Degeneracy-aware Imaging Sonar SLAM

Eric Westman and Michael Kaess

***Abstract*—High-frequency imaging sonar sensors have recently been applied to aid underwater vehicle localization, by providing frame-to-frame odometry measurements or loop closures over large time-scales. Previous methods have often assumed a planar environment, thereby restricting the use of such algorithms mostly to seafloor mapping. We propose an algorithm to generate pose-to-pose constraints for pairs of sonar images, which may also be applied to larger sets of images, that makes no assumptions about the environmental geometry. The algorithm is sensitive to the inherent degeneracies of the imaging sonar sensor model, and may be tuned to trade off between providing more constraints on the sensor motion and not over-fitting to noise in the measurements. For real-time localization, we fuse the resulting pair-wise sonar pose constraints with vehicle odometry in a pose graph optimization framework. We rigorously evaluate the proposed method and demonstrate improvement in accuracy over previously proposed formulations both in simulation and real-world experiments.**

I. Introduction

Acoustic sonar sensors have long been used for underwater sensing on ships and submarines, and more recently, autonomous underwater vehicles (AUVs). Side-scan sonar, synthetic aperture sonar, and multibeam echo-sounders are often placed on AUVs or surface vessels facing downward to image the seafloor on a large scale. These sonars have been utilized for robotic tasks such as navigation [8], mapping [6], and object tracking [25]. However, these sensors are not well-suited for small-scale or complex 3D environments. Autonomous exploration and inspection of scenes such as bridge and pier pilings, shipwrecks and archaeological sites, and other natural and man-made underwater structures requires a different kind of underwater sensor.

The emergence of high-frequency acoustic imaging sonars, or forward-looking sonars (FLS), in recent years (e.g. the SoundMetrics Aris[1] and DIDSON[2] and the Teledyne BlueView[3]) has afforded AUVs much greater sensing capabilities in such environments. The tasks that these sensors have enabled AUVs to perform include localization [15, 3], mapping structured environments [33, 31], image mosaicing [13], obstacle avoidance, and path planning [26]. These sensors have proven to be particularly useful in turbid waters where underwater optical cameras fail to see beyond very short ranges.

The focus of this work is using imaging sonar for high-accuracy *localization*, or pose estimation, in previously unmapped environments by means of a simultaneous localization and mapping (SLAM) framework. The ultimate goal of a localization algorithm is to estimate the 6 degree-of-freedom (DOF) transformation that describes the sensor position and rotation relative to some global coordinate frame. Many approaches to imaging sonar localization simplify the problem formulation to solving for the relative transformation between a pair of sonar images. These transformations may be estimated sequentially and composed in a dead-reckoning framework, or they may be used to provide loop-closures to reduce drift over long-term operation. We primarily consider this two-view approach in this work for the sake of simplicity and computational efficiency. However, our proposed methods easily generalize to systems consisting of more than two sonar views, which provide stricter constraints on the sensor poses at the expense of greater computational complexity.

The standard imaging sonar sensor model is analogous to a monocular camera – both provide 2D images of a 3D environment. Under the pinhole camera model, each pixel corresponds to a ray in 3D space that passes through the camera center and the 3D point location that is imaged by the pixel. The point's range from the camera center is lost due to perspective projection, but the azimuth and elevation angles are directly measured by the pixel coordinates. In contrast, for an image generated an imaging sonar, each pixel provides a direct measurement of the bearing (azimuth) angle and range, but the elevation angle is lost in projection, as depicted in Fig. 1. Disambiguating the elevation angle of features is a fundamental challenge for acoustic localization and mapping, just as disambiguating the range is for the optical case.

Despite this analog to monocular cameras, there exist several other factors that make imaging sonar SLAM generally a more difficult problem than optical SLAM. First, imaging sonar sensors have considerably lower resolution and signal to noise ratio than optical cameras. As active sensors comprised of multiple transducers, imaging sonars do not homogeneously insonify the environment, which often creates unwanted patterns or artifacts in the images that ought to be removed by a pre-processing step. Additionally, in contrast to optical cameras, there does not exist a one-to-one pixel to surface patch correspondence. Any surface patch that exists along a pixel's corresponding elevation arc may contribute to the intensity measured at that pixel. A consequence of this is that the same 3D scene may appear very different in sonar images taken from different viewpoints. These factors all significantly increase the difficulty of extracting point features from sonar images to use in a feature-based SLAM system. However, recent works have demonstrated the effectiveness of automatically detecting and corresponding features from multiple viewpoints by means of anisotropic diffusion [18, 30]. This makes it possible to use point features in our proposed localization and SLAM algorithm.

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. {westman, kaess}@cmu.edu

This work was partially supported by the Office of Naval Research under awards N00014-16-1-2103 and N00014-16-1-2365.

[1]http://www.soundmetrics.com/Products/ARIS-Sonars

[2]http://www.soundmetrics.com/Products/DIDSON-Sonars/

[3]http://www.teledynemarine.com/blueview

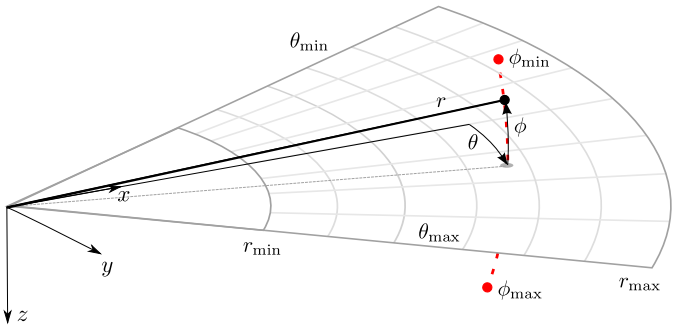


Figure 1: The imaging sonar sensor model. Each pixel provides direct measurements of the bearing angle $\theta \in [\theta_{\min}, \theta_{\max}]$ and range $r \in [r_{\min}, r_{\max}]$, but the elevation angle $\phi \in [\phi_{\min}, \phi_{\max}]$ is lost due to the projection onto the zero-elevation plane.

In this work we propose a novel imaging sonar SLAM algorithm, with the end goal of high-accuracy sensor localization. We build upon the previous methods of acoustic structure from motion (ASFM) [11, 12, 35], which propose bundle adjustment-inspired frameworks for jointly optimizing sensor poses and 3D landmark positions. While these approaches sufficiently address the mapping aspect of the SLAM framework, we demonstrate that there remain degeneracies in these previously proposed SLAM frameworks that result in increased errors in the sensor pose estimation. The advantages of our proposed SLAM framework are that it:

- makes no assumptions about the scene geometry
- is sensitive to the degeneracies of the imaging sonar sensor model in the SLAM framework
- generalizes to systems consisting of any number of desired poses
- may be easily incorporated in a pose graph-based SLAM solution for long-term, high-accuracy localization with loop closures

The remainder of this paper is organized as follows. Section II introduces relevant imaging sonar localization algorithms and relates these previous approaches to our proposed method. Section III introduces maximum a posteriori estimation and nonlinear least squares optimization which underlie our proposed methods. Section IV presents the two-view acoustic bundle adjustment problem, previous solutions to the problem, and our novel, fully degeneracy-aware solution. Section V describes how the result of this two-view bundle adjustment may be incorporated into a pose-graph framework for real-time, large-scale localization. Our simulated and experimental results are shown and discussed in Section VI, and Section VII closes with our concluding remarks.

II. Related work

A fundamental component of any feature-based SLAM system is feature detection. A variety of algorithms have been used for detecting and computing descriptors for features in optical images, such as SIFT [19], SURF [4], and ORB [27]. However, these methods tend to not translate very well to sonar images due to the high levels of speckle noise [30]. Many previous works have simply bypassed this problem by requiring manual extraction of feature points in order to perform SLAM or 3D reconstruction [5, 11, 20, 22]. Other feature detection methods have been developed for specific objects of interest, such as using Canny edge detection and the Hough transform to detect corners of a customized target in a test tank environment [14]. Yet another approach has been to find clusters of high intensity or high gradient pixels, which often correspond to rocks or other small distinct objects in a scene [15, 3]. Alternatively, man-made targets have been placed in the environment to provide reliable, easily detected blob-like features [9]. These methods mostly apply to seafloor mapping scenarios, where objects protrude from the smooth seafloor and induce shadows. A robust, general-purpose, feature point detection algorithm for imaging sonar has thus far not been proposed to the best of our knowledge. However, A-KAZE feature detection, which relies on anisotropic diffusion for denoising, has been used previously on sonar imagery to provide features for SLAM algorithms [18, 30]. Although feature detection for sonar imagery remains an open research topic, the A-KAZE algorithm suffices for the purposes of this work and is utilized in our field experiments.

In order to handle the elevation ambiguity inherent to the imaging sonar, many previous imaging sonar localization algorithms have made use of some form of planar environment assumption. One of the early works in this direction was [28], 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 our proposed method is built upon, it suffers from several sources of error. First, the planar approximation introduces error when the detected features do not lie on a planar surface. Second, the proposed framework optimizes only over the pose and surface normal parameters – it does not explicitly include the 3D landmark positions in the state of the optimization. [30] utilizes a similar bundle adjustment type of optimization, but also assumes a globally planar surface to estimate the 3D position of landmarks.

In [15], the planar assumption is also used to solve for a 3-DOF sensor motion between two sonar frames. As in [28], extracted feature points are assumed 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 as feature points, and a normal distribution transform (NDT) is used to iteratively align the two images. While this method provides great flexibility for the frontend as explicit feature correspondences are not required, its application is restricted to the case of mapping flat seafloors and does not provide a full 6-DOF pose constraint.

Further improvements in sonar pose estimation using the planar assumption were made [2, 3, 21]. 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 dis-

ambiguate 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. However, this method also utilizes a planar surface approximation, and is therefore restricted to cases where the imaged volume is a mostly flat surface, such as a seafloor.

The ASFM algorithm [11, 12] 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 3D landmark locations. This backend optimizer may be used with any frontend module that extracts and corresponds features across different sonar frames. The effectiveness of the ASFM algorithm in recovering the 3D positions of landmarks was demonstrated with both simulated data and real-world sonar images using manually-selected feature points, complemented with vehicle odometry measurements.

ASFM has recently been applied to localization during ship hull inspection, in which common features observed over a group of three frames (a "clique") are used to perform a local ASFM optimization [18]. 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 [17] 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.

The effect of the ASFM algorithm on sensor localization was investigated in [35]. This work demonstrated that point landmarks may often be under-constrained in the elevation angle depending on the sensor viewpoints, and that this degeneracy may lead to large errors in the sensor state estimate. A semi-parametric representation of 3D landmarks was proposed to handle this degeneracy – however, the degeneracy of the sensor motion was not addressed.

Similarly to the ASFM algorithm, [36] 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 3D 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.

In contrast to all of the described previous works, our proposed algorithm makes no assumptions about the scene geometry and takes steps to account for the degeneracy of both the full 6-DOF pose transformation as well as the landmark positions. Additionally, we propose a probabilistically sound framework for fusing pose constraints between pairs (or higher-order sets) of sonar images with measurements from other sensors in a pose graph framework.

III. Maximum a posteriori estimation

In this section we derive the nonlinear least squares (NLS) optimization that is used to solve the maximum a posteriori estimation framework of the SLAM / bundle adjustment problem. The optimization we derive underlies the methods used in our two-view bundle-adjustment presented in Section IV as well as the pose graph for large-scale localization, which is presented in Section V.

Maximum a posteriori (MAP) estimation attempts to find the most likely state $\mathbf{x}$ of the modeled system given a set of measurements $\mathbf{z} = \{\mathbf{z}_1, \dots \mathbf{z}_N\}$. We represent this type of optimization graphically using factor graphs, as in Fig. 2. Using this representation, the large, clear circles represent the state variables $\mathbf{x}$ to be optimized. Small, colored circles are factors, which represent the measurements $\mathbf{z}$ which constrain the variables to which they are connected.

Following [7], the MAP estimation problem may be formulated as

$$\mathbf{x}^* \quad = \quad \underset{\mathbf{x}}{\operatorname{argmax}} \; p\left(\mathbf{x}\right) \prod_{i=1}^{N} p\left(\mathbf{z}_i|\mathbf{x}\right) \tag{1}$$

where $p(\mathbf{z}_i|\mathbf{x})$ is the measurement model for measurement $\mathbf{z}_i$. Here we assume conditional independence of measurements, which is encoded in the connectivity of the factor graph. Note that although we use the notation $p\left(\mathbf{z}_i|\mathbf{x}\right)$, the measurement $\mathbf{z}_i$ is only conditioned on the subset of variables from the state $\mathbf{x}$ to which it is connected in the factor graph. If there is no prior knowledge of the state, which we will assume here, $p\left(\mathbf{x}\right)$ may be dropped. As is standard in the SLAM literature, we assume additive Gaussian noise in all measurement models:

$$p(\mathbf{z}_i|\mathbf{x}) \quad = \quad \mathcal{N}(\mathbf{h}_i(\mathbf{x}), \mathbf{\Sigma}_i) \tag{2}$$

Here $\mathbf{h}_i\left(\mathbf{x}\right)$ is the prediction function, which predicts a value $\hat{\mathbf{z}}_i$ of the measurement $\mathbf{z}_i$ based on the state estimate $\mathbf{x}$. The covariance matrix $\mathbf{\Sigma}_i$ represents the uncertainty of the measurement $\mathbf{z}_i$ and may be derived from sensor specifications or determined empirically. In principle, these are the three components that the corresponding factor defines: (1) the measurement itself (2) the prediction function and (3) the noise model (taking the form of a covariance matrix or, as we 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:

$$\mathbf{x}^* \quad = \quad \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \mathbf{h}_i\left(\mathbf{x}\right) - \mathbf{z}_i \right\|^2_{\mathbf{\Sigma}_i} \tag{3}$$

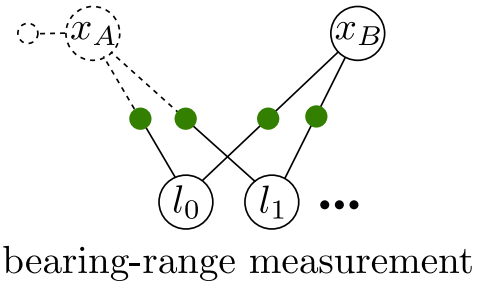


Figure 2: Factor graph representation of the two-view sonar optimization. In our two-view configuration, we optimize the relative 6-DOF transformation between the two views and the positions of all observed landmarks. x_A is dotted to signify that it is treated as a constant and is therefore not explicitly modeled in the bundle-adjustment optimization.

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

The Gauss-Newton algorithm (GN) is commonly used to solve the nonlinear least squares problem in Equation 3 by iteratively solving linear approximations of the nonlinear system. Given some initial state estimate $\mathbf{x}^0$, the prediction function is linearized as

$$\mathbf{h}_i(\mathbf{x}) = \mathbf{h}_i\left(\mathbf{x}^0 + \mathbf{\Delta}\right) \approx \mathbf{h}_i\left(\mathbf{x}^0\right) + \mathbf{H}_i\mathbf{\Delta} \tag{4}$$

$$\mathbf{H}_i = \left.\frac{\partial \mathbf{h}_i(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}^0} \tag{5}$$

where $\mathbf{\Delta} = \mathbf{x} - \mathbf{x}^0$ is the state update vector and $\mathbf{H}_i$ is the *Jacobian* matrix of the prediction function $\mathbf{h}_i(\mathbf{x})$. Substituting this linearized approximation into Equation 3 yields

$$\mathbf{\Delta}^* = \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\|\mathbf{h}_i\left(\mathbf{x}^0\right) + \mathbf{H}_i\mathbf{\Delta} - \mathbf{z}_i\right\|^2_{\mathbf{\Sigma}_i} \tag{6}$$

$$= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\|\mathbf{A}_i\mathbf{\Delta} - \mathbf{b}_i\right\|^2 \tag{7}$$

$$= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \left\|\mathbf{A}\mathbf{\Delta} - \mathbf{b}\right\|^2 \tag{8}$$

where $\mathbf{A}_i = \mathbf{\Sigma}_i^{-1/2}\mathbf{H}_i$ and $\mathbf{b}_i = \mathbf{\Sigma}_i^{-1/2}\left(\mathbf{z}_i - \mathbf{h}_i\left(\mathbf{x}^0\right)\right)$ 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.

Setting the derivative of Equation 8 to zero results in the so-called normal equations:

$$\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{\Delta}^* = \mathbf{A}^T\mathbf{b} \tag{9}$$

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

$$\mathbf{\Delta}^* = \mathbf{A}^\dagger\mathbf{b} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{10}$$

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 [7].

This nonlinear least squares optimization for MAP estimation underlies both of the frameworks we present in this paper: (1) the two-view acoustic bundle adjustment and (2) the pose-graph framework for large-scale localization. These two optimizations are distinct processes, and we will use the general notation introduced in this section to discuss each framework individually. For both frameworks, we will describe the specific factors utilized in the optimization and define their required components: the measurement, the prediction function (and its corresponding Jacobian matrix), and the noise model.

IV. Two-view acoustic bundle adjustment

A. Setup

The goal of our 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 [11, 12, 35], which include odometry measurements in the optimization. We 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. Fig. 2 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 3D point landmarks $\mathbf{l}_1, \ldots, \mathbf{l}_N$. A 6DOF pose x_A may be represented as a transformation matrix

$$\mathbf{T}_{x_A} = \begin{bmatrix} \mathbf{R}_{x_A} & \mathbf{t}_{x_A} \\ \mathbf{0} & 1 \end{bmatrix} \tag{11}$$

where $\mathbf{R}_{x_A}$ is the rotation matrix and $\mathbf{t}_{x_A}$is the translation vector. While x_A is not a vector, it may be updated by vector values using the exponential map, as discussed in greater detail in Section V-B as well as in [1].

A standard approach to solving a feature based SLAM problem is to place a prior measurement on the first pose, as represented in the dotted portion of the Fig. 2. However, since we are seeking to solve for a single relative 6-DOF transformation between the two poses, we eliminate x_A 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 our optimization: only pose x_B and the landmarks are explicitly represented as variables in the state. The sensor pose

x_A is treated as constant in the corresponding bearing-range measurements.

Therefore the state which we wish to optimize is $\mathbf{x} = \{x_B, \mathbf{l}_1, \ldots, \mathbf{l}_N\}$. We follow [12] 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 we define as the reference coordinate system. We 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 bearing-range measurement corresponding to $\mathbf{l}_i$ taken from pose P. We assume feature correspondences are provided by some frontend module.

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:

$$\boldsymbol{\Sigma_i} = \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}. \tag{12}$$

The Gaussian noise model for bearing and range measurements is not based on empirical characterization of the noise, but is a rough approximation. This is assumed primarily for theoretical convenience to fit the NLS optimization framework. We leave it to future work to investigate the effects of different noise models for bearing-range measurements in this framework.

The last component to define is the prediction function. Here we 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 $\mathbf{h}_i(\mathbf{x})$), we 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

$$\mathbf{h}_i^A(\mathbf{l}_i) = \begin{bmatrix} \theta_i \\ r_i \end{bmatrix} \tag{13}$$

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

$$\mathbf{h}_i^B(x_B, \mathbf{l}_i) = \boldsymbol{\pi}(\mathbf{q}_i) = \begin{bmatrix} \operatorname{atan2}(q_{i,y}, q_{i,x}) \\ \sqrt{q_{i,x}^2 + q_{i,y}^2 + q_{i,z}^2} \end{bmatrix} \tag{14}$$

$$\mathbf{q}_i = T_{x_B}(\mathbf{p}_i) = \mathbf{R}_{x_B}^T(\mathbf{p}_i - \boldsymbol{t}_{x_B}) \tag{15}$$

$$\mathbf{p}_i = C(\mathbf{l}_i) = \begin{bmatrix} r_i \cos\theta_i \cos\phi_i \\ r_i \sin\theta_i \cos\phi_i \\ r_i \sin\phi_i \end{bmatrix} \tag{16}$$

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. $\mathbf{R}_{x_B}$ and $\mathbf{t}_{x_B}$ are the rotation matrix and translation vector for pose x_B. Finally, we must be able to evaluate the Jacobian matrices $\mathbf{H}_i^A$ and $\mathbf{H}_i^B$. While these may be computed numerically, it is usually more computationally efficient to use analytically derived Jacobians. These closed-form Jacobians are presented in Appendix A for reference.

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 state estimate.

B. Landmark elevation degeneracy

The first type of degeneracy we consider in the two-view acoustic bundle adjustment optimization is that of a point landmark's elevation angle. Our previous work in [35] 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. We 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. We 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 $\mathbf{h}_i^B(x_B, \mathbf{m}_i)$. We address this performing a direct search over the valid range of ϕ_i to find the elevation angle with the lowest reprojection error:

$$\mathbf{h}_i^B(x_B, \mathbf{m}_i) = \boldsymbol{\pi}(T_{x_B}(c_{i,\phi^*})) \tag{17}$$

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

where $\Phi = \{\phi_{\min}, \phi_{\min} + \Delta\phi, \ldots, \phi_{\max} - \Delta\phi, \phi_{\max}\}$, $\Delta\phi$ is selected such that we select n_{elv} uniformly spaced angles 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 two-view scenario we consider.

C. 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.

We adopt the general approach of *solution remapping* in nonlinear optimization, as presented in [37]. This technique operates on the linear approximation of the nonlinear system at each step in the optimization. Therefore, we follow the same formulation of the two-view optimization presented in Section III, up until the linear approximation in Equation 8. We make use of the singular-value decomposition (SVD) of the whitened $m \times n$ 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 $\begin{bmatrix} 1/\sigma_1 & \dots & 1/\sigma_n \end{bmatrix}$. Using this decomposition, the linearized least squares problem in Equation 8 may be solved as $\boldsymbol{\Delta}^* = \mathbf{V}\mathbf{S}^\dagger\mathbf{U}^T\mathbf{b}$, which yields the same update vector $\boldsymbol{\Delta}^*$ 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 $\sigma_{\min}$ below which a singular value and its corresponding update direction will not be added to $\boldsymbol{\Delta}^*$. In this formulation, we solve the linear least squares problem using a modified pseudoinverse

$$\boldsymbol{\Delta}^* = \mathbf{A}_D^\dagger\mathbf{b} = \mathbf{V}\mathbf{S}_D^\dagger\mathbf{U}^T\mathbf{b} \tag{18}$$

where $\mathbf{S}_D^\dagger$ is an $n \times m$ diagonal matrix with diagonal $\begin{bmatrix} 0 & \dots & 1/\sigma_s & \dots & 1/\sigma_n \end{bmatrix}$ and σ_s is the smallest singular value greater than the threshold $\sigma_{\min}$. This procedure generates an update vector $\boldsymbol{\Delta}^*$ only using the well-constrained directions of the state [37]. 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. We x_B^* to denote the final optimized pose.

V. 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 measurements in the overall factor graph, the landmarks are marginalized out locally in our 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 [11, 35]. A visual representation of such a pose-graph with a large-scale loop closure is shown in Fig. 3.

Similar 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 III. The state consists only of vehicle and sonar poses $\mathbf{x} = \{x_0, \dots, x_n, s_0, \dots, s_n\}$ and no landmarks are explicitly modeled. We utilize three different 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 our data sets.

A. Odometry constraints

Many different types of odometry measurement constraints may be utilized in conjunction with our pose-to-pose sonar constraints. Here, we will describe the specific odometry measurements that we use with our robotic platform, which is described in detail in Section VI-B. We follow [33, 34] in using two types of measurements to model the onboard, odometry-based state estimate: (1) a relative 3-DOF pose-to-pose constraint on x and y translation and yaw rotation (heading), abbreviated as XYH and (2) a unary 3-DOF constraint on z translation and pitch and roll rotations, abbreviated as ZPR. The use of an IMU allows for globally observable, drift-free (but noisy) observations in the ZPR directions, but the XYH directions will drift over large time-scale operation. See [34] for additional information on these measurements.

B. Sonar constraint - measurement

We model the two-view sonar constraint between poses x_i and x_j as

$$p(\mathbf{z}_{ij}|x_i, x_j) = \mathcal{N}(\mathbf{h}_{ij}(x_i, x_j), \boldsymbol{\Xi}_{ij}) \tag{19}$$

$$\mathbf{h}_{ij}(x_i, x_j) = x_i^{-1}x_j \tag{20}$$

where the measurement $\mathbf{z}_{ij}$ is an element of the $SE\,(3)$ Lie group that represents the pose-to-pose sonar constraint from x_i to x_j and $\boldsymbol{\Xi}_{ij}$ is the corresponding covariance matrix. $\mathbf{h}_{ij}(x_i, x_j)$ generates the measurement prediction by computing the relative 6-DOF pose transformation from x_i to x_j. Following a linearization procedure similar to that taken in Equation 4, we express the squared error term corresponding to this factor as

$$\left\|\mathbf{H}_{ij}\boldsymbol{\Delta} - \log\left(\mathbf{z}_{ij}^{-1}\mathbf{h}_{ij}\left(\mathbf{x}^0\right)\right)\right\|_{\boldsymbol{\Xi}_{ij}}^2 \tag{21}$$

or more explicitly as

$$\left\|\mathbf{F}_{ij}\boldsymbol{\xi}_i + \mathbf{G}_{ij}\boldsymbol{\xi}_j - \log\left(\mathbf{z}_{ij}^{-1}\mathbf{h}_{ij}\left(\mathbf{x}^0\right)\right)\right\|_{\boldsymbol{\Xi}_{ij}}^2 \tag{22}$$

where $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ are minimal 6-DOF local vectors that represent updates to x_i and x_j, respectively, which are computed using the exponential map, as defined in [1]. $\mathbf{F}_{ij}$ and $\mathbf{G}_{ij}$ are the Jacobian matrices of the exponential map with respect to $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$, respectively. $\log(\cdot)$ denotes the logarithm map, the

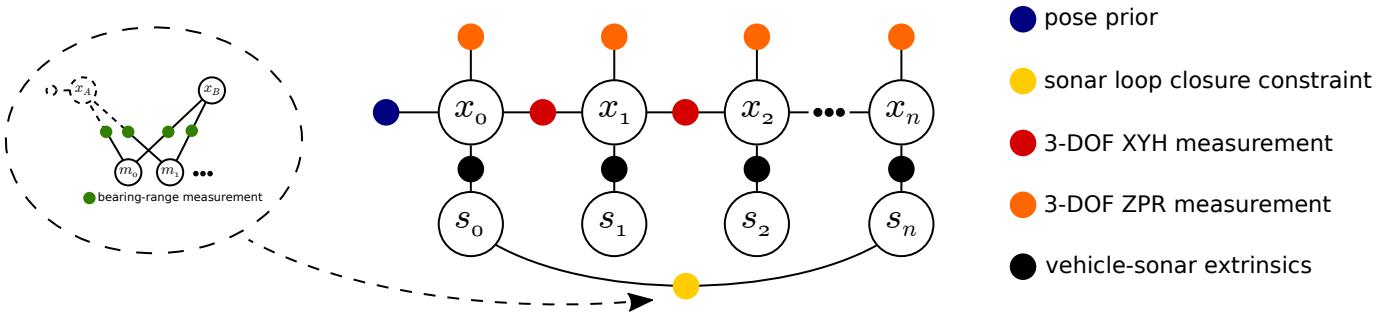


Figure 3: The pose graph framework we propose for long-term navigation. A two-view bundle adjustment problem is optimized, and the resulting pose-to-pose loop closure constraint is added to the pose graph along with the XYH and ZPR odometry measurements obtained from the onboard navigation.

inverse of the exponential map, which computes a minimal 6-DOF vector in tangent space. Using this linear approximation, the remainder of the NLS derivation in Section III may be followed, proceeding with the "whitening" step and ultimately solving the optimization with GN or a similar algorithm. See [7] for a comprehensive treatment of this formulation of the 6-DOF poses.

Note that while we define the measurement model for the two-view sonar constraint using a 6-DOF pose constraint $\mathbf{z}_{ij}$ and a covariance matrix $\mathbf{\Xi}_{ij}$, the NLS optimization underlying the pose graph actually requires the square root information matrix $\mathbf{\Xi}_{ij}^{-1/2}$. While the mean of the sonar measurement is simply the optimized relative pose from the two-view bundle adjustment problem, recovering the square-root information matrix is significantly more involved, and is detailed in the following subsection.

C. 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 IV-C:

$$\begin{aligned}\mathbf{\Gamma} &= \mathbf{A}_D^T\mathbf{A}_D && (23)\\ &= \mathbf{V}\mathbf{S}_D^T\mathbf{U}^T\mathbf{U}\mathbf{S}_D\mathbf{V}^T && (24)\\ &= \mathbf{V}\mathbf{S}_D^T\mathbf{S}_D\mathbf{V}^T && (25)\\ &= \begin{bmatrix}\mathbf{\Gamma}_{11} & \mathbf{\Gamma}_{12}\\ \mathbf{\Gamma}_{21} & \mathbf{\Gamma}_{22}\end{bmatrix}. && (26)\end{aligned}$$

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 marginalize out the landmark variables. This is done using the Schur complement:

$$\mathbf{\Lambda} = \mathbf{\Gamma}_{11} - \mathbf{\Gamma}_{12}\mathbf{\Gamma}_{22}^{-1}\mathbf{\Gamma}_{21}. \quad (27)$$

The block $\mathbf{\Gamma}_{22}$ is always invertible due to our 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:

$$\begin{aligned}\mathbf{\Lambda} &= \mathbf{P}\mathbf{L}\mathbf{D}\mathbf{L}^T\mathbf{P}^T && (28)\\ &= \left(\mathbf{P}\mathbf{L}\mathbf{D}^{1/2}\right)\left(\mathbf{D}^{T/2}\mathbf{L}^T\mathbf{P}^T\right) && (29)\\ &= \mathbf{R}^T\mathbf{R} && (30)\end{aligned}$$

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 compatible with the nonlinear least squares optimization and may be used to "whiten" the Jacobian matrices and error vectors, as in Equation 7.

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 V-B, using $\mathbf{z}_{ij} = x_B^*$ and $\mathbf{\Xi}_{ij}^{-1/2} = \mathbf{R} = \mathbf{D}^{T/2}\mathbf{L}^T\mathbf{P}^T$. The pose graph may be solved efficiently using the state-of-the-art iSAM2 algorithm [16] 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 Equation 8, 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 the directions that are not constrained by the two-view sonar measurements. Our proposed framework always meets this criterion, as the combination of the XYH and ZPR odometry measurements fully constrain each pose.

D. Frontend: feature detection and association

All of the work discussed thus far has dealt with the *backend* of our 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, we propose a novel implementation for associating point features between two sonar frames.

Joint compatibility branch and bound (JCBB) [23] has often been considered the gold standard algorithm for probabilistic association of landmarks in a SLAM context. Several more recent works have made improvements to the original JCBB algorithm for the purpose of feature cloud matching [24, 29]. 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 VI-B, we use the joint compatibility framework described in [29] for efficient data association between the two sonar frames in our two-view bundle adjustment problem. In Section IV, our 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 $\mathbf{z}_{j_i}^B$:

$$f_{ij_i}\left(x_B, \mathbf{z}_i^A, \mathbf{z}_{j_i}^B\right) = \mathbf{h}_{ij_i}\left(x_B, \mathbf{z}_i^A\right) - \mathbf{z}_{j_i}^B. \quad (31)$$

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:

$$\mathbf{h}_{ij_i}\left(x_B, \mathbf{z}_i^A\right) = \boldsymbol{\pi}\left(T_{x_B}\left(c_{i,\phi^*}\right)\right) \quad (32)$$

$$\phi_i^* = \underset{\phi\in\Phi}{\operatorname{argmin}} \left\|\boldsymbol{\pi}\left(T_{x_B}\left(c_{i,\phi}\right)\right) - \mathbf{z}_{j_i}^B\right\|_{\boldsymbol{\Sigma}_{j_i}}^2 \quad (33)$$

where $c_{i,\phi}$ denotes the Cartesian coordinates corresponding to the spherical coordinates $\begin{bmatrix} z_{\theta,i}^A & z_{r,i}^A & \phi_i \end{bmatrix}^T$. We implement the joint compatibility framework described in [29] 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 our experiments (up to a few dozen features per frame), the algorithm is very quick and finds a robust correspondence between the feature clouds in real-time. This is approximately the maximum number of reliable features that are expected in a sonar image, considering the low resolution and poor signal-to-noise ratio.

VI. Results and discussion

The two-view sonar bundle adjustment is implemented in C++ using the Eigen library [10] for efficient matrix operations and decompositions. We implement the pose graph framework using the GTSAM library,[4] which includes an implementation of iSAM2 that we utilize for optimization. To evaluate our proposed algorithms we conduct simulated experiments as well as real-world experiments in both a test tank environment and in the field. All computation is performed on a consumer laptop computer with a 4-core 2.50GHz Intel Core i7-6500U CPU and 8 GB RAM.

A. Simulation: two-view

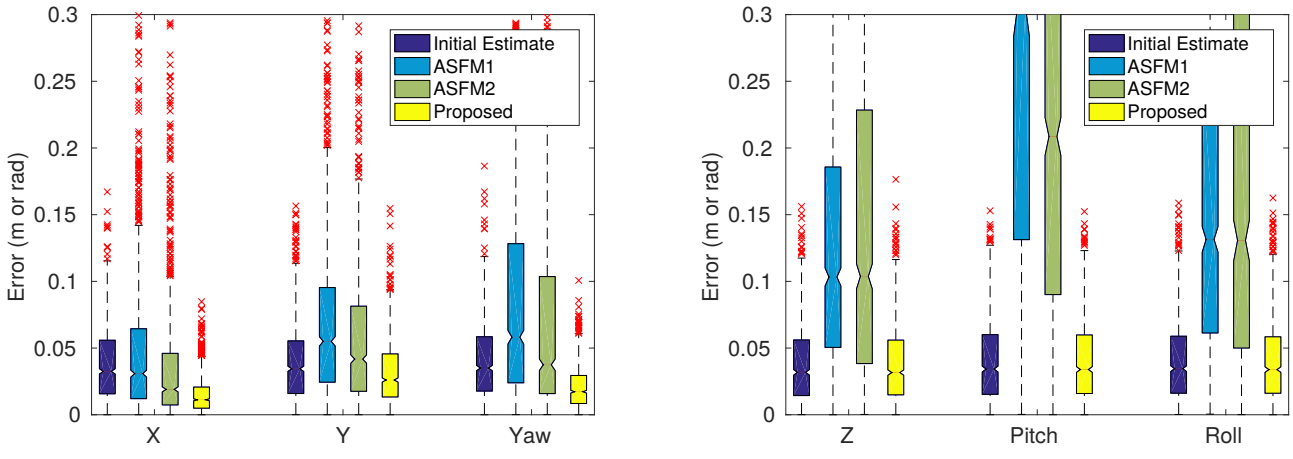
We conduct various Monte Carlo simulations of our two-view sonar bundle adjustment algorithm. In all of our simulations we assume ground-truth feature correspondences are known between the two sonar frames, which allows us to isolate the bundle adjustment algorithm in the evaluation. Gaussian noise is added to the bearing-range feature measurements ($\sigma_\theta = 0.01$ rad, $\sigma_r = 0.01$ m). We simulated the characteristics of the DIDSON imaging sonar used in our 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 3D point landmarks are generated, with a minimum of 6 points and an average of 12 viewed per two-view optimization. We use a constant threshold of $\sigma_{min} = 50$ throughout all simulations.

The first quantity that we sample in our simulations is the ground-truth relative pose transformation. We sample random small transformations with Euler angles drawn from $\mathcal{U}(-0.3 \text{ rad}, 0.3 \text{ rad})$ and translation components drawn from $\mathcal{U}(-0.3 \text{ m}, 0.3 \text{ m})$. 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 our proposed degeneracy-aware algorithm over two previous approaches. The evaluated approaches are:

- **ASFM1** - The formulation presented in Section IV-A and [11, 12], which solves the optimization via LM.
- **ASFM2** - The formulation presented in Section IV-B and [35] which models the elevation angle non-parametrically and solves the optimization via LM.
- **Proposed** - Our formulation presented in Section IV-C, 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: $\mathcal{N}(0, 0.05 \text{ rad})$ in the three Euler angles and $\mathcal{N}(0, 0.05 \text{ m})$ in the three translation directions. The box and whisker plots in Fig. 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 Fig. 4a and the poorly constrained DOF in Fig. 4b. In the well-constrained XYH directions, the proposed

[4] https://bitbucket.org/gtborg/gtsam/



(a) (b)

Figure 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.

method significantly decreases the error compared to the initial estimate and the previous methods ASFM1 and ASFM2. In the poorly constrained ZPR directions, our proposed method makes hardly 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, our method cautiously provides updates to the state estimate in only the directions that are well-constrained by the underlying geometry.

We repeat the previous simulations, but vary the noise levels of the initial pose estimate in rotation and translation over a wide range. Fig. 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 our proposed method outperforms the previously proposed algorithms in terms of the mean and variance of the pose estimation error.

B. Experimental: test tank

We validate our 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 we use is the hovering autonomous underwater vehicle (HAUV)[5], as shown in Fig. 6. 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 our case, the test tank floor. The vehicle also has a SoundMetrics DIDSON 300m sonar[6] mounted on the front of the vehicle, with 90° range of tilt motion controlled by an actuator. Lastly, there is an optical stereo camera pair mounted beside the DVL.

For ground-truth sonar localization, we use the fiducial-based visual SLAM algorithm presented in [34]. 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 data sets we use in these experiments. The extrinsics are then modeled as constant when collecting ground-truth data for our experiments. For these experiments, we 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 toward the right and z axis down. We 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).

[5]https://gdmissionsystems.com/products/underwater-vehicles/bluefin-hauv

[6]http://www.soundmetrics.com/Products/DIDSON-Sonars/DIDSON-300m

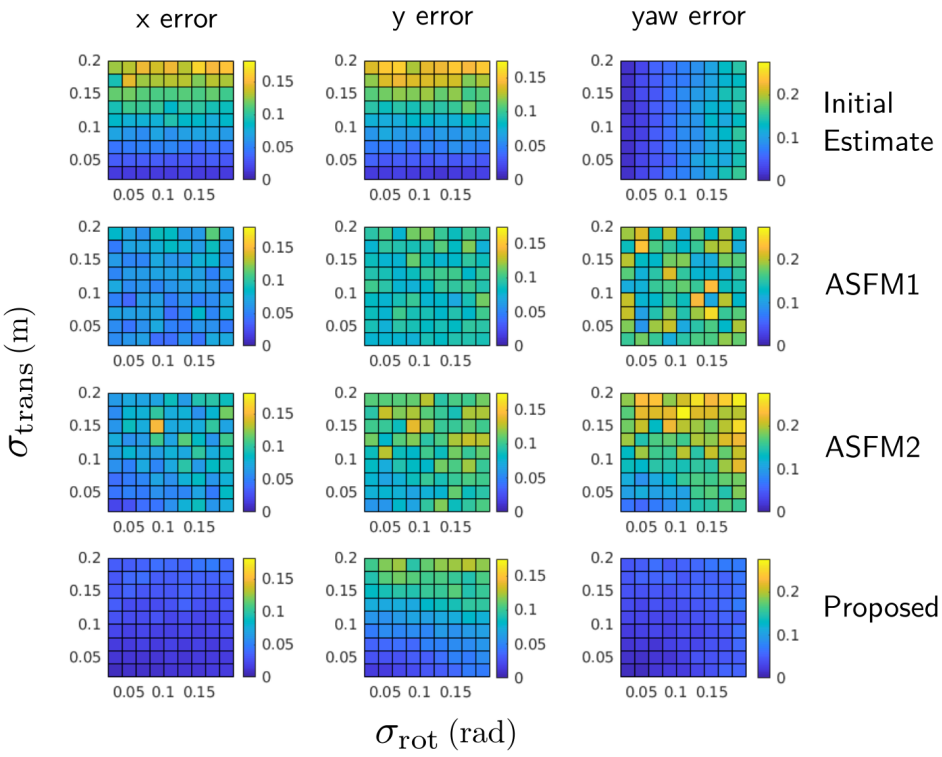


Figure 5: Average error over 100 Monte Carlo simulations for various levels of noise in the initial pose estimate. All plots in a row show the errors for a single pose estimation method and each column shows the error in one particular degree of freedom. The x-axis for each plot shows the value of the standard deviation σ_{rot} corresponding to the distribution of the noise $\mathcal{N}(0, \sigma_{\text{rot}})$ that is added to all rotation degrees of freedom, and likewise for the y-axis and σ_{trans}.

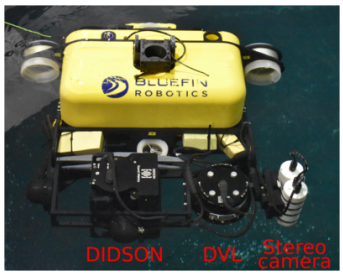


Figure 6: The Bluefin HAUV which is used in our real-world experiments. The DVL and DIDSON imaging sonar are mounted in front of the vehicle. The DVL points downward to measure velocity relative to the tank floor or seafloor, while the DIDSON may be tilted through a 90° field of view.

In these experiments we evaluate four different localization methods. In all of these methods, we 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: $\mathcal{N}(0, 0.02 \text{ m/s})$ in the XY directions and $\mathcal{N}(0, 0.02 \text{ rad/s})$ in the yaw direction. The four localization methods are as follows:

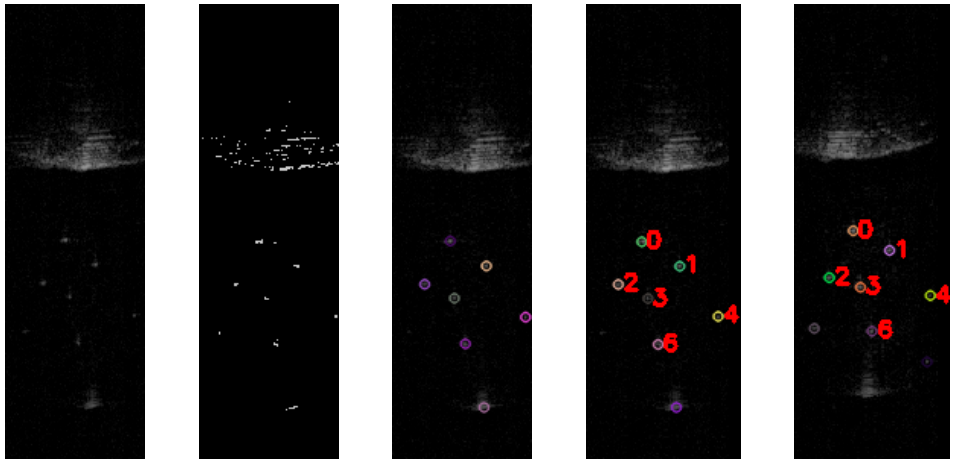
- **Dead reckoning** - Using the noisy odometry measurements.
- **Li modified** - We consider a modified version of the method proposed by Li et al. [18]. In the original work, Li et al. perform an optimization using the original ASFM formulation [11] using cliques of 3 imaging sonar frames. This framework also includes the odometry measurements in the ASFM optimization, thereby double counting the odometry information. We 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** - Our novel, fully degeneracy aware method detailed in Section V.

As our test tank environment consists of very smooth surfaces and lacks distinguishable features, we added features artificially. We placed an aluminum frame near the surface of the water. Magnets were placed protruding from 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 Fig. 7.

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

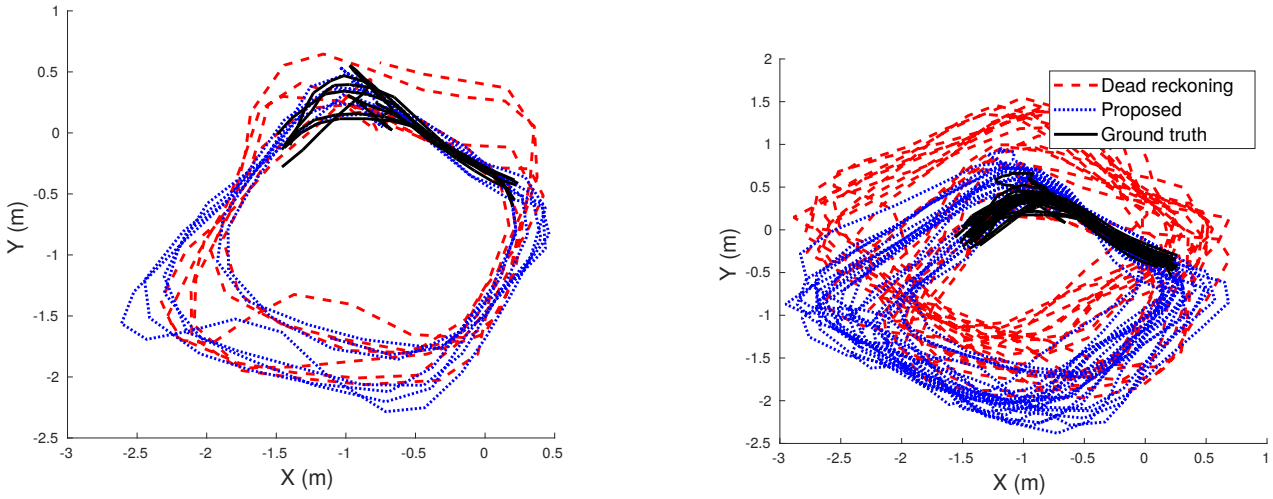
	Dead reckoning	Li modified	ASFM2	Proposed
Short trajectory	0.230	0.290	0.558	**0.074**
Long trajectory	0.519	0.252	0.769	**0.159**

Table I: The localization error (ATE) in units of meters for the three tested data sets. The short and long data sets used 37 and 66 loop closures, respectively, for all evaluated algorithms since the frontend feature detection and association is distinct from the backend optimization. Our proposed method produces a significantly more accurate localization result than the other evaluated methods for both data sets. The overall lengths of the full trajectories are approximately 60 and 180 meters.



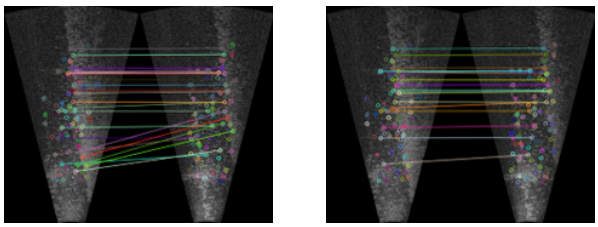
(a) (b) (c) (d) (e)

Figure 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 V-D.



(a) (b)

Figure 8: Top-down view of the trajectories of the dead-reckoning and proposed pose graph SLAM solution as compared to the ground truth for Short data set (a) and Long data set (b). The pose-graph significantly reduces drift throughout the sequence despite only making loop closures at one corner of the rectangular trajectory.



(a) (b)

Figure 9: (a) Feature matches between two frames in a loop closure proposal clique, as determined by Li et al [18]. (b) Our proposed joint compatibility-based feature association method rejects several incorrect feature matches, resulting in a significantly more reliable set of feature correspondences.

We recorded two data sets, 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 two seconds, and loop closures are added between sonar poses when a positive association is made between sonar frames with at least five 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 one second is required in order to add a loop closure constraint to the pose graph. Fig. 8 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 we 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.

Table I shows the absolute trajectory error (ATE) [32] 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. ASFM2 degrades localization accuracy compared to dead-reckoning due to pose degeneracy in the two-view optimizations. Finally, our proposed method decreases the localization error compared to all other methods as we 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.

C. Experimental: ship hull

To demonstrate its applicability to real-world applications, we test our pose graph localization algorithm on the ship hull inspection data sets presented by Li et al. in [18]. We compare our pose graph optimization method to both dead reckoning localization as well as Li et al.'s proposed approach. In [18], 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, we add relative noise between poses in the X, Y, and yaw directions and noise to global observations in the Z, pitch, and roll directions [34].

We utilize several components of the frontend presented in [18] to allow for a direct comparison of our proposed work's contributions. First, we only consider sonar images that are deemed sufficiently salient for potential loop closures. We 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, we simply uniformly sample poses that are close to the current pose for potential loop closures. We use the feature matches resulting from Li's method, which utilizes descriptor and geometric information, as input to our 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 Fig. 9. 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 our degeneracy-aware solution makes this clique formulation unnecessary, we still consider cliques of three sonar images for loop closures, but we treat each clique as two pairs (1-2 and 1-3). If at least seven features are matched between both pairs of images, we perform two separate acoustic bundle adjustment optimizations and add both resulting constraints into the overall pose graph.

Table II shows the localization error metrics used to evaluate (1) dead reckoning localization (2) the method of Li et al. and (3) our proposed method on the six ship hull data sets. We 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. Our 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, our 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 our joint compatibility feature association framework, which rejects a large number of potential loop closures due to poor feature matching.

Fig. 10 visualizes top-down views of several trajectories resulting from dead reckoning and our 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 our acoustic bundle adjustment algorithm, these results demonstrate the advantages of our formulation of acoustic bundle adjustment over previous attempts.

VII. Conclusion

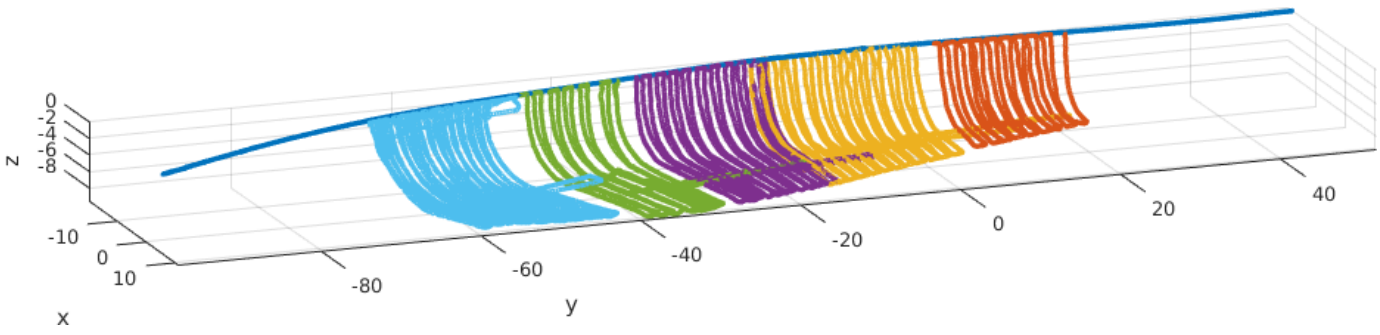
In this paper we have presented a novel solution to the feature-based imaging sonar bundle adjustment problem that emphasizes accurate pose estimation. We focus on analyzing the case of pairwise bundle adjustment, but our framework is easily applicable to systems with three or more sonar viewpoints. We also propose a pose graph framework that efficiently combines odometry measurements with pose-to-pose constraints derived from our 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 dead-reckoning localization. Our two-view bundle adjustment algorithm is evaluated extensively in simulation and is proven to outperform previous algorithms [18, 35]. We use test tank experiments and field tests to demonstrate the effectiveness of our 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 our acoustic bundle adjustment algorithm in environments where distinctive point features are present.

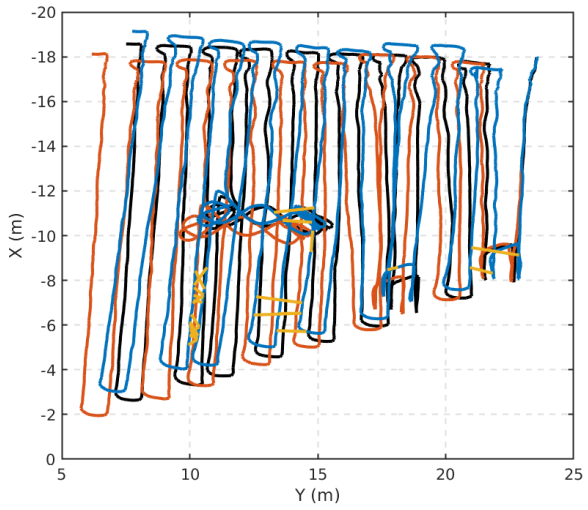
Ship hull localization error

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

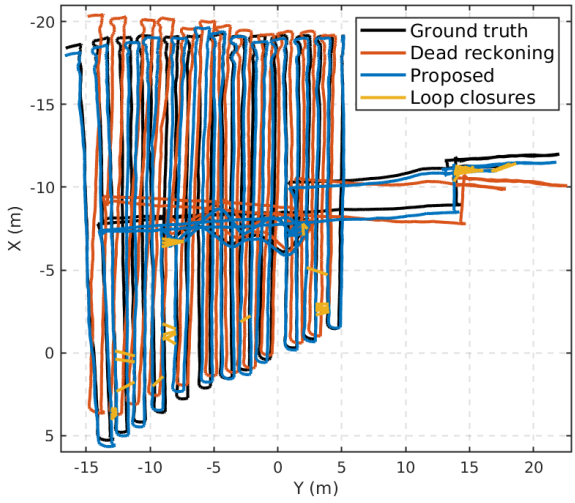
Table II: The localization errors of dead reckoning (DR), the method of Li et al. (Li) and our proposed method (Prop.) on the six ship hull data sets presented in [18]. Each error metric denotes the mean error of all poses in the trajectory over 10 trials of each data set. Our proposed method achieves the lowest error on 14 out of 18 metrics.



(a)



(b)



(c)

Figure 10: (a) Isometric view of the six ship-hull data sets, each plotted with a distinct color. (b) and (c) show sample ground truth, dead reckoning, and SLAM trajectories for data sets 2 and 3, respectively.

APPENDIX

In this appendix we present the Jacobians of the sonar prediction functions $\mathbf{h}_i^A(\mathbf{l}_i)$ and $\mathbf{h}_i^B(x_B, \mathbf{l}_i)$. Note that while in Section III we use the notation $\mathbf{H}_i = \frac{\partial \mathbf{h}_i(\mathbf{x})}{\partial \mathbf{x}}$, the partial derivatives of the measurement functions w.r.t. all landmarks except $\mathbf{l}_i$ are zero. Therefore, we will only examine the block components of $\mathbf{H}_i^A$ and $\mathbf{H}_i^B$ corresponding to the partial derivative w.r.t. x_B and $\mathbf{l}_i$.

First, we examine the Jacobians of the measurement function utilizing the full 3-vector parameterization $\mathbf{l}_i$ of a landmark. Since $\mathbf{l}_i$ is parameterized in spherical coordinates relative to pose x_A, the Jacobians of the measurement function $\mathbf{h}_i^A(\mathbf{l}_i) = \begin{bmatrix} \theta_i & r_i \end{bmatrix}^T$ are trivial:

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

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

rule:

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

where

$$\begin{aligned}\frac{\partial \hat{\mathbf{z}}}{\partial \mathbf{q}} &= \begin{bmatrix} \frac{-q_y}{\sqrt{q_x^2+q_y^2}} & \frac{q_x}{\sqrt{q_x^2+q_y^2}} & 0 \\ \frac{q_x}{\sqrt{q_x^2+q_y^2+q_z^2}} & \frac{q_y}{\sqrt{q_x^2+q_y^2+q_z^2}} & \frac{q_z}{\sqrt{q_x^2+q_y^2+q_z^2}} \end{bmatrix} \\ \frac{\partial \mathbf{q}}{\partial x_B} &= \begin{bmatrix} [\mathbf{q}]_x & -\mathbf{I}_{3\times 3} \end{bmatrix} \\ \frac{\partial \mathbf{q}}{\partial \mathbf{p}} &= \begin{bmatrix} \mathbf{R}_{x_B}^T \end{bmatrix} \\ \frac{\partial \mathbf{p}}{\partial \mathbf{l}_i} &= \begin{bmatrix} -r_i \sin\theta_i \cos\phi_i & \cos\theta_i \cos\phi_i & -r_i \cos\theta_i \sin\phi_i \\ r_i \cos\theta_i \cos\phi_i & \sin\theta_i \cos\phi_i & -r_i \sin\theta_i \sin\phi_i \\ 0 & \sin\phi_i & r_i \cos\phi_i \end{bmatrix}\end{aligned}$$

Here $[\cdot]_x$ denotes the 3×3 skew-symmetric cross-product matrix of a 3-vector. With these computed, the Jacobians of $\mathbf{h}_i^A(\mathbf{m}_i)$ and $\mathbf{h}_i^B(x_B, \mathbf{m}_i)$ are trivial, as we simply remove the last column of the appropriate Jacobians that corresponds to ϕ_i, which is no longer part of the state:

$$\begin{aligned}\frac{\partial \mathbf{h}_i^A(\mathbf{m}_i)}{\partial x_B} &= \mathbf{0} \\ \frac{\partial \mathbf{h}_i^A(\mathbf{m}_i)}{\partial \mathbf{m}_i} &= \mathbf{I}_{2\times 2} \\ \frac{\partial \mathbf{h}_i^B(x_B, \mathbf{m}_i)}{\partial x_B} &= \frac{\partial \hat{\mathbf{z}}}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial x_B} \\ \frac{\partial \mathbf{h}_i^B(x_B, \mathbf{m}_i)}{\partial \mathbf{m}_i} &= \frac{\partial \hat{\mathbf{z}}}{\partial \mathbf{q}}\frac{\partial \mathbf{q}}{\partial \mathbf{p}}\frac{\partial \mathbf{p}}{\partial \mathbf{m}_i}\end{aligned}$$

where

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

REFERENCES

[1] 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)*, 2006, pp. 1891–1897.

[2] 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.

[3] M. D. Aykin and S. Negahdaripour, "On feature matching and image registration for two-dimensional forward-scan sonar imaging," *Journal of Field Robotics*, vol. 30, no. 4, pp. 602–623, 2013.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[5] N. Brahim, D. Guériot, 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*, 2011, pp. 1–6.

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

[7] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017, http://dx.doi.org/10.1561/2300000043.

[8] M. Fallon, M. Kaess, H. Johannsson, and J. J. Leonard, "Efficient AUV navigation fusing acoustic ranging and side-scan sonar," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 2398–2405.

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

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

[11] 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)*, 2015, pp. 758–765.

[12] T. Huang, "Acoustic structure from motion," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, 2016.

[13] 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)*, 2012, pp. 5298–5305.

[14] 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)*, 2016, pp. 224–230.

[15] 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)*, 2010, pp. 4396–4403.

[16] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Intl. J. of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[17] 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 *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 1230–1237.

[18] J. Li, M. Kaess, R. Eustice, and M. Johnson-Roberson, "Pose-graph SLAM using forward-looking sonar," *IEEE Robotics and Automation Letters*, 2018.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] 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," *International Federation of Automatic Control PapersOnLine*, vol. 50, no. 1, pp. 1043–1049, 2017.

[21] 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, 2013.

[22] ——, "Application of forward-scan sonar stereo for 3-D scene reconstruction," *IEEE J. of Oceanic Engineering*, 2018.

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

[24] 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)*, 2012, pp. 3467–3474.

[25] 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.

[26] Y. Petillot, I. T. Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *IEEE J. of Oceanic Engineering*, vol. 26, no. 2, pp. 240–251, 2001.

[27] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Intl. Conf. on Computer Vision (ICCV)*, vol. 11, no. 1, 2011, p. 2.

[28] 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.

[29] 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.

[30] 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*, 2015, pp. 1–6.

[31] Y.-e. Song and S.-J. Choi, "Underwater 3D reconstruction for underwater construction robot based on 2D multibeam imaging sonar," *Journal of Ocean Engineering and Technology*, vol. 30, no. 3, pp. 227–233, 2016.

[32] 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)*, 2012, pp. 573–580.

[33] 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), 2016, pp. 4288–4295.

[34] E. Westman and M. Kaess, "Underwater AprilTag SLAM and extrinsics calibration for AUV localization," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-18-43, 2018.

[35] 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)*, 2018.

[36] Y. Yang and G. Huang, "Acoustic-inertial underwater navigation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 4927–4933.

[37] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2016, pp. 809–816.