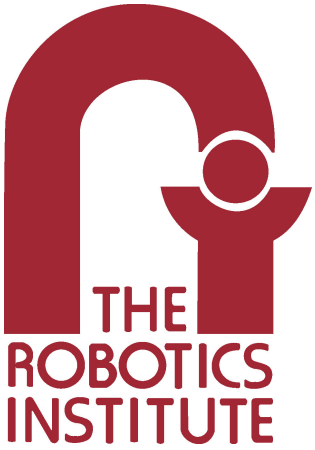
Acoustic Structure from Motion

Tiffany A. Huang

May 2016



Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
CMU-RI-TR-16-08

Thesis Committee
Prof. Michael Kaess, Chair
Prof. David Wettergreen
Sanjiban Choudhury

Contents

Abstract

Although the ocean spans most of the Earth's surface, our ability to explore and perform tasks underwater is still limited by high costs and slow, inefficient 3D mapping and localization techniques. Due to the short propagation range of light underwater, imaging sonar or forward looking sonar (FLS) is commonly used for autonomous underwater vehicle (AUV) navigation and perception. A FLS provides bearing and range information to a target, but the elevation of the target is unknown within the sensor's field of view. Hence, current state-of-the-art techniques commonly make a flat surface (planar) assumption so that the FLS data can be used for navigation. Towards expanding the possibilities of underwater operations, a novel approach, entitled acoustic structure from motion (ASFM), is presented for recovering 3D scene structure from multiple 2D sonar images, while at the same time localizing the sonar. Unlike other methods, ASFM does not require a flat surface assumption and is capable of utilizing information from many frames, as opposed to pairwise methods that can only gather information from two frames at once. The optimization of several sonar readings of the same scene from different poses, the acoustic equivalent of bundle adjustment, and automatic data association is formulated and evaluated on both simulated data and real FLS sonar data.

1 Introduction

1.1 Motivation

Mapping and state estimation have been widely explored for autonomous vehicles that operate on land and in the air. However, for an environment that spans the majority of our planet Earth, surprisingly little progress has been made towards the same autonomous abilities underwater. For instance, the rift valley of the Mid-Atlantic Ridge, an underwater mountain range and one of the largest geographical features in the world, was not explored by humans until 1973, four years after the first humans landed on the Moon [15]! Currently, most underwater tasks are performed by human divers or remotely operated vehicles (ROVs). Autonomous underwater vehicles (AUVs) open the door to exciting new possibilities for underwater exploration such as venturing into areas too dangerous for human divers or exploring large areas much faster and more efficiently. Furthermore, AUVs have the potential to eliminate the tedium and high costs of ROV missions.

More specifically, in this work we focus on the problem of simultaneous localization and mapping (SLAM) for AUVs, or building a map and pinpointing the vehicle's location without any prior knowledge about the environment. One particular challenge underwater is the necessity of non-conventional sensors such as sonar. Due to the turbidity of some water environments as well as the short propagation range of light in water, more common and well-studied sensors such as cameras and LIDAR do not work well underwater. As for localization, GPS cannot be used since radio waves do not travel well in water.

Feature extraction and data association, or finding which measurements from different views correspond to the same object, make up the first part of the SLAM problem. Once feature correspondences are known, the constraints can then be fed into the second part, the optimization, to find the maximum likelihood set of robot poses and landmark positions. Data association is crucial because incorrect correspondences can drastically degrade the quality of the resulting map and trajectory. An erroneous data association will pull the poses and landmarks in the optimization out of their correct positions in an attempt to satisfy incorrect constraints. Thus, it is important for the data association algorithm to be as accurate and robust as possible.

Due to the challenges mentioned above, SLAM algorithms using sonar have not been well-studied or developed for general underwater environments. Towards real-time autonomous navigation and creating a faster and more accurate 3D map with sonar, we introduce the concept of acoustic structure from motion (ASFM),

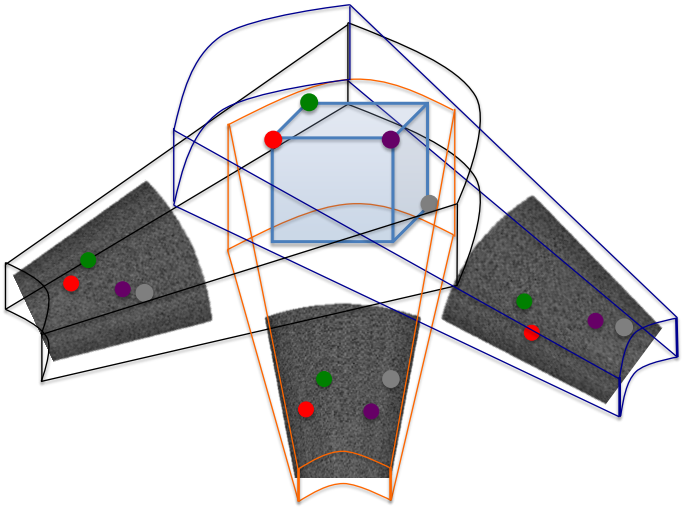


Figure 1.1: Multiple imaging sonar views of a scene allow recovery of 3D position of point features, even though the individual views themselves do not provide elevation information about the features.

using multiple, general sonar viewpoints of the same scene to reconstruct the 3D structure of select point features while minimizing the effects of accumulating error (Fig. 1.1). In this work, we formulate much of the theoretical basis of the approach and focus on its integration with odometry measurements received from other on-board sensors. Furthermore, we explore the acoustic equivalent of bundle adjustment [22], the geometric optimization in traditional structure from motion for cameras. Much of this work can also be found in our conference publication [10]. To address the challenge of ensuring the data association is accurate and robust, we additionally introduce a novel algorithm that uses a tree of correspondences similar to that of Joint Compatibility Branch and Bound [17].

ASFM has applications in real-time navigation for AUVs in general 3D environments. Unlike previous approaches, our solution does not make any assumptions about the planarity of the environment in order to localize the sonar. Additionally, ASFM is able to use information gathered from multiple sonar images to constrain the 3D geometry of the scene and the motion of the vehicle better than a pairwise approach.

In this work, forward-looking sonar (FLS) is used, but ASFM is not limited to this type of sonar. An interesting step for future work would be applying ASFM to other types of sonar such as side-scan sonar [6]. FLS is an obvious choice because it is one of the less expensive types of sonar and its larger field of view allows for faster imaging of an environment. Currently, beam-steering 3D forward-

looking sonar sensors are available (e.g. Blueview 3DFLS), but they are both more expensive and slower to image a given volume (because of the low speed of sound in water), requiring up to 4 seconds for a single sweep at a short 6 m range, and more time for larger ranges. Thus, for many applications, it is advantageous to apply a 3D reconstruction technique with a FLS rather than utilize a 3D sonar directly.

We explore the optimization and automatic data association of ASFM. Most sections will be split into these two parts to discuss the methodology and results behind each component.

1.2 Problem Statement

In order to explore and work in the watery environments that cover the vast majority of our planet, it is necessary to have a reliable way to image and map general scenes underwater. While several methods exist to process sonar images, most require a planar assumption about the scene. How can we extend our understanding of sonar images to encompass general scenes for 3D mapping and AUV navigation?

Our work makes the following contributions:

1. **We present a novel method, acoustic structure from motion, for localization and general-scene 3D mapping underwater using sonar.**
2. **We present a novel automatic data association algorithm for sonar images.**
3. **We demonstrate our ability to localize the sonar and recover 3D structure from simulation and real data sequences.**

2 Related Work

2.1 Localization and 3D Reconstruction from Imaging Sonar

Various other works have explored different ways to localize the AUV from sonar images, but most current methods require a planar scene assumption. Johannsson et al. [12] and Hover et al. [9] extract points with high gradients from the sonar image and cluster the points to use as features. Next, a normal distribution transform algorithm is applied to serve as a model for image registration. The entire trajectory of the AUV is put into a pose-graph smoothing algorithm, and the optimized trajectory shows significant improvements over dead reckoning from the Doppler Velocity Log (DVL). However, to solve the ambiguity in elevation of the points presented by sonar, the points are assumed to lie on a plane that is level with the vehicle. This planar assumption works well for the non-complex areas of a ship hull, the main application of their work, but induces large errors for many other environments. ASFM does not require this assumption, making it useful for a wider range of applications. Hurtos et al. [11] explore a different approach, using Fourier-based techniques instead of feature points for registration. However, the authors primarily focus on applications in 2D mapping, so do not address 3D geometry in detail.

To recover 3D geometry of a scene using imaging sonar, most techniques employ a pairwise registration approach. Babaee et al. [4] use a stereo imaging system composed of one sonar and one optical camera where the centers of the two sensors' coordinate systems and their axes align. The trajectory of the stereo system is calculated using opti-acoustic bundle adjustment. Assalih [1] once again exploits the stereo idea, but instead uses two imaging sonars placed one on top of the other. In contrast, ASFM requires only one sensor and water turbidity is not an issue because no optical cameras are involved. Our work is more similar to Brahim et al. [5] where point-based features are used with evolutionary algorithms to recover 3D geometry from pairs of sonar frames. Unlike Assalih and Brahim however, ASFM is capable of using information from multiple viewpoints as opposed to only pairs of images. Multiple viewpoints add more information and can further constrain the problem to result in more accurate reconstruction than pairwise matching.

Aykin and Negahdaripour [3] relax the planar assumption for pairwise matching of sonar frames but still assume a locally planar surface in order to include shadow information. They show improvements over Johannsson [12] by instead applying a Gaussian distribution transform to the images. Negahdaripour [16] extends this

work to feature tracking and visual odometry in sonar video. The same authors present a space-carving method [2] for recovering 3D geometry from multiple 2D forward-looking sonar images at known poses. Finding the closest edge of an object in multiple sonar images provides information about the occupancy of 3D voxels in the sonar field of view. This method achieves 3D reconstruction without the need for data association and feature extraction. However, ASFM constrains both the motion of the sonar as well as landmark positions, so unlike the space-carving method, the sonar poses do not need to be known a priori.

2.2 Data Association for Imaging Sonar

Not much previous work has been done on automatic data association for imaging sonar. Most of the related ideas once again rely on a planar assumption of the scene. For instance, Leonard et al. [14] use a multiple hypothesis tracking (MHT) algorithm to perform data association and reconstruct the geometry of a static, rigid, 2D environment. Similar to our algorithm, MHT creates a tree of possible hypotheses matching measurements to features, and the tree is pruned based on the likelihood of each hypothesis. MHT in this work assumes that the landmark can be initialized with one measurement, which is not true for 3D scenes. Ribas et al. [20] use an individual compatibility test with a χ^2 threshold to determine which previous features could be correspondences. A nearest neighbor criterion is then applied to select the previously seen feature with the smallest Mahalanobis distance to the current test point. Once again, the authors in this paper make a planar assumption by only imaging planar objects.

Petillot et al. [19] present methods for 2D obstacle mapping and avoidance, the main application being surveys of the seabed. The authors use a single Kalman filter, which does not include the vehicle state, to track objects in the forward-looking sonar images. For data association, the authors use a nearest neighbor algorithm that takes into account both the position and area of the observations. The nearest neighbor criterion does not take into account the joint hypothesis of the entire set of features like our data association algorithm and therefore is more susceptible to accepting spurious features and producing incorrect correspondences.

In [7], the authors discuss a system that uses FLS sonar images to find and navigate to a previously mapped target. For data association, a scoring algorithm was used that takes into account positive information of features detected by the sonar and negative information of features that were expected to be seen but were not detected. Our work is similar to some of the ideas such as scoring SLAM graph hypotheses, but our data association applies to matching more generally with previous sonar images for 3D reconstruction instead of against a prior map for localization.

3 Sonar Geometry

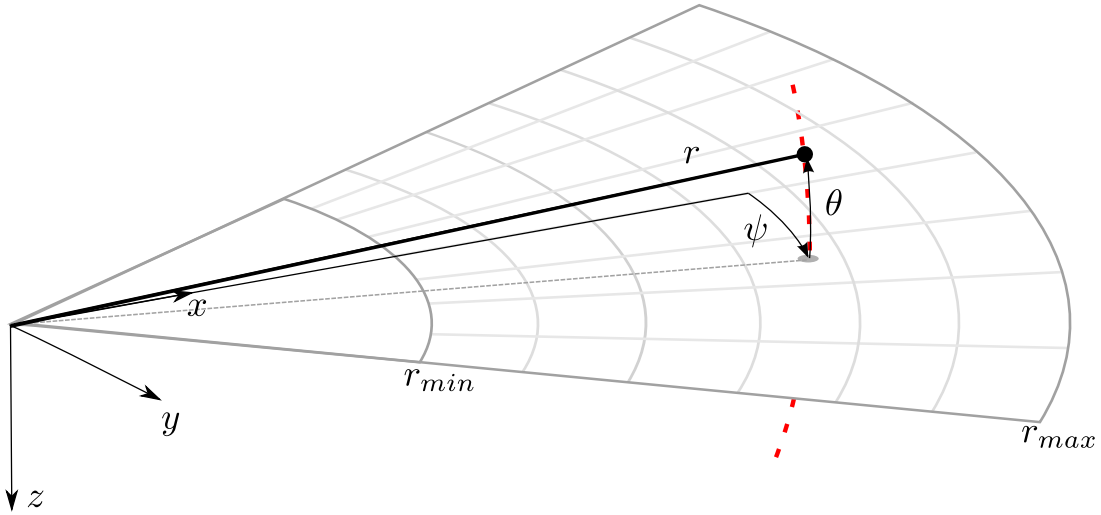


Figure 3.1: Imaging sonar geometry. Any 3D point along the dashed red elevation arc will appear as the same image point in the $x-y$ plane. Bearing angle ψ and range r are measured, but the elevation angle θ is lost in the projection process.

Before discussing ASFM further, it is important to understand the information provided in a FLS sonar image. The imaging sonar sends out an acoustic ping and measures the intensity of acoustic waves reflected back from objects inside of a frustum defined by the sonar's bearing field of view (FOV) (deg), elevation FOV (deg), and minimum and maximum range (m). The returns from one ping are put together to form an intensity image, where each pixel represents a bearing and range bin, discretized per the specifications of the sonar. As seen in Fig. 3.1, the sonar only provides partial information about a feature (bearing ψ and range r) and does not provide its elevation angle θ. In a 1-D array of receivers, which is typical for a FLS, the difference between the time it takes for one receiver to detect a signal and another receiver to detect the same signal denotes the bearing of the feature. The range is determined by the time of flight of the sound wave. The elevation of the point is lost, as all points along an elevation arc will collapse to the same pixel in the sonar image. Since one dimension of the feature is missing, one sonar image is not sufficient to recover 3D geometry.

3.1 Cartesian to Polar Transformation

In all of the imaging sonar data sequences we use for our experiments, features are extracted from the Cartesian sonar image. The original polar (bearing/range) image returned by the sonar is converted to a Cartesian sonar image by finding and solving an analytic function that describes the mapping between the pixels of a Cartesian image of a given width in the sonar frame and the corresponding pixels for the polar image in the sonar frame. This mapping is also used to convert features in the Cartesian image to bearing/range measurements.

All points in the sonar field of view are projected along a circular arc onto the plane of the sonar, so points returned by the sonar can lie anywhere along an arc spanning the vertical aperture of the sensor. Note that this projection implies that in the sonar point of view, all points have $z_s = 0$. The following equations describe the mapping between the Cartesian image coordinates (u, v) and the polar image bearing and range bin (n_b, n_r).

$$\gamma = \frac{w}{2r_{max}\sin(\frac{\psi_{max}}{2})} \tag{3.1}$$

$$x_s = \frac{u - \frac{w}{2}}{\gamma} \tag{3.2}$$

$$y_s = r_{max} - \frac{v}{\gamma}. \tag{3.3}$$

$$r = \sqrt{x_s^2 + y_s^2} \tag{3.4}$$

$$\psi = \frac{180}{\pi}\text{atan2}(x_s,\, y_s) \tag{3.5}$$

$$n_r = \frac{N_r(r - r_{min})}{r_{max} - r_{min}} \tag{3.6}$$

$$n_b = M_4(N_b, \psi) \tag{3.7}$$

where γ is a constant, w is the width of the Cartesian image in pixels, r_{min} is the minimum range of the sonar, r_{max} is the maximum range of the sonar, N_r is the number of range bins, ψ_{max} is the bearing field of view of the sonar, and $M_4(N_b, \psi)$ is a third-order polynomial (with 4 coefficients determined by the number of bearing bins (N_b)) given by the sonar manufacturer that accounts for lens distortion. In our experiments with the Sound Metrics DIDSON 300 m sonar, $w = 200$ pixels, $r_{min} = 0.75$ m, $r_{max} = 5.25$ m, $N_r = 512$, $N_b = 96$, and $\psi_{max} = 28.8°$.

The bearing and range bins are then converted to bearing and range measure-

ments (ψ, r) for input into the optimization:

$$\psi = \psi_{max}(\frac{n_b}{N_b} - 0.5) \tag{3.8}$$

$$r = \frac{(r_{max} - r_{min})n_r}{N_r} + r_{min} \tag{3.9}$$

3.2 Sonar and Odometry Models

To evaluate the probability of a sensor measurement for a given variable configuration, we need to define a generative sensor model. The generative model consists of a geometric prediction given a configuration of poses and points with added noise. As is standard in the literature, we assume a Gaussian noise model.

The measurement model for odometry measurements is

$$g(x_{i-1}, x_i) + \mathcal{N}(0, \Lambda_i) \tag{3.10}$$

where $g(x_{i-1}, x_i)$ predicts the odometry measurement between poses x_{i-1} and x_i. $\mathcal{N}(0, \Lambda_i)$ represents the noise sampled from a Gaussian distribution with 0 mean and covariance Λ_i.

Similarly, we define the measurement model for sonar measurements by

$$h(x_i, l_j) + \mathcal{N}(0, \Xi_k) \tag{3.11}$$

where $h(x_i, l_j)$ predicts the sonar measurement (ψ, r) between pose x_i and landmark l_j. $h(x_i, l_j)$ first transforms the landmark location $l_j = (x_g, y_g, z_g)$ into the sonar frame based on pose x_i, obtaining the local coordinates (x_s, y_s, z_s). Bearing ψ and range r are then obtained by

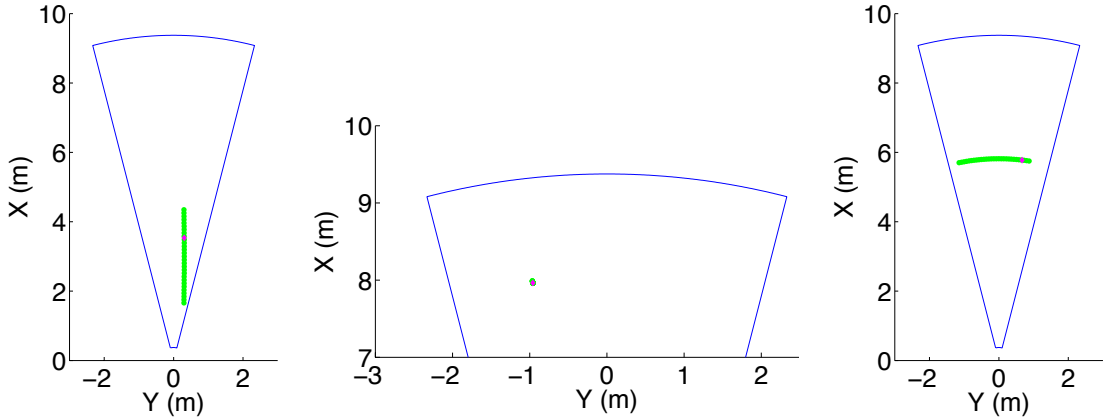
$$r = \sqrt{x_s^2 + y_s^2 + z_s^2} \tag{3.12}$$

$$\psi = \text{atan2}(y_s, x_s). \tag{3.13}$$

$\mathcal{N}(0, \Xi_k)$ represents the noise sampled from a Gaussian distribution with 0 mean and covariance Ξ_k.

3.3 Arc Reprojections

An interesting question is whether sonar has some kind of geometry similar to the epipolar geometry found in cameras. For cameras, one point seen in one image can be reprojected onto a single line, the epipolar line, in another image. Unfortunately for sonar, the reprojection of a point into another image is not



(a) Pitch $-90°$ **(b)** Forward x **(c)** Roll $45°$

Figure 3.2: Elevation arc reprojections (green points) for (a) -90° pitch, (b) forward x, and (c) 45° roll motion. The magenta diamond is the true measurement of the 3D point in the current pose.

that simple. The examples shown in Fig. 3.2 demonstrate several different possible geometries resulting from the elevation arc of one point reprojected into another sonar image.

For $-90°$ pitch (accompanied by forward x and upward z motion so that the sonar FOVs would overlap), one could imagine that instead of the sonar rotating, the elevation arc pitches $-90°$ in the viewpoint of the new sonar frame. Consequently, the elevation arc becomes a distribution of 3D points that looks like a hill at similar bearing but different ranges. Mapped onto the 2D sonar image, this looks like a nearly vertical line, as evidenced by Fig. 3.2a. For the forward x motion case, the top half of the elevation arc above 0 elevation will map to the same points as the bottom half of the elevation arc, so we see a small vertical line, which should be proportional to the arc's curvature. In this example (Fig. 3.2b) the curvature was quite small, resulting in a very short vertical line. Finally, for the 45° roll example, one could once again imagine that instead of the sonar rotating, the elevation arc rolls 45° in the viewpoint of the new sonar frame. The resulting arc (Fig. 3.2c) is now a horizontal arc instead of a vertical arc, and it is shorter than the original arc. The new horizontal arc would be the same length as the original arc if we had rolled 90° instead. From these examples, it is clear that the reprojection of one point into another sonar image does not result in a simple geometry that can be easily exploited. The elevation arc reprojections can appear as many different geometries depending on the motion between sonar poses.

4 Acoustic Structure from Motion

ASFM is inspired by a related problem in computer vision called structure from motion (SFM), which uses multiple camera images of a scene to recover 3D geometry as well as camera locations [8]. Much of the high-level formulation of the two problems are similar because like sonar images, camera images only give 2D information about the scene. However, a critical difference between the two sensors highlights the novelty and challenges of ASFM. Cameras provide elevation and bearing of a feature, but not the depth, while as mentioned before, sonars provide bearing and depth, but not elevation. This difference implies that new sensor models, parameterizations, and degenerate cases will have to be explored before ASFM can be used successfully.

4.1 Pose Graph Formulation

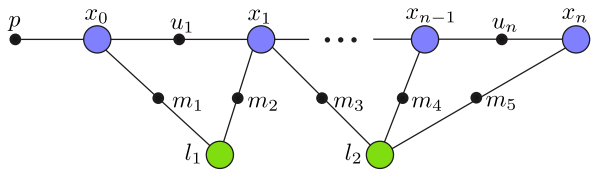


Figure 4.1: Factor graph representation of the acoustic structure from motion problem. Variable nodes consist of the underwater vehicle poses x_i and the point features l_j. The black dots represent factor nodes, which are derived from odometry measurements u_i and feature observations m_k. The unary factor p represents a prior on the first pose that defines the reference frame.

We represent the ASFM problem as a factor graph [13] (Fig. 4.1). A factor graph is a bipartite graph with two node types: variable nodes that represent the poses x_i and landmarks l_j to be estimated, and factor nodes that represent odometry u_i and point feature sonar measurements m_k. An edge in the factor graph connects one factor node with two variable nodes. Here, almost all factors are binary, i.e. they connect only two variables. Only one factor, p, is unary, and it is a prior that defines a reference frame, eliminating otherwise unconstrained degrees of freedom.

The factor graph was chosen because it captures the underlying dependence structure of the ASFM estimation problem. Since the measurements u_i and m_k are known, they are represented as factors of the joint probability over the unknowns, the variable nodes x_i and l_j. In fact, the goal of ASFM is to find the maximum

probability set of landmarks and vehicle poses $\Theta = \{x_i, l_j\}$ given all measurements $Z = \{u_i, m_k\}$. The set Θ^* that satisfies this criteria is defined as

$$\begin{aligned}\Theta^* &= \underset{x}{\operatorname{argmax}}\ p(\Theta|Z) \\ &= \underset{x}{\operatorname{argmax}}\ p(\Theta)p(Z|\Theta) \\ &= \underset{x}{\operatorname{argmax}}\ p(x_0)\prod_{k=1}^{M} p(m_k|x_i, l_j) \\ &\quad \cdot \prod_{i=1}^{N} p(u_i|x_{i-1}, x_i). \end{aligned} \tag{4.1}$$

Here we have used Bayes Theorem to obtain a maximum a posteriori (MAP) solution for Θ^*. We have also exploited the factorization defined by the factor graph, where each term in Eq. 4.1 corresponds to one of the factors in Fig. 4.1.

4.2 Nonlinear Least-Squares

Given a set of measurements from different viewpoints, the most likely set of vehicle poses and landmark positions can be found by solving a nonlinear least squares problem. Nonlinear least squares suffers from several disadvantages including the need for a good initial estimate and the possibility that the solution converges to a local, not global, minimum. However, this type of problem has a simple, known solution. Additionally, alternatives such as an extended Kalman filter (EKF) are too inefficient to run in real-time.

The nonlinear least-squares problem follows directly from the MAP problem of Eq. 4.1 under the assumption of Gaussian noise. Here we use Mahalanobis distance notation defined as:

$$\|x\|_{\Sigma}^2 = x^T\Sigma^{-1}x. \tag{4.2}$$

The nonlinear least-squares problem becomes:

$$\begin{aligned}\Theta^* &= \underset{x}{\operatorname{argmin}}[-\log p(x_0)\prod_{k=1}^{M} p(m_k|x_i, l_j) \\ &\quad \cdot \prod_{i=1}^{N} p(u_i|x_{i-1}, x_i)] \\ &= \underset{x}{\operatorname{argmin}}[\ \|x_0\|_{\Lambda}^2 + \sum_{k=1}^{M} \|h(x_i, l_j) - m_k\|_{\Xi_k}^2 \\ &\quad + \sum_{i=1}^{N} \|g(x_{i-1}, x_i) - u_i\|_{\Lambda_i}^2]. \end{aligned} \tag{4.3}$$

Here we have made use of the monotonicity of the logarithm function.

We find an initial estimate for the feature points by backprojection of the sonar measurements. We use the first observation of each feature, consisting of a range r and bearing ψ measurement. We apply the backprojection function

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = r \begin{bmatrix} \cos\psi\cos\theta \\ \sin\psi\cos\theta \\ \sin\theta \end{bmatrix} \tag{4.4}$$

where we set the unknown elevation angle θ to 0. The sonar pose x_i is then used to convert the point from sonar Cartesian coordinates (x_s, y_s, z_s) to world Cartesian coordinates (x_g, y_g, z_g), which serve as initial guesses for the 3D position of the features.

Starting from this initial estimate, the nonlinear least-squares problem is solved by iterative linearization. For nonlinear measurement functions, nonlinear optimization methods such as Gauss-Newton or the Levenberg-Marquardt algorithm solve a succession of linear approximations in order to approach the minimum. At each iteration of the nonlinear solver, we linearize around the current estimate Θ to get a new, linear least-squares problem in $\mathbf{\Delta}$

$$\underset{\Delta}{\operatorname{argmin}} \left\| A\mathbf{\Delta} - b \right\|^2, \tag{4.5}$$

where $A \in \mathbb{R}^{U \times V}$ is the measurement Jacobian consisting of $U = 6N + 2M$ measurement rows, and $\mathbf{\Delta}$ is an V-dimensional vector, where $V = 6N + 3M$. Note that each odometry measurement has 6 degrees of freedom (DOF) and each sonar measurement has 2, while each vehicle pose has 6 DOF and each landmark has 3 DOF. Note that the covariances Σ_i, which represent covariances such as Λ_i and Ξ_k in Eq. 4.3, have been absorbed into the corresponding block rows of A, making use of

$$\|\mathbf{\Delta}\|_\Sigma^2 = \mathbf{\Delta}^T\Sigma^{-1}\mathbf{\Delta} = \mathbf{\Delta}^T\Sigma^{-\frac{T}{2}}\Sigma^{-\frac{1}{2}}\mathbf{\Delta} = \left\|\Sigma^{-\frac{1}{2}}\mathbf{\Delta}\right\|^2. \tag{4.6}$$

Once $\mathbf{\Delta}$ is found, the new estimate is given by $\Theta \oplus \mathbf{\Delta}$, which is then used as the linearization point in the next iteration of the nonlinear optimization. The operator $\oplus$ is often simple addition, but for overparametrized quantities such as 3D rotations, an exponential map is used instead to locally obtain a minimal representation.

The minimum of the linear system $A\mathbf{\Delta} - \mathbf{b}$ is obtained by Cholesky factorization. By setting the derivative in $\mathbf{\Delta}$ to zero we obtain the normal equations $A^TA\mathbf{\Delta} = A^T\mathbf{b}$. Cholesky factorization yields $A^TA = R^TR$, and a forward and backsubstitution on $R^T\mathbf{y} = A^T\mathbf{b}$ and $R\mathbf{\Delta} = \mathbf{y}$ first recovers $\mathbf{y}$, then the actual solution, the update $\mathbf{\Delta}$.

4.3 Existence of Solution

We discuss under which conditions the system of equations is solvable by analyzing the number of feature points that need to be observed to fully constrain the system. Let N be the number of poses, and M be the number of points to reconstruct. For every pose, there are 6 unknowns $(x, y, z, yaw, pitch, roll)$ and for every point there are 3 unknowns (x, y, z). The first pose is fixed using a prior, so there are 0 degrees of freedom for the first pose. In the case where all features are visible from each pose, there are $2N$ equations for each point, and the system is fully constrained iff:

$$6(N-1) + 3M \leq 2MN \tag{4.7}$$

Since we are not restricted to pairs of sonar views, our simulated examples in later sections use information from 3 sonar viewpoints. From Eq. 4.7 we see that for 3 sonar views, a minimum of 4 points are needed to fully constrain the estimation problem. In our real sonar data experiments, features from 5 poses are used; thus, a minimum of 4 points are needed to make 3D reconstruction possible.

4.4 Relative Parameterization

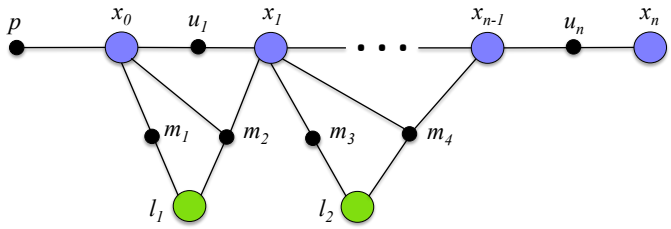


Figure 4.2: The SLAM factor graph using a relative parameterization. All of the landmark measurements are represented relative to the first sonar pose that has seen that landmark.

Depending on the shape of the optimization function and the quality of the initial estimate, the nonlinear optimization can take a long time to converge. In ASFM, this problem is exacerbated by complicated posterior densities created from the parameterization of the landmarks in Cartesian coordinates. As the elevation of a landmark is being optimized, the landmark must move along an elevation arc, which is nonlinear. Additionally, three Cartesian coordinates have to be changed each time the landmark is moved. A similar issue exists in optical SFM, and to improve convergence properties, homogeneous coordinates were introduced as a solution. Along the same lines, we explore an alternative parameterization of

the sonar measurements in hopes of reducing the nonlinearity of the optimization function. As sonar measurements naturally arrive in polar bearing-range coordinates, we investigate a spherical parameterization for features relative to the first sonar pose that has seen that particular landmark. The factor graph with this parameterization is shown in Fig. 4.2.

For this new relative parameterization, the optimization still takes on the form of Eq. 4.3. However, the landmark positions are now stored in spherical coordinates in the frame of the first sonar pose that saw that landmark. Thus, $h(x_i, l_j)$ now involves first converting the landmark position from spherical coordinates to Cartesian coordinates, both in the first sonar pose's frame, as in Eq. 4.4. The landmark is then transformed to the new poses's frame and projected into the new pose to predict the landmark measurement (Eq. 3.12 and Eq. 3.13). In this new relative parameterization, when the landmark's elevation is being optimized, only one coordinate has to change (the elevation angle). Consequently, the spherical parameterization should be much more linear compared to the Cartesian case.

4.5 Degenerate Cases

As is the case for optical structure from motion, there are situations in which a unique solution does not exist. We now discuss three such cases that we have also included in our simulation evaluation.

One of these cases is pure translation in the x-direction. This scenario does not allow us to recover elevation of the point features because the circular arc containing the set of possible 3D points in the sonar geometry for the first pose will intersect the circular arc of the same point seen in the next pose, which differs only in x, at two points. These two intersections cause an ambiguity in the elevation of the points symmetric about the zero plane (Fig. 6.1).

Another case is pure pitch rotation. Since all points lying along a circular arc map to the same point in a sonar image, all of the images from this case would be the same. Consequently, we would not have enough information to recover elevation. However, if the sonar pitched so much that the vehicle would have to translate in the z-direction as well to see the same scene, this motion would be able to recover the points well because the overlapping arcs would overlap in a small region.

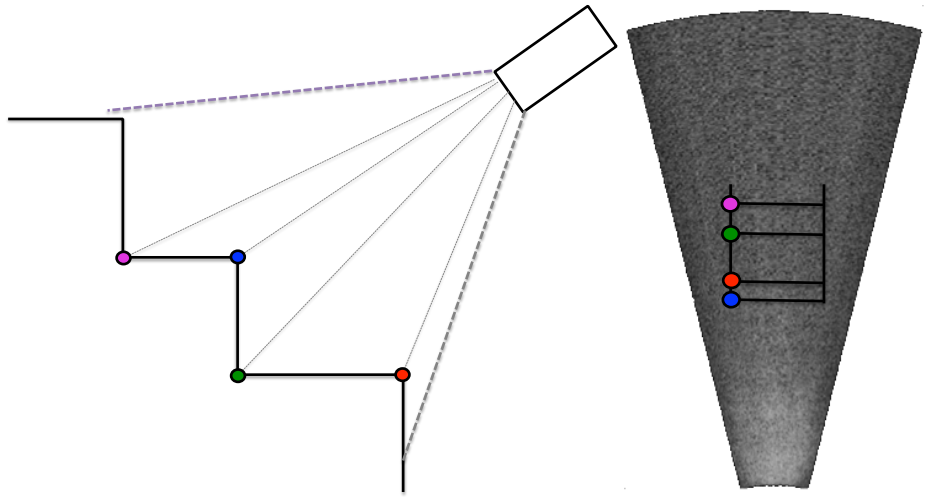
The third situation that results in multiple solutions is pure yaw and y-translation. Like the other two cases discussed, in this kind of trajectory, the elevation arcs would have a large overlap region. The correct elevation of the feature point could lie anywhere in this overlapping region.

5 Automatic Data Association

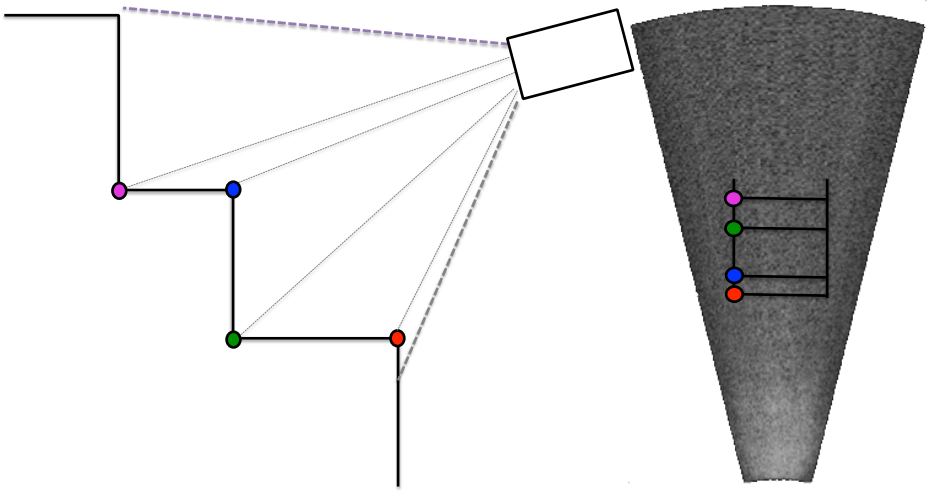
5.1 Data Association Challenges

Unlike camera images, sonar images are much less intuitive to understand and interpret. An example is given in Fig. 5.1. Assume the AUV is imaging a stair-like structure underwater and we have manually picked out some point features that intuition would lead us to believe are stable, like the corners along one edge of the stairs. Since the vertical axis of the image denotes distance along the viewing axis of the sonar and the blue feature appears to be closest to the sonar, the blue point appears as the bottom-most feature in the sonar image. The next closest point to the sonar looks to be the red feature, then the green, then the purple. Note that just looking at the final ordering of the feature points in the sonar image does not give a helpful indication of the true 3D structure. From only the sonar image, it would be almost impossible to tell that in 3D, the blue feature point is in fact between the green and the purple feature points.

To confuse data association further, moving the sonar angle changes the ordering of the feature points because the distance between the features and the sonar changes. Therefore, in (b) of Fig. 5.1, the sonar moves and the resulting sonar image contains a different ordering of feature points. In this case, the sonar moves closer to the red feature point and further from the blue feature point. Consequently, the blue and red features switch places in the second image. Without knowing the exact motion of the AUV, even manually assigning feature correspondences becomes difficult. It would be very challenging to correctly associate the blue feature point in the first image to the blue feature point in the second image.



(a)



(b)

Figure 5.1: Data association for imaging sonar presents several challenges. First, sonar images are very non-intuitive to interpret. The representation of structure in the image does not follow the visual image projection that we are familiar with in camera images. This can be seen in (a) where the order of the colored points do not agree with our intuition based on visual imagery. Second, different angles of sonar viewing could produce similar images but with different correspondences between feature points and real 3D points. The difference between (a) and (b) serves as an example. This complication makes even manual data association difficult.

5.2 Incremental Data Association Algorithm

To initialize the algorithm, the first pose in the data sequence is fixed using a prior in the ASFM factor graph. Additionally, all landmark measurements from the first pose are regarded as new landmarks and stored in a landmark history. Next, incrementally, a new odometry measurement will arrive along with a set of feature measurements from this new pose. If feature measurements are too close to each other, we will discard one to avoid ambiguity. Close feature points often do not add much information about the scene, so not much value is lost by discarding one of the features. In our experiments, if two measurements were within 1° bearing and 0.2 m range of each other, one was discarded.

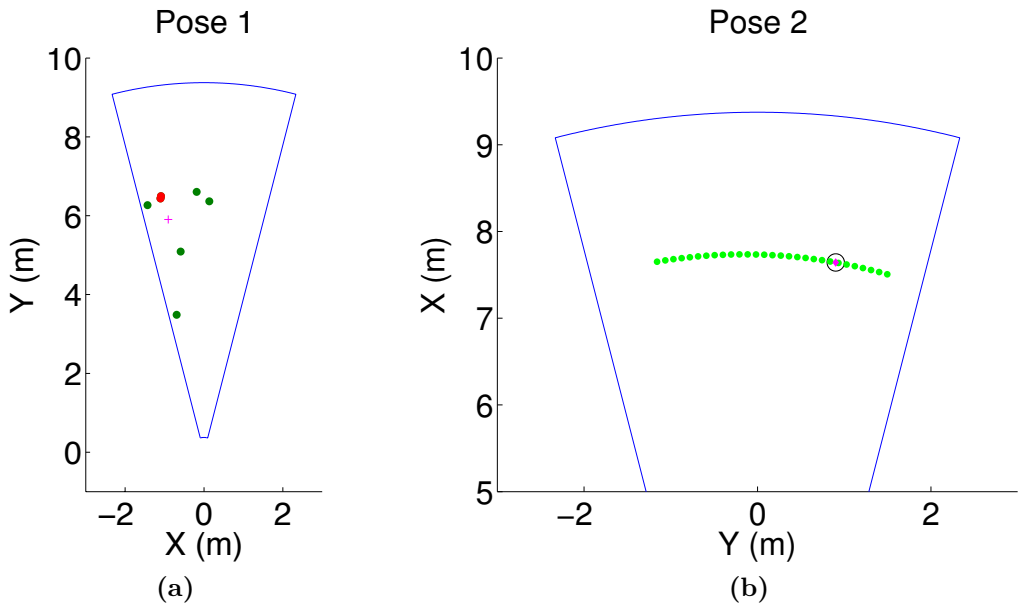


Figure 5.2: An example of finding matches for a given landmark measurement. (a) In the first pose, the two red landmarks are too close together, so one is discarded. The magenta cross is the landmark that is visualized in the next pose. (b) The marked feature is backprojected into an arc of possible 3D points and reprojected into the new pose as the green points. If any of these green points lie within a gating threshold (the circle) of the current feature measurement (diamond), then the landmark seen in pose 1 is considered a possible match.

For each new feature measurement, we look at the reprojection error of each landmark in the stored landmark history to prune possible matches. Each landmark seen so far is backprojected through the most recent pose that has seen the landmark into 1-degree interval 3D points along the elevation arc of possible 3D

points (Fig. 3.1). These 3D points for each landmark are then reprojected into the current pose and the landmark is accepted as a possible match if any of its associated reprojected 3D points lies within a gating threshold of the current feature measurement (Fig. 5.2). The gating threshold, which is Euclidean distance in bearing-range space between the reprojected landmark and the current feature measurement, is set to 0.1 in our experiments. To keep the number of landmark reprojections (and consequently computational time) from growing unboundedly, a landmark is not tested if it has not been seen in the last two frames.

Once all possible matches have been determined for each new feature measurement using the gating threshold, a brute-force exhaustive search over possible match combinations is utilized to find the correct data association hypothesis. More specifically, we build a tree where each level represents a feature observation and its potential matches (including a null match) as determined by the gating threshold. Each path from a node in the top level to a leaf node at the bottom level represents a potential data association hypothesis. The data association hypothesis is accepted if the sum of squared residuals for the posterior position of the landmarks and the robot pose given this hypothesis falls under a $\chi^2_{d,\alpha}$ threshold:

$$\begin{aligned} \|x_0\|^2_\Lambda + \sum_{k=1}^{M} \|h(x_i, l_j) - m_k\|^2_{\Xi_k} \quad + \\ \sum_{i=1}^{N} \|g(x_{i-1}, x_i) - u_i\|^2_{\Lambda_i} \quad < \quad \chi^2_{d,\alpha} \end{aligned} \tag{5.1}$$

Feature measurements and landmarks are not added to the optimization if they are not fully constrained, i.e. they have not been seen by at least two different poses. The $\chi^2_{d,\alpha}$ threshold is determined using d, the degrees of freedom of the factor graph (number of measurements minus number of variables) and α, the confidence parameter (set to 0.99 in our experiments). Null matches, or declaring feature measurements to be new landmarks, are penalized such that the algorithm picks the hypothesis that fits the $\chi^2_{d,\alpha}$ criterion with the fewest null matches. If there are multiple hypotheses that fit the $\chi^2_{d,\alpha}$ criterion with the same number of null matches, the algorithm picks the hypothesis with the smallest optimization residual.

A downside of our algorithm is that the posterior of the entire set of landmarks and vehicle poses has to be calculated for each hypothesis. However, before performing the brute-force exhaustive search, the data association hypotheses are sorted in order of increasing number of null matches. Therefore, if small sonar motion is assumed, which is not necessary for the algorithm to work, but applicable to most situations, then many features will overlap and the correct data association hypothesis will contain few null matches. Thus, the algorithm will not have to search through many hypotheses because once it finds the correct hypothesis, it

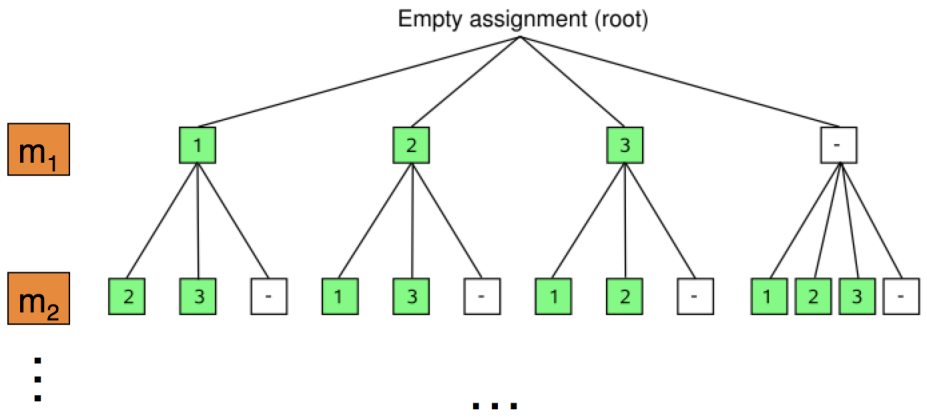


Figure 5.3: The hypothesis tree that is searched through to find the correct data association hypothesis. Each path from the root to a leaf represents one possible data association hypothesis. Each level in the tree represents one measurement (the orange boxes on the left). The nodes at each level are the possible matches determined by the arc reprojection error of the existing landmarks and the gating threshold. In addition, each level has a null-match node signifying the possibility that the measurement corresponds to a new landmark.

will not continue to search through the other possibilities with a larger number of null matches.

In summary, the algorithm goes through the following steps:

1. Set first pose as prior, insert all sonar measurements from first pose into landmark history.

2. For each new pose, insert odometry measurement into the SLAM factor graph. Discard a new sonar measurement if it is too close to another feature seen in this pose.

3. For each sonar measurement from the new pose, backproject each landmark in the landmark history to a set of possible 3D points. Project these candidate 3D points into the new sonar pose and accept the landmark as a possible match if at least one of its candidate 3D points lies within a gating threshold of the current sonar measurement.

4. Find all combinations of data association hypotheses including null matches and sort them in order of increasing number of null matches.

5. Perform a brute-force search through the data association hypotheses. Accept the hypothesis with the smallest posterior residual less than the $\chi^2_{d,\alpha}$ threshold with the fewest number of null matches.

6. Discard landmarks in the landmark history that have not been seen in the last two frames.

6 Experimental Results

6.1 ASFM Optimization and 3D Reconstruction

6.1.1 Simulation Setup

Table 6.1: ASFM Simulated Data Experimental Design

	Value
Number of Monte Carlo samples	1000
Orientation: stddev (deg)	1
Translation: stddev (m)	0.01
Bearing: stddev (deg)	0.2
Range: stddev (m)	0.005
Minimum range of sonar (m)	0.375
Maximum range of sonar (m)	9.375
Bearing FOV of sonar (deg)	28.8
Elevation FOV of sonar (deg)	28
Number of bearing bins	96
Number of range bins	512

We present statistical results on 3D reconstruction using ASFM for multiple types of vehicle motion. The simulation data for this experiment was generated by selecting three sonar poses containing overlapping regions in their fields of view and randomly creating 3D points until at least 15 points were visible in all three sonar frames. Gaussian noise was added to the bearing ($\sigma = 0.2°$) and range ($\sigma = 0.005$ m) components of the ground truth sonar measurements. Similarly, Gaussian noise was added to both rotational ($\sigma = 1°$) and translational components ($\sigma = 0.01$ m) of the odometry between consecutive poses. For the 3D reconstruction simulations, a pose at the origin $(0, 0, 0, 0, 0, 0)$ was added to the factor graph with a prior factor. The prior had the same uncertainty as the odometry and the pose at the origin did not have any landmark measurements. The simulated sonar and environment specifications are listed in Tab. 6.1. Five different sonar trajectories were analyzed:

1. *General Motion*: In this trajectory, the sonar undergoes an x, y, and z-translation as well as changes in yaw, pitch, and roll.

2. *Pitch + Z*: To represent a well-constrained case, we have the sonar go through purely pitch and z-translation motion. This configuration is particularly

well-constrained because the different arcs along which a point could lie intersect with very small overlapping regions.

3. *Forward Motion*: One degenerate case is shown through this trajectory of pure x-translation (2 m total) (Fig. 6.1). For this motion, the arcs along which the points could lie intersect in two regions, which creates an ambiguity as to whether the point lies in an elevation above or below the zero elevation plane.

4. *Yaw + Y*: Another degenerate case is explored using a pure yaw and y-translation trajectory. The elevation arcs in this case have a large overlapping region, making the z-coordinate of feature points difficult to recover accurately.

5. *Roll*: For this trajectory, the sonar undergoes pure roll motion, 45° in total. This case is fairly well-constrained because the motion rotates the elevation arc about the actual elevation point.

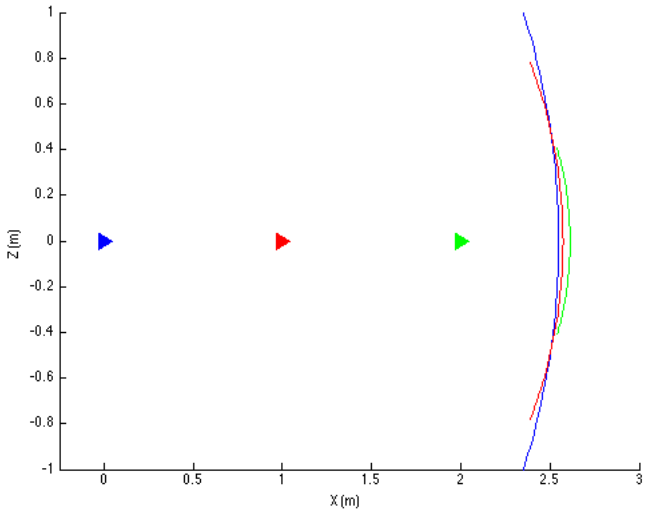
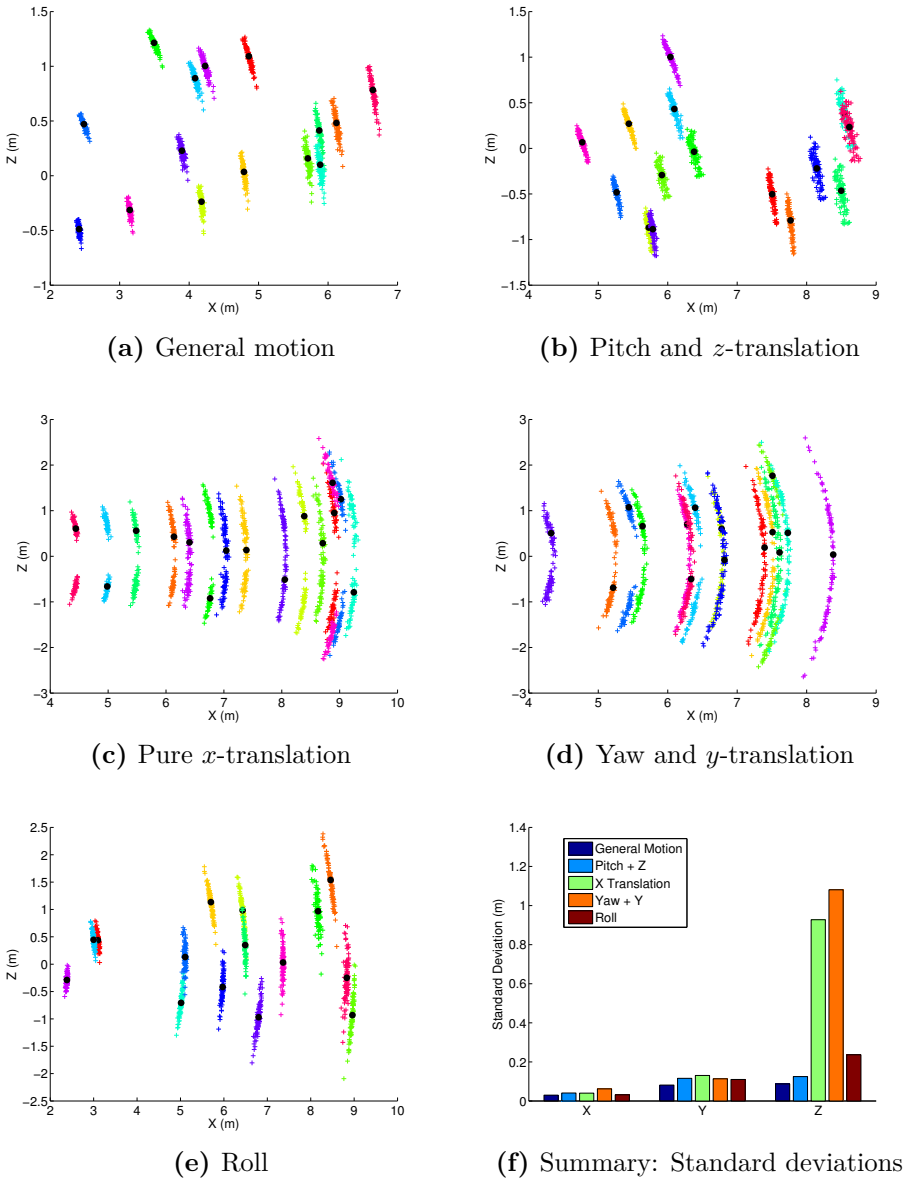


Figure 6.1: An example of a degenerate case, pure forward motion, where the elevation arcs from the different sonar poses overlap in multiple regions, creating an ambiguity of the elevation of the feature point symmetric about the sonar plane.

We use Monte Carlo sampling to compare the variations in recovered point features for each motion type. Each sonar trajectory was simulated 1,000 times with the same 3D points and noise randomly sampled each time from the same Gaussian distribution with $\mu = 0$ and σ as described above.

6.1.2 Simulation Results



(a) General motion **(b)** Pitch and z-translation

(c) Pure x-translation **(d)** Yaw and y-translation

(e) Roll **(f)** Summary: Standard deviations

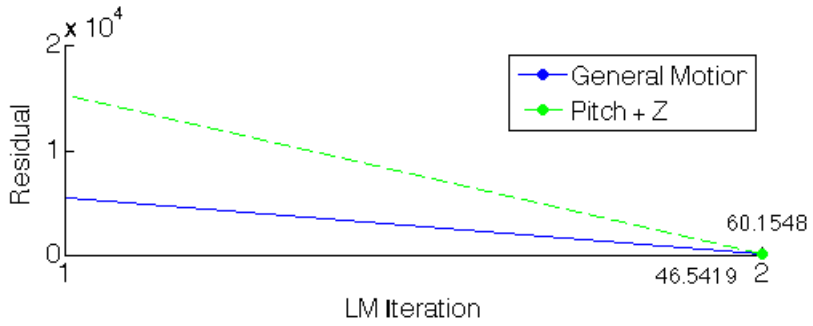
Figure 6.2: Monte Carlo simulation results for the different motion sequences. (a-e) Cluster of point estimation results from 100 random noise trials of the five different simulations. The black dots denote ground truth. (f) Standard deviation of the error for the recovered x, y, and z coordinates over 1,000 Monte Carlo simulations, clearly indicating degenerate motion cases for pure x translation as well as for yaw + y motion.

The standard deviation for the recovered points for each case over the 1,000 runs can be seen in Fig. 6.2f. The variation in z is greater than the variations in x and y for all the cases as expected. The ambiguity in elevation that is not resolved from the information gained in the degenerate cases causes the variation in z for those situations to be much greater than the other two trajectories. Variations in x and y-coordinates can be attributed to the optimization changing the odometry slightly to meet the measurement constraints.

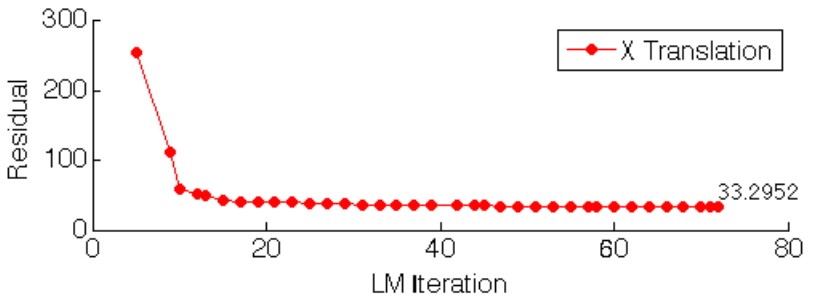
A visualization of the variation for each individual motion example provides further insights. The x, z-coordinate distributions for 100 runs of each of the 15 points in each simulation are shown in Fig. 6.2 with the ground truth marked for comparison. Only 100 runs are shown to avoid cluttering the graph. As seen by the thin bands, the elevation varies much more than the x-coordinate for each point. Note also that the bands are not vertical, but rather trace an arc, which is the elevation arc along which all the points would map to the same point in the sonar image. For the degenerate cases of pure x-translation and pure yaw and y-translation, the symmetric ambiguity about the zero plane is clearly seen. Points were equally likely to appear at the correct elevation or at the same elevation on the opposite side of the zero plane.

Another insight into how well-constrained a trajectory and set of landmarks are is the number of Levenberg-Marquardt (LM) iterations needed until convergence, and the resulting residual. Over 1,000 simulations, the general motion case converged in an average of 2 LM iterations, the pitch and z case converged in an average of 2 LM iterations, and the pure x translation case converged in an average of 74 LM iterations. The yaw and y example converged in an average of 95 LM iterations and the roll case converged in an average of 3 LM iterations. A representative example of the residuals after each LM iteration for the first three simulation trajectories can be seen in Fig. 6.3. The residuals for the first two simulation cases started off high, but quickly dropped after just one LM iteration to 46.5 for the general motion case and 60.2 for the pitch and z case. This indicates that the cost functions for these two trajectories are close to quadratic near the minimum. The high number of LM iterations needed for the pure x-translation trajectory, which eventually reaches a residual of 33.3, as well as the yaw and y case implies that the optimization function is not quadratic, but presumably close to flat in at least one direction. The flatness is due to the degenerate geometric configuration, which leads to much slower convergence.

The poses and overall error (geometric distance from the estimated point to the true point) for each simulation can be found in Tab. 6.2. Note that the point errors for fully constrained situations are less than 0.23 m and general motion has the smallest geometric error of only about 0.11 m. The point errors are much larger in the degenerate cases because the z-coordinates of the recovered points



(a) General motion and pitch + z



(b) x-translation

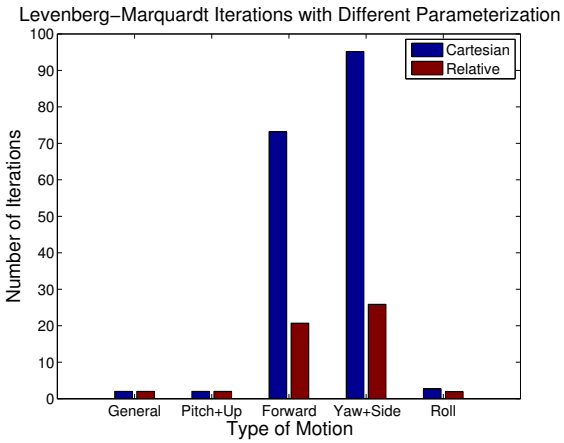
Figure 6.3: Sample residuals for simulation data after each Levenberg-Marquardt iteration for a representative run of (a) the general motion case and the pitch + z trajectory and (b) the pure x translation case. The final residual for each case is labeled. Iterations without a residual in (b) indicate a rejected LM step due to an increase in error.

were not able to be uniquely resolved. The odometry is recovered well, largely due to a good initial guess (perfect odometry with added Gaussian noise (σ listed in Tab. 6.1)). A promising result is that the general motion simulation performs very well, suggesting that ASFM could work well for inspection and surveying applications.

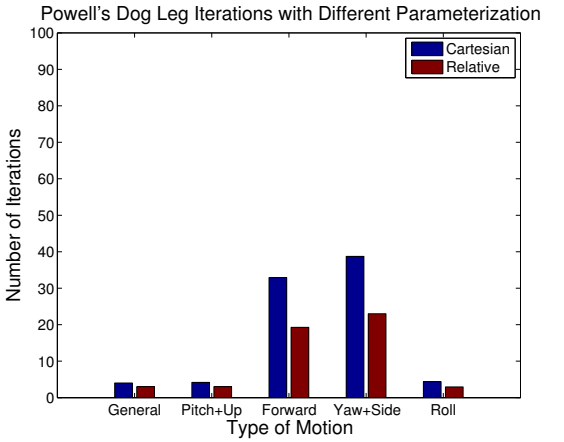
Table 6.2: Monte Carlo Simulation Results

	General Motion	Pitch + z	x	Yaw + y	Roll
Feature mean error (m)	0.109	0.155	0.943	1.05	0.227
Feature stddev (m)	0.0662	0.0888	0.834	0.812	0.159
Pose 1 (m, m, m, deg, deg, deg)	(0, 0, -1, 0, -22.5, 0)	(0, 0, -2, 0, -22.5, 0)	(0, 0, 0, 0, 0, 0)	(0, 0, 0, 0, 0, 0)	(0, 0, 0, 0, 0, 0)
Pose 2 (m, m, m, deg, deg, deg)	(-1, 0, 0, 0, 0, 15)	(0, 0, 0, 0, 0, 0)	(1, 0, 0, 0, 0, 0)	(0, 2, 0, -15, 0, 0)	(0, 0, 0, 0, 0, 22.5)
Pose 3 (m, m, m, deg, deg, deg)	(-0.5, 2, 2, -22.5, 22.5, 0)	(0, 0, 3, 0, 30, 0)	(2, 0, 0, 0, 0, 0)	(0, 4, 0, -22.5, 0, 0)	(0, 0, 0, 0, 0, 45)
Pose position mean error (m)	0.0144	0.0153	0.0133	0.0132	0.0103
Pose position stddev (m)	0.0062	0.0065	0.0062	0.0062	0.0052
Pose orient. mean error (deg)	0.808	0.774	1.21	1.37	1.33
Pose orient. stddev (deg)	0.470	0.487	0.556	0.630	0.584
Avg. number of LM iterations	2	2	74	95	3

6.2 Relative Parameterization



(a)



(b)

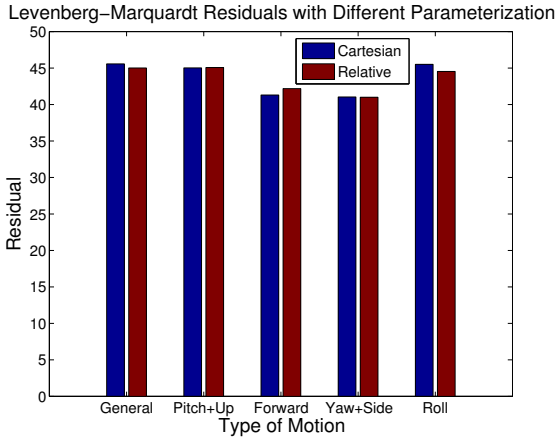
Figure 6.4: Average number of iterations until convergence for all five motion cases over 1,000 runs without (blue) and with (red) relative parameterization using (a) Levenberg-Marquardt and (b) Powell's Dog Leg to solve the nonlinear least squares.

A relative spherical parameterization was tested on the same simulation as described above with the five different vehicle motions in an attempt to improve the optimization convergence properties. The simulation was run 1,000 times as before with and without relative parameterization using both Levenberg-Marquardt and Powell's Dog Leg (PDL), two different nonlinear least-squares algorithms. As

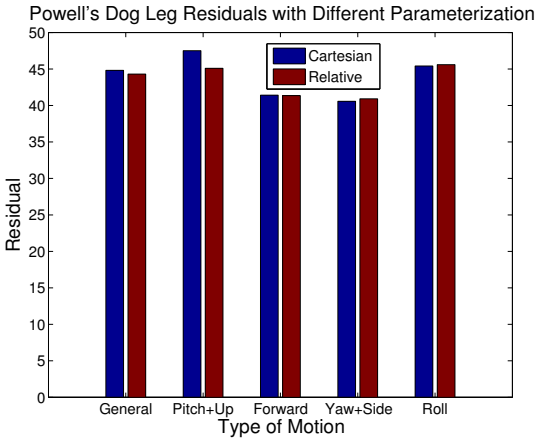
shown in Fig. 6.4, these two graphs show that in the well-defined cases, general, pitch+z, and roll motion, relative parameterization results in the same number or slightly fewer iterations. However, the number of iterations required was small regardless, so there was not much room for reduction. For the degenerate cases however, the two methods both show a drastic reduction in the number of iterations required for the relative parameterization. In general, the degenerate cases, forward and yaw+y motions, result in many more iterations due to the ambiguity of the elevation symmetric about zero elevation. Since we initialize the landmarks at zero elevation, the optimization starts on a hill in the cost function between two local minima. There the gradient is close to 0, so progress towards a minimum will initially be slow to reach one of the solutions. Note that which solution is reached will only depend on the measurement noise since one solution is not more likely than the other. As we expect, it seems that the relative parameterization reduces the nonlinearity of these degenerate cost functions and allows the optimization to converge with up to about 67% fewer iterations in the case of Levenberg-Marquardt.

Taking a look at the optimization residuals (Fig. 6.5), we predict that the residuals would not be dependent on the parameterization. The optimization should still find the same solution, but the hope is that the relative parameterization will allow the optimization to converge faster. Our results show that indeed, for both LM and PDL, the residuals for both parameterizations are not significantly different.

The simulations overall demonstrate a very promising potential benefit for using relative parameterization in ASFM. It's important to note that during our experiments, we found that the degenerate cases can lead to very poorly conditioned square root information matrices that will cause numerical issues during nonlinear optimization. The ill conditioning stems from the high uncertainty of the 3D position of the landmarks due to the elevation arcs having large regions of overlap. In addition, we found that in some instances of the degenerate cases, near the zero elevation hill in the cost function where the 3D points are initialized, LM steps will keep getting rejected until the update vector becomes so small that the optimization stops without moving very much toward either solution symmetric about the zero plane.



(a)



(b)

Figure 6.5: Average optimization residual for all five motion cases over 1,000 runs without (blue) and with (red) relative parameterization using (a) Levenberg-Marquardt and (b) Powell's Dog Leg to solve the nonlinear least squares.

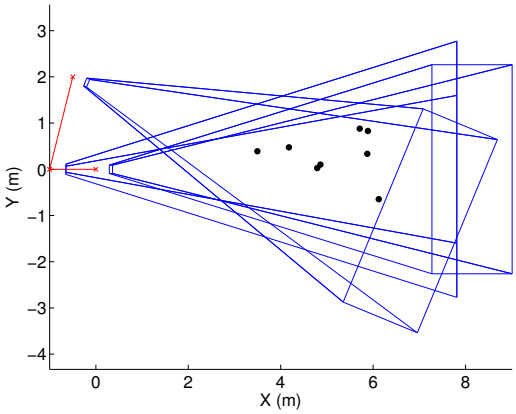
6.3 Data Association

6.3.1 Simulation Setup

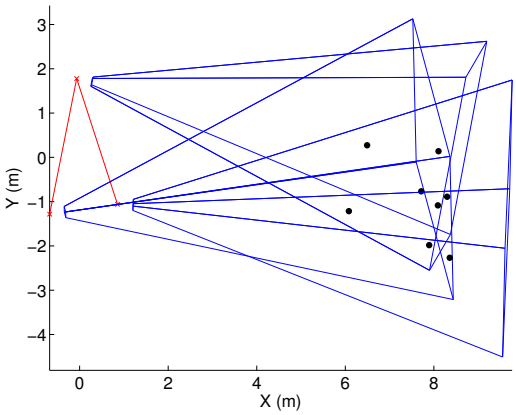
To test our data association algorithm, we ran simulations for general vehicle motion using simulated data. The simulation data was generated by randomly creating three sonar poses and 3D points until at least eight 3D points were visible in all three sonar frames. Gaussian noise was added to the bearing ($\sigma = 0.2°$)

and range ($\sigma = 0.005$ m) components of the ground truth sonar measurements. Similarly, Gaussian noise was added to both rotational ($\sigma = 1^\circ$) and translational components ($\sigma = 0.01$ m) of the odometry between consecutive poses. The simulated sonar and environment specifications are the same as those listed in Tab. 6.1 except we only used 100 Monte Carlo samples. Ten different sonar and point environments were randomly generated for the simulation experiments.

We use Monte Carlo sampling to analyze the accuracy and robustness of our automatic data association algorithm. Two experiments were performed: one including spurious feature measurements and one without. Each sonar trajectory was simulated 100 times with noise randomly sampled each time from the same Gaussian distribution ($\mu = 0$ and σ as described above). For each pose in each trial, five of the eight 3D points that could have been seen were randomly chosen to be measured. Consequently, not all of the poses saw all of the same points and the correct data association hypothesis often contained several null matches. For the spurious feature experiment, a random number of spurious features (0-2) were added for each pose. The spurious features were generated by randomly creating measurements in the sonar field of view.



(a) Environment 2



(b) Environment 4

Figure 6.6: Top views of examples of a random environment generated for the Monte Carlo simulations. The red line shows the trajectory of the sonar, the blue frustums demonstrate the frustum fields of view of the sonars, and the black dots are the eight randomly generated 3D points that can be seen by all three sonar poses. Environment 2 had consistently high accuracy and small landmark residuals while environment 4 had lower accuracy and higher landmark residuals. The AUV in environment 4 goes through larger changes in translation and orientation.

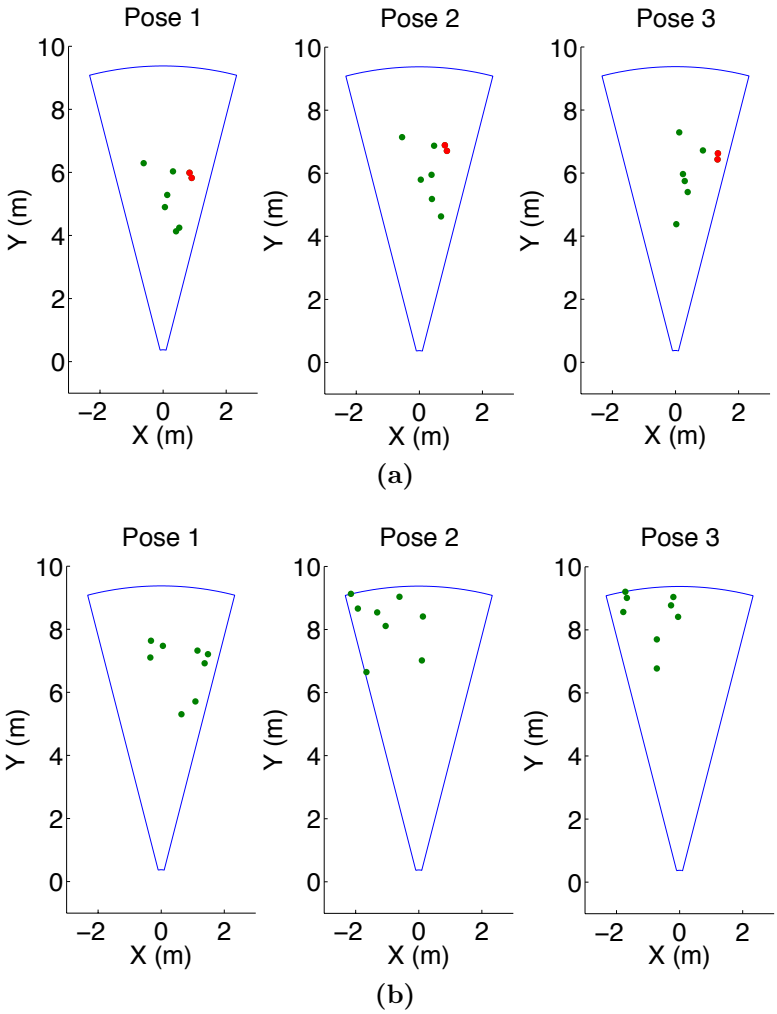


Figure 6.7: Feature measurements for environments (a) 2 and (b) 4. In environment 2, the two red points were deemed too close, and one of the points was discarded to avoid ambiguity.

6.3.2 Simulation Results

Examples of two environments are shown in Fig. 6.6 and Fig. 6.7. Error analysis on our simulation data showed that another gating threshold needs to be applied before data association begins to ensure that all feature measurements are sufficiently far apart from each other to avoid ambiguity. We discard one measurement if it is within 1° bearing or 0.2 m range of another feature measurement. Without this additional threshold, the incorrect data association confusing these features often has a similarly small, if not better, posterior than the correct hypothesis.

Additionally, we found that it is important to set up the factor graph in a batch manner when testing a hypothesis. If the graph is built incrementally, meaning when new measurements arrive, they are put in on top of an existing graph for optimization, the optimization can get trapped in the wrong local minimum if there are insufficient constraints. In this situation, even if the new measurements pull the optimization in the correct direction, leaving the wrong local minimum will result in a higher residual so the optimization will stay trapped. Therefore, we solve this problem by inserting all measurements and correspondences from the entire history of the trajectory into a new graph all at once.

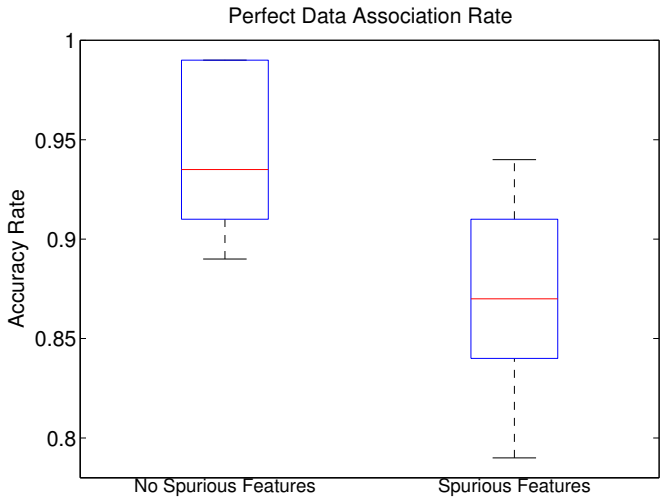


Figure 6.8: Rate at which our data association algorithm found the exact correct hypothesis with no mistakes. The mean accuracy rate without spurious features was 0.941 with a standard deviation of 0.04. The mean accuracy rate including spurious features was 0.867 with a standard deviation of 0.05.

The average accuracy of our data association algorithm over 100 runs each of 10 environments without the inclusion of spurious features was 94.1%, meaning that the algorithm found the exactly correct data association 94.1% of the time. With spurious measurements, the accuracy drops to 86.7% (Fig. 6.8). Many of the errors were caused by the algorithm finding a data association hypothesis that had fewer null matches than the correct one but still had a residual below the $\chi^2_{d,\alpha}$ threshold. Since adding spurious features increases the number of null matches in the correct hypothesis, more of these kinds of errors were made, reducing the accuracy rate. A possible solution to reducing these mistakes would be to tighten the $\chi^2_{d,\alpha}$ threshold by decreasing the confidence level α. However, there is a trade-off because decreasing α means a larger percentage of the distribution will be

thrown out as outliers, increasing the likelihood that a correct hypothesis will be rejected.

A box plot of the average landmark factor squared residuals with and without spurious features for each simulation environment is shown in Fig. 6.9. The landmark factor residual is essentially a reprojection error between the final 3D point recovered and the original feature measurement. The average landmark factor squared residual over all trials without spurious features was 0.50 with a standard deviation of 0.27. Including spurious features, the average reprojection error was 0.45 with a standard deviation of 0.29. As expected, the addition of spurious features does not affect the landmark residuals very much because the spurious features should rarely ever be added to the factor graph, as they do not correspond to real landmarks. The chance that two poses both have spurious measurements that could correspond to the same 3D point is small, and therefore the spurious feature will rarely be fully constrained and added to the graph.

One of the environments that consistently has lower accuracy rates and large landmark residual outliers is environment 4. As shown in Fig. 6.6, the sonar motion between poses is quite large. Compared to a more well-behaved environment such as environment 2, the largest motion between poses in environment 4 is about 0.5 m larger in the $x-$direction, 0.7 m larger in the $y-$direction, and has about 40° more change in roll. These larger changes in sonar motion between frames could contribute to more ambiguity amongst features even if they do not appear very close to each other because the geometry and correlation of the features can change significantly between more radically different points of view. In a real mission, it is usually safe to assume that the sonar motion between frames is small because for mapping purposes AUVs typically move very slowly. For instance, the Bluefin Hovering Autonomous Underwater Vehicle (HAUV) we use in our real data experiments usually travels at speeds of about 0.5 m/s.

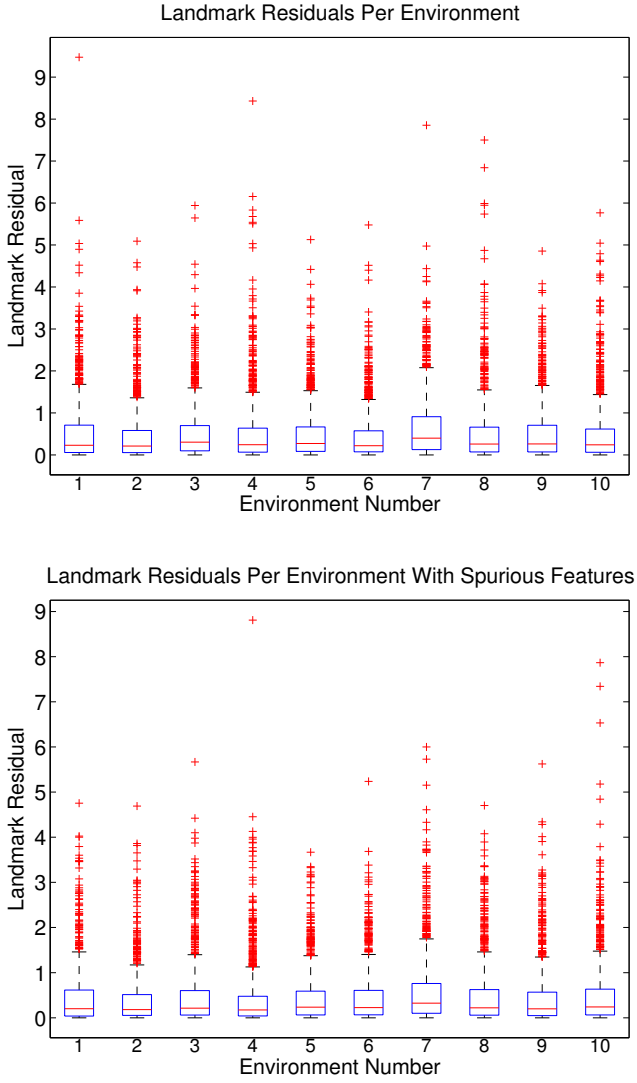


Figure 6.9: Box plots of individual landmark squared residuals (reprojection error in bearing-range space) per environment. The average landmark squared residual over all environments without spurious features was 0.50 with a standard deviation of 0.27. Including spurious features, the average landmark squared residual was 0.45 with a standard deviation of 0.29.

Runtime of our algorithm over the different randomly generated environments is shown in Fig. 6.10. Even though we use a brute-force exhaustive search, the order of the inputs into the search reduces the runtime significantly given that we favor hypotheses with fewer null matches. Without spurious features, the average runtime using a C++ implementation in Linux on a 2.5 GHz Intel i7 processor

was $15ms$ with a standard deviation of $22ms$. Including spurious features, the average runtime was $16ms$ with a standard deviation of $27ms$. Environment 7 stands out in terms of longer runtime because it needed to throw out two feature measurements to avoid ambiguities so the correct hypothesis always had more null matches than the other environments, which usually only needed to discard one measurement at the most.

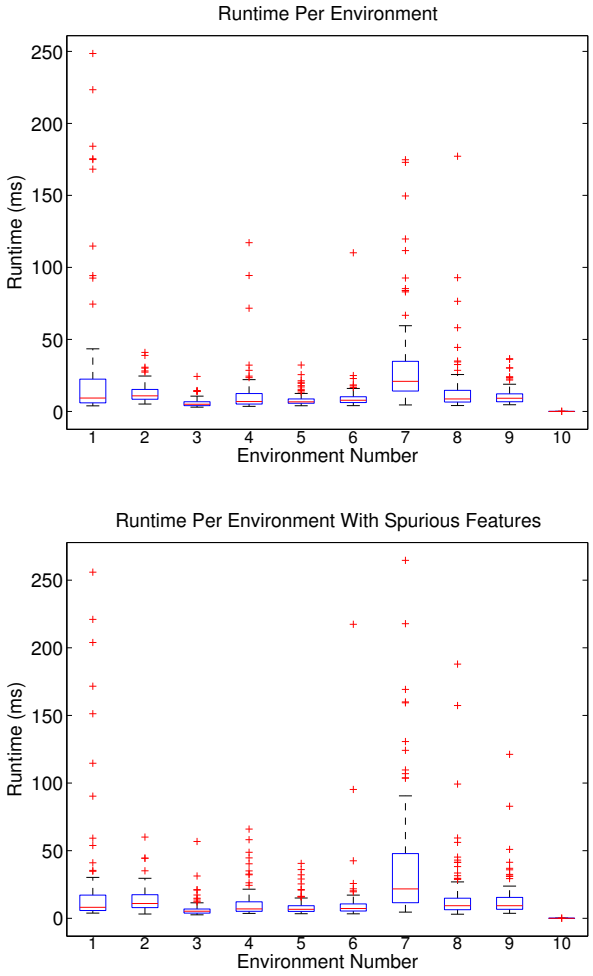


Figure 6.10: Box plots of runtime in milliseconds per environment for the data association algorithm using a C++ implementation on a 2.5GHz Intel i7 processor in Linux. The average runtime over all environments without spurious features was $15ms$ with a standard deviation of $22ms$. With spurious features, the average runtime was $16ms$ with a standard deviation of $27ms$.

6.4 Imaging Sonar Sequence

6.4.1 Imaging Sonar Experimental Setup

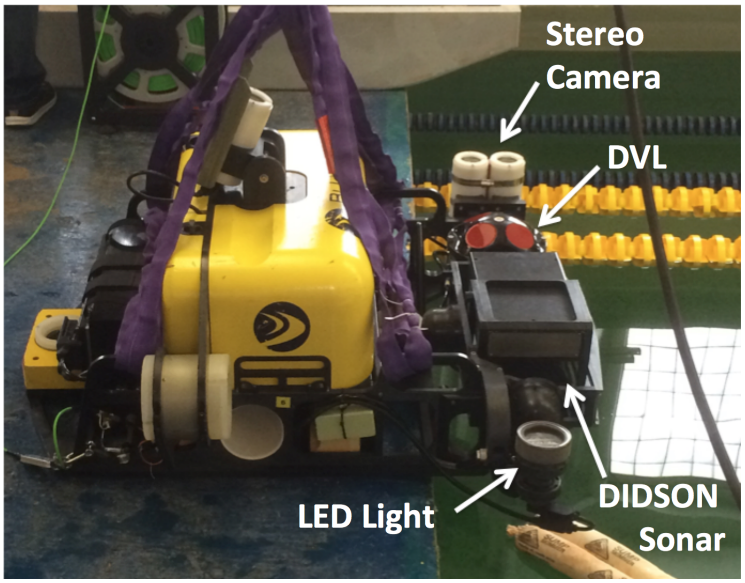


Figure 6.11: Bluefin Hovering Autonomous Underwater Vehicle (HAUV) used in our real data experiments. The DIDSON sonar and Doppler Velocity Log (DVL) are pictured attached to the front of the vehicle.

We demonstrate 3D structure recovery with automatic data association from several imaging sonar frames recorded with a Bluefin Hovering Autonomous Underwater Vehicle (HAUV) (Fig. 6.11) in Boston, Massachusetts. Five sonar frames were selected from the dataset to perform ASFM and point features were manually selected from all five sonar images. We also randomly generated a random number of spurious features (0-2) in each sonar image to test the algorithm's robustness on real data (Fig. 6.12). Although features were extracted manually, point correspondences were found automatically using our data association algorithm.

In our experiments, we use a Sound Metrics DIDSON 300m forward-looking sonar [21]. It has a $\psi_{max} = 28.8°$ bearing field of view (FOV) and a 28° vertical FOV (using a spreader lens). The DIDSON sonar discretizes returns into $N_b = 96$ bearing bins and $N_r = 512$ range bins. The DIDSON mode used for this dataset provides a minimum range of $r_{min} = 0.75$ m and a maximum range of $r_{max} = 5.25$ m.

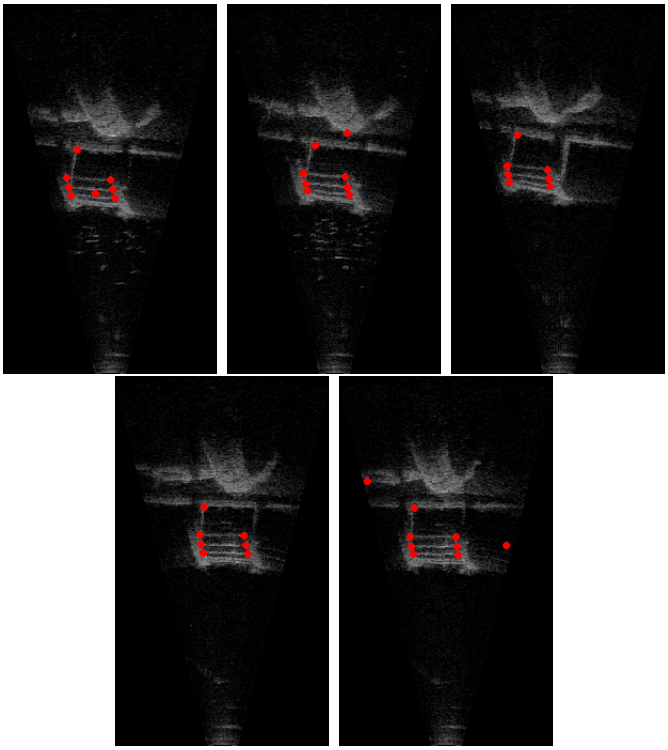


Figure 6.12: Manually marked features (red circles) for the five raw sonar frames that were used to reconstruct the ladder geometry with the addition of 0 - 2 randomly generated spurious features.

6.4.2 3D Reconstruction with Automatic Data Association

The feature measurements were incrementally introduced to the automatic data association algorithm and a hypothesis for the feature correspondences was found. Using this hypothesis, the measurements were placed into the factor graph optimization for 3D reconstruction. Odometry readings from the vehicle were also used in the optimization to further constrain the problem. The odometry was collected from a Doppler Velocity Log (DVL), which uses acoustic pings to measure the velocity of the vehicle. The orientation of the HAUV is measured using the DVL's on-board compass, pitch, and roll sensors. We chose odometry uncertainties of $\sigma = 1^\circ$ for rotation and $\sigma = 0.1$ m for translation. For bearing and range measurements from the DIDSON sonar we use $\sigma = 0.2^\circ$ and $\sigma = 0.005$ m respectively.

Fig. 6.13 shows how the optimization reduces errors in the location of the point features initially very quickly over a few iterations. Near the minimum, each iteration reduces errors more slowly. The final reprojection error is shown in the last frame. Since no ground truth is available for this dataset, we use reprojection

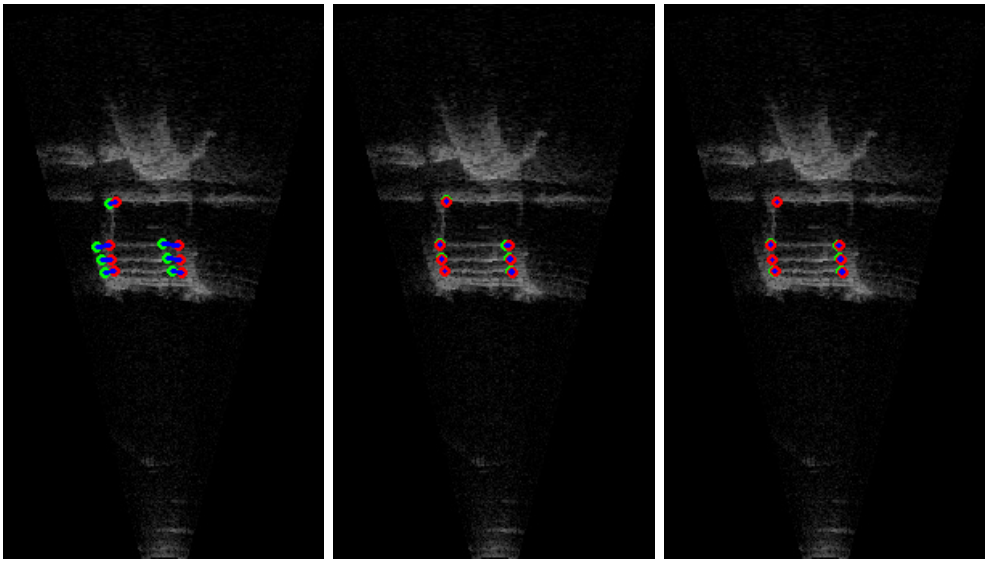
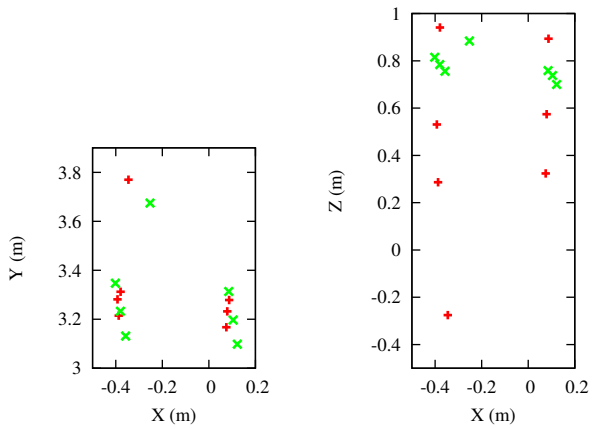


Figure 6.13: Reprojection error for the last sonar frame (left) from initialization, (center) after 5 LM iterations, and (right) after a solution was found (27 LM iterations). The red circles indicate the manually selected features and the green circles indicate the reprojected features. The blue lines show the reprojection error used in the ASFM optimization.

error on the Cartesian image as one indicator for ASFM's performance. As seen in Fig. 6.13, each recovered point is very close to the manually selected point. The optimization for this imaging sonar sequence took 27 LM iterations and had an ending residual of 52.8.

The 3D geometry of the ladder in the imaging sonar sequence was recovered as shown in Fig. 6.14. Before optimization, the ladder is initialized as a flat object lying in the $x-y$ plane. The structure in the $x-y$ plane looks convincing, but from the $x-z$ view, it is clear that the initialization does not capture the reality that the ladder's rungs are at different z elevations. The algorithm correctly ignored the spurious features and found the true hypothesis in $232ms$. Without spurious features, we were able to generate the correct data association in $230ms$. The extra computational time needed by introducing spurious features ended up being very small as the features should have very few potential matches. Thus, the spurious features should not increase the data association hypothesis search space by a significant amount.



(a) Top view **(b)** Front view

Figure 6.14: (a) Top and (b) front views of the 3D ladder structure before (green '×') and after (red '+') optimization from five imaging sonar frames.

Without ground truth, it is difficult to determine the geometric error between the recovered points and the true 3D points. Going off the assumption that the steps are spaced evenly on the ladder, we can estimate our maximum error to be about 0.2 m given that the top point on the left side of the ladder is spaced about 0.2 m farther than the spacing between the other points.

7 Conclusion

We have presented a novel algorithm for recovery of 3D point features from multiple sonar views, while also constraining the poses from which the images are taken. In contrast to previous solutions, we do not make any planar surface assumptions. Simulations of several types of sonar trajectories show the ability of ASFM to recover 3D structure with low uncertainty for general trajectories. They also show a limitation of ASFM in its failure to recover elevation of points for motions that provide poor constraints such as in the case of pure x-translation. An experiment with real sonar data and manually extracted feature points further demonstrates ASFM's 3D reconstruction capabilities.

Furthermore, we have presented a novel automatic data association algorithm for finding point correspondences between multiple 2D sonar images. Simulations of randomly generated sonar trajectories show the ability of our algorithm to find the correct data association hypothesis with a high success rate. The inclusion of spurious measurements in our simulation experiments further demonstrates the robustness of our data association algorithm. An experiment with real sonar data containing spurious features and manually extracted feature points shows the successful incorporation of the algorithm into the ASFM pipeline for 3D geometry recovery.

The nonlinear least-squares optimization used in ASFM has two main disadvantages: the solution returned may only be a locally optimal solution and our assumption that the posterior distribution is Gaussian may not hold. For the first issue, we have investigated a relative parameterization of the sonar measurements that showed very promising results for reducing the nonlinearity of the optimization function and increasing the rate of convergence. More experiments in simulation and on real data will need to be done to verify the benefits of this new parameterization. As for the second problem, it is clear that in some cases, such as the degenerate cases where the elevation remains ambiguous symmetric about the zero plane, the posterior is not Gaussian. In fact, for the degenerate cases, the distribution is bimodal. A possible solution would be to use multi-modal inference to capture an arbitrary distribution using a combination of many different Gaussians.

A possible improvement to our automatic data association algorithm is the use of the Incremental Posterior Joint Compatibility Test (IPJC) [18], which uses the same ideas as our current algorithm by searching a tree of hypotheses and computing a posterior compatibility cost. However, IPJC approximates the $\chi^2_{d,\alpha}$ error with an Extended Kalman Filter (EKF) update step instead of using a full optimization. If the correct data association most often includes very few null matches,

our brute-force search would already be very fast. Nevertheless, more generally if the sonar has many spurious features or not many overlapping features with recent sonar poses, IPJC could reduce the computational time of our algorithm significantly.

Of course, the automatic ASFM pipeline would not be complete and practical for real-time applications without an automatic feature extractor. What kind of features are most useful in imaging sonar images remains an open problem. Many computer vision features have been developed for cameras, but they might not be the best fit for the unique projective geometry of the sonar. Further research is needed to determine suitable features for sonar images.

Acknowledgments

I would like to thank my advisor, Prof. Michael Kaess, for his invaluable guidance and patience throughout my time at CMU. Thank you for answering all of my questions, setting aside so much time for your students, and for teaching us so much every week. I truly appreciate your hard work, dedication, and support. Thank you to my thesis committee members Prof. David Wettergreen and Sanjiban Choudhury for overseeing my research. I would also like to acknowledge my fellow group members, Eric Westman, Ming Hsiao, Garrett Hemann, and Puneet Puri. Thanks for asking the tough questions, giving valuable feedback, and for the jokes and great conversation. I feel very lucky to have been part of such a great group. Thanks also go to Dr. Jason Stack for his support on the project and Pedro Vaz Teixeira for recording the ladder sonar sequence in Boston. Finally, I would also like to thank my parents for tirelessly supporting my journey in research and robotics. I couldn't have done it without them.

Bibliography

[1] H. Assalih, "3D reconstruction and motion estimation using forward looking sonar," Ph.D. dissertation, Heriot-Watt University, 2013.

[2] 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*, May 2015, pp. 1949–1958.

[3] ——, "On feature matching and image registration for two-dimensional forward-scan sonar imaging," *J. of Field Robotics*, vol. 30, no. 4, pp. 602–623, Jul. 2013.

[4] M. Babaee and S. Negahdaripour, "3-D object modeling from occluding contours in opti-acoustic stereo images," in *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*, Sep. 2013, pp. 1–8.

[5] 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, Jun. 2011, pp. 1–6.

[6] E. Coiras, Y. Petillot, and D. Lane, "Mutliresolution 3-D reconstruction from side-scan sonar images," *IEEE Trans. on Image Processing*, vol. 16, no. 2, pp. 382–390, Feb. 2007.

[7] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard, "Relocating underwater features autonomously using sonar-based SLAM," *IEEE J. Ocean Engineering*, vol. 38, no. 3, pp. 500–513, 2013.

[8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[9] 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," *Intl. J. of Robotics Research*, vol. 31, no. 12, pp. 1445–1464, Oct. 2012.

[10] 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)*, Oct. 2015, pp. 758–765.

[11] N. Hurtos, X. Cufi, 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)*, Oct. 2012, pp. 5298–5305.

[12] 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.

[13] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[14] J. J. Leonard, B. A. Moran, I. J. Cox, and M. L. Miller, "Underwater sonar data fusion using an efficient multiple hypothesis algorithm," in *Proc. IEEE Int. Conf. Robotics and Automation*, May 1995, pp. 2995–3002.

[15] K. Macdonald, "Exploring the global mid-ocean ridge," *Oceanus*, vol. 41, no. 1, Mar. 1998.

[16] S. Negahdaripour, "On 3-D motion estimation from feature tracks in 2-D FS sonar video," *IEEE Trans. Robotics*, vol. 29, no. 4, pp. 1016–1030, Aug. 2013.

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

[18] 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)*, Oct. 2012, pp. 3467–3474.

[19] Y. Petillot, I. T. Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *Journal of Oceanic Engineering*, vol. 26, pp. 240–251, Apr. 2001.

[20] D. Ribas, P. Ridao, J. Neira, and J. Tardós, "SLAM using an imaging sonar for partially structured underwater environments," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 5040–5045.

[21] Sound Metrics Corporation, "SoundMetrics Didson 300 specifications," http://www.soundmetrics.com/products/DIDSON-Sonars/DIDSON-300-m/.

[22] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. LNCS,

W. Triggs, A. Zisserman, and R. Szeliski, Eds., vol. 1883. Springer Verlag, 2000, pp. 298–372.

Nomenclature

Abbreviations

ASFM	Acoustic structure from motion
AUV	Autonomous underwater vehicle
DIDSON	Sound Metrics DIDSON 300 m sonar
DOF	Degrees of freedom
DVL	Doppler velocity log
FLS	Forward looking sonar
FOV	Field of view
GPS	Global positioning system
HAUV	Hovering autonomous underwater vehicle
LIDAR	Light radar, or light detection and ranging
LM	Levenberg-marquardt
MAP	Maximum a posteriori
MHT	Multiple hypothesis tracking
PDL	Powell's dog leg
ROV	Remotely operated vehicle
SFM	Structure from motion
SLAM	Simultaneous localization and mapping

Mathematical Symbols

α	X^2 confidence level parameter
γ	Transformation constant
$\mathcal{N}(0, \Lambda_i)$	Gaussian noise for odometry between pose x_{i-1} and x_i with 0 mean and covariance Λ_i

$\mathcal{N}(0, \Xi_k)$	Gaussian noise for sonar measurement k with 0 mean and covariance Ξ_k
ψ	Bearing angle
ψ_{max}	Bearing field of view of the sonar
θ	Elevation angle
Θ^*	Maximum probability set of landmarks and poses
A	Measurement Jacobian
d	Number of degrees of freedom of the SLAM factor graph
$g(x_{i-1}, x_i)$	Odometry measurement between poses x_{i-1} and x_i
$h(x_i, l_j)$	Sonar measurement of landmark l_j from pose x_i
l_j	Landmark j
M	Total number of landmarks
$M_4(N_b, \psi)$	DIDSON lens distortion function
m_k	Sonar measurement k
N	Total number of poses
N_b	Total number of bearing bins
n_b	Bearing bin in polar sonar image
N_r	Total number of range bins
n_r	Range bin in polar sonar image
p	Prior factor
r	Range
r_{max}	Maximum range of the sonar
r_{min}	Minimum range of the sonar
u, v	Cartesian image coordinates
u_i	Odometry measurement i

w	Width of the Cartesian sonar image
x_g, y_g, z_g	Global coordinates
x_i	Pose i
x_s, y_s, z_s	Local sonar frame coordinates
deg	Degrees
m	Meters