximate surface normal of the patch at $I_{i,j}$ may be computed using the cross product of the neighboring 3-D points:

$$\mathbf{v}_{ij} = \mathbf{d}_{ij} \times \mathbf{e}_{ij} \tag{5.3}$$

$$\hat{\mathbf{n}}_{ij} = \frac{\mathbf{v}_{ij}}{\left\|\mathbf{v}_{ij}\right\|_2}. \tag{5.4}$$

Here, $\mathbf{d}_{ij} = \mathbf{p}_{i+1,j} - \mathbf{p}_{i,j}$ and $\mathbf{e}_{ij} = \mathbf{p}_{i,j-1} - \mathbf{p}_{i,j}$ or $\mathbf{e}_{ij} = \mathbf{p}_{i,j+1} - \mathbf{p}_{i,j}$, depending on which pixel in neighboring columns is identified as a surface pixel. Then using the vector corresponding to the ray of incident sound from the sensor $\hat{\mathbf{p}}_{ij} = \mathbf{p}_{ij}/\left\|\mathbf{p}_{ij}\right\|_2$, the angle of incidence may be computed as:

$$\alpha = \operatorname{acos}\left(\left|\hat{\mathbf{n}}_{ij} \cdot \hat{\mathbf{p}}_{ij}\right|\right). \tag{5.5}$$

Then the generative model in Eq. 5.2 is used to compute the model-predicted image intensity for the given elevation angle.

I perform a search of discrete elevation angles taken at uniform intervals from the range of feasible elevation angles: $\left[\phi_{i,j}, \min\left(\phi_{i,j} + \Delta\phi_{\max}, \phi_{\max}\right)\right]$, where the maximum change in elevation angle from pixel to pixel $\Delta\phi_{\max}$ may be manually tuned. I set $\phi_{i+1,j}$ to the elevation angle with the smallest absolute error between the actual image measurement and model-predicted intensity. If there are not sufficient neighboring surface pixels to solve for $\mathbf{p}_{i+1,j}$, I assume that $\phi_{i+1,j} = \phi_{i,j}$. This procedure proves to work quite well for continuous surfaces, but may fail for images with more complex, disjointly segmented shapes.

If the trailing object edge is known as well as the leading edge, then the parameters of the generative model k and m may be iteratively refined until the trailing object edge determined by integrating the generative model aligns with the known trailing edge, as in [10]. I leave this to future work, however, as the trailing edges of the pilings in my experiments are difficult to consistently detect and reconstruct.

5.4.3 TSDF integration

Given the high levels of noise in the sonar image that remain after denoising and various unmodeled effects, the 3-D point cloud generated by a single image may be quite noisy and inaccurate, even for simple structures such as pilings.

A truncated signed distance field (TSDF) is a volumetric map representation that has been used to generate high quality surface reconstructions from multiple noisy 3-D scans generated by RGB-D cameras [84] and laser scanners [23]. Since point measurements typically correspond to rays of light or a laser beam, voxels are updated by stepping along the ray from the sensor to the point measurement. Each voxel tracks a value that is updated with a weighted, signed distance of the voxel from the surface along the line of sight. The zero crossings denote the estimated surface and a point cloud or triangle mesh may be generated from the TSDF.

While the TSDF is a quite intuitive choice of map representation for RGB-D sensors, in which each pixel corresponds to a ray-based surface observation, it is not so obvious a choice for the imaging sonar, where pixels correspond to elevation arcs. However, it is a good fit for my framework in which dense 3-D surface measurements are generated for each pixel. Furthermore, each surface measurement is made by acoustic waves propagating along the ray between the sensor and surface patch. This allows us to use the standard TSDF ray casting updates to fuse multiple surface measurements into a single cohesive global model.

5.5 Experimental results

To evaluate the proposed system, I quantitatively and qualitatively compare my 3-D reconstructions from real-world test tank and field datasets to the 3-D reconstructions resulting from the two previously discussed baseline methods: space carving (SC) and occupancy grid mapping (OGM). SC and OGM are considered the leading state-of-the-art algorithms for real-time 3-D

	MAE (m)			RMSE (m)		
Dataset	SC	OGM	Mys	SC	OGM	Mys
Tank piling	0.033	0.038	**0.0176**	0.035	0.047	**0.022**
Field piling	0.136	0.152	**0.039**	0.168	0.207	**0.047**

Table 5.1: Quantitative evaluation of three-dimensional object reconstructions from the test tank experiment. The two metrics I use are mean absolute error (MAE) and root mean square error (RMSE). My surface reconstruction method results in considerably more accurate surface models than the baseline methods.

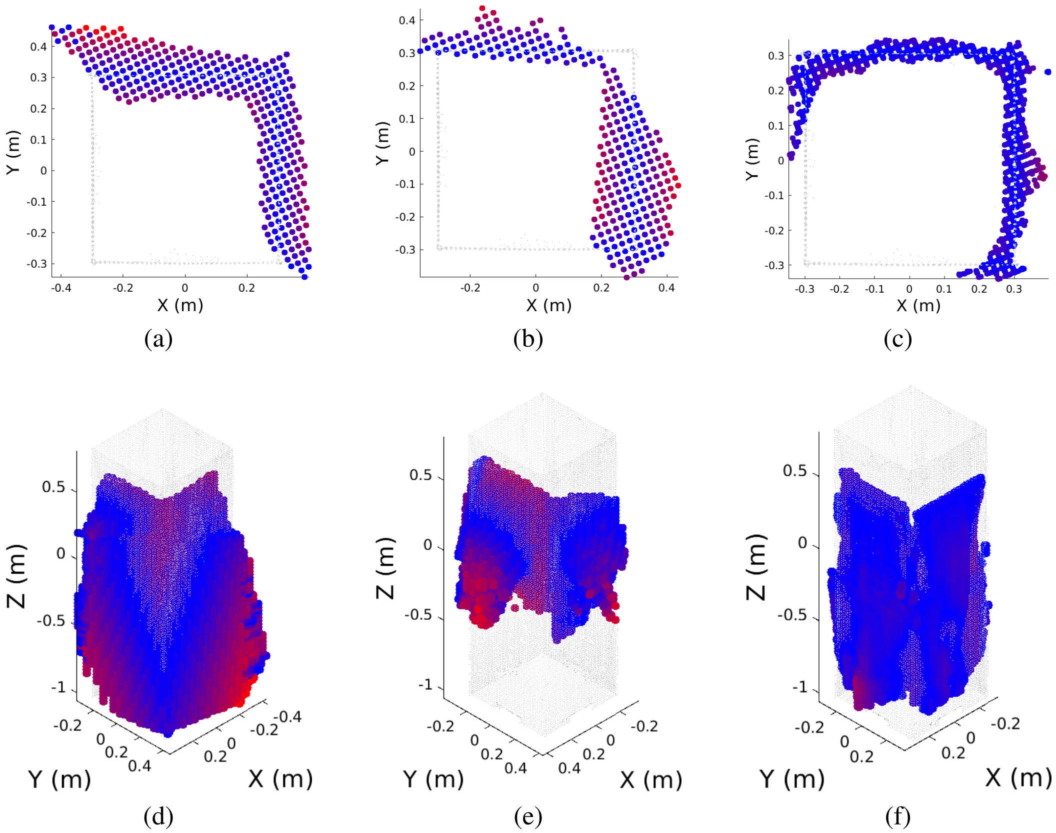
imaging sonar reconstruction. I compare my proposed method to my own implementations of SC and OGM, which use fixed-size voxel grids for mapping. My implementation of SC uses an OGM-like tracking of the probability of occupancy, rather than minimum-filtering. This allows for using a tunable threshold to acquire object points and to make the algorithm more robust to errors in the sensor pose and image segmentation. For both baseline methods, the threshold to distinguish occupied from free voxels was tuned to generate the best reconstruction. My proposed framework uses Voxblox [85] which implements spatially-hashed voxels [60, 84] for memory-efficient TSDF integration. For my proposed reconstruction method, I discard surface measurements from the outer 20% of columns on either side of the image, as the image intensity does not adhere to the generative model well due to anisotropic emission by the sonar.

The imaged targets in these experiments are a mock-up and a real-world piling. While these objects consist of approximately planar segments, my proposed method does not make any planarity assumptions.

Both the test tank and field datasets were recorded using a SoundMetrics DIDSON imaging sonar [104] mounted on a Bluefin Hovering Autonomous Underwater Vehicle (HAUV) [36]. An actuator allows the DIDSON to tilt through a $90°$ range of motion. A spreader lens is used to increase the elevation aperture $\phi_{\max} - \phi_{\min}$ from $14°$ to $28°$. Poses are acquired from the proprietary vehicle navigation and the actuator encoders. The vehicle odometry is highly accurate for short-term localization but inevitably drifts over time. For this reason, I only image two faces of each piling – the drift that accumulates from dead reckoning navigation while circumnavigating the entire piling is too great for mapping with known poses.

5.5.1 Test tank experiments

I imaged a mock-up piling of dimensions approximately 0.61 m x 0.61 m x1.83 m. I image two faces of the piling, tilting the sonar between approximately $20°$ and $50°$ from the horizontal,



(a) (b) (c)

(d) (e) (f)

Figure 5.4: 3D reconstructions of the mock-up piling from my test tank experiment. The gray cloud depicts the ground-truth model generated by a survey laser scanner. Colored points are the sonar reconstruction, with blue encoding low point-to-plane alignment error and red encoding high error. (a) - (c) Top-down views of the reconstructed point clouds of the **SC**, **OGM**, and **Proposed** algorithms, respectively, compared to the ground truth model. (d) - (f) Isometric views of the same reconstructions.

which allows the sonar to cover the entirety of the piling, except the small portion that passes through the top of the viewing frustum. Voxel grids for all reconstruction methods use a voxel size of 2.5 cm, including the TSDF, to produce reconstructions in real time at 5-10 frames per second. The generative model parameters $k = 0.37$ and $m = 1$ were used to model the acoustic reflection properties of the mock-up piling. Upon generating the 3-D reconstruction, the surface points from each model are extracted and aligned to a ground truth model with ICP. The ground truth model was obtained using a FARO Focus3D survey laser scanner.

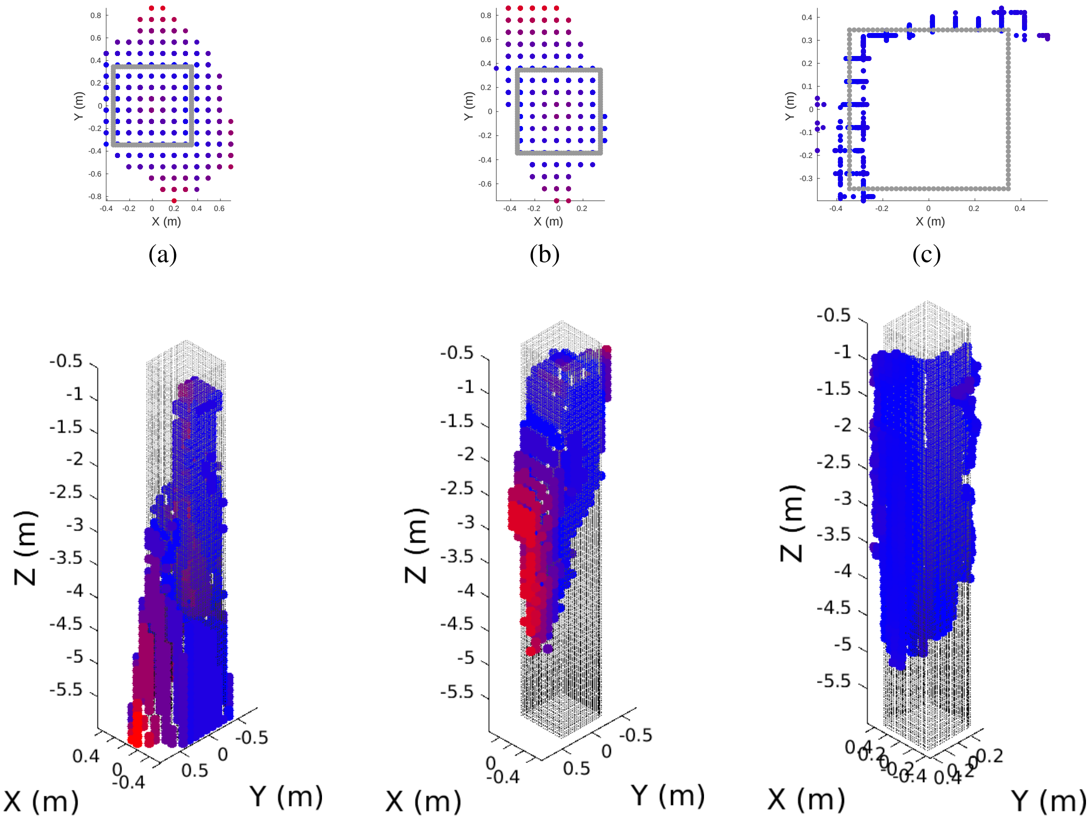
Figure 5.4 shows top-down and isometric views for the three evaluated reconstruction methods. The point clouds are colored according to the point-to-plane error evaluated during ICP alignment, with the same color scale used across all three models. The top-down views show how SC and OGM fail to “carve out” space in front of each piling face. This causes the reconstructed surface to bulge out towards the bottom of the piling. On the other hand, my proposed method fuses the estimated 3-D point clouds from each input image to generate a rather accurate estimate of the surface. While some inaccuracies in the resulting surface exist, there is no prominent bulge or spreading of the surface towards the bottom of the piling - both faces of the reconstructed piling are quite close to vertical and planar.

Furthermore, I quantitatively evaluate the error of the resulting models using mean absolute error (MAE) and root mean square error (RMSE) of the point-to-plane error metric. Table 5.1 shows that my method significantly increases the accuracy of the surface estimate compared to SC and OGM.

5.5.2 Field experiments

As the ultimate goal of this work is to enable robotic mapping and inspection of real-world environments, I conducted field tests to reconstruct a pier piling in a harbor environment. Since the piling is larger than the one used in my test tank, voxel grids for all algorithms use a voxel size of 10 cm to maintain real-time reconstruction. The generative model parameters $k = 0.28$ and $m = 2$ were used to model the acoustic reflection properties of the piling. A photo of the piling and the mesh reconstruction generated by my algorithm are shown in Figure 5.1. As a ground-truth model is not available for such a piling, I manually measured the width of the piling underwater (69 cm), and assume a purely rectangular prism shape. The shape of the piling is somewhat distorted by biofouling, as is visible in the photo, but the rectangular prism model remains a rather accurate estimate. Similar to the tank piling, I imaged two faces of the piling, as the vehicle state estimate drifted too much for a full circumnavigation.

Figure 5.5 shows the same top-down and isometric views as for the tank piling dataset. SC and OGM clearly cannot accurately reconstruct the piling surface below a depth of 2.5 m, while



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.5: 3D reconstructions of the real-world piling in the field. The gray-scale cloud depicts the ideal model according to my measurements of the piling. Colored points are the sonar reconstruction, with blue denoting low point-to-plane alignment error and red denoting high error. (a) - (c) Top-down views of the reconstructed point clouds of the **SC**, **OGM**, and **Proposed** algorithms, respectively, compared to the ground truth model. (d) - (f) Isometric views of the same reconstructions.

my algorithm reconstructs a rather planar surface all the way to a depth of 5 m. Table 5.1 demonstrates the quantitative improvement in my reconstruction's accuracy, as evaluated against the ideal piling model.

It should be noted that while SC and OGM may theoretically be able to generate surface estimates of these simple structures with accuracy comparable to the proposed method, this would require obtaining a much wider variety of viewpoints. For real-world experiments, this would mean longer mission times, higher state estimate uncertainty, and potentially infeasible sensor-vehicle configurations.

5.6 Conclusion

In this chapter I have presented an algorithm for mapping with known poses using imaging sonar and a generative sensor model. Using very general prior information about the environment, the 3-D location of the leading object edge may be accurately determined. Using this edge as initialization, the generative model may be used to fill-in the rest of the object surface. This single image reconstruction procedure follows the spirit of optical shape from shading [93]. Using known sensor poses, the point clouds resulting from each input image are fused in a global model using a TSDF to smooth the surface estimate. I have demonstrated experimentally that my proposed method can outperform the existing state-of-the-art algorithms in terms of accuracy and that it requires fewer viewpoints and images to generate a surface model.

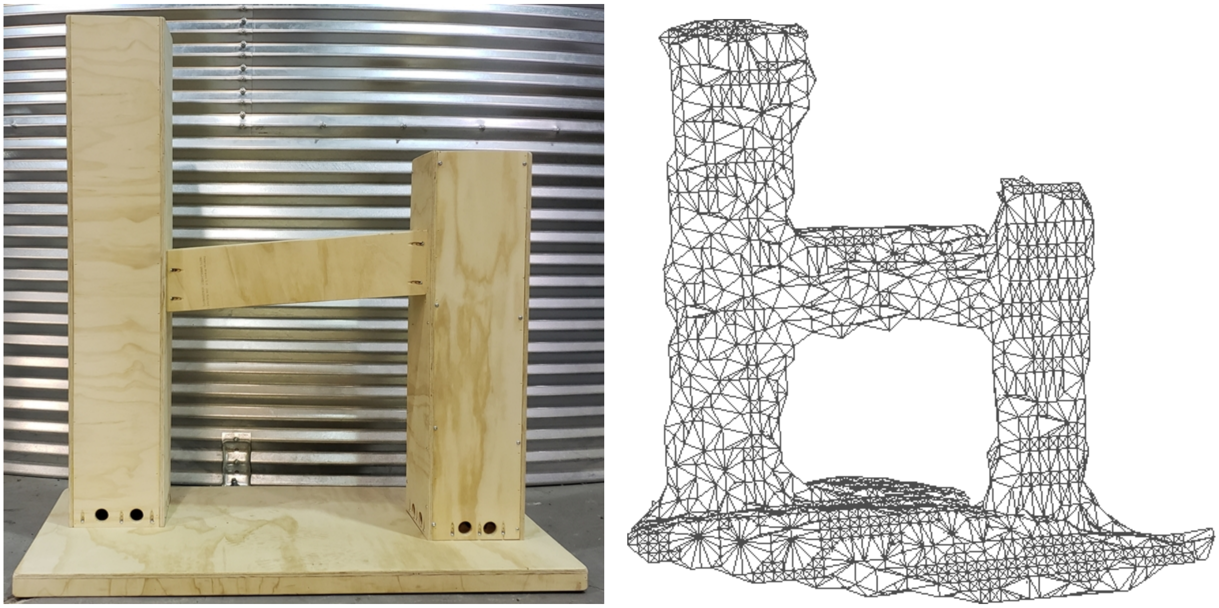
Chapter 6

Volumetric albedo

6.1 Introduction

Several recent works on imaging sonar mapping seek to infer or disambiguate the 3-D structure of a scene using particular motions. Space carving [7, 11] and occupancy grid mapping [112, 113] both rely on rolling the sensor around its forward-facing axis at different positions in order to generate an object surface model or occupancy model. Guerneve et al. [40] approximate the elevation aperture as linear, effectively modeling each pixel's elevation arc as a vertical line segment parallel to the sensor's z-axis. By restricting motion to pure translation along the z-axis, 3-D volumetric reconstruction is framed as a blind deconvolution with a spatially varying kernel that captures the surfaces' reflection properties. Assuming uniform reflectivity, the problem is reduced to a linear least squares problem, solved with ℓ_1 regularization by means of an interior point method [59]. This approach is similar to the method proposed in this chapter, but the linear approximation and sensor motion restriction severely limit its practical application. My proposed method may be viewed as a generalization of, and improvement over, this approach, as we do not introduce linearization errors or place any restrictions on the sensor motion. Additionally, unlike occupancy grid mapping, which updates all voxels independently, our proposed method does not assume conditional independence of measurements. Rather, all correlations between voxels due to pixel measurements are maintained and a joint optimization is carried out to solve for a scene model that more accurately reflects the measurements.

In looking to the field of computer vision for inspiration for multiview 3-D imaging sonar reconstruction, the problem of multiview stereo (MVS) seems like a naturally starting point. However, MVS techniques rely on photometric consistency to triangulate surface patches, an assumption for optical cameras that does not apply to imaging sonar sensors. Rather than looking to classical MVS techniques to generate 3-D reconstructions with imaging sonar, we draw inspi-



(a)

(b)

Figure 6.1: (a) Photograph of the test structure that was custom made for our test tank experiments. (b) A 3-D mesh reconstruction generated from a pointcloud extracted from our volumetric albedo scene representation.

ration from the problem of non-line-of-sight (NLOS) reconstruction. This problem, discussed in detail in Chapter 2.2.3, consists of reconstructing a scene from indirect active illumination, by looking through a diffusing medium or imaging based on reflections off of a matte surface, such as a wall.

The specific contributions of this chapter are as follows:

- a novel volumetric albedo framework for imaging sonar reconstruction, inspired by recent works in the field of NLOS scene reconstruction
- a solution to the proposed framework utilizing a convex optimization algorithm that allows for incorporating a variety of priors
- evaluation of our proposed reconstruction algorithm on simulated and real-world datasets.

6.2 Volumetric albedo framework

In this section we discuss how imaging sonar reconstruction with known poses is an inverse problem with the same structure as NLOS reconstruction and present a volumetric albedo framework for 3-D reconstruction. First, we summarize the important characteristics of NLOS reconstruction.

6.2.1 NLOS reconstruction as volumetric albedo

Consider the NLOS scenario of reconstructing a scene around a corner by imaging diffuse reflections off of a wall. A point $\mathbf{l}$ on the line-of-sight (LOS) planar wall is illuminated, often by a laser pulse, at time $t = 0$. Light is scattered in all directions according to some unknown bidirectional reflection distribution function (BRDF), and bounces off of some NLOS surfaces at various points in time, and possibly multiple surfaces at the same instant in time. The third and final bounce occurs when the light reflects off the LOS wall again, and back toward the sensor. The sensor detects light reflected from a distinct sensing point along the LOS wall $\mathbf{s}$, resulting in the so-called 5D light transient $i\left(t; \mathbf{l}, \mathbf{s}\right)$, which is the intensity (or photon count) as a function of time and points $\mathbf{l}$ and $\mathbf{s}$. Fourth and higher-order bounces are usually ignored due to the difficulty of detecting them and for the sake of simplifying the imaging model.

In the confocal case, the illumination and sensing point on the wall are the same – that is, $\mathbf{l} = \mathbf{s}$, resulting in a 3-D light transient as a function of t and $\mathbf{s}$ [88]. Since the speed of light and relative position of the sensor to $\mathbf{l}$ are known, a transient measured at point $\mathbf{l}$ is equivalent to a series of *range-only* measurements taken by a virtual sensor located at $\mathbf{l}$, where the intensities of the measurements are determined by the reflectance properties and geometry of the NLOS surface. The azimuth and elevation angles of the measured surfaces are lost due to scattering. Thus, NLOS reconstruction is an ill-posed inverse problem wherein multiple 1-D range measurements must be sampled from a variety of locations on the LOS scene in order generate constraints on the azimuth and elevation angles of the NLOS surfaces.

Due to the complexity of this inverse problem, the scene is often modeled as a volume wherein each point $\mathbf{p}$ is described by a directionally uniform albedo $\rho\left(\mathbf{p}\right)$. This simplified model greatly reduces the complexity of the inverse problem, as compared to attempting to account for surface normals and BRDF [42]. Under this assumption, the forward measurement model for the transient in the confocal case is

$$i\left(t; \mathbf{l}\right) = \iiint\limits_{\Omega_{\mathbf{p}}} \rho\left(\mathbf{p}\right) \frac{\delta\left(\mathbf{p} \in \mathcal{S}_{ct}\right)}{\left\|\mathbf{p} - \mathbf{l}\right\|_2^4} \, \mathrm{d}\mathbf{p} \tag{6.1}$$

where $\Omega_{\mathbf{p}}$ denotes the 3-D volume of the NLOS scene, $\mathcal{S}_{ct}$ denotes the sphere of radius ct centered at $\mathbf{l}$, and the quartic term accounts for the spatial propagation of scattered light. For more details on the problem of NLOS reconstruction, we refer the reader to [3, 17, 41, 110].

6.2.2 Imaging sonar reconstruction as volumetric albedo

In the imaging sonar case, each column in an image is analogous to the 1-D series of range-only measurements in the NLOS scenario. However, the azimuth angle is disambiguated by an array of transducers, constraining measured surfaces to lie upon a 1-D elevation arc rather than a 2-D sphere. Thus, under the volumetric albedo model, the forward measurement model for the imaging sonar is given by

$$I(\theta, r) = \iiint\limits_{\Omega_{\mathbf{p}}} \rho(\mathbf{p})\, \delta(\mathbf{p} \in A_{\theta,r})\, \mathrm{d}\mathbf{p} \tag{6.2}$$

where $A_{\theta,r}$ denotes the 1-D elevation arc with limited aperture that corresponds to an image pixel (θ, r). Since the measurement is linear in the albedo, this model may be discretized as a linear system

$$\mathbf{b} = \mathbf{A}\mathbf{x} \tag{6.3}$$

where $\mathbf{b} \in \mathbb{R}_+^N$ is the vector representation of all N discrete image measurements $I(\theta, r)$ from all images, and $\mathbf{x} \in \mathbb{R}_+^{n_x n_y n_z}$ is the vector representation of the discretized albedo volume, with n_x, n_y, and n_z representing the size of the voxel grid in the corresponding dimensions. $\mathbf{A}$ is a sparse binary matrix that corresponds pixel measurements to voxels that lie on the corresponding elevation arc. If the sonar images are not pre-processed to compensate for the spatial spreading of sound, then a range-based gain may be applied to (6.2), which would scale entries in $\mathbf{A}$ accordingly. This linear system directly follows from the discretized albedo volumes used for NLOS reconstruction [3, 41, 42, 110]. However, the sonar linear systems are significantly sparser than those in the NLOS scenario, since each measurement corresponds to a 1-D elevation arc manifold, rather than a 2-D ellipsoidal or spherical manifold.

To compute the correspondence matrix $\mathbf{A}$, a forward projection (projecting center points of voxels into image pixels) or back projection (projecting pixels along their elevation arc into the voxel grid) procedure may be used. For our experiments, we use forward projection, noting that both projection procedures may be parallelized to improve computational efficiency.

Some of the first attempts to solve this large, sparse system in the NLOS case approximated the solution using backprojection:

$$\mathbf{x}_{\text{bp}} = \mathbf{A}^T\mathbf{b}. \tag{6.4}$$

A commonly used heuristic is to apply a filter after backprojection, such as the Laplacian filter, to sharpen the result [110]. My proposed method attempts to solve (6.3) via regularized optimization, which is inspired by similar works in the NLOS literature [41]. Several convolutional approximations have been proposed for the NLOS problem that greatly increase the computational efficiency of these optimization-based approaches [3, 88]. While similar approximations may be made in the imaging sonar case, we find that the sparsity of $\mathbf{A}$ and relatively limited resolution of the sonar sensor make standard optimization procedures much more efficient for imaging sonar reconstruction than for NLOS reconstruction.

A key shortcoming of this framework is that it does not capture the effects of occlusion. One possible way to address this is to only use low-intensity pixels in each image column from the shortest range until the first high-intensity pixel. These correspond to free space under the image formation model. Low-intensity pixels at ranges beyond the first imaged surface may correspond to actual surfaces in the scene that are occluded by surfaces closer to the sensor.

One of the benefits of this framework is that the albedo of the entire scene may be solved for jointly. This contrasts with prior works utilizing occupancy grid mapping [112] or minimum filtering [39], which update each voxel independently. Furthermore, formulating the forward sensing model as a sparse linear system enables the use of convex optimization methods to guarantee convergence to a global minimum.

6.3 ADMM optimization

The linear system derived from the volumetric albedo formulation of imaging sonar reconstruction may be solved using a least squares optimization:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \ \frac{1}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|_2^2 + \Gamma\left(\mathbf{x}\right) \tag{6.5}$$

where $\Gamma\left(\mathbf{x}\right)$ is a term that combines all priors or regularization terms. If no regularization is used, the solution is trivial to obtain but may be rather inaccurate. We propose utilizing three separate priors that are commonly used in the NLOS volumetric albedo literature [41]: non-negativity, weighted ℓ_1 regularization, and total variation regularization. These may be expressed as:

$$\begin{aligned} \mathbf{x}^* = \ & \underset{\mathbf{x}}{\operatorname{argmin}} \ \frac{1}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{b} \right\|_2^2 + \mathcal{I}_{\mathbb{R}_+}\left(\mathbf{x}\right) + \\ & \qquad \lambda_1 \left\| \mathbf{W}\mathbf{x} \right\|_1 + \lambda_{\mathrm{TV}} \left\| \nabla \mathbf{x} \right\|_1 \end{aligned} \tag{6.6}$$

This may be reformulated as a separable objective function with linear constraints:

$$\begin{aligned} \mathbf{x}^* = \ & \underset{\mathbf{x}}{\operatorname{argmin}} \ g_1(\mathbf{z}_1) + g_2(\mathbf{z}_2) + g_3(\mathbf{z}_3) + g_4(\mathbf{z}_4) \\ & \text{subject to} \quad \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{I} \\ \mathbf{W} \\ \nabla \end{bmatrix}}_{\mathbf{C}} \mathbf{x} - \underbrace{\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix}}_{\mathbf{z}} = \mathbf{0}. \end{aligned} \tag{6.7}$$

This may be solved using the alternating direction method of multipliers (ADMM) algorithm, which is often used to solve the volumetric albedo problem in the NLOS literature [41, 42]. Then, the augmented Lagrangian is

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \sum_{i=1}^{4} g_i(\mathbf{z}_i) + \mathbf{y}^T(\mathbf{Cx} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{Cx} - \mathbf{z}\|_2^2. \tag{6.8}$$

We proceed using the notation of scaled ADMM, where $\mathbf{u} = \mathbf{y}/\rho$. We compute the update for $\mathbf{x}$ using gradient descent, which is much faster than inverting $\mathbf{C}$:

$$\mathbf{x} \leftarrow \mathbf{x} - \frac{\rho}{\mu}\mathbf{C}^T(\mathbf{Cx} - \mathbf{z} + \mathbf{u}) \tag{6.9}$$

where μ controls the step size. The update for each component of $\mathbf{z}$ utilizes the proximal operators corresponding to each $g_i(\mathbf{z}_i)$:

$$\begin{aligned} \mathbf{z}_1 \leftarrow & \underset{\mathbf{z}_1}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{z}_1 - \mathbf{b}\|_2^2 + \frac{\rho}{2}\|\mathbf{v} - \mathbf{z}_1\|_2^2, \ \mathbf{v} = \mathbf{Ax} + \mathbf{u}_1 \\ = & \frac{\mathbf{b} + \rho\mathbf{v}}{1 + \rho} \\ \mathbf{z}_2 \leftarrow & \underset{\mathbf{z}_2}{\operatorname{argmin}} \ \mathcal{I}_{\mathbb{R}_+}(\mathbf{z}_2) + \frac{\rho}{2}\|\mathbf{v} - \mathbf{z}_2\|_2^2, \ \mathbf{v} = \mathbf{x} + \mathbf{u}_2 \\ = & \max(0, \mathbf{v}) \\ \mathbf{z}_3 \leftarrow & \underset{\mathbf{z}_3}{\operatorname{argmin}} \ \lambda_1\|\mathbf{z}_3\|_1 + \frac{\rho}{2}\|\mathbf{v} - \mathbf{z}_3\|_2^2, \ \mathbf{v} = \mathbf{Wx} + \mathbf{u}_3 \\ = & S_{\lambda_1/\rho}(\mathbf{v}) \\ \mathbf{z}_4 \leftarrow & \underset{\mathbf{z}_4}{\operatorname{argmin}} \ \lambda_{\mathrm{TV}}\|\nabla\mathbf{z}_4\|_1 + \frac{\rho}{2}\|\mathbf{v} - \mathbf{z}_4\|_2^2, \ \mathbf{v} = \nabla\mathbf{x} + \mathbf{u}_4 \\ = & S_{\lambda_1/\rho}(\mathbf{v}) \end{aligned} \tag{6.10}$$

where $S_\kappa(a) = (a-\kappa)_+ - (-a-\kappa)_+$ is the soft threshold function. Finally, the dual variable update is

$$\mathbf{u} \leftarrow \mathbf{u} + \mathbf{C}\mathbf{x} - \mathbf{z}. \tag{6.11}$$

This iterative ADMM procedure is performed until convergence of the primal and dual residuals, as defined in [14].

The entire ADMM optimization is performed during each iteration of an iteratively reweighted ℓ_1 minimization procedure (IRL), in order to further enhance sparsity [18, 41]. We initialize the ℓ_1 weighting matrix $\mathbf{W}$ as identity for the first IRL iteration, and update it at each IRL iteration as

$$\mathbf{W}^{j+1} := \operatorname{diag}\left(\frac{1}{|\mathbf{x}^j| + \epsilon}\right). \tag{6.12}$$

6.4 Evaluation

In evaluating our proposed imaging sonar reconstruction framework, we are primarily concerned with the results on real-world datasets. However, we find it helpful to evaluate the results on simulated datasets as well, due to the availability of ground-truth models and sensor poses.

We evaluate our proposed method against backprojection for comparison (denoted ADMM and BP, respectively). BP is an approximate solution to the inverse problem computed using (6.4). This is a commonly used benchmark solution in the NLOS literature. It is also akin to the occupancy grid mapping (OGM) approach to imaging sonar reconstruction [112, 113]. Both OGM and BP disregard the correlation between voxels corresponding to the same measurement — for each measurement, both methods update each corresponding voxel independently based on some function of the pixel intensity (e.g. inverse sensor model).

In all experiments we use a voxel grid with 2.5 cm resolution. For the ADMM optimization, we use $\rho = 1$, $\mu = \|\mathbf{C}\|_2^2$ (estimated by the power method), and $\epsilon = 0.01$.

6.4.1 Metrics

For qualitative evaluation, we show the maximum intensity projection (MIP) images of evaluated volumes. A MIP image shows the maximum intensity of all voxels along one particular direction and is a useful tool for visualization.

There are a variety of ways to quantitatively evaluate the accuracy of a volumetric reconstruction against a ground-truth pointcloud. We choose to extract pointclouds from the volume by thresholding the albedo and taking the centers of voxels that exceed the threshold as surface points. The pointcloud is aligned to the ground-truth model using a known transformation for

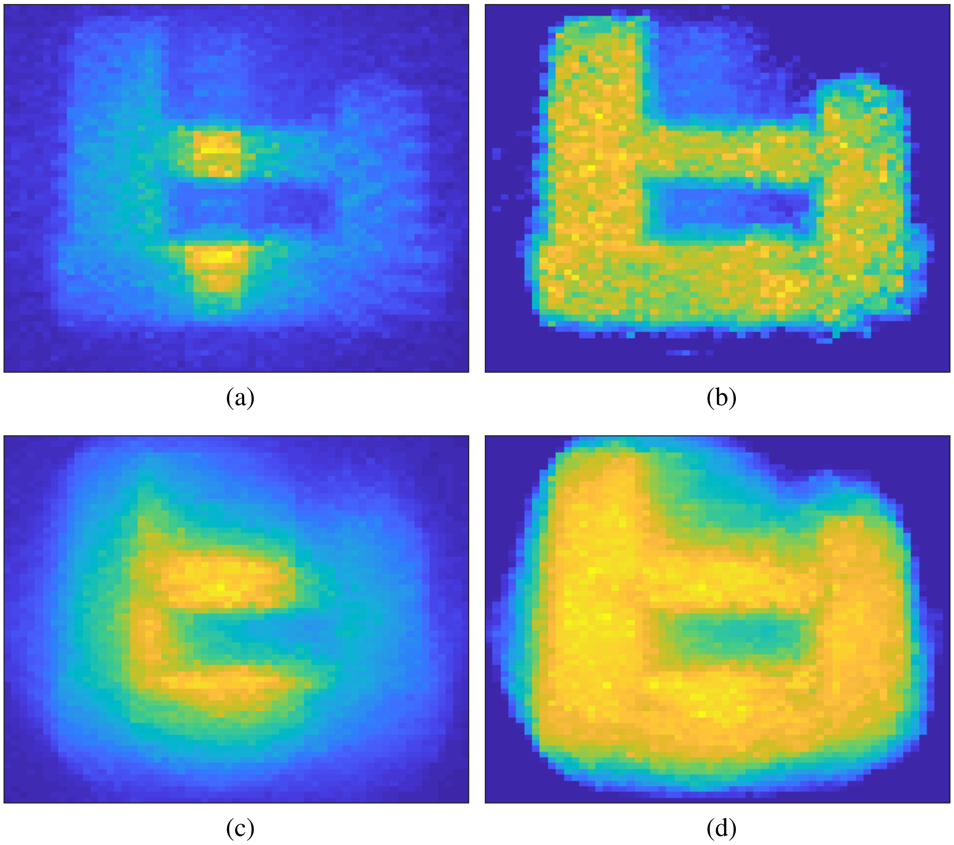


Figure 6.2: Maximum intensity projection (MIP) images for (a) BP and (b) ADMM on a simulated dataset with $3°$ elevation aperture. (c) and (d) show the MIP images for BP and ADMM, respectively, on a simulated dataset with $10°$ elevation aperture. Blue regions correspond to low albedo, and yellow regions to high albedo.

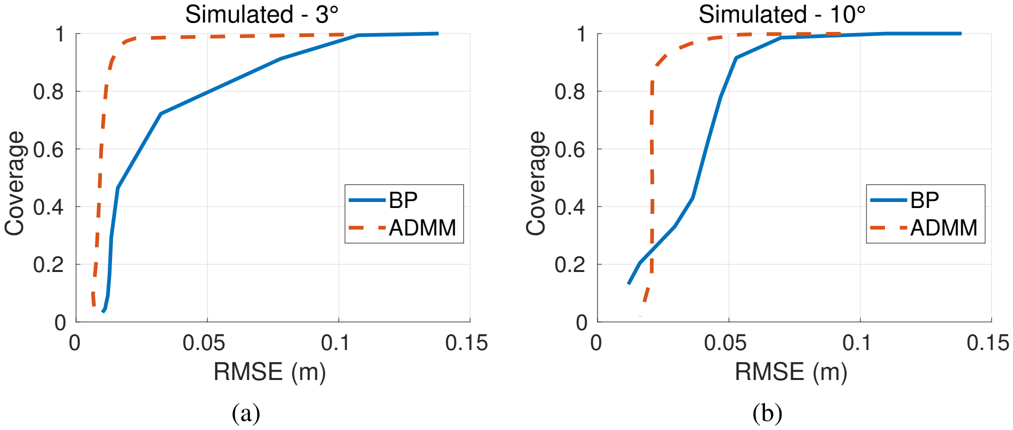


Figure 6.3: Coverage vs. error curves for simulated datasets using (a) $3°$ elevation aperture and (b) $10°$ elevation aperture.

simulated datasets and a manually tuned transformation for test tank datasets, since ground-truth alignment is not available. We evaluate two metrics using the aligned pointclouds: coverage and error. The coverage is defined as the ratio of points in the ground-truth model for which the closest point in the reconstructed model is within a certain Euclidean distance. We use the length of a voxel diagonal as the threshold when computing coverage. For the error metric, we compute the root-mean-square error (RMSE) of the Euclidean distance from all points from the reconstructed model to the closest ground-truth point. Varying the threshold throughout the feasible range allows for trading off between the coverage and error of the reconstructed pointcloud and yields a curve much like the receiver operating characteristic (ROC) curve of a binary classifier. We present these curves for quantitative evaluation of our reconstructed volumes.

6.4.2 Simulation

We simulate sonar images of the structure shown in Figure 6.1a, utilizing a ground-truth point-cloud scan scan collected using a Faro Focus 3-D laser scanner. We use a set of sonar poses that based on a realistic set of viewpoints that could be imaged using an AUV or ROV. The datasets consist of 180 images with motion between sensor poses limited to the sensor's $x - y$ plane. An image is generated at each of 18 different roll angles induced at 10 different points in the $x - y$ plane, which could be acquired on an AUV by simply translating and yawing the vehicle. Images are generated by projecting all points lying in the sonar field of view into the sonar image, with the intensity of each pixel proportional to the number of imaged points. A more accurate pixel intensity model may also be used. However, the precise intensity as a function of the sensor and surface geometry has minimal effect on our results, as the volumetric albedo framework does not consider surface reflection properties. Surfaces that would actually be occluded are still visible in the simulated images. Finally, normally distributed noise is added to each pixel to simulate the low SNR of real acoustic sensors.

Figure 6.2 shows the MIP images for the resulting BP and ADMM volumes for simulated datasets with $3°$ and $10°$ elevation apertures. Compared to the BP volumes, the ADMM volumes show much more distinct surfaces and lower intensities in sections that correspond to free space. Naturally, the reconstructions are less precise with greater elevation ambiguity, but the general shape of the structure is still clearly visible in the ADMM MIP for the $10°$ dataset. Likewise, the coverage vs. error curve for our ADMM reconstruction achieves significantly lower error for the same amount of coverage as BP, as shown in Figure 6.3. Note that using a discretized volume representation limits the minimum possible achievable error. In these simulated experiments, we use $\lambda_1 = \frac{\phi_{\text{fov}}}{5}$ and $\lambda_{\text{TV}} = \frac{\phi_{\text{fov}}}{20}$, where $\phi_{\text{fov}} = \phi_{\text{max}} - \phi_{\text{min}}$ is the elevation field of view in degrees. The regularization coefficients ought to increase with the elevation aperture to account for the

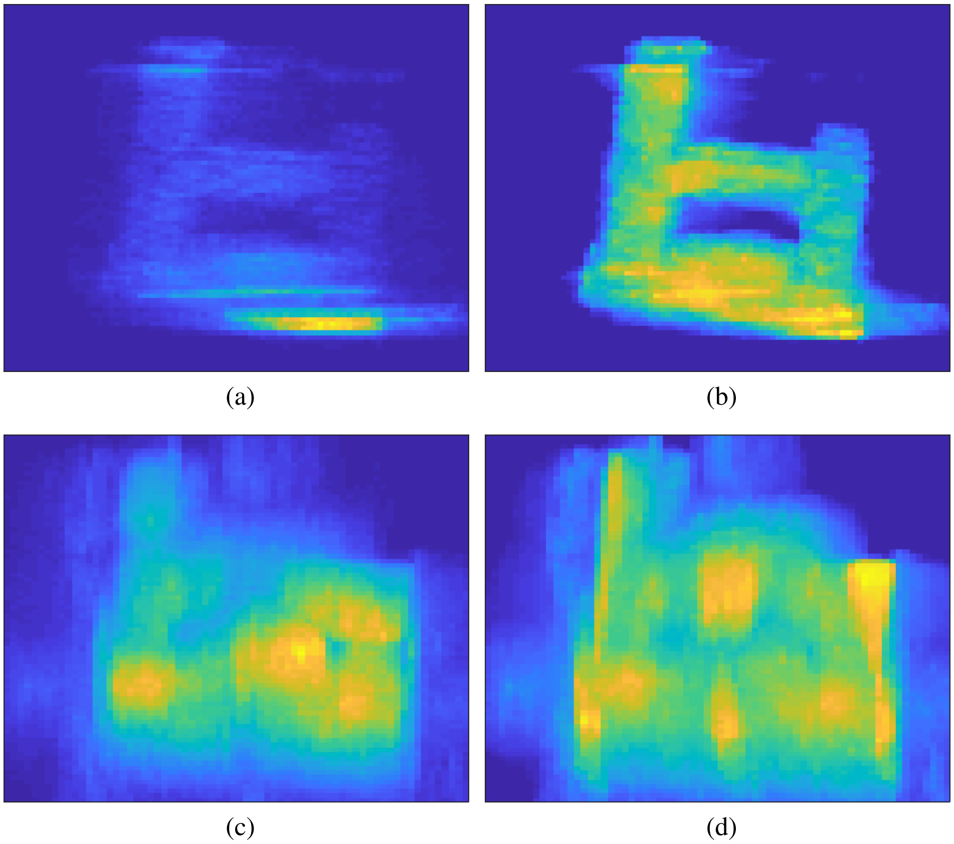

(a) (b) (c) (d)

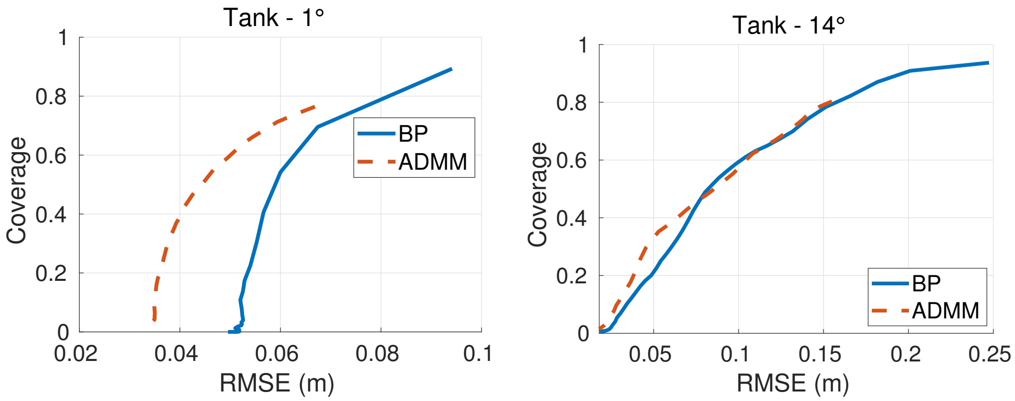
Figure 6.4: Maximum intensity projection (MIP) images for (a) BP and (b) ADMM on a test tank dataset with $1°$ elevation aperture. (c) and (d) show the MIP images for BP and ADMM, respectively, on a test tank dataset with $14°$ elevation aperture. Blue regions correspond to low albedo, and yellow regions to high albedo.

increased number of voxels observed per measurement.

6.4.3 Test tank

We also evaluate our proposed method on real-world datasets collected in a test tank environment using a SoundMetrics DIDSON imaging sonar mounted on a Bluefin Hovering Autonomous Underwater Vehicle (HAUV) [36]. Due to the limited size of the test tank, we keep the DIDSON fixed and pointed directly downward from the vehicle, scanning the submerged structure in Figure 6.1a from above. We collect datasets with a concentrator lens to narrow the elevation aperture to approximately $1°$ and with no lens for a $14°$ elevation aperture. Over 900 images are required with $1°$ elevation aperture to achieve full coverage of the structure, and over 400 images for $14°$ elevation aperture.

The HAUV's onboard odometry measurements are used to provide the pose estimates for



(a)

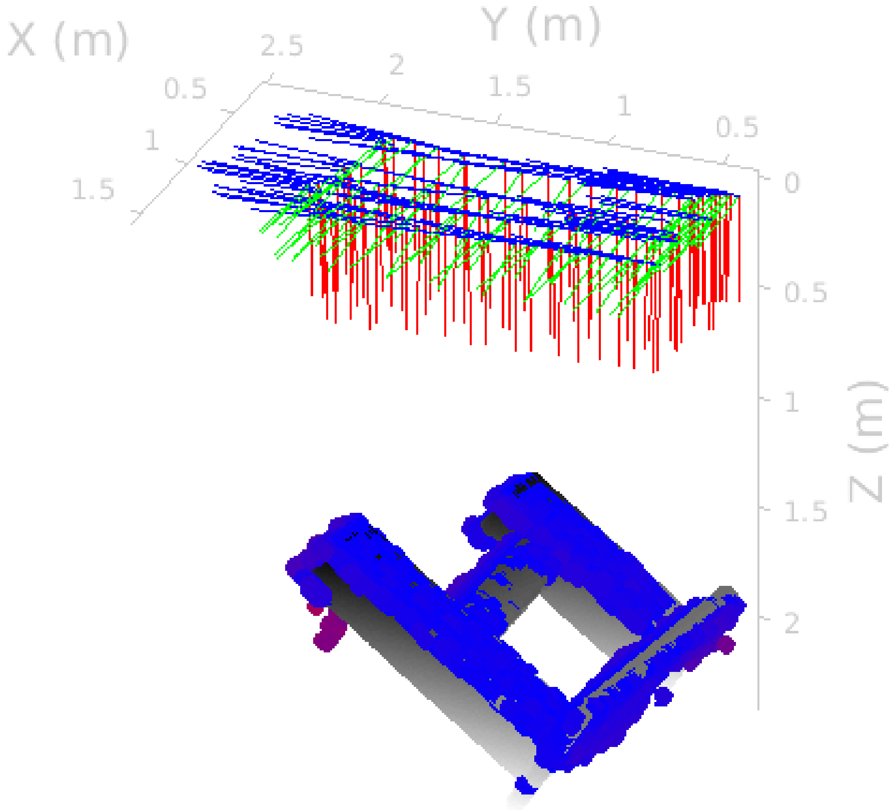
(b)

Figure 6.5: Coverage vs. error curves for test tank datasets using (a) $1°$ elevation aperture and (b) $14°$ elevation aperture.

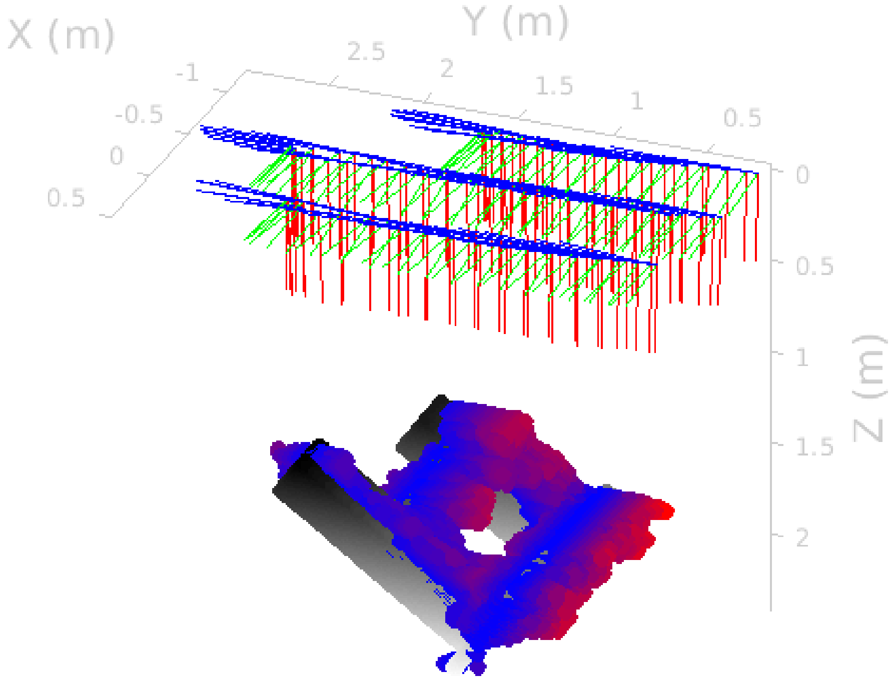
both BP and ADMM. Although the odometry measurements are quite accurate due to the combination of a high-end IMU and a Doppler velocity log (DVL), the pose estimate inevitably drifts after prolonged use. To maintain the integrity of the pose estimates for the reconstruction procedures, we limit the length of the datasets to only a few minutes long. The vehicle is remotely controlled to perform three to four sweeps parallel to the sensor's z-axis, offset along the sensor's y-axis, to ensure full coverage of the structure.

Figure 6.4 shows the MIP images for the $1°$ and $14°$ test tank datasets. Much like the simulated results, the structure is more well-defined and clearly visible in the ADMM MIP images than BP. The coverage-error curves for both resulting volumes are shown in Figure 6.5. As in simulation, the ADMM results show marked improvement over BP for the narrow aperture dataset. The difference in performance is not significant for the wide aperture dataset, which is due to the limited set of viewpoints, particularly the lack of roll rotation (around the sonar's x-axis).

Sample pointclouds extracted from both test tank datasets are shown in Figure 6.6. The narrow aperture reconstruction is highly accurate and covers the entire top surface of the structure, with only a few outlier points due to multipath reflections from the interior of the hollow structure. The wide aperture reconstruction cannot fully disambiguate the structure due to the lack of rich viewpoints, but still captures the main components: the base, two vertical pilings, and crossbar.



(a)



(b)

Figure 6.6: Sample reconstruction of the imaged structure on (a) 1° and (b) 14° elevation aperture test tank datasets. Reconstructed points are colored blue to red with increasing error and are overlaid on a gray ground-truth pointcloud. Downsampled sensor poses are shown with the red, green, and blue coordinates axes corresponding to the sensor's x, y, and z axes, respectively.

6.5 Conclusion

In this chapter, we have connected the problems of NLOS reconstruction and imaging sonar reconstruction. We have presented a volumetric albedo framework for imaging sonar reconstruction with known poses that generalizes previous work to arbitrary sensor motion and does not make any linearization approximations. The proposed framework is solved via ADMM and may incorporate a variety of priors and regularization terms. We demonstrate our algorithm's improvement over previous methods using simulated and real-world data, with several different elevation apertures.

In future work, more extensive evaluation of the proposed method in comparison to occupancy grid mapping [112, 113] and space carving [11] ought to be carried out with a wide variety of underwater structures. A drift-free SLAM method such as [114] should be utilized to increase the accuracy of sensor pose estimates and to allow for longer, richer datasets to be collected. The proposed method could be adapted to attempt to explicitly account for occlusions, for example by incorporating a non-convex prior as in [41].

Chapter 7

Fermat paths

7.1 Introduction

In this chapter we present a novel method for reconstructing particular points on imaged surfaces, which shall be called *Fermat points*. We derive this from the fundamental geometric principles of the imaging sonar sensor. In contrast to previous methods, the proposed framework cannot be categorized as inherently sparse or dense – the density of reconstructed points depends on the geometry of the imaged surfaces relative to the sensor. The proposed method may be viewed as an improvement on and generalization of space carving.

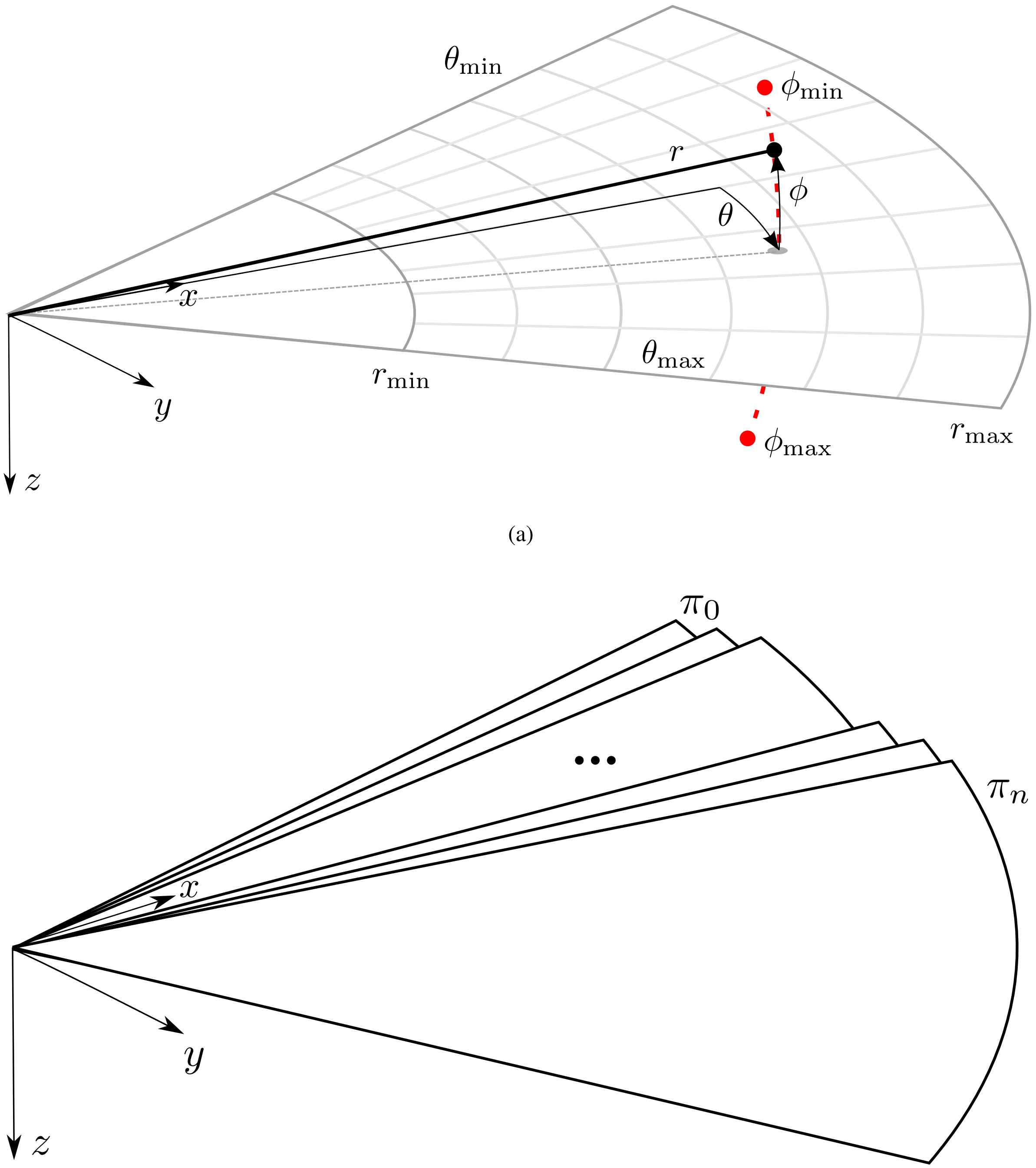
The specific contributions of this work are:

- relating the problems of imaging sonar reconstruction and non-line-of-sight reconstruction
- a derivation of the 2-D Fermat flow equation, which describes a novel method for estimating the 3-D location of surface points
- a framework for estimating the spatial gradients of Fermat pathlengths to solve the 2-D Fermat flow equation,
- an evaluation of the proposed method on simulated and experimental datasets.

7.2 Fermat paths for imaging sonar

7.2.1 Imaging sonar sensor model

The imaging sonar is an active acoustic sensor that emits pulsed sound waves and measures the intensity of sound reflected back towards it. The known speed of sound in water is used to measure the range r of returning sound waves and a 1-D array of transducers is utilized to



(a)

(b)

Figure 7.1: (a) The imaging sonar sensor model. The coordinate frame of the sensor is defined with the x-axis pointing forward and z-axis pointing downward. An imaged 3-D point (black dot) is projected into the zero-elevation plane, where it is imaged by the pixel corresponding to its azimuth angle θ and range r. (b) The ith image column corresponds to a 2-D plane π_i which contains the sensor's z-axis.

disambiguate the azimuth angle θ. This results in a two-dimensional image where the rows correspond to discrete range bins and columns to discrete azimuth angle bins. Figure 7.1a shows the image pixels projected onto the zero-elevation plane, and an imaged 3-D point projected into its corresponding pixel. The sensor has a limited field of view in azimuth, range, and elevation. A single sensor origin is considered the source of emitted sound and the point of detection of reflected sound, with the x-axis pointing forward towards the imaged volume and z-axis pointed downward.

Since the sensor disambiguates the azimuth angle θ, we consider the image formation model for an arbitrary azimuth angle θ_i. Although each azimuthal beare images a slice of volume with a small non-zero width, we can approximate the image formation model in two dimensions. We consider the 2-D plane π_i corresponding to azimuthal beare θ_i, which intersects the sensor's z-axis and is rotated off of the xz plane by θ_i, as shown in Figure 7.1b. The rest of the discussion on Fermat paths takes place within the 2-D plane. Then, using $\mathcal{X}$ to denote the 2-D cross-section of the 3-D scene defined by π_i, $\mathbf{c}$ to denote the imaging sonar origin, and $\mathbf{x}$ to denote a surface point along $\mathcal{X}$, the image formation model for column i may be expressed as

$$I_i\left(r;\mathbf{c}\right) = \int_{\mathcal{X}} f\left(\mathbf{x};\mathbf{c}\right)\delta\left(r - r\left(\mathbf{x};\mathbf{c}\right)\right)\mathrm{d}l\left(s\right) \tag{7.1}$$

where $f\left(\mathbf{x};\mathbf{c}\right)$ captures visibility, reflectivity, shading, and the spatial propagation of sound, $s \in [0,1]$ is a parameterization of the surface and $\mathrm{d}l$ is the differential width measure. $r\left(\mathbf{x};\mathbf{c}\right)$ denotes the range of a surface point $\mathbf{x}$ from the sensor origin.

For the sake of simplicity, we consider only single bounce returns, disregarding that sound waves may be reflected off of multiple surfaces before returning to the sensor, falsely contributing intensity to pixels at longer ranges. Additionally, we define a coordinate system in the plane π_i as follows: the z-axis is aligned with the sensor's z-axis, and the forward-facing w-axis lies in the sensor's xy-plane, rotated from the x-axis by θ_i.

7.2.2 Fermat paths in imaging sonar sensing

My theory follows the work of Xin et al. [119], which derives the 3-D Fermat flow equation for non-line-of-sight (NLOS) reconstruction. Under this model, recovering a point's elevation angle is equivalent to 2-D range-only mapping. This is akin to the problem of NLOS reconstruction, which may be viewed as 3-D range-only mapping, as the azimuth angle is not disambiguated.

Consider a scene $\mathcal{X}$ that is formed as the union of smooth surfaces. This theory focuses on a subset of surface points, defined as follows, which possess unique properties that make them amenable to 3-D reconstruction.

Definition 1. *Let $\mathbf{x} \in \mathcal{X}$ be a point on the scene surface, $\mathbf{c}$ the sonar sensor origin, and $r(\mathbf{x}; \mathbf{c}) = \|\mathbf{x} - \mathbf{c}\|$ the range (or pathlength) of the surface point $\mathbf{x}$ from the sensor. Then, the Fermat set $\mathcal{F}(\mathbf{c})$ is the set of all points $\mathbf{x}$ for which the range function $r(\mathbf{x}; \mathbf{c})$ is a local extremum or a saddle point.*

We refer to points in $\mathcal{F}(\mathbf{c})$ as Fermat points because they correspond to paths that satisfy Fermat's principle.

Proposition 2. *The Fermat set consists of two disjoint sets, the boundary set $\mathcal{B}(\mathbf{c})$ and the specular set $\mathcal{S}(\mathbf{c})$, such that $\mathcal{F}(\mathbf{c}) \triangleq \mathcal{B}(\mathbf{c}) \cup \mathcal{S}(\mathbf{c})$. $\mathcal{B}(\mathbf{c})$ contains all points on $\mathcal{X}$ for which a normal vector is not defined. $\mathcal{S}(\mathbf{c})$ consists of points that create a specular reflection.*

Proof: Let $s \in [0, 1]$ be a parameterization of the object surface $\mathcal{X}$. By fundamental principles, the range function $r(\mathbf{x}(s); \mathbf{c})$, has extrema at $s = 0$, $s = 1$. Other extrema or saddle points only occur where $\frac{\partial r(\mathbf{x}(s);\mathbf{c})}{\partial s} = 0$. Boundary points correspond to $s = 0$ and $s = 1$ and are thus clearly extrema. For a specular point $\mathbf{x}(s) \in \mathcal{S}(\mathbf{c})$, consider the derivative of the range:

$$\frac{\partial r(\mathbf{x}(s); \mathbf{c})}{\partial s} = \left\langle \frac{\mathbf{x}(s) - \mathbf{c}}{\|\mathbf{x}(s) - \mathbf{c}\|}, \mathbf{x}_s(s) \right\rangle \tag{7.2}$$

$$= \frac{1}{r(\mathbf{x}(s); \mathbf{c})} \langle \mathbf{x}(s) - \mathbf{c}, \mathbf{x}_s(s) \rangle \tag{7.3}$$

where $\mathbf{x}_s(s) = \frac{\partial \mathbf{x}(s)}{\partial s}$, and $\|\cdot\|$ denotes the ℓ_2-norm. The vector $\mathbf{x}_s(s)$ is by definition parallel to the tangent of the curve $\mathcal{X}$ at $\mathbf{x}$. The surface normal at $\mathbf{x}(s)$ must be parallel or anti-parallel to $\mathbf{x}(s) - \mathbf{c}$ in order to create a specular reflection. Therefore, $\mathbf{x}(s) - \mathbf{c}$ is orthogonal to the tangent vector, so $\langle \mathbf{x}(s) - \mathbf{c}, \mathbf{x}_s(s) \rangle = 0$ and

$$\frac{\partial r(\mathbf{x}(s); \mathbf{c})}{\partial s} = 0. \tag{7.4}$$

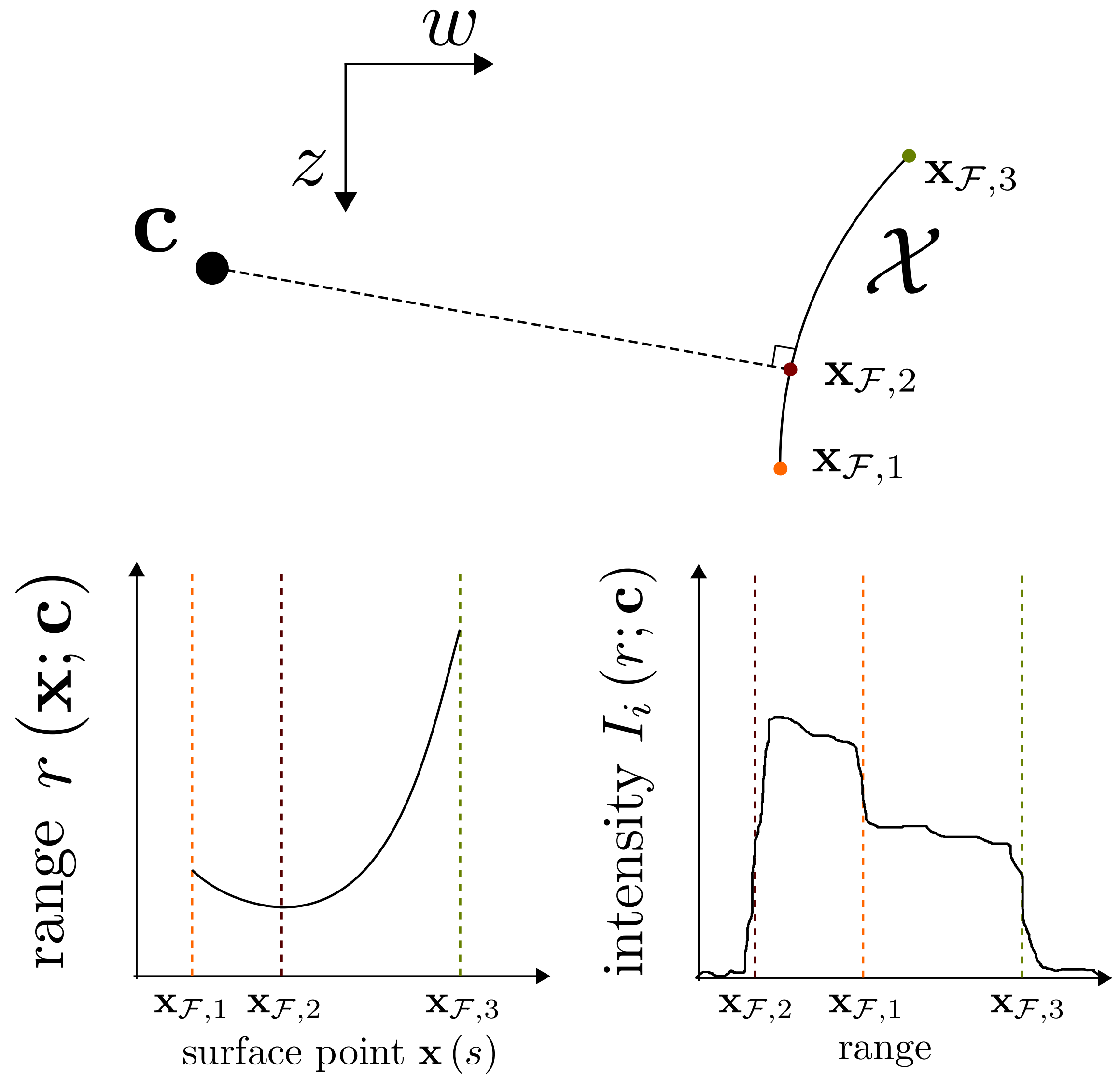
Therefore, $\mathbf{x}(s)$ is either a local extremum or saddle point in the range function. ■

A simple example is shown in Figure 7.2. A convex surface is imaged, resulting in three Fermat paths. The two boundary points correspond to extrema in the range from the sensor. The first-detected surface is the specular point $\mathbf{x}_{\mathcal{F},2}$, which is a local minimum in the range function.

Note that points in the specular set are not necessarily points of specular reflection in 3-D. The orthogonality condition need only hold in the 2-D cross section of the 3-D surface defined by the azimuthal plane of interest.

Next we describe how Fermat points may be detected in a sonar image.

Proposition 3. *Assume that the reflection of sound off the surface $\mathcal{X}$ is non-zero in the specular*



w
z
$\mathbf{c}$
$\mathbf{x}_{\mathcal{F},3}$
$\mathcal{X}$
$\mathbf{x}_{\mathcal{F},2}$
$\mathbf{x}_{\mathcal{F},1}$
range $r\left(\mathbf{x};\mathbf{c}\right)$
$\mathbf{x}_{\mathcal{F},1}$ $\mathbf{x}_{\mathcal{F},2}$ $\mathbf{x}_{\mathcal{F},3}$
surface point $\mathbf{x}\left(s\right)$
intensity $I_i\left(r;\mathbf{c}\right)$
$\mathbf{x}_{\mathcal{F},2}$ $\mathbf{x}_{\mathcal{F},1}$ $\mathbf{x}_{\mathcal{F},3}$
range

Figure 7.2: A surface $\mathcal{X}$ observed by an imaging sonar located at $\mathbf{c}$ in the 2-D plane corresponding to a particular azimuth angle. The first return is generated by $\mathbf{x}_{\mathcal{F},2}$, which is the only point in the specular set in this example. The two boundary points are $\mathbf{x}_{\mathcal{F},1}$ and $\mathbf{x}_{\mathcal{F},3}$. Below the drawing, the range measurement is plotted as a function of the surface point and the image intensity is shown as a function of the range.

direction. Then, for all $\mathbf{x} \in \mathcal{F}(\mathbf{c})$*, the image intensity* $I_i(r;\mathbf{c})$ *will have a discontinuity at pathlength* $r(\mathbf{x};\mathbf{c})$.

Let $\mathcal{C}(\rho;\mathbf{c})$ be the circle of radius ρ centered at $\mathbf{c}$. The surface $\mathcal{X}$ may be reparameterized from $s \in [0,1]$ to $\rho \in [0,\infty]$. Note that this parameterization is only continuously-differentiable locally, as occluding contours, surface boundaries and surface discontinuities induce separate manifolds $\mathcal{M}_j$ within which this condition holds. Then, the image intensity model in Eq. 7.1 may be expressed as

$$I_i(r;\mathbf{c}) = \sum_{\mathcal{M}_j} \int_{\mathcal{M}_j} f(\mathbf{x};\mathbf{c})\,\delta(r - r(\mathbf{x};\mathbf{c}))\left(\frac{\mathrm{d}\rho}{\mathrm{d}s}\right)^{-1} \mathrm{d}l(\rho)\,. \tag{7.5}$$

Boundary points correspond to the limits of integration under this parameterization, and therefore generate discontinuities in the image intensity. For a specular point$\mathbf{x}_{\mathcal{S}} \in \mathcal{S}(\mathbf{c})$, note that the radius by which we parameterize the surface is equal to the range measurement of that surface, i.e. $\rho(\mathbf{x}_{\mathcal{S}}) = r(\mathbf{x}_{\mathcal{S}};\mathbf{c})$. Then we have from Eq. 7.4 that $\frac{\mathrm{d}}{\mathrm{d}s}\rho(\mathbf{x}_{\mathcal{S}}) = 0$ for any specular point. The image intensity in Eq. 7.5 converges to infinity at $\mathbf{x}_{\mathcal{S}}$, producing a discontinuity. ■

7.3 Fermat flow equation

7.3.1 Fermat flow derivation

Here we derive the Fermat flow equation, which is stated as follows and can be used to solve for the 3-D locations of Fermat points.

Proposition 4. *Consider a range measurement* $r_{\mathcal{F}}(\mathbf{c})$ *corresponding to a Fermat point. Assume that there is a single unique point* $\mathbf{x}_{\mathcal{F}} \in \mathcal{F}(\mathbf{c})$ *with* $r(\mathbf{x}_{\mathcal{F}};\mathbf{c}) = r_{\mathcal{F}}(\mathbf{c})$ *. Then,*

$$\mathbf{x}_{\mathcal{F}} = \mathbf{c} - r_{\mathcal{F}} \nabla_{\mathbf{c}} r_{\mathcal{F}}(\mathbf{c})\,. \tag{7.6}$$

Proof. We prove this for a point $\mathbf{x}_{\mathcal{F}} \in \mathcal{S}(\mathbf{c})$ in the specular set, and omit the proof for the boundary set for brevity. We will use $\mathbf{v} = [v^w, v^z]^T$ to denote the 2-D coordinates of a point $\mathbf{v}$ in the plane. Let $s \in [0,1]$ be a parameterization of the surface $\mathcal{X}$ in the neighborhood around $\mathbf{x}_{\mathcal{F}}$, so that $\mathbf{x}_{\mathcal{F}} = \mathbf{x}\left(s\left(\mathbf{c}\right)\right)$. Then, considering the derivative of $r_{\mathcal{F}}\left(\mathbf{c}\right)$ with respect to c^{w}, we have

$$\begin{aligned}
\frac{\partial r_{\mathcal{F}}\left(\mathbf{c}\right)}{\partial c^{w}} &= \frac{\partial \left\|\mathbf{x}_{\mathcal{F}} - \mathbf{c}\right\|}{\partial c^{w}} && (7.7)\\
&= \left\langle \frac{\mathbf{x}_{\mathcal{F}} - \mathbf{c}}{\left\|\mathbf{x}_{\mathcal{F}} - \mathbf{c}\right\|}, \frac{\partial \left(\mathbf{x}_{\mathcal{F}} - \mathbf{c}\right)}{\partial c^{w}} \right\rangle && (7.8)\\
&= \left\langle \frac{\mathbf{x}_{\mathcal{F}} - \mathbf{c}}{\left\|\mathbf{x}_{\mathcal{F}} - \mathbf{c}\right\|}, \frac{\partial \left(\mathbf{x}\left(s\left(\mathbf{c}\right)\right) - \mathbf{c}\right)}{\partial c^{w}} \right\rangle && (7.9)\\
&= \left\langle \frac{\mathbf{x}_{\mathcal{F}} - \mathbf{c}}{\left\|\mathbf{x}_{\mathcal{F}} - \mathbf{c}\right\|}, \mathbf{x}_{s}\left(s\left(\mathbf{c}\right)\right)\frac{\partial s\left(\mathbf{c}\right)}{\partial c^{w}} - \left[1, 0\right]^{T} \right\rangle && (7.10)
\end{aligned}$$

where $\mathbf{x}_{s}\left(s\left(\mathbf{c}\right)\right)$ is tangent to the surface $\mathcal{X}$ at $\mathbf{x}_{\mathcal{F}}$ by definition. Since $\mathbf{x}_{\mathcal{F}} \in \mathcal{S}\left(\mathbf{c}\right)$ and from the definition of the specular set $\mathcal{S}\left(\mathbf{c}\right)$, the vector $\mathbf{x}_{\mathcal{F}} - \mathbf{c}$ is parallel to the surface normal at $\mathbf{x}_{\mathcal{F}}$. Therefore, $\mathbf{x}_{\mathcal{F}} - \mathbf{c}$ is orthogonal to $\mathbf{x}_{s}\left(s\left(\mathbf{c}\right)\right)$, and (7.10) becomes

$$\frac{\partial r_{\mathcal{F}}\left(\mathbf{c}\right)}{\partial c^{w}} = -\frac{\left(\mathbf{x}_{\mathcal{F}} - \mathbf{c}\right)^{w}}{\left\|\mathbf{x}_{\mathcal{F}} - \mathbf{c}\right\|}. \tag{7.11}$$

The same derivation may be used to show that (7.11) holds for the z-coordinate as well, and therefore that

$$\nabla_{\mathbf{c}} r_{\mathcal{F}}\left(\mathbf{c}\right) = -\frac{\mathbf{x}_{\mathcal{F}} - \mathbf{c}}{\left\|\mathbf{x}_{\mathcal{F}} - \mathbf{c}\right\|}. \tag{7.12}$$

This may be rearranged in the form of (7.6). ■

The significance of this result is that Fermat points may be reconstructed in the 2-D plane, and therefore in 3-D, with “single-shot” estimation. Only the gradient of the Fermat pathlength is required, which can be estimated with short baseline motion. This yields what may be thought of as a “differential stereo” algorithm for reconstructing points in $\mathcal{F}\left(\mathbf{c}\right)$.

7.3.2 Fermat flow estimation

An interesting result of (7.12) is that the gradient of the Fermat pathlength is a unit vector, with just one degree of freedom. This gradient with respect to the acoustic center of the sensor may be computed as

$$\nabla_{\mathbf{c}} r_{\mathcal{F}}\left(\mathbf{c}\right) = \left.\left(\sqrt{1 - \left(\frac{\partial r_{\mathcal{F}}}{\partial z}\right)^{2}}, \frac{\partial r_{\mathcal{F}}}{\partial z}\right)\right|_{\mathbf{c}} \tag{7.13}$$

where, $\frac{\partial r_{\mathcal{F}}}{\partial z}$ is the gradient along the sensor’s z-axis, estimated by translating the sensor in that direction. The direction of the gradient in the plane’s w-axis is inferred, using the unit norm property of $\nabla_{\mathbf{c}} r_{\mathcal{F}}\left(\mathbf{c}\right)$. Alternatively, $\frac{\partial r_{\mathcal{F}}}{\partial w}$ could be estimated by translating in plane’s the w-axis.

However, since the 3-D direction of the w-axis for each azimuthal plane in the sonar image is different, each translational direction would yield a direct gradient estimation for only one image column. Additionally, as the sensor translates, the surface point imaged by the discontinuity in each image column may change due to translation outside the plane. Therefore, it is best to translate the sensor along its z-axis so that the azimuthal plane for each column does not change, and directly estimate $\frac{\partial r_{\mathcal{F}}}{\partial z}$ for all image columns simultaneously.

To generate a gradient estimate that is robust to noise in the Fermat pathlength measurement and the sensor pose, we fit a quadratic polynomial to the Fermat pathlength measurements $r_{\mathcal{F}}$ as a function of z:

$$r_{\mathcal{F}}\left(z\right) = a_2 z^2 + a_1 z + a_0 \tag{7.14}$$

and compute the gradient as

$$\frac{\partial r_{\mathcal{F}}}{\partial z} = 2a_2 z + a_1. \tag{7.15}$$

This smoothing procedure is applied to a window of local values around each point of interest. Note that this is a heuristic used to provide robust gradient estimates in the presence of noise, and a variety of other filtering techniques could be used instead. Fitting a quadratic to the Fermat pathlength can also be viewed as placing a prior on the smoothness or curvature of the surface.

7.3.3 Boundary points

A significant distinction between points in $\mathcal{B}\left(\mathbf{c}\right)$ and those in $\mathcal{S}\left(\mathbf{c}\right)$ is that in the 2-D plane, $\mathcal{B}\left(\mathbf{c}\right)$ consists of only up to two points for each continuous surface for all sensor locations $\mathbf{c}$. These same boundary points in $\mathcal{B}\left(\mathbf{c}\right)$ are observed repeatedly from different viewpoints and correspondence may be established across these viewpoints. This is in contrast to points in $\mathcal{S}\left(\mathbf{c}\right)$, of which there are infinitely many and which cannot be corresponded between different viewpoints. If the same point $\mathbf{b} \in \mathcal{B}\left(\mathbf{c}\right)$ is observed from multiple sensor locations, a nonlinear least squares optimization may be used to estimate its location in the plane:

$$\min_{\mathbf{b}} \sum_{k} \left\| r_{\mathbf{b}}\left(\mathbf{c}_k\right) - \sqrt{\left(b^w - c_k^w\right)^2 + \left(b^z - c_k^z\right)^2} \right\|^2 \tag{7.16}$$

where $r_{\mathbf{b}}\left(\mathbf{c}_k\right)$ is the range measurement of the point from sensor location $\mathbf{c}_k$.

This estimation method may be thought of as a 2-D equivalent of the nonlinear optimizations previously used to solve for the 3-D locations of sparse features under general sensor motion [46, 76, 117], and may be solved efficiently using Gauss-Newton. The main drawback of these previous methods is reliably detecting and corresponding feature points between different viewpoints. However, restricting the sensor motion to translation in the z-axis provides an implicit solution to this problem: boundary points may be detected as discontinuities in the image and tracked within the same image column as the sensor moves.

The main limitation of this approach is that discontinuities corresponding to boundary points can generally not be distinguished from discontinuities corresponding to specular points without the aid of prior information.

7.3.4 Field of view

In our derivation of the Fermat flow equation, we disregard the sensor's field of view, assuming that either the sensor has an unlimited elevation field of view or that the entire surface $\mathcal{X}$ lies within the sensor's frustum. In reality, imaging sonars have a limited elevation field of view. For example, the SoundMetrics DIDSON [104] can be configured for up to $28°$ elevation aperture and the Oculus M-series [13] sonars for up to $20°$. If a surface lies partially within the sensor's field of view, our analysis remains intact and visible specular and boundary points may still be reconstructed using the Fermat flow equation. The one complicating effect is that the intersection of the surface with the end of the sensor's field of view in the elevation direction may generate additional discontinuities in the image that correspond neither to specular points nor boundary points. If the Fermat flow equation is applied to all discontinuities in a sequence of sonar images, the presence of such a geometry could potentially introduce false estimated surface points, since the Fermat flow equation does not hold for these points.

7.3.5 Relation to space carving

The application of the Fermat flow equation to reconstruct specular points may be viewed as a generalization of the theory of space carving for imaging sonar reconstruction [7, 11]. Space carving considers the first high-intensity return for each image column, which is generated by the closest surface in the 2-D plane defined by the azimuthal beam. All volume along the entire elevation field of view between the sensor and this range measurement is "carved out" as free space, while all volume behind this measurement remains potentially occupied. While the closest high-intensity measurement is generally caused by one surface point, the surface point is not explicitly determined or solved for. The estimated surfaces are simply the exterior of the potentially occupied regions that remain after repeating the carving process from a variety of viewpoints.

The point generating the first return is, by definition, one of the many Fermat points of a surface. When the surface is convex (e.g., a cylinder, sphere, or plane), then all of its Fermat

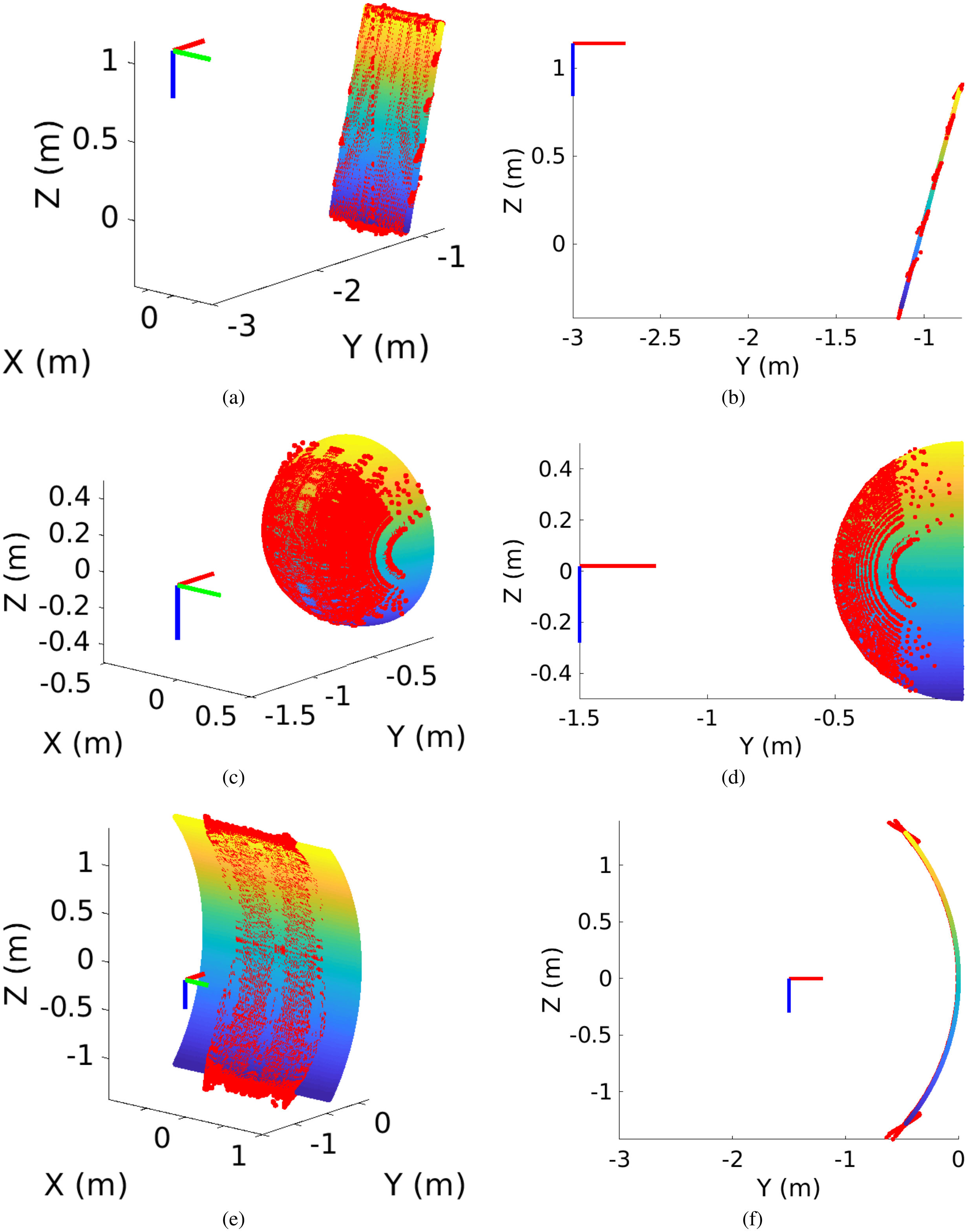


Figure 7.3: Reconstructed points of surfaces from simulated datasets using our solution to the Fermat flow equation. (a), (c), (e) depict isometric views of the ground truth surface points (yellow to blue colored) and the reconstructed surface points (red) for sections of a plane, sphere, and concave cylinder, respectively. (b), (d), (f) depict side views of the same reconstructions.

points will also be points generating first returns. In this case, it can be shown that, assuming an infinite density of sensors, our method and space carving will generate the same surface. However, when a surface is *sufficiently* concave, it will contain Fermat points that do not generate first returns. In this case, space carving will generate a hull containing the surface within its interior, but will fail to reconstruct its concave parts. By contrast, our method will successfully reconstruct the entire surface. This relationship is discussed, in the context of NLOS imaging, in [108, 119].

7.4 Evaluation

We present results from simulation and real-world datasets to demonstrate and evaluate the proposed framework for imaging sonar reconstruction. For both simulated and real-world sonar images, we detect discontinuities using Canny edge detection to detect discontinuities and use several heuristics to reject false positives.

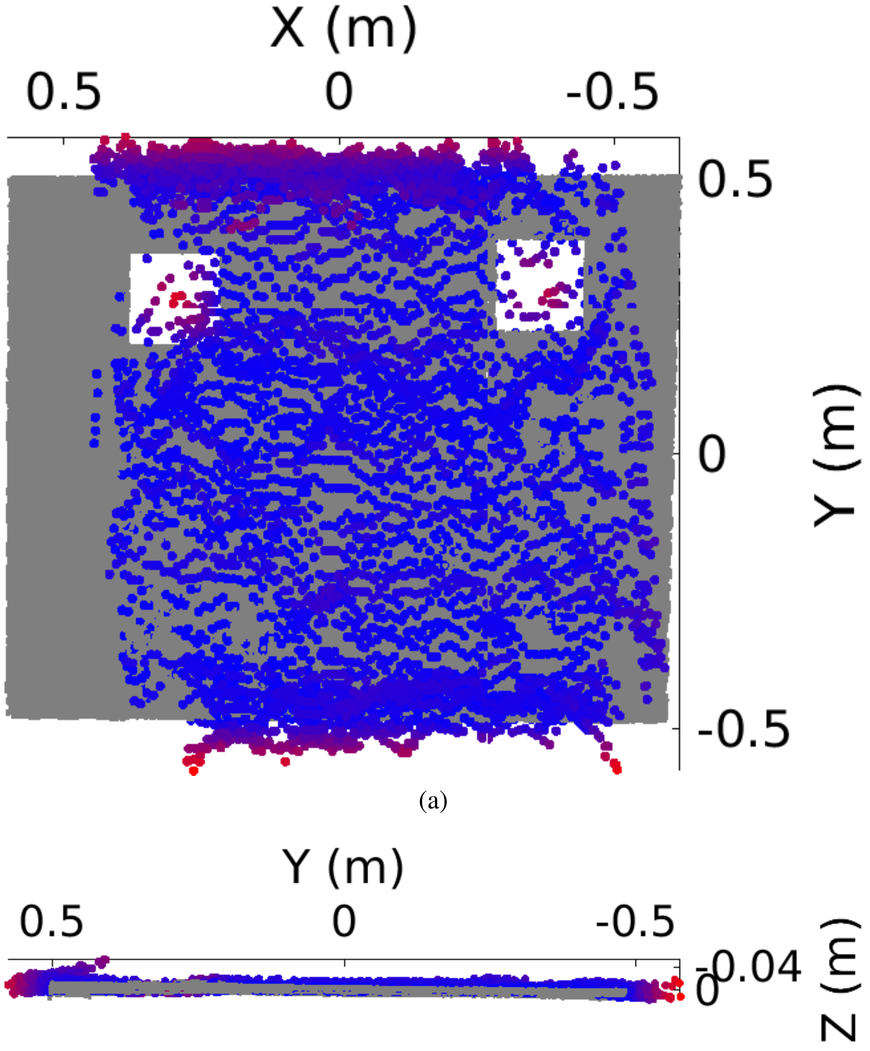
7.4.1 Simulation

We generate simulated datasets of a variety of simple surfaces. We simulate an imaging sonar with an artificially wide elevation aperture of $180°$ for two purposes. First, a wider elevation aperture allows for great coverage of surfaces during a single sweep along the z-axis. Second, this also demonstrates that wider elevation apertures do not affect the accuracy of our proposed algorithm, as long as the discontinuities are detectable in the image. This contrasts starkly with volumetric albedo [40, 116] or occupancy grid [112, 113] methods which perform significantly worse with wider elevation apertures. We use a simple projective image formation model that does not model occlusions or shading, as the precise pixel intensity is not used to solve the Fermat flow equation. Pixel intensities are only used for detecting discontinuities that correspond to points in the specular or boundary sets.

Figure 7.3 visualizes the reconstructed specular and boundary points compared to the ground truth surface models. The specular points are reconstructed nearly perfectly while there is a small amount of noise in the boundary point estimation, due to the smoothness prior enforced in estimating the Fermat pathlength gradient.

7.4.2 Real-world experiments

We also conducted real-world experiments using a Bluefin HAUV [36] robotic test platform in a test tank environment. Data was collected using a SoundMetrics DIDSON imaging sonar



(a)

(b)

Figure 7.4: (a) Top-down view and (b) side view of a planar target imaged in a test tank. The ground-truth point cloud (gray) is shown alongside the reconstructed surface points for comparison, which are colored blue for lower error and red for higher error. The reconstruction does not cover the entire width of the target due to the limited azimuthal field of view of the sonar (28.8°).

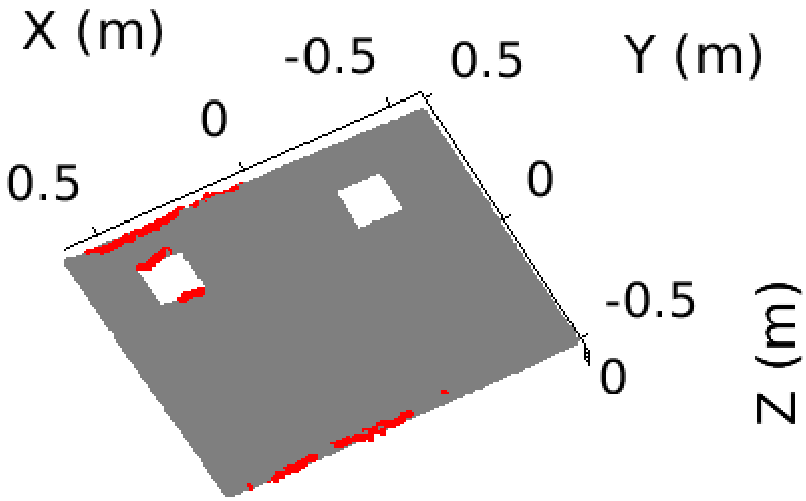


Figure 7.5: Reconstructed boundary points from a single sweep along the sensor z-axis.

with a spreader lens to increase the elevation aperture to $28°$. The vehicle's odometry, which makes use of a high-end IMU and Doppler velocity log (DVL), is used for sensor pose estimates at each sonar frame.

We image a planar plywood target, as shown in Figure 7.4, with the sonar oriented close to perpendicular to the target. A single sweep along the sonar's z-axis allows for reconstructing a dense set of points on along the surface and boundary of the target. The boundaries of the square cut-outs are not easily detected due to the perpendicular viewing angle. Some surface points are reconstructed in these gaps due to the violation of the assumption in Proposition 4 that a detected discontinuity is due to a single surface point – some discontinuities are the result of simultaneous returns from the plane above and below the cut-out. Nevertheless, the side view of the reconstruction demonstrates that the elevation angle of the surface points is estimated quite accurately and the planar structure is recovered despite making no assumptions regarding planarity of the scene. As expected, many noisier boundary points are reconstructed due to repeated observation. The mean absolute error of the reconstructed points is 1.3 cm, which is mostly due to the noisy estimation of boundary points. The range resolution of the sonar in this experiment is approximately 0.5 cm per pixel. Figure 7.5 shows detected boundary points reconstructed using Eq. 7.16, with the plane imaged at a steeper angle to aid the detection of the interior boundaries.

We also imaged an H-shaped, custom-made structure in the same test tank. In a single vertical sweep along the sensor's z-axis, several long edges can be detected as discontinuities. Figure 7.6 visualizes reconstructions of the edge (boundary) points based on three estimation methods: raw Fermat path reconstruction, averaged Fermat path points in a single image column, and points jointly optimized using Eq. 7.16. Table 7.1 shows quantitative error metrics of the reconstructed

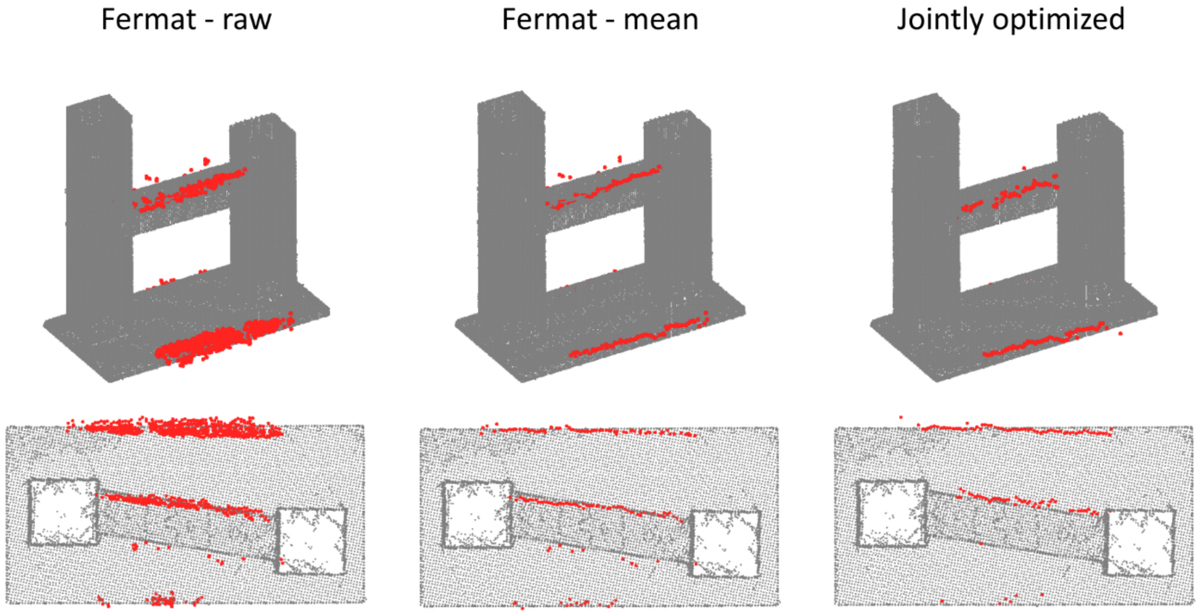


Figure 7.6: Isometric (top) and top-down (bottom) views of reconstructed edge / boundary points based on three methods: raw, unfiltered points using the Fermat flow equation (left), average position of reconstructed Fermat points in the same image column (middle), and points optimized using Eq. 7.16 (right).

points. Both the qualitative and quantitative evaluations demonstrate that even points estimated using single-shot Fermat path reconstruction are quite accurate. Averaging multiple such estimates provides quite comparable, or even better, results in comparison to jointly optimizing the points.

7.5 Conclusion

In this chapter we have presented a novel framework for reconstructing surface points observed by an imaging sonar sensor. We have derived the 2-D Fermat flow equation, which may be applied to reconstruct 3-D surface points from single observations by simply translating the sensor to estimate spatial gradients. While this approach is primarily useful for reconstructing points of specular reflection in the 2-D cross-sectional plane, we describe how the same sensor motion may be used to accurately reconstruct boundary points based on multiple observations. We demonstrate the effectiveness of our proposed algorithm in simulation as well as on real-world datasets collected in a test tank environment.

In future work, a wider variety of objects with convex and concave surfaces should be re-

Method	MAE (m)	RMSE (m)	Max (m)
Fermat - raw	0.017	0.021	0.053
Fermat - mean	**0.011**	**0.013**	0.048
Opt	0.013	0.015	**0.042**

Table 7.1: Quantitative evaluation of the Fermat flow equation and nonlinear optimization method for reconstructing boundary points from the H-structure test tank target dataset. The mean absolute error (MAE), root mean square error (RMSE) and maximum absolute error (Max) are the error metrics used to evaluate the reconstructed edges.

constructed with our proposed method in real-world experiments to highlight its advantage over space carving. We note that the detection of discontinuities corresponding to Fermat points is one of the main sources of error in our real-world experiments and could be improved in future works as well.

We have treated image columns as separate sensors in this work and solve for the 3-D location of points across image columns independently. Information from Fermat paths in neighboring image columns could be used as a form of regularization to improve the estimation of surface points. Additionally, Xin et al. [119] describe an optimization procedure to generate more accurate surface estimates in the presence of noise in the Fermat pathlength gradient estimates in the NLOS scenario, which could be applied to the problem of imaging sonar reconstruction.

Chapter 8

Discussion

8.1 Summary

In this dissertation I have explored the problems of underwater localization and mapping with imaging sonar. I have sought to adapt relevant problems and techniques from the field of computer vision to underwater robotics.

As a solution to localization with imaging sonar, I have proposed acoustic bundle adjustment, a degeneracy-aware solution, and a method of incorporating the resulting constraints into a pose graph optimization for efficient and accurate long-term navigation.

For imaging sonar mapping with known poses, I have proposed three distinct methodologies. First, I demonstrate that a generative sensor model may be utilized to infer a full 3-D surface estimate given an initial edge. I propose a framework in which multiple such surface estimates may be fused together into a single coherent global map. Second, inspired by the analogous problem of NLOS reconstruction, I propose a volumetric albedo framework for sonar reconstruction and a solution based on convex optimization that can incorporate a variety of priors. Lastly, I derive the 2-D Fermat flow equation and present a practical solution that generates 3-D point reconstructions with small baseline motion.

8.2 Significance and future work

In developing algorithms for any task of an autonomous robot, maximal robustness and generality are always desired. However, in an uncertain and variable world, a single algorithm that is guaranteed or empirically proven to provide accurate results in any unknown environment often proves to be quite elusive. While the proposed methodologies in this work do not provide such panaceas to the problems of localization and mapping with imaging sonar, they represent significant steps towards achieving this ultimate goal.

8.2.1 Localization

The framework proposed for imaging sonar localization in Chapter 4 offers marked improvements over past formulations of feature-based SLAM [46, 47, 48]. It is robust to noise and the inherent degeneracies associated with the sensor, which allows for extracting maximal geometric constraints from the detected features while avoiding overfitting any poorly constrained degrees of freedom to noise. In my estimation, this rounds out the analysis of the backend optimization of feature-based SLAM. However, this backend optimization relies entirely on receiving accurately detected and corresponded features from the frontend. I suggest that the frontend of such acoustic SLAM or localization systems is the limiting factor in achieving robust and general purpose acoustic localization for underwater vehicles. As such, improving feature detection and correspondence ought to be the focus of future research efforts in the domain of acoustic feature-based SLAM and localization.

8.2.2 Mapping

The methods I propose in this work for imaging sonar mapping mark significant improvements over previous efforts. The generative model-based reconstruction algorithm in Chapter 5 extends the original sonar shape from shading work [10] to non-seafloor scenarios and robotic mapping. The volumetric albedo formulation in Chapter 6 is similar to volumetric occupancy grid mapping [112, 113] but allows for performing joint inference over the entire volume, rather than assuming conditional independence of voxels. It is also a generalization of the blind-deconvolution method of [40], allowing for arbitrary sensor motion rather than just pure translation. Finally, the theory of Fermat paths and the Fermat flow equation derived in Chapter 7 may be viewed as a generalization of and improvement on the theory of space carving [7, 11], which is able to utilize positive information rather than just the negative information. While none of these methods offer a foolproof solution to acoustic mapping in arbitrary scenes, it is worthwhile to consider their strengths and weaknesses, and how this may inform future research in the field.

Generative model-based reconstruction from a single image alone cannot generate a 3-D map without some prior knowledge, due to the elevation symmetry of the imaging sonar sensor. It may be quite sensitive to noise in the sonar images and to inaccuracies in the calibration of the generative sensor model, which may be difficult to know precisely in real-world environments. However, the general approach of inferring structure from pixel intensity may eventually prove to be the most complete solution to the problem of imaging sonar mapping because it directly utilizes all of the information encoded in an image. The main difficulties in advancing this line of work seem to lie in map representations and optimization techniques that allow for performing joint inference over multiple images, in order to resolve ambiguities and provide more robust solutions in the presence of noise. Nevertheless, this is a compelling line of work that ought to be pursued in future research.

The volumetric albedo approach to imaging sonar mapping may be the most widely applicable to different structures and scenes, as it makes minimal assumptions regarding the scenes and is flexible to incorporate a variety of priors that may be appropriate. However, it suffers from poor performance as the elevation aperture and ambiguity increases. Wide aperture sonars offer richer information and greater coverage per image. This introduces an inherent trade-off when mapping with the volumetric albedo framework between coverage and ambiguity. I generally recommend this approach for small elevation apertures up to about $5°$, and especially in scenarios where a wide variety of viewpoints may be gathered, including different sonar rotations and positions. A particularly interesting direction would be to apply this technique using sonar rotator devices that allow for rotating the sensor through $180°$ of actuation, as in [112, 113]. Then, a fair comparison could be made between the occupancy grid mapping framework of [112, 113] and the batch convex optimization approach I propose, with the expectation that my proposed method would generate more accurate reconstructions at the expense of greater computational complexity.

One advantage of the theory of Fermat paths for sonar mapping is that is relies solely upon basic geometric principles and range measurements – the precise pixel intensity, and our model of how it is formed, does not directly influence the resulting reconstructions. Therefore, it is less sensitive to deviations of measurements from the ideal model-predicted measurement. Of course, this is also a limitation of the method, as it does not utilize all of the information encoded in an image. As with the feature-based acoustic bundle adjustment approach to sonar localization, the main source of error in this approach to reconstruction is the frontend: detecting discontinuities in the image. The relatively low resolution and signal-to-noise ratio of imaging sonars limit the accuracy of the reconstructions that may be generated and increase the required baseline of motion to generate robust gradient estimates. Future research efforts ought to focus on this frontend detection task as well as ways to extend this method to more generalized motion, rather than pure z-translation. Interestingly, the main benefit of this approach is that the accuracy does not degrade as the elevation aperture of the sensor increases, in contrast to the methods of volumetric albedo and occupancy grid mapping. In fact, wider elevation apertures allow for more complete reconstructions, as the wider frustum allows for the possibility of more surface points to fall into the sensor's set of Fermat points. This informs the development of future generations of imaging sonar: it may indeed be beneficial to develop sonars that may be configured to have much wider elevation apertures than the standard $10° - 20°$. Finally, this method of surface reconstruction is best suited for smooth surfaces, which offer continuous Fermat path gradients that may be accurately estimated by filtering noisy discontinuity measurements.

Appendix A

AprilTag SLAM and calibration

A.1 Introduction and background

Many underwater robotic tasks require high-precision vehicle localization. Vehicle odometry may be measured by an inertial measurement unit (IMU) or a Doppler velocity log (DVL), among other sensors. However, these pose estimates will drift unboundedly with time, as they rely on dead reckoning (integration of odometry measurements). For traditional non-underwater robotics, ground-truth trajectories of robots or sensors are typically acquired by a camera-based motion capture system or laser surveying equipment. While such motion capture systems exist for underwater tracking [94], the inherent difficulties presented by underwater optics and electronics make such systems cost-prohibitive for many applications. Furthermore, these systems are only practically usable in a controlled laboratory setting, and not in the field. I aim to provide a localization solution that:

- corrects drift that accumulates with dead reckoning and bounds the pose uncertainty
- incorporates any localization information from multiple on-board sensors
- automatically solves for the extrinsics between the camera and odometry coordinate frames
- is significantly less costly than an underwater motion capture system
- is highly reconfigurable and requires minimal labor to setup and operate in the laboratory and in the field

Various visual SLAM systems have proven capable of satisfying all of the above requirements, both in standard open-air environments as well as underwater, with various limitations and precision [45]. In uncontrolled environments, natural features are often detected and used as the landmarks in a SLAM formulation. A variety of feature descriptors have been formulated in order to perform the critical task of *data association*: matching features locally for feature tracking or globally for loop closure [4]. While descriptors aid the matching process, outlier rejection algorithms such as RANSAC usually must be employed to reject incorrect correspondences. Nevertheless, incorrect feature correspondences may persist and negatively affect the SLAM result.

Visual fiducials (easily identifiable, artificial markers placed in the robot's environment) are often used to provide strong features and a robust solution to the data association problem. Various types of fiducials and detection methods have been proposed in recent years and have been widely used in the field of robotics [34, 87, 111]. In my proposed visual SLAM framework, I utilize the AprilTag system [87], as it provides particularly robust data association correspondences and can even identify partially occluded fiducials. Since the proposed algorithm performs online mapping, the fiducials may be placed anywhere in the environment, as long as they remain stationary. However, placing the AprilTags where they will be viewed most frequently over the course of the vehicle's mission will help the SLAM algorithm to generate the best localization, mapping, and calibration results.

The proposed system can accommodate the use of individual AprilTag fiducials as the landmarks in the SLAM system as well as my custom-made AprilTag boards, which are shown in Figure A.1. I printed four AprilTags of the same size in a square pattern on each aluminum dibond board. This aids the SLAM process by reducing the number of degrees of freedom that need to be estimated. If eight AprilTags are utilized in the form of two boards, then only 12 DOF must be estimated for the landmarks (two 6-DOF poses) in contrast to the 48 DOF that would be required to model the poses of eight individual AprilTags. In the remainder of this work, I will describe my system as it pertains to my custom-made AprilTag boards.

Several recent works have proposed using visual fiducials to achieve high-precision, drift-free localization of an underwater vehicle using an EKF framework [54] and a particle filter framework [55]. These systems have proven successful at reducing localization error over dead reckoning, but do not allow for simultaneously solving for the camera extrinsics (relative to odometry coordinate frame), and presumably utilize manual extrinsics measurements. I do not explicitly compare my localization method to these in this work, as each system is highly tailored to the specific underwater vehicle and testing environment.

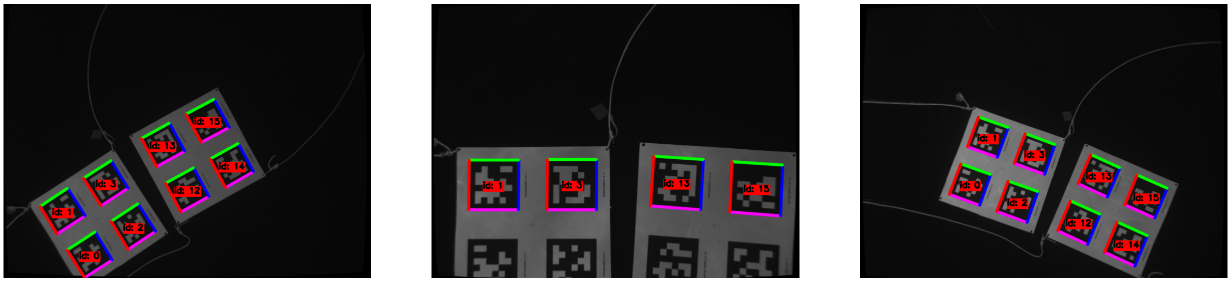


Figure A.1: Sample images from one of my test tank datasets, showing the AprilTags detected from various vehicle depths and viewing angles.

A.2 Proposed SLAM algorithm

A.2.1 Factor graph SLAM

I model the SLAM problem with a factor graph formulation, which is shown in Figure A.2a. A factor graph is a bipartite graph with two types of vertices: nodes that represent the *variables* in the optimization and factors that represent the *measurements* that provide constraints. The edges from factors to nodes describe the dependency structure of the optimization: the nodes connected to a factor are constrained by that factor's measurement.

In the standard SLAM formulation, the variables consist of the poses in the trajectory and the observed landmarks. The state is comprised of all of the variables: $\Theta = \{x_1, \ldots, l_1, \ldots\}$, where x_i is the ith vehicle pose and l_j is the jth landmark pose. Note that I explicitly model the vehicle poses rather than cameras poses in the factor graph. However, the camera poses may be computed using the vehicle-camera extrinsics. In this simple SLAM formulation, I model the vehicle-camera extrinsics as an *a priori constant*. A landmark represents one of my custom-made boards that has four AprilTag fiducials printed in precisely known locations. The measurement vector is comprised of all measurements $\mathcal{Z} = \{r_0, u_1, \ldots v_1, \ldots, m_1, \ldots\}$, where u_i is an XYH odometry measurement that constrains poses x_{i-1} and x_i, v_i is a ZPR measurement that constrains pose x_i, and m_k is an observation of any AprilTag. Finally, I denote with r_0 a prior measurement placed on the first pose to tie down the trajectory to a global coordinate frame.

A.2.2 Factor graph SLAM with extrinsics calibration

The typical SLAM formulation presented in the previous sub-section assumes the camera-vehicle extrinsics are known a prior and are treated as constant in the optimization. However, these extrinsics may be very difficult to measure precisely by hand. However, I may incorporate this variable seamlessly into the factor graph formulation, as shown graphically in

Figure A.2b. I simply add the extrinsics e as an additional 6-DOF pose to the state, so that $\Theta = \{e, x_1, \ldots, l_1, \ldots\}$. This variable is then constrained by the AprilTag measurement factors, but not by any odometry factors. Note that I assume this transformation is constant throughout the entire operating sequence of the vehicle (the camera is fixed relative to the odometry frame). This assumption is crucial in order to get sufficient constraints to accurately estimate the extrinsics.

A.2.3 Measurements

To solve the SLAM + calibration problem, I use the MAP estimation framework and nonlinear least squares optimization presented in Chapter 3.2. For pose priors and odometry constraints, I utilize the measurement functions described in Chapter 3.3 and Chapter 3.4.2, respectively. The likelihood function for the tag measurement is:

$$l(x_{i_k}, l_{j_k}, e; m_k) \quad \propto \quad \exp\left\{-\frac{1}{2}\left\|h_k(x_{i_k}, l_{j_k}, e) - m_k\right\|^2_{\Xi_k}\right\}. \tag{A.1}$$

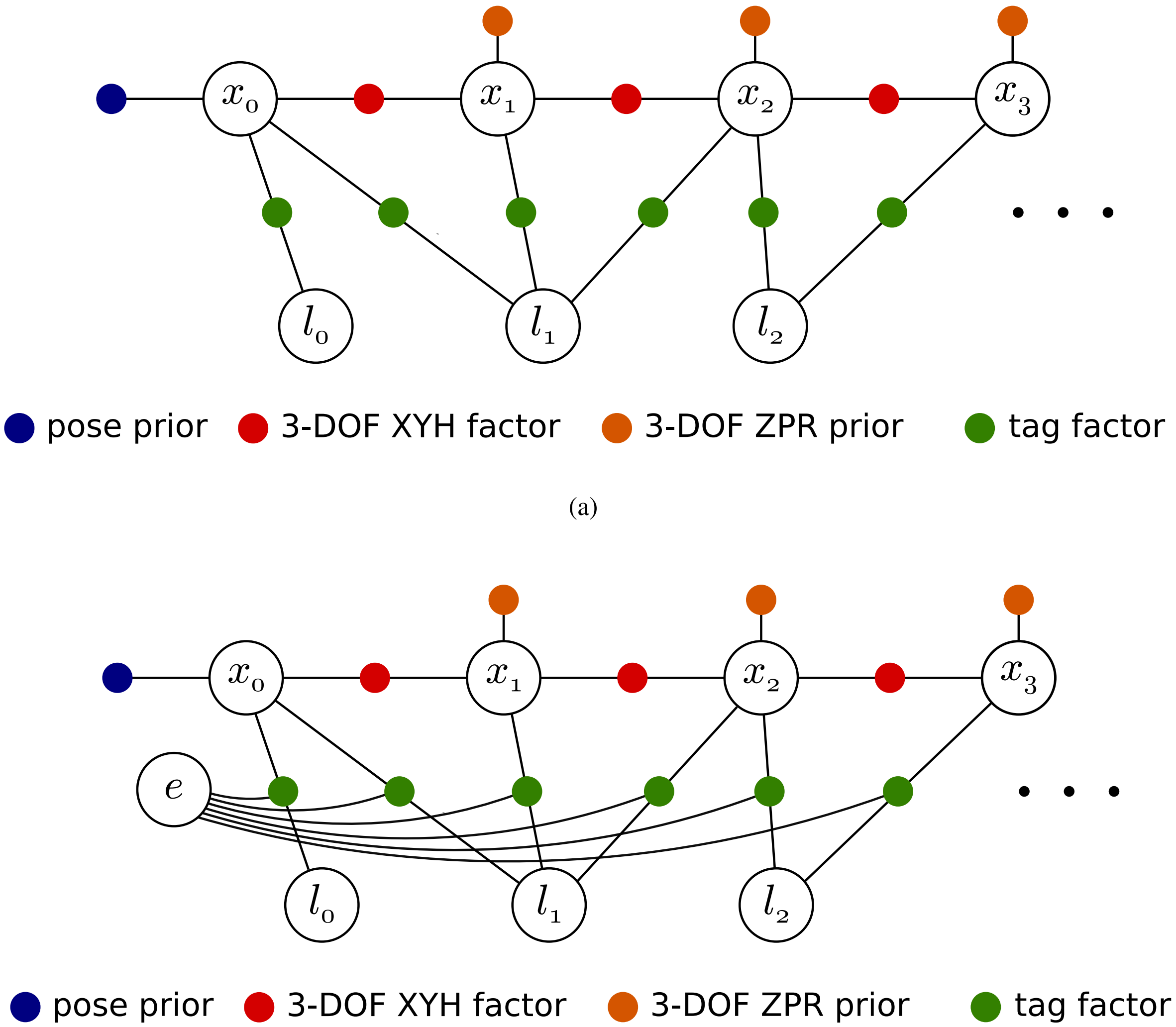
Here I use $h(x_{i_k}, l_{j_k}, e)$ to denote the tag measurement function. A tag measurement m_k is an 8-vector comprised of the (u, v) pixel coordinates of the four corners of a detected AprilTag:

$$m_k = \begin{bmatrix} c^u_{k,1} & c^v_{k,1} & c^u_{k,2} & c^v_{k,2} & c^u_{k,3} & c^v_{k,3} & c^u_{k,4} & c^v_{k,4} \end{bmatrix}^\top. \tag{A.2}$$

Since the side length s and placement of each AprilTag on its board is known *a priori*, the 3-D corner points are treated as constants relative to the board's coordinate frame. The origin of the board's frame is at the center of the board and the x and y axes are parallel to the sides of the AprilTags. The homogeneous 3-D coordinates of AprilTag corner i in the board frame corresponding to the measurement m_k is denoted as $d_{k,i} = \begin{bmatrix} d^x_{k,i} & d^y_{k,i} & 0 & 1 \end{bmatrix}^T$. Using the pinhole camera model with a calibrated intrinsics matrix $\mathbf{K}$, I define q_k as the relative transformation from l_{j_k} to $x_{i_k} * e$. That is, q_k is the pose of the camera at timestep i_k relative to the frame of AprilTag l_{j_k}. Under this projection model, the tag measurement function is:

$$h(x_{i_k}, l_{j_k}, e) = \Omega\left(\mathbf{KT}\left(q_k\right)\begin{bmatrix} d_{k,1} & d_{k,2} & d_{k,3} & d_{k,4} \end{bmatrix}\right) \tag{A.3}$$

where the function $\Omega\left(\cdot\right)$ normalizes each homogeneous 3-vector column of the input matrix and reshapes the entire matrix to an 8-vector, in the form of m_k. $\mathbf{T}\left(q_k\right)$ creates the 4×4 transformation matrix that corresponds to the 6-DOF relative pose q_k. The covariance of the measurement



(a)

(b)

Figure A.2: (a) Factor graph representation of AprilTag SLAM (b) Factor graph representation of AprilTag SLAM with automatic calibration of vehicle-camera extrinsics. A node is represented as a large, uncolored circle labeled with the corresponding variable name. A factor is represented as a small, colored circle, with the color corresponding to the category of measurement, as indicated by the legend. Factor labels are omitted for simplicity.

is

$$\Xi_k = \sigma_c^2 I_8 \tag{A.4}$$

where $\sigma_c = 1$ pixel in my experiments. A more sophisticated noise model for the corner detection may be implemented, but I found a constant uncertainty to work well in my experiments.

A.2.4 Implementation

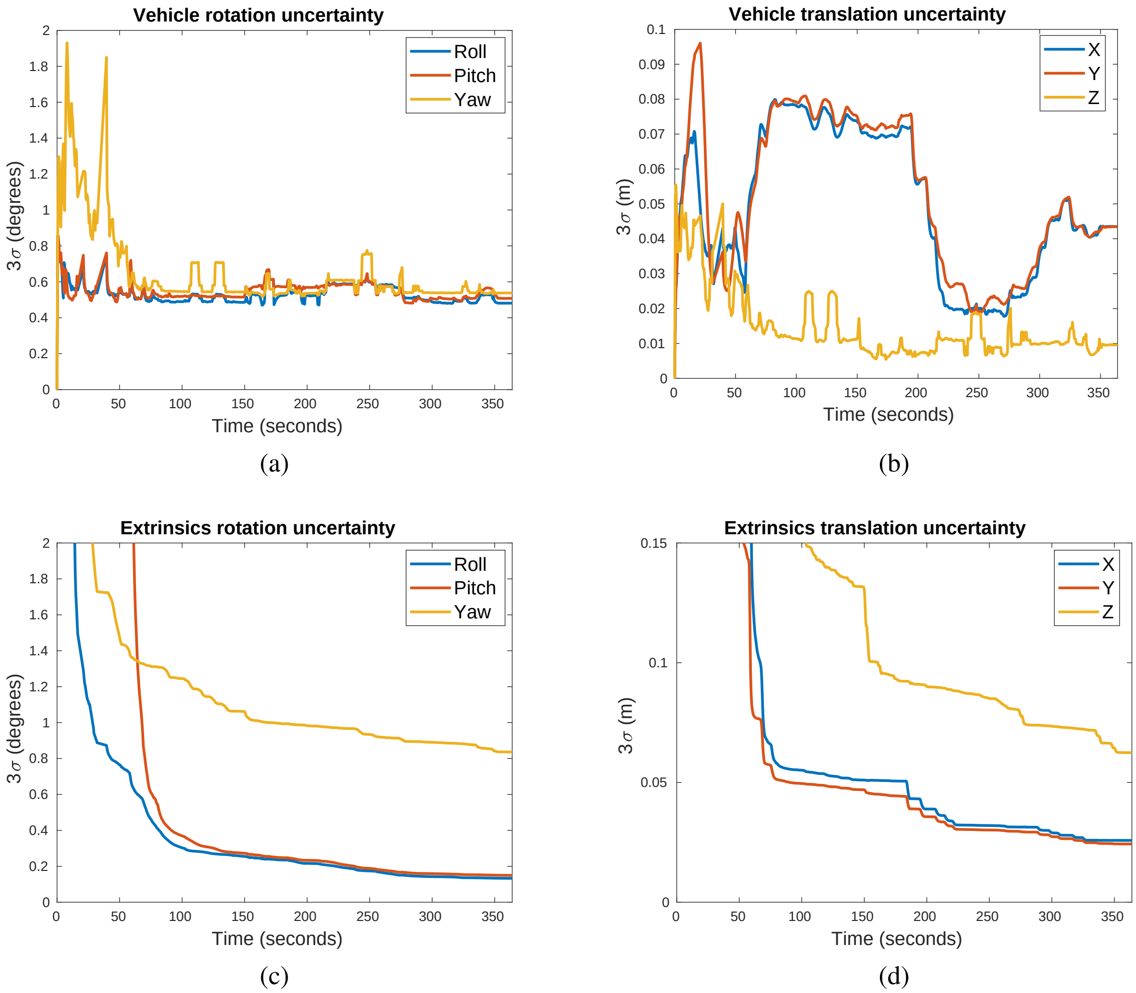
The proposed SLAM framework is implemented using the GTSAM library for optimization and on-manifold operations [27]. Since odometry and camera measurements arrive asynchronously, I consider each camera measurement as a single timestep in my framework and interpolate the odometry pose estimates using the camera measurement's timestamp. The factor graph is optimized using the iSAM2 algorithm for efficient, real-time state estimation [56]. Analytical Jacobians are implemented for all factors in the nonlinear least-squares optimization except the XYH and ZPR factors, for which numerical Jacobians are computed.

A.3 Experimental results

A.3.1 Setup and evaluation metrics

The proposed SLAM system is evaluated using the HAUV platform described in Section 3.4 in an indoor 7m-diameter, 3m-deep test tank. I placed two of my custom-made boards on the tank floor, spanning an area of approximately 2x1 meters. Both the DVL and camera were fixed pointing downward. It is difficult to evaluate the accuracy of the localization of my algorithm without an underwater motion capture system. Previous works have used manual measurements [55] or ceiling-mounted vision systems [54] to obtain ground-truth trajectories of the AUV. However, my method is expected to be *more* accurate than manual measurements or a simple external vision system (except a highly calibrated, multi-camera motion capture system) because it optimizes over high-quality odometry from the IMU and DVL as well as direct measurements of the AprilTag fiducials. Therefore, I validate this system statistically, by demonstrating with repeated trials that the resulting camera-vehicle extrinsics estimates have low variance and are consistent with the corresponding uncertainty in the factor graph optimization.

Additionally, I evaluate the system using two different types of odometry: (1) the proprietary vehicle odometry that fuses IMU, depth sensor, and DVL measurements and (2) the proprietary vehicle odometry with random Gaussian noise added in the XYH directions (zero-mean, with



(a)

(b)

(c)

(d)

Figure A.3: Three-sigma uncertainty bounds on: (a) the vehicle rotation (b) the vehicle translation (c) the extrinsics rotation and (d) the extrinsics translation. All quantities are evaluated on the same dataset using the proprietary vehicle odometry with no noise added.

DVL-Camera Extrinsics

Trial	ψ (deg)	θ (deg)	ϕ (deg)	x (cm)	y (cm)	z (cm)
Accurate 1	-0.30	0.62	90.47	6.62	-23.67	11.75
Accurate 2	-0.28	0.52	90.06	5.94	-23.36	8.30
Accurate 3	-0.47	0.55	90.24	6.20	-23.35	13.08
Mean	**-0.35**	**0.57**	**90.26**	**6.25**	**-23.46**	**11.04**
Noisy 1	-0.301	0.62	90.30	6.03	-24.37	20.87
Noisy 2	-0.27	0.52	90.03	11.11	-26.50	23.11
Noisy 3	-0.46	0.55	90.56	8.51	-25.66	19.21
Mean	**-0.34**	**0.56**	**90.30**	**8.55**	**-25.51**	**21.06**
Manual	0	0	90	6	-23	8

Table A.1: Estimated vehicle-camera extrinsics parameters from my real-world experiments. Extrinsics are shown in yaw, pitch, roll, and x, y, and z translation. The mean of each set of three datasets is bolded, and my manual measurement is shown for comparison.

DVL-Camera Extrinsics: Deviation from mean (σ)

Trial	ψ	θ	ϕ	x	y	z
Accurate 1	1.26	1.25	0.72	0.35	-0.21	0.31
Accurate 2	1.62	-0.95	-0.67	-0.30	0.10	-1.20
Accurate 3	-2.88	0.30	-0.05	-0.05	0.11	-0.89
Noisy 1	0.89	1.12	-0.01	-2.37	1.13	-0.08
Noisy 2	1.89	-1.26	-0.89	2.41	-0.98	0.85
Noisy 3	-2.79	0.13	0.89	-0.04	-0.15	-0.77

Table A.2: Standard deviations of my experimental results from the sample mean, evaluated using the standard deviations derived from the corresponding marginal uncertainty in the overall factor graph optimization. All estimates are within 3σ of the mean.

standard deviation of 0.01 radians and 0.01 meters per frame). These are the degrees of vehicle motion that are not directly observable by the IMU or depth sensor. Therefore, this noisy odometry simulates an estimate that would be provided by just an IMU and depth sensor, without the DVL.

I recorded three datasets with which to perform SLAM and calibration. The vehicle was remotely operated and its motion consisted of translation along the x and y vehicle axes at various depths between 0 and 1.5 meters, and rotation about the z-axis (yaw rotation). This utilizes all controllable degrees of freedom of HAUV motion available, as the pitch and roll of the vehicle are not controllable by the thrusters. It is important to utilize all possible degrees of freedom of motion to provide as many constraints on the camera-vehicle extrinsics as possible.

A.3.2 Uncertainty

To demonstrate the bounded vehicle pose uncertainty and the convergence of the extrinsics estimate, I examine the marginal covariance of these variables at every step in the optimization. Figure A.3 shows plots of the 3σ bounds of both the vehicle pose and the extrinsics for one of my experimental datasets, separating the values into rotation and translation uncertainty. The vehicle pose estimates are tightly bounded ($0.5°$ rotation and 0.03m), except for times when the AprilTags either briefly go out of the field of view of the camera, or when the vehicle is too close to the tags to observe more than one or two in a single frame. The latter case is clearly visible in the vehicle translation uncertainty from $100 - 300$ seconds, when the uncertainty in x and y translation rises as the vehicles dives close to the tank floor.

A.3.3 Consistency

Table A.1 shows the resulting extrinsics estimates for my six datasets. “Accurate” denotes the datasets using the vehicle odometry with no noise added. “Noisy” denotes the datasets using the vehicle odometry with noise added in the XYH directions. The mean extrinsics for both categories are also shown. In order to demonstrate the consistency of the extrinsics estimate, we examine the extrinsics estimate from these repeated trials with respect to the estimated uncertainty from the overall factor graph optimization. Since the extrinsics uncertainties at the end of the optimization are very similar across all three datasets, I arbitrarily use the uncertainty estimate from the first dataset. Table A.2 shows the deviation of each dataset’s extrinsics estimates from the sample mean, normalized by the uncertainty. Most of the values lie within 2σ of the mean value, with all lying within 3σ. While three datasets is a small sample size, this confirms that my method is likely to provide a good upper-bound on the uncertainty of the extrinsics estimate.

Finally, I show in Figure A.4 a comparison of the dead reckoning and SLAM trajectories, both utilizing the noisy odometry estimates. The dead reckoning estimate clearly drifts by tens of centimeters if not meters over the course of the six-minute dataset. The same trend may be seen using the accurate odometry estimates, but the difference between the trajectories is less pronounced.

A.4 Conclusion

I have presented a novel formulation of simultaneous underwater localization, mapping, and extrinsics calibration using a camera and one or more odometry sensors, such as an IMU and

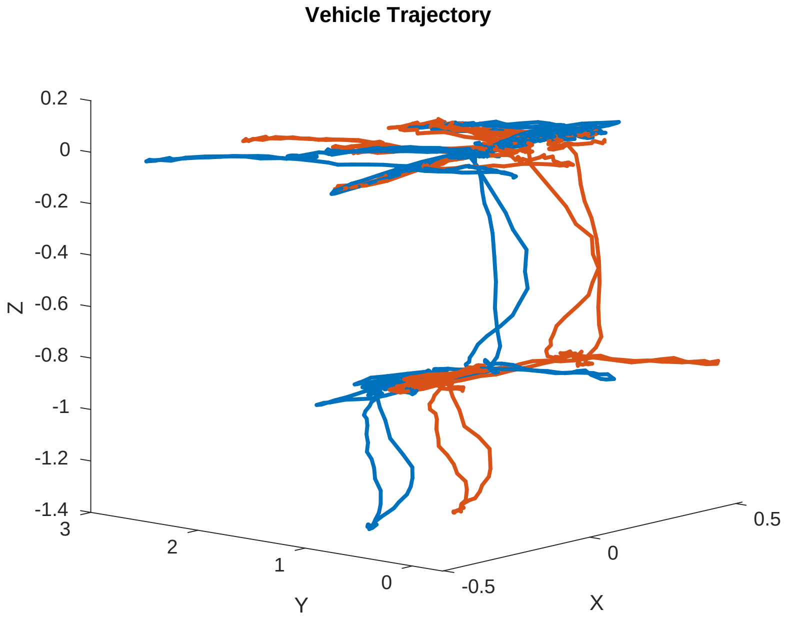


Figure A.4: Sample dead reckoning trajectory (red) and SLAM trajectory (blue) using noisy odometry measurements. The dead reckoning trajectory clearly drifts from the SLAM solution even over the course of a short, six-minute dataset.

DVL. I utilize AprilTag fiducials to make my SLAM solution highly reconfigurable, inexpensive, and robust. The resulting extrinsics are consistent with the optimization's uncertainty model, and very accurate in rotation.

One limitation of the proposed framework is the effect of timing synchronization errors on the system accuracy. This will present itself most significantly when the vehicle undergoes relatively high accelerations, as the velocity measurements made by the IMU and DVL may line up poorly with the camera measurements. A possible improvement would be to rigorously characterize the noise of the DVL measurements and develop a more accurate noise model. Additionally, it may be beneficial to explicitly model more vehicle dynamics, such as the mechanical slop in the roll cage on which the sensors are mounted, as in [89]. Finally, exploring recent extensions to AprilTag fiducials could make the system more robust to lighting changes that result in poor corner estimation [73].

Bibliography

[1] 2G Robotics. *ULS-500 PRO Dynamic Underwater Laser Scanner*. http://www.tritech.co.uk/product/gemini-720is-1000m-or-4000m.

[2] M. Agrawal. “A Lie algebraic approach for consistent pose registration for general euclidean motion”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Oct. 2006, pp. 1891–1897.

[3] B. Ahn, A. Dave, A. Veeraraghavan, I. Gkioulekas, and A. C. Sankaranarayanan. “Convolutional Approximations to the General Non-Line-of-Sight Imaging Operator”. In: *Intl. Conf. on Computer Vision (ICCV)*. 2019.

[4] P. F. Alcantarilla, J. Nuevo, and A. Bartoli. “Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces”. In: *British Machine Vision Conf. (BMVC)*. Bristol, UK, 2013.

[5] H. Assalih. “3D Reconstruction and Motion Estimation Using Forward Looking Sonar”. PhD thesis. Heriot-Watt University, 2013.

[6] J. Aulinas, A. Fazlollahi, J. Salvi, X. Lladó, Y Petillot, J. Sawas, and R. Garcıa. “Robust automatic landmark detection for underwater SLAM using side-scan sonar imaging”. In: *Proc. of the 11th Intl. Conf. on Mobile Robots and Competitions*. 2011, pp. 21–26.

[7] M. Aykin and S. Negahdaripour. “On 3-D Target Reconstruction from Multiple 2-D Forward-Scan Sonar Views”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. Genova, Italy, May 2015, pp. 1949–1958.

[8] M. Aykin and S. Negahdaripour. “On feature extraction and region matching for forward scan sonar imaging”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2012, pp. 1–9.

[9] M. Aykin and S. Negahdaripour. “On Feature Matching and Image Registration for Two-dimensional Forward-scan Sonar Imaging”. In: *J. of Field Robotics* 30.4 (2013), pp. 602–623.

[10] M. D. Aykin and S. Negahdaripour. “Forward-look 2-D sonar image formation and 3-D reconstruction”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2013, pp. 1–10.

[11] M. D. Aykin and S. Negahdaripour. “Three-dimensional target reconstruction from multiple 2-d forward-scan sonar views by space carving”. In: *IEEE J. of Oceanic Engineering* 42.3 (2016), pp. 574–589.

[12] M. D. Aykin and S. S. Negahdaripour. “Modeling 2-D lens-based forward-scan sonar imagery for targets with diffuse reflectance”. In: *IEEE J. of Oceanic Engineering* 41.3 (2016), pp. 569–582.

[13] Blueprint Subsea. *Oculus Multibeam Sonars*. `https://www.blueprintsubsea.com/oculus/index.php`.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122.

[15] N. Brahim, D. Gueriot, S. Daniel, and B. Solaiman. “3D reconstruction of underwater scenes using DIDSON acoustic sonar image sequences through evolutionary algorithms”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. Santander, Spain, June 2011.

[16] N. Brahim, D. Guériot, S. Daniely, and B. Solaiman. “3D reconstruction of underwater scenes using image sequences from acoustic camera”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2010, pp. 1–8.

[17] M. Buttafava, J. Zeman, A. Tosi, K. Eliceiri, and A. Velten. “Non-line-of-sight imaging using a time-gated single photon avalanche diode”. In: *Optics express* 23.16 (2015), pp. 20997–21011.

[18] E. J. Candes, M. B. Wakin, and S. P. Boyd. “Enhancing sparsity by reweighted ℓ_1 minimization”. In: *J. of Fourier Analysis and Applications* 14.5-6 (2008), pp. 877–905.

[19] R. Cerqueira, T. Trocoli, G. Neves, S. Joyeux, J. Albiez, and L. Oliveira. “A novel GPU-based sonar simulator for real-time applications”. In: *Computers & Graphics* 68 (2017), pp. 66–76.

[20] R. Cerqueira, T. Trocoli, G. Neves, L. Oliveira, S. Joyeux, and J. Albiez. “Custom Shader and 3D Rendering for computationally efficient Sonar Simulation”. In: *29th Conference on Graphics, Patterns and Images - SIBGRAPI*. 2016.

[21] H. Cho, B. Kim, and S.-C. Yu. “AUV-Based Underwater 3-D Point Cloud Generation Using Acoustic Lens-Based Multibeam Sonar”. In: *IEEE J. of Oceanic Engineering* (2017).

[22] E. Coiras, Y. Petillot, and D. M. Lane. “Multiresolution 3-D reconstruction from side-scan sonar images”. In: *IEEE Trans. on Image Processing* 16.2 (2007), pp. 382–390.

[23] B. Curless and M. Levoy. “A Volumetric Method for Building Complex Models from Range Images”. In: *SIGGRAPH*. 1996, pp. 303–312.

[24] R. DeBortoli, A. Nicolai, F. Li, and G. A. Hollinger. “Assessing perception quality in sonar images using global context”. In: *Proc. IEEE Conference on Intelligent Robots and Systems Workshop on Introspective Methods for Reliable Autonomy*. 2017.

[25] R. DeBortoli, A. Nicolai, F. Li, and G. A. Hollinger. “Real-Time Underwater 3D Reconstruction Using Global Context and Active Labeling”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2019, pp. 6204–6211.

[26] F. Dellaert and M. Kaess. “Factor Graphs for Robot Perception”. In: *Foundations and Trends in Robotics* 6.1-2 (Aug. 2017). http://dx.doi.org/10.1561/2300000043, pp. 1–139.

[27] F. Dellaert. *Factor Graphs and GTSAM: A Hands-on Introduction*. https://bitbucket.org/gtborg/gtsam/.

[28] F. Dellaert. *Factor Graphs and GTSAM: A Hands-on Introduction*. Tech. rep. GT-RIM-CPR-2012-002. GT RIM, Sept. 2012.

[29] K. J. DeMarco, M. E. West, and A. M. Howard. "A computationally-efficient 2D imaging sonar model for underwater robotics simulations in Gazebo". In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2015, pp. 1–7.

[30] E. Eade. *Lie Groups for 2D and 3D Transformations*. Tech. rep. 2017. URL: http://www.ethaneade.org/lie_groups.pdf.

[31] A. Elfes. "Using occupancy grids for mobile robot perception and navigation". In: *Computer* 22.6 (1989), pp. 46–57.

[32] R. Eustice, H. Singh, J. J. Leonard, M. R. Walter, and R. Ballard. "Visually Navigating the RMS Titanic with SLAM Information Filters." In: *Robotics: Science and Systems (RSS)*. 2005, pp. 57–64.

[33] M. Fallon, M. Kaess, H. Johannsson, and J. Leonard. "Efficient AUV Navigation Fusing Acoustic Ranging and Side-scan Sonar". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Shanghai, China, May 2011, pp. 2398–2405.

[34] M. Fiala. "ARTag, a fiducial marker system using digital techniques". In: *Proc. IEEE Intl. Conf. Computer Vision and Pattern Recognition*. June 2005, pp. 590–596.

[35] J. Folkesson, J. Leonard, J. Leederkerken, and R. Williams. "Feature tracking for underwater navigation using sonar". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2007, pp. 3678–3684.

[36] General Dynamics Mission Systems. *Bluefin HAUV*. https://gdmissionsystems.com/products/underwater-vehicles/bluefin-hauv.

[37] G. Guennebaud, B. Jacob, et al. *Eigen v3*. http://eigen.tuxfamily.org. 2010.

[38] D. Gueriot and C. Sintes. “Forward looking sonar data simulation through tube tracing”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2010, pp. 1–6.

[39] T. Guerneve and Y. Petillot. “Underwater 3d reconstruction using blueview imaging sonar”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2015, pp. 1–7.

[40] T. Guerneve, K. Subr, and Y. Petillot. “Three-dimensional reconstruction of underwater objects using wide-aperture imaging SONAR”. In: *J. of Field Robotics* 35.6 (2018), pp. 890–905.

[41] F. Heide, L. Xiao, W. Heidrich, and M. B. Hullin. “Diffuse mirrors: 3D reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors”. In: *Proc. IEEE Intl. Conf. Computer Vision and Pattern Recognition*. 2014, pp. 3222–3229.

[42] F. Heide, M. O’Toole, K. Zang, D. B. Lindell, S. Diamond, and G. Wetzstein. “Non-line-of-sight imaging with partial occluders and surface normals”. In: *ACM Transactions on Graphics* 38.3 (2019), p. 22.

[43] B. Henson. “Image registration for sonar applications”. PhD thesis. University of York, 2017.

[44] B. T. Henson and Y. V. Zakharov. “Attitude-Trajectory Estimation for Forward-Looking Multibeam Sonar Based on Acoustic Image Registration”. In: *IEEE J. of Oceanic Engineering* 99 (2018), pp. 1–14.

[45] F. Hover, R. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. Leonard. “Advanced Perception, Navigation and Planning for Autonomous In-Water Ship Hull Inspection”. In: *Intl. J. of Robotics Research* 31.12 (Oct. 2012), pp. 1445–1464.

[46] T. A. Huang and M. Kaess. “Towards Acoustic Structure from Motion for Imaging Sonar”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Hamburg, Germany, Oct. 2015, pp. 758–765.

[47] T. Huang and M. Kaess. “Incremental Data Association for Acoustic Structure from Motion”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Daejeon, Korea, Oct. 2016, pp. 1334–1341.

[48] T. Huang. “Acoustic Structure from Motion”. MA thesis. Pittsburgh, PA: Carnegie Mellon University, May 2016.

[49] N. Hurtós, S. Nagappa, X. Cufí, Y. Petillot, and J. Salvi. “Evaluation of registration methods on two-dimensional forward-looking sonar imagery”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2013, pp. 1–8.

[50] N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi. “Fourier-based Registration for Robust Forward-looking Sonar Mosaicing in Low-visibility Underwater Environments”. In: *J. of Field Robotics* 32.1 (2014), pp. 123–151.

[51] N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi. “Fourier-Based Registrations for Two-Dimensional Forward-Looking Sonar Image Mosaicing”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Vilamoura, Portugal, Oct. 2012, pp. 5298–5305.

[52] Y. Ji, S. Kwak, A. Yamashita, and H. Asama. “Acoustic camera-based 3D measurement of underwater objects through automated extraction and association of feature points”. In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. Sept. 2016, pp. 224–230.

[53] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard. “Imaging Sonar-Aided Navigation for Autonomous Underwater Harbor Surveillance”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Taipei, Taiwan, Oct. 2010, pp. 4396–4403.

[54] J. Jung, Y. Lee, D. Kim, D. Lee, H. Myung, and H.-T. Choi. “AUV SLAM using forward/downward looking cameras and artificial landmarks”. In: *Proc. of Intl. Symp. of Underwater Technology*. 2017, pp. 1–3.

[55] J. Jung, J.-H. Li, H.-T. Choi, and H. Myung. “Localization of AUVs using visual information of underwater structures and artificial landmarks”. In: *Intelligent Service Robotics* 10.1 (2017), pp. 67–76.

[56] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. “iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree”. In: *Intl. J. of Robotics Research, IJRR* 31.2 (Feb. 2012), pp. 216–235.

[57] J. Kim and S.-C. Yu. “Convolutional neural network-based real-time ROV detection using forward-looking sonar image”. In: *IEEE/OES Autonomous Underwater Vehicle (AUV) Symposium*. 2016, pp. 396–400.

[58] K Kim, N Neretti, and N Intrator. “Mosaicing of acoustic camera images”. In: *IET Proceedings Radar, Sonar & Navigation* 152.4 (2005), pp. 263–270.

[59] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. “A method for large-scale ℓ_1-regularized least squares”. In: *IEEE J. on Selected Topics in Signal Processing* 1.4 (2007), pp. 606–617.

[60] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao. “Chisel: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device using Spatially Hashed Signed Distance Fields”. In: *Robotics: Science and Systems (RSS)*. Vol. 4. 2015, p. 1.

[61] S. Kwak, Y. Ji, A. Yamashita, and H. Asama. “3-D reconstruction of underwater objects using arbitrary acoustic views”. In: *Mechatronics (MECATRONICS)/17th International Conference on Research and Education in Mechatronics (REM), 2016 11th France-Japan & 9th Europe-Asia Congress on*. 2016, pp. 74–79.

[62] S. Kwak, Y. Ji, A. Yamashita, and H. Asama. “Development of acoustic camera-imaging simulator based on novel model”. In: *Environment and Electrical Engineering (EEEIC), 2015 IEEE 15th International Conference on*. 2015, pp. 1719–1724.

[63] S. Kwon, J. Park, and J. Kim. “3D reconstruction of underwater objects using a wide-beam imaging sonar”. In: *Proc. of Intl. Symp. of Underwater Technology*. 2017, pp. 1–4.

[64] G. Lamarche, X. Lurton, A.-L. Verdier, and J.-M. Augustin. “Quantitative characterisation of seafloor substrate and bedforms using advanced processing of multibeam backscatter—Application to Cook Strait New Zealand”. In: *Continental Shelf Research* 31.2 (2011), S93–S109.

[65] Y. Lee, T. Kim, J. Choi, and H.-T. Choi. “Preliminary result on object shape reconstruction using an underwater forward-looking sonar”. In: *Ubiquitous Robots and Ambient Intelligence (URAI), 13th International Conference on*. 2016, pp. 7–10.

[66] J. Li, P. Ozog, J. Abernethy, R. M. Eustice, and M. Johnson-Roberson. “Utilizing high-dimensional features for real-time robotic applications: Reducing the curse of dimensionality for recursive Bayesian estimation”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2016, pp. 1230–1237.

[67] J. Li, M. Kaess, R. Eustice, and M. Johnson-Roberson. “Pose-graph SLAM using forward-looking sonar”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 2330–2337.

[68] T. I. Limited. *Gemini 720is - Multibeam Imaging Sonar*. http://www.tritech.co.uk/product/gemini-720is-1000m-or-4000m.

[69] J. Long, E. Shelhamer, and T. Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proc. IEEE Intl. Conf. Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[70] N. T. Mai, H. Woo, Y. Ji, Y. Tamura, A. Yamashita, and H. Asama. “3-D Reconstruction of Underwater Object Based on Extended Kalman Filter by Using Acoustic Camera Images”. In: *International Federation of Automatic Control PapersOnLine* 50.1 (2017), pp. 1043–1049.

[71] N. T. Mai, H. Woo, Y. Ji, Y. Tamura, A. Yamashita, and H. Asama. “3D reconstruction of line features using multi-view acoustic images in underwater environment”. In: *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2017 IEEE International Conference on*. 2017, pp. 312–317.

[72] N. T. Mai, Y. Ji, H. Woo, Y. Tamura, A. Yamashita, and H. Asama. “Acoustic Image Simulator Based on Active Sonar Model in Underwater Environment”. In: *15th International Conference on Ubiquitous Robots (UR)*. 2018, pp. 775–780.

[73] J. G. Mangelson, R. W. Wolcott, P. Ozog, and R. M. Eustice. “Robust visual fiducials for skin-to-skin relative ship pose estimation”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2016.

[74] J. Matas, O. Chum, M. Urban, and T. Pajdla. “Robust wide-baseline stereo from maximally stable extremal regions”. In: *Image and Vision Computing* 22.10 (2004), pp. 761–767.

[75] S. Negahdaripour. “On 3-D Motion Estimation From Feature Tracks in 2-D FS Sonar Video”. In: *IEEE Trans. Robotics* 29.4 (Aug. 2013), pp. 1016–1030.

[76] S. Negahdaripour. “Application of Forward-Scan Sonar Stereo for 3-D Scene Reconstruction”. In: *IEEE J. of Oceanic Engineering* (2018).

[77] S. Negahdaripour, M. Aykin, and S. Sinnarajah. “Dynamic scene analysis and mosaicing of benthic habitats by FS sonar imaging-issues and complexities”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2011, pp. 1–7.

[78] S. Negahdaripour, P. Firoozfam, and P. Sabzmeydani. “On processing and registration of forward-scan acoustic video imagery”. In: *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*. 2005, pp. 452–459.

[79] S. Negahdaripour, V. M. Milenkovic, N. Salarieh, and M. Mirzargar. “Refining 3-D object models constructed from multiple FS sonar images by space carving”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2017, pp. 1–9.

[80] J. Neira and J. D. Tardós. “Data association in stochastic mapping using the joint compatibility test”. In: *IEEE Trans. Robotics* 17.6 (2001), pp. 890–897.

[81] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. “DTAM: Dense tracking and mapping in real-time”. In: *Intl. Conf. on Computer Vision (ICCV)*. 2011, pp. 2320–2327.

[82] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, et al. “Navigating, recognizing and describing urban spaces with vision and lasers”. In: *Intl. J. of Robotics Research* 28.11-12 (2009), pp. 1406–1433.

[83] P. M. Newman, J. J. Leonard, and R. J. Rikoski. “Towards Constant-Time SLAM on an Autonomous Underwater Vehicle Using Synthetic Aperture Sonar”. In: *Proc. of the Intl. Symp. of Robotics Research (ISRR)*. 2005, pp. 409–420.

[84] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. “Real-time 3D reconstruction at scale using voxel hashing”. In: *ACM Transactions on Graphics (ToG)* 32.6 (2013), p. 169.

[85] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto. “Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2017, pp. 1366–1373.

[86] E. Olson and Y. Li. “IPJC: The Incremental Posterior Joint Compatibility Test for Fast Feature Cloud Matching”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Vilamoura, Portugal, Oct. 2012, pp. 3467–3474.

[87] E. Olson. “AprilTag: A Robust and Flexible Visual Fiducial System”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2011.

[88] M. O’Toole, D. B. Lindell, and G. Wetzstein. “Confocal non-line-of-sight imaging based on the light-cone transform”. In: *Nature* 555.7696 (2018), p. 338.

[89] P. Ozog, M. Johnson-Roberson, and R. M. Eustice. “Mapping underwater ship hulls using a model-assisted bundle adjustment framework”. In: *Robotics and Autonomous Systems, Special Issue on Localization and Mapping in Challenging Environments* 87 (2017), pp. 329–347.

[90] P. Perona and J. Malik. “Scale-space and edge detection using anisotropic diffusion”. In: *IEEE Trans. Pattern Anal. Machine Intell.* 12.7 (1990), pp. 629–639.

[91] Y. Petillot, S. Reed, and J. Bell. “Real time AUV pipeline detection and tracking using side scan sonar and multi-beam echo-sounder”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2002, pp. 217–222.

[92] M. Pizzoli, C. Forster, and D. Scaramuzza. “REMODE: Probabilistic, monocular dense reconstruction in real time”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2014, pp. 2609–2616.

[93] E. Prados and O. Faugeras. “Shape from shading”. In: *Handbook of Mathematical Models in Computer Vision*. 2006, pp. 375–388.

[94] Qualisys AB. *Oqus Underwater*. https://www.qualisys.com/cameras/oqus-underwater/.

[95] I. T. Ruiz, S. De Raucourt, Y. Petillot, and D. M. Lane. “Concurrent mapping and localization using sidescan sonar”. In: *IEEE J. of Oceanic Engineering* 29.2 (2004), pp. 442–456.

[96] H. Sac, M. K. Leblebicioğlu, and G. Akar. “2D high-frequency forward-looking sonar simulator based on continuous surfaces approach”. In: *Turkish Journal of Electrical Engineering & Computer Sciences* 23.Sup. 1 (2015), pp. 2289–2303.

[97] D. T. Sandwell, R. D. Müller, W. H. F. Smith, E. Garcia, and R. Francis. “New global marine gravity model from CryoSat-2 and Jason-1 reveals buried tectonic structure”. In: *Science* 346.6205 (2014), pp. 65–67.

[98] H. Sekkati and S. Negahdaripour. “3-D motion estimation for positioning from 2-D acoustic video imagery”. In: *Iberian Conf. on Pattern Recognition and Image Analysis*. 2007, pp. 80–88.

[99] X. Shen, E. Frazzoli, D. Rus, and M. H. Ang. “Fast Joint Compatibility Branch and Bound for feature cloud matching”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2016, pp. 1757–1764.

[100] Y. S. Shin, Y. Lee, H. T. Choi, and A. Kim. “Bundle adjustment from sonar images and SLAM application for seafloor mapping”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. Oct. 2015, pp. 1–6.

[101] G. Shippey, M. Jonsson, and J. N. Pihl. “Position correction using echoes from a navigation fix for synthetic aperture sonar imaging”. In: *IEEE J. of Oceanic Engineering* 34.3 (2009), pp. 294–306.

[102] G. Sibley, L. Matthies, and G. Sukhatme. “Sliding window filter with application to planetary landing”. In: *J. of Field Robotics* 27.5 (2010), pp. 587–608.

[103] Sound Metrics Corporation. *SoundMetrics Aris*. `http://www.soundmetrics.com/Products/ARIS-Sonars`.

[104] Sound Metrics Corporation. *SoundMetrics Didson 300 Specifications*. `http://www.soundmetrics.com/Products/DIDSON-Sonars/DIDSON-300-m/`.

[105] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. “A Benchmark for the Evaluation of RGB-D SLAM Systems”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Oct. 2012, pp. 573–580.

[106] M. Sung, H. Cho, H. Joe, B. Kim, and S.-C. Yu. “Crosstalk Noise Detection and Removal in Multi-beam Sonar Images Using Convolutional Neural Network”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2018, pp. 1–6.

[107] P. Teixeira, M. Kaess, F. Hover, and J. Leonard. “Underwater inspection using sonar-based volumetric submaps”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Daejeon, Korea, Oct. 2016, pp. 4288–4295.

[108] C.-Y. Tsai, K. N. Kutulakos, S. G. Narasimhan, and A. C. Sankaranarayanan. “The geometry of first-returning photons for non-line-of-sight imaging”. In: *Proc. IEEE Intl. Conf. Computer Vision and Pattern Recognition*. 2017, pp. 7216–7224.

[109] M. VanMiddlesworth. “Toward autonomous underwater mapping in partially structured 3D environments”. MA thesis. Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2014.

[110] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. G. Bawendi, and R. Raskar. “Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging”. In: *Nature Communications* 3 (2012), p. 745.

[111] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. “Pose tracking from natural features on mobile phones”. In: *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*. 2008, pp. 125–134.

[112] Y. Wang, Y. Ji, H. Woo, Y. Tamura, A. Yamashita, and A. Hajime. “3D Occupancy Mapping Framework Based on Acoustic Camera in Underwater Environment”. In: *IFAC-PapersOnLine* 51.22 (2018), pp. 324–330.

[113] Y. Wang, Y. Ji, H. Woo, Y. Tamura, A. Yamashita, and H. Asama. “Three-dimensional Underwater Environment Reconstruction with Graph Optimization Using Acoustic Camera”. In: *IEEE/SICE Intl. Symp. on System Integration (SII)*. 2019, pp. 28–33.

[114] E. Westman and M. Kaess. *Underwater AprilTag SLAM and Extrinsics Calibration for AUV Localization*. Tech. rep. CMU-RI-TR-18-43. Robotics Institute, Carnegie Mellon University, Sept. 2018.

[115] E. Westman, I. Gkioulekas, and M. Kaess. “A theory of Fermat paths for 3-D imaging sonar reconstruction”. In: *submission to IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020.

[116] E. Westman, I. Gkioulekas, and M. Kaess. “A volumetric albedo framework for 3D imaging sonar reconstruction”. In: *submission to IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020.

[117] E. Westman, A. Hinduja, and M. Kaess. “Feature-Based SLAM for Imaging Sonar with Under-Constrained Landmarks”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Brisbane, Australia, May 2018, pp. 3629–3636.

[118] E. Westman and M. Kaess. “Degeneracy-Aware Imaging Sonar Simultaneous Localization and Mapping”. In: *IEEE J. of Oceanic Engineering* (2019). DOI: 10.1109/JOE.2019.2937946.

[119] S. Xin, S. Nousias, K. N. Kutulakos, A. C. Sankaranarayanan, S. G. Narasimhan, and I. Gkioulekas. “A theory of Fermat paths for non-line-of-sight shape reconstruction”. In: *Proc. IEEE Intl. Conf. Computer Vision and Pattern Recognition*. 2019, pp. 6800–6809.

[120] Y. Yang and G. Huang. “Acoustic-inertial underwater navigation”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Singapore, May 2017, pp. 4927–4933.

[121] B. Zerr and B. Stage. “Three-dimensional reconstruction of underwater objects from a sequence of sonar images”. In: *Proc. of the Intl. Conf. on Image Processing*. Vol. 3. 1996, pp. 927–930.

[122] J. Zhang, M. Kaess, and S. Singh. “On Degeneracy of Optimization-based State Estimation Problems”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Stockholm, Sweden, May 2016, pp. 809–816.

[123] J. Zhang, F. Sohel, H. Bian, M. Bennamoun, and S. An. “Forward-looking sonar image registration using polar transform”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2016, pp. 1–6.