

Long-Term Mapping Techniques for Ship Hull Inspection and Surveillance Using an Autonomous Underwater Vehicle

Paul Ozog

Department of Electrical Engineering and Computer Science
University of Michigan
paulozog@umich.edu

Nicholas Carlevaris-Bianco

Department of Electrical Engineering and Computer Science
University of Michigan
carlevar@umich.edu

Ayoung Kim

Department of Civil and Environmental Engineering
Korea Advanced Institute of Science and Technology
ayoungk@kaist.ac.kr

Ryan M. Eustice

Department of Naval Architecture and Marine Engineering
University of Michigan
eustice@umich.edu

Abstract

This paper reports on a system for an autonomous underwater vehicle to perform *in-situ*, multiple session, hull inspection using long-term simultaneous localization and mapping (SLAM). Our method assumes very little *a-priori* knowledge, and does not require the aid of acoustic beacons for navigation, which is a typical mode of navigation in this type of application. Our system combines recent techniques in underwater saliency-informed visual SLAM and a method for representing the ship hull surface as a collection of many locally planar surface features. This methodology produces accurate maps that can be constructed in real-time on consumer-grade computing hardware. A single-session SLAM result is initially used as a prior map for later sessions, where the robot automatically merges the multiple surveys into a common hull-relative reference frame. To perform the re-localization step, we use a particle filter that leverages the locally planar representation of the ship hull surface, and a fast visual descriptor matching algorithm. Finally, we apply the recently-developed graph sparsification tool, generic linear constraints (GLC), as a way to manage the computational complexity of the SLAM system as the robot accumulates information across multiple sessions. We show results for 20 SLAM sessions for two large vessels over the course of days, months, and even up to three years, with a total path length of approximately 10.2 km.

1 Introduction

Periodic maintenance, structural defect assessment, and harbor surveillance of large ships is of great importance in the maritime industry because structural wear and defects are a significant source of lost revenue (Boon et al., 2009). Until recently, these tasks were carried out by periodic dry-docking, where a vessel is removed from the water for easy access to the entire ship hull. This task has recently evolved to include the deployment of a variety of unmanned underwater robots to map and inspect the hull *in-situ*. Due to the periodic nature of the inspection, surveys are often conducted every few months. Having a single autonomous underwater vehicle (AUV) map an entire vessel within a short time frame may be impossible. However, using multi-session mapping techniques would parallelize the inspection process, or allow partial maps to be built as time allows. In addition, the sheer size of the final map would prevent a state-of-the-art simultaneous localization and mapping (SLAM) system from performing in real-time. Therefore, a computationally efficient and compact representation of the map is critical to maintain real-time map-merging performance in the long term.

The goal of this work is to extend our Hovering Autonomous Underwater Vehicle (HAUV) graph-based visual SLAM system to support automated registration from multiple maps via a method called multi-session SLAM. To maintain real-time performance, redundant or unnecessary nodes are marginalized from the factor-graph once the sessions are aligned into a common hull-relative reference frame. An overview of our multi-session SLAM framework is illustrated in Fig. 1, including a visualization of how the estimated trajectory would align with a computer aided design (CAD) hull model. This framework is consistent, computationally efficient, and leverages recent techniques in SLAM research to create an extensive map. Our method has several advantages over long-baseline (LBL) and global positioning system (GPS) navigation methods. In essence, long-term hull-inspection with an AUV demands a hull-relative method for navigation because a vessel’s berth may change with time.

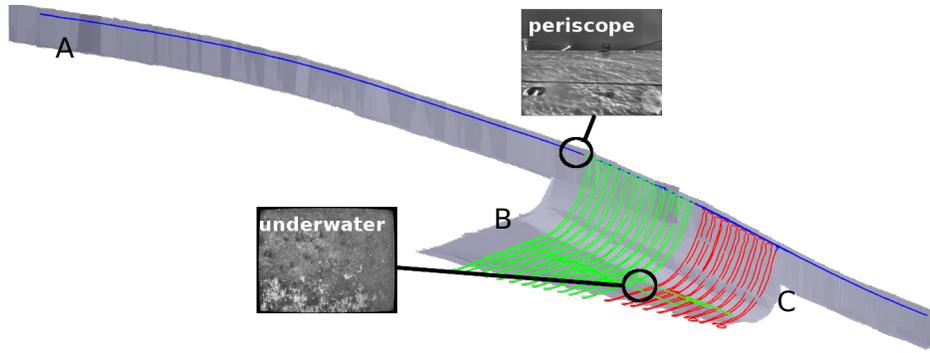
We have laid the foundation of this paper in our earlier publications (Ozog and Eustice, 2013b, 2014). Here, we offer several improvements to our previous methodology, including:

- The use of a constraint between a plane and individual Doppler velocity log (DVL) beams.
- Periscope re-localization with the aid of pose-constrained correspondence search (PCCS).
- Exemplar views to maintain visual diversity in the merged graph.
- A quantitative evaluation of the effect of repeated application of generic linear constraints (GLC) for our application.
- Significantly more field data used in our evaluation. The surveys now span three years, with a total path length exceeding 10 km.

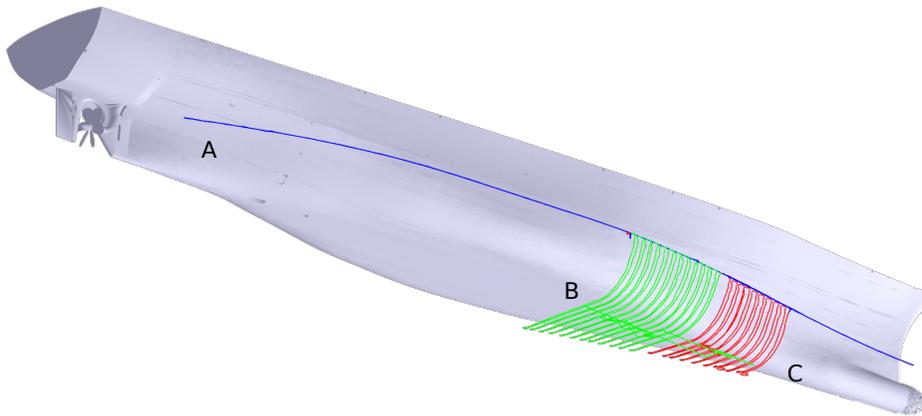
We start with a brief overview of relevant literature in Section 2. In Section 3, we summarize our single session SLAM framework and introduce our planar factor constraint, which substantially improves our survey results. We describe our overall approach for long-term hull-relative SLAM using the HAUV in Section 4. Here, we present how our SLAM system for the HAUV can be adapted for the GLC framework, thus enabling long-term mapping capabilities. Finally, in Section 5, we experimentally evaluate our approach using real-world data, and conclude with a discussion in Section 6.

2 Related Work

We cover three broad areas of related work: (i) underwater surveillance using robots and underwater sensors, (ii) managing graph complexity for long-term SLAM, and (iii) merging intermediate SLAM graphs into a larger map without a fixed absolute reference frame, for example, as provided by a GPS.



(a) *SS Curtiss* multisession SLAM result with three sessions (A,B,C)



(b) Merged graphs overlaid on CAD model

Figure 1: Multi-session SLAM overview. In (a), we show three different sessions aligned to a common frame using our multi-session SLAM framework, with each color denoting a different session. A surface mission ('A'), in blue, uses the periscope camera to capture images of the ship's above-water superstructure. Two underwater missions ('B' and 'C'), in red and green, capture underwater images of the ship hull. The gray patches denote the planar features used in our novel planar constraints framework, described in §3.4. Two representative images from the periscope and underwater camera are shown. In (b), we manually aligned a low fidelity CAD model of the ship for the sake of visual clarity, with the corresponding sessions colored appropriately. Note that this CAD model is not used in our SLAM system.

2.1 Robotic Underwater Surveillance

Early work by Harris and Slate (1999) aimed to assess superficial defects caused by corrosion in large ship hulls using the *Lamp Ray* remotely operated vehicle (ROV). A non-contact underwater ultrasonic thickness gauge measured the plate thickness of the outer hull, and an acoustic beacon positioning system was used for navigation. Since the early 2000's, some focus has shifted from structural defect detection to the identification of foreign objects like limpet mines. Sonar imaging sensors, in particular, have been successfully deployed to detect foreign objects underwater (Belcher et al., 2001). In work by Trimble and Belcher (2002), a similar imaging technique was successfully deployed using the free-floating *CetusII* AUV. Like the *Lamp Ray*, this system used a specifically-designed LBL acoustic beacon system for navigation around ship hulls and similar underwater structure. Additionally, Carvalho et al. (2003) used a probability-of-detection metric to assess the effectiveness of these sorts of acoustic-based sensors. Regardless of the specific robot used, the significant amount of infrastructure required for beacon-based navigation (Milne, 1983) hinders the widespread use of these robotic systems in large shipping ports.

More recently, techniques in computer vision and image processing have been applied to various applications in underwater inspection. Negahdaripour and Firoozfam (2006) used a stereo vision system to aid in the navigation of a ROV, and provide coarse disparity maps of underwater structure. Ridaoui et al. (2010) used both a camera and sonar to inspect dams using an ROV, and employed bundle-adjustment techniques and image blending algorithms to create mosaics of wall-like structures of a hydroelectric dam. Some efforts have even attempted to identify physical defects only from a single image. Bonnín-Pascual and Ortiz (2010) applied various image processing techniques to automatically identify the portions of a single image that appear to contain corrosion.

In addition to free-floating vehicles, there are some approaches that use robots that are physically attached to the ship hull itself. Menegaldo et al. (2008, 2009) use magnetic tracks to attach the robot to the hull, but their hardware is limited to above-water portions of the hull. By contrast, Ishizu et al. (2012) developed a robot that uses underwater, magnetically adhesive spider-like arms to traverse the hull. Like previously mentioned free-floating vehicles, this robot employs a stereo camera to assess the amount of corrosion. Hull inspection with high frequency profiling sonar is presented by VanMiddlesworth et al. (2013) where they build a 3D map and the entire pose-graph is optimized using submaps.

Although the main focus of this paper is on in-water hull inspection, there are other applications for long-term AUV-based surveillance and mapping in large area underwater environments. Autonomous vehicles present significant potential for efficient, accurate and quantitative mapping in arctic exploration (Kunz et al., 2009), Mariana Trench exploration (Bowen et al., 2009), pipeline inspection (Gustafson et al., 2011) and coral reef habitat characterization (Singh et al., 2004). These studies were presented within the context of a single-session SLAM application, however, the repeated monitoring and multi-session SLAM framework we present here could be beneficial in those application domains as well.

2.2 Long-term SLAM for Extensive Mapping

Graph-based SLAM solvers (Dellaert and Kaess, 2006; Grisetti et al., 2007; Kaess et al., 2008; Kummerle et al., 2011) are popular tools in SLAM, however they are limited by computational complexity in many long-term applications. As a robot explores an area and localizes within its map, pose nodes are continually added to the graph. Therefore, optimizing a graph of modest spatial extent becomes intractable if the duration of the robot's exploration becomes sufficiently large. A number of proposed methods attempt to address this problem by removing nodes from the graph, thus reducing the graph's size. Recent emphasis has been placed on measurement composition using full-rank factors, such as laser scan matching or odometry (Konolige and Bowman, 2009; Eade et al., 2010; Kretzschmar and Stachniss, 2012). Measurement composition synthesizes virtual constraints by compounding full-rank 6-degree of freedom (DOF) transformations, as shown in Fig. 2(a), however this operation is ill-defined for low-rank measurements, such as those produced by a monocular camera, shown in Fig. 2(b). This is one of the primary motivations for the development of the factor-graph marginalization framework described by Carlevaris-Bianco and Eustice (2013a), called generic linear constraints (GLC), which address the low-rank problem.

The GLC method avoids measurement composition and works equally well using full-rank and low-rank loop-closing measurements. This is a critical requirement for the HAUV application because low-rank factors from either a monocular camera or an imaging sonar are the primary means of correcting navigational error from the DVL. Furthermore, measurement composition tends to re-use measurements when synthesizing virtual constraints. Doing so double-counts information and does not produce the correct marginalization in the reduced SLAM graph.

Previous studies of large-scale SLAM suggest solving SLAM using multiple maps to efficiently and consistently optimize very large graphs. The first method for constructing a large-scale map involves the use of submapping techniques, i.e., building local maps and then stitching the local maps together to build larger maps. This type of submap technique was originally proposed by Leonard and Feder (1999) to reduce the computational cost by decoupling the larger maps into submaps. This work was further developed into

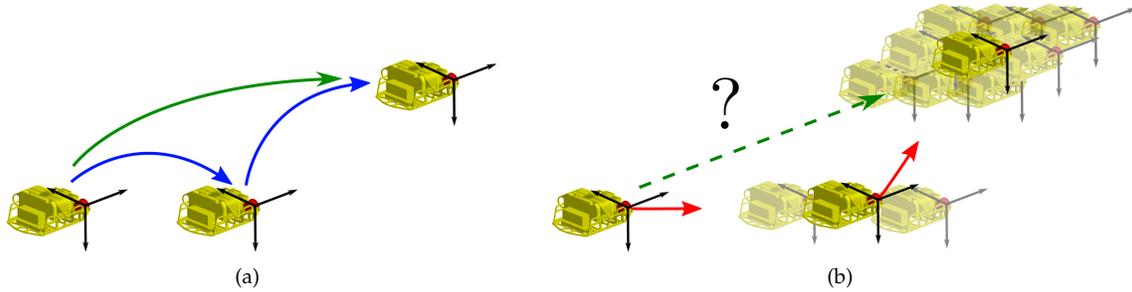


Figure 2: This figure shows that the measurement composition technique, often used in variable marginalization in graph-based SLAM, is undefined for low-rank monocular camera measurements, as used in this work. For generating the green end-to-end 6-DOF transformation in (a), one simply has to compound the intermediate 6-DOF relative-pose transformations, in blue. This cannot be done for the intermediate 5-DOF measurements shown as red arrows in (b). In this case, computing the end-to-end baseline direction of motion is ill-posed because it depends on the scale at each camera measurement, which is not known.

the *Atlas* framework by Bosse et al. (2004) where each submap is represented as a coordinate system in a global graph. For example, Bosse and Zlot (2008) developed a two-level loop-closing algorithm based on the *Atlas* framework, while Neira and Tardos (2001) introduced a way to measure joint compatibility, which resulted in optimal data association. Examples of underwater submap applications include vision (Pizarro et al., 2009) and bathymetry maps from acoustic sensors (Roman and Singh, 2005). Kim et al. (2010) used a similar submapping approach, except in the context of a factor-graph representation of cooperative SLAM. They used so-called *anchor nodes*, which are variable nodes representing the transformation from a common reference frame to the reference frame of each vehicle’s SLAM graph. Anchor nodes are nearly identical to *base nodes* introduced by Ni et al. (2007). They are a very simple way to extend observation models to perform submap alignment, which is why our multi-session SLAM approach (discussed in §4) formulates the session alignment using anchor nodes.

To efficiently perform multi-session SLAM in the absence of a readily-available absolute position reference requires an initial data-association step. By doing so, each map can be expressed into a common frame of reference, thus bounding the search space for identifying additional data associations. Recently, visual information has been exploited in data association by providing an independent loop-closing ability based upon appearance. Nistér and Stewénus (2006) introduced a framework that used a bag-of-words (BoW) representation of images to quickly compare the histogram of words between two images using vocabulary trees. Cummins and Newman (2008) used a generative model to describe the co-occurrence of words in the training data. They developed the popular Fast Appearance-Based Mapping (FAB-MAP) algorithm, which frames the problem as a recursive Bayes filter and showed that their system can leverage temporal information with or without a topological motion prior. Both of these methods have become a common practice for loop-closure detection in real-time SLAM front-ends.

Our approach also shares some similarity with methods that maintain visual diversity in long-term SLAM graphs. GraphSLAM by Thrun and Montemerlo (2006) presents some early work on a large urban area with visual features. The method, however, was an offline approach with conventional optimization methods. Our work has more connection to recent online approaches such as the one introduced by Konolige and Bowman (2009). They represent a place in the SLAM graph as a collection of exemplar views. This encourages visually diverse keyframes in the SLAM graph, thus increasing potential matchability as more images are collected by the robot. Furthermore, this approach limits the number of views to be less than a user-defined parameter, which keeps the graph size bounded.

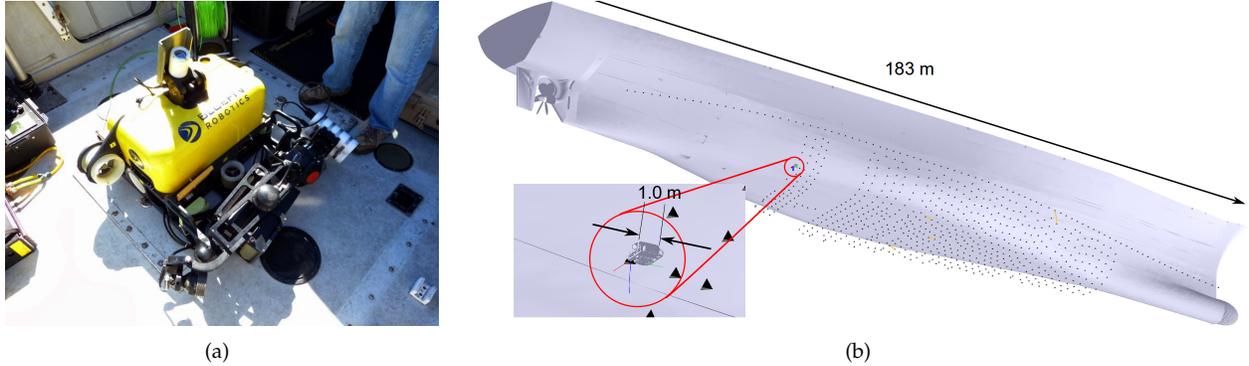


Figure 3: The Bluefin Robotics HAUV (shown in (a)) is specifically designed for *in-situ* underwater hull and harbor inspection and requires no pre-existing infrastructure (e.g., acoustic beacons), to localize. Instead, it uses a DVL for hull-relative navigation, which differentiates it from other similar inspection robots. In (b), we show the relative size of the vehicle compared to a typically-sized vessel.

3 Underwater Visual SLAM with Planar Constraints

In our trials, we use the HAUV (Fig. 3) developed by Bluefin Robotics. The HAUV is a free-floating vehicle that requires no existing LBL infrastructure for localization (Hover et al., 2007). Instead, the HAUV uses a DVL as its primary navigation sensor, which can operate in a hull-relative or seafloor-relative mode. This allows the robot to inspect a variety of underwater infrastructure: ship hulls, pilings, or harbor surveillance. The vehicle is equipped with two monocular cameras so that it can switch between below-water and above-water images. The periscope camera is fixed with a static angle allowing the robot to capture superstructure and above-water features. The underwater camera is mounted on a tray along with the imaging sonar and DVL. During a mission, the tray is servoed such that both the DVL and camera always point nadir to the hull while the robot keeps a fixed distance.

In this section, we introduce the overall system, including the SLAM back-end, and camera and plane constraints. The illustration of the pose-graph framework is given in Fig. 4. The robot is provided with camera, planar, and odometry measurements together with absolute measurements on depth and attitude. Following a brief illustration of each module, we specifically focus on the planar constraints that substantially improve mapping and localization performance.

3.1 State Representation

For the remainder of this section, let $\mathbf{x}_{ij} = [x_{ij}, y_{ij}, z_{ij}, \phi_{ij}, \theta_{ij}, \psi_{ij}]^T$ be the 6-DOF relative-pose of frame j as expressed in frame i , where x, y, z are the Cartesian translation components, and $\phi_{ij}, \theta_{ij},$ and ψ_{ij} denote the roll (x -axis), pitch (y -axis), and yaw (z -axis) Euler angles, respectively. R_j^i is the rotation matrix that rotates a vector in j 's frame to a vector in i 's frame, and \mathbf{t}_{ij}^i is the translation from i to j as expressed in i 's frame. Finally, g will refer to the global, or common, reference frame.

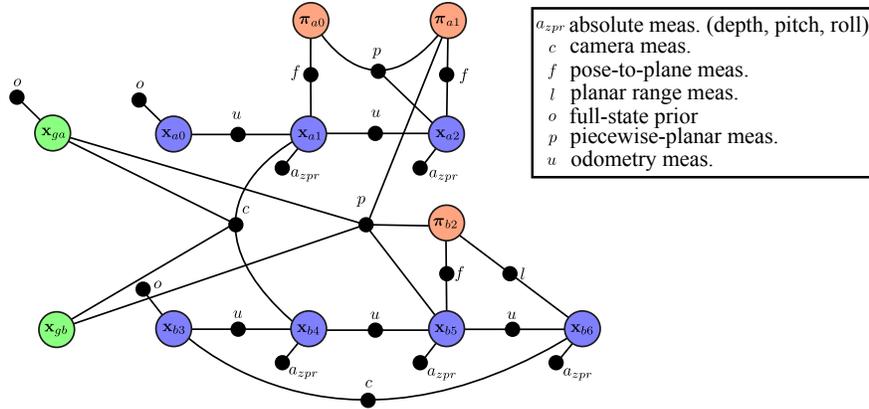


Figure 4: An illustration of a multi-session factor-graph with camera, planar and other measurements. Orange, blue, and green variable nodes correspond to planes, vehicle poses, and anchor nodes, respectively. The top half nodes correspond to a prior session (reference frame a), while the bottom half belong to a subsequent session (reference frame b). Factor nodes are shown as black circles, along with the type of observation encoded as a script character. The use of odometry and prior factors is common-practice in pose-graph SLAM so these are not covered in detail in this paper. The camera and planar measurements are discussed in more detail in §3.3 and §3.4, respectively.

3.2 SLAM Back-end

The HAUV is equipped with a variety of sensors: a DVL and inertial measurement unit (IMU) for odometry, a pressure depth sensor, two cameras for 3D bearing measurements, and an imaging sonar. The use of these sensors for SLAM is described in detail by Johannsson et al. (2010) and Kim and Eustice (2009). SLAM aims to correct for any accumulated navigational error when the robot senses environmental features it has seen previously. The DVL measures four beam ranges, along with four body-frame velocities. These range measurements can also be converted to a sparse 3D point set, which allows us to fit local planar patches to the DVL data.

The SLAM system used on the HAUV has evolved from an offline filtering-based approach with manual data association (Walter et al., 2008), to a real-time factor-graph optimization framework that supports both sonar and monocular camera measurements automatically (Hover et al., 2012). Currently, our SLAM system uses the incremental smoothing and mapping (iSAM) algorithm as the optimization back-end (Kaess et al., 2008; Kaess and Dellaert, 2009; Kaess et al., 2010).

3.3 Camera Constraints

The previously-mentioned underwater and periscope cameras can both be used to derive spatial constraints from two-view feature correspondence. The underwater environment is particularly challenging because it does not always contain visually useful features for camera-derived measurements. To alleviate this issue, Kim and Eustice (2013) allowed for greatly increased computational performance by only considering visually salient keyframes in the visual SLAM pipeline. They derived a local saliency score using a coarse BoW representation of each keyframe from speeded-up robust features (SURF) descriptors (Bay et al., 2006). To derive two-view camera constraints from overlapping images, the system searches for correspondence between keyframes using the scale-invariant feature transform (SIFT) descriptor (Lowe, 2004). A scene depth prior based upon the DVL and a relative-pose prior based on the SLAM estimate offer a constrained putative correspondence search that is faster and more consistent than typical appearance-only random sample consensus (RANSAC)-based approaches (Eustice et al., 2008; Carlevaris-Bianco and Eustice, 2011).

From two overlapping images taken from our monocular camera, we can derive a spatial constraint, modulo scale, between the vehicle poses from which those images were taken. This measurement therefore has five DOFs: three rotations, and a vector representing the direction of translation that is parameterized by azimuth and elevation angles.

We have found that adding camera measurements with low baseline can potentially produce outlier measurements. These outliers are occasionally not identified with a Mahalanobis distance check prior to their addition to the SLAM graph. We therefore use the robust back-end method known as Dynamic Covariance Scaling (DCS) (Agarwal et al., 2013). This method is based on the work by Sunderhauf and Protzel (2012), which employs switching variables to turn on and off measurements by scaling the information matrix for loop-closing measurements. DCS does this in closed form, which avoids having to increase the dimensionality of the state to estimate the scale variables. Thus, the robust monocular camera factor between poses \mathbf{x}_{gi} and \mathbf{x}_{gj} is:

$$\chi_{DCS_{ij}}^2 = \left\| \mathbf{z}_{ij}^{5\text{dof}} - h^{5\text{dof}}(\mathbf{x}_{gi}, \mathbf{x}_{gj}) \right\|_{\frac{1}{s_{ij}} \Xi_{\mathbf{z}_{ij}}}^2, \quad (1)$$

where $h^{5\text{dof}}(\mathbf{x}_{gi}, \mathbf{x}_{gj}) = [\alpha_{ij}, \beta_{ij}, \phi_{ij}, \theta_{ij}, \psi_{ij}]$ is the predicted baseline direction of motion azimuth α_{ij} , elevation β_{ij} , and the relative Euler angles $\phi_{ij}, \theta_{ij}, \psi_{ij}$, and $\frac{1}{s_{ij}} \Xi_{\mathbf{z}_{ij}}$ is the DCS scaled measurement covariance matrix. Agarwal et al. (2013) show that the covariance scale variable, s_{ij} , is given by

$$s_{ij} = \min \left(1, \frac{2\Phi}{\Phi + \chi_i^2} \right),$$

where χ_i^2 is the unscaled chi-squared error, computed from (1) using the unscaled measurement covariance, $\Xi_{\mathbf{z}_{ij}}$. Higher values of the parameter Φ make it more likely to accept loop closures. For our field trials, we choose $\Phi = 5$; however, values between 1 and 10 produce nearly identical results.

3.4 SLAM with Planar Segments

To further constrain the vehicle’s trajectory, we use planar patches as nodes in the SLAM graph using a piecewise-planar model (Ozog and Eustice, 2013b), which is summarized in Fig. 5. This method provides an explicit representation of the ship hull surface in the factor graph, and produces accurate and efficient maps using only a sparse 3D point cloud (such as one produced by the underwater DVL navigation sensor). Furthermore, this method preserves the overall curvature of the ship’s exterior hull, assuming that the robot has coarse prior knowledge on the side-to-side and top-to-bottom characteristic curvature of the surface. This distinguishes our work from other planar SLAM approaches (Weingarten and Siegwart, 2006; Trevor et al., 2012) in two ways. First, other methods make use of a much more data-rich 3D laser scanners or depth cameras. Second, these other approaches are experimentally evaluated in structured environments, like office buildings, that contain many planar walls at sharp angles to each other. Unfortunately, these approaches require that the planes be directly observed from two or more views. However, this is often not achievable in ship hull inspection because (i) the hull is smoothly curved, and (ii) the DVL’s field of view is both more narrow and more sparse than a typical laser scanner or depth camera. In this section, we explain the plane constraints from a single session point of view, but another significant strength of these constraints are in robust registration that allows for multi-session SLAM, which will be discussed in §4.

3.4.1 Plane Composition Operators

The planar observation models used in this paper involve expressing planes in multiple frames of reference. In this section, we describe these operations in detail. Let π_{ik} be the plane indexed by k , expressed with respect to frame i . When i is the global frame, g , this plane corresponds to a variable node in our factor-graph. We parametrize the plane $\pi_{gk} = [n_{gk}^x, n_{gk}^y, n_{gk}^z]^\top$ using three numbers, where n_{gk}^x , n_{gk}^y , and n_{gk}^z

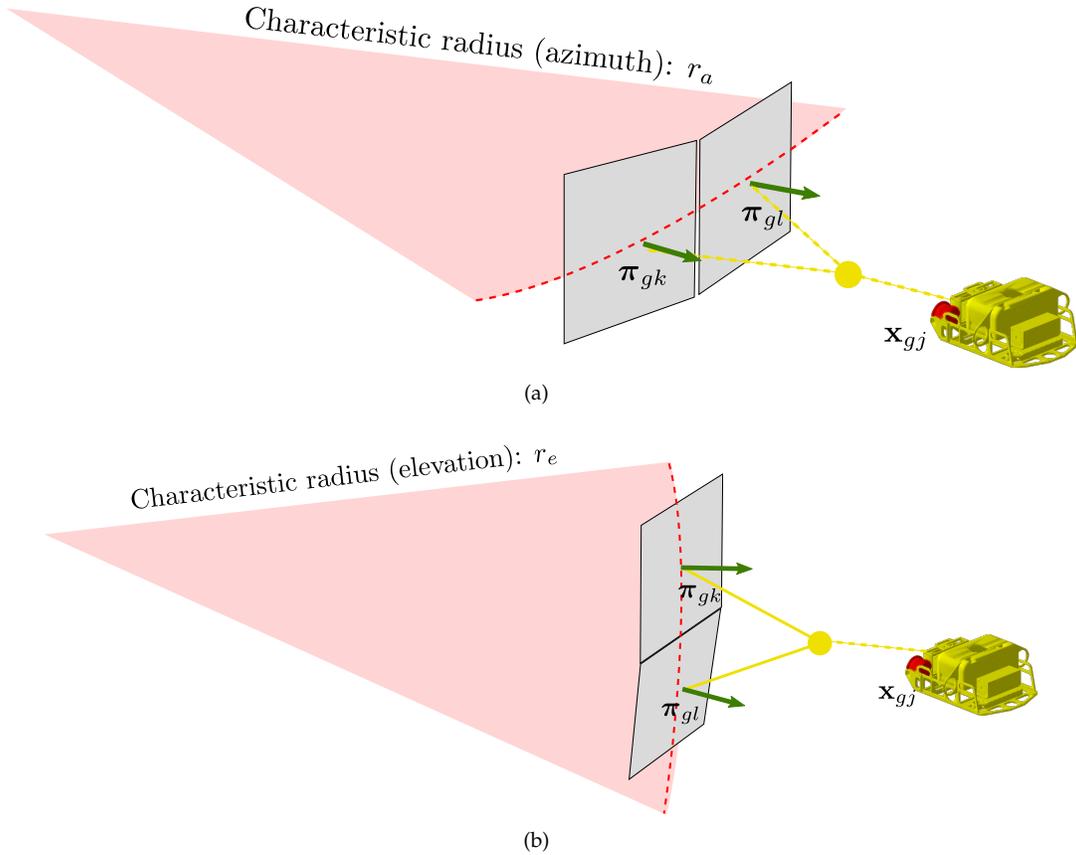


Figure 5: The method in this paper represents the ship hull surface as a collection of locally planar feature patches (gray), with a simple assumed prior on the characteristic curvature model. The residual error of two co-registered planes is weighted in such a way so that only errors that extend outside of the deviation explained by the characteristic curvature are significant. The method relies on two characteristic radii parameters: one for the azimuth (side-to-side variation, shown in (a)), and one for elevation (top-down variation, shown in (b)). Ozog and Eustice (2013b) show that the performance of this approach is not overly sensitive to these user-defined parameters.

are the x , y , and z components of the (non-unit) normal vector to the plane k , expressed in the global-frame, g . We do not use the popular Hessian normal form simply because it is parameterized by four numbers, despite having only *three* degrees of freedom. This over-parameterization would therefore result in a singular covariance matrix. The Hessian normal form is given by: $\hat{\mathbf{n}}^\top \mathbf{x} = -p$ for any point \mathbf{x} that lies on the plane, where $\hat{\mathbf{n}}$ is the unit normal of the plane, and p is the distance of the plane to the origin. Conversely, our parameterization uses only three numbers. We simply scale the unit-normal by the distance to the origin, yielding what we denote as the π symbol. With our parameterization, we have $\pi^\top \mathbf{x} = -\|\pi\|^2$ for any point \mathbf{x} on the plane.

The frame of reference of the plane is denoted with a subscript whereby the plane indexed by k as expressed with respect to some frame i is denoted π_{ik} . If one wishes to express a plane in a different frame of reference, one simply applies the \boxminus and \boxplus operators, which take as input a 6-DOF pose and 3-DOF plane. Let $\mathbf{x}_{ij} \boxminus \pi_{ik} = \pi_{jk}$, where

$$\pi_{jk} = \mathbf{x}_{ij} \boxminus \pi_{ik} = \frac{(\mathbf{t}_{ij}^\top \pi_{ik} + \|\pi_{ik}\|^2) \mathbf{R}_i^j \pi_{ik}}{\|\pi_{ik}\|^2}.$$

The \boxminus operator is commonly used to evaluate factor potentials, discussed in the next section. We also define the \boxplus operator, where $\mathbf{x}_{ij} \boxplus \boldsymbol{\pi}_{jk} = \boldsymbol{\pi}_{ik}$. This can be expressed in terms of the \boxminus operator by

$$\mathbf{x}_{ij} \boxplus \boldsymbol{\pi}_{jk} = \ominus \mathbf{x}_{ij} \boxminus \boldsymbol{\pi}_{jk},$$

where \ominus is the pose-inversion operation (Smith et al., 1986). The \boxplus operator is commonly used to transform plane measurements from the vehicle frame to the global, but not to evaluate factor potentials.

3.4.2 Planar Constraints between Poses and other Planes

In this section we describe observation models used to add measurements to our SLAM graph. A binary factor potential of the form

$$\Psi(\mathbf{x}_{gi}, \boldsymbol{\pi}_{gl}; \mathbf{z}_{\pi_{il}}, \Sigma_{\mathbf{z}_{\pi_{il}}}) = \|\mathbf{z}_{\pi_{il}} - (\mathbf{x}_{gi} \boxminus \boldsymbol{\pi}_{gl})\|_{\Sigma_{\mathbf{z}_{\pi_{il}}}}^2 \quad (2)$$

is used when the DVL sensor determines a good planar fit of a sliding time-window of 3D points. The covariance matrix, $\Sigma_{\mathbf{z}_{\pi_{il}}}$, is determined by first-order approximation of a function that fits a plane from the 3D points, which are assumed to be corrupted by Gaussian noise.

A piecewise-planar factor potential is similarly used to constrain the surface normals of two plane nodes with respect to a common pose node. It is given by the expression

$$\Omega(\mathbf{x}_{gi}, \boldsymbol{\pi}_{gk}, \boldsymbol{\pi}_{gl}; W_{ijkl}) = \|\mathbf{x}_{gi} \boxminus \boldsymbol{\pi}_{gk} - \mathbf{x}_{gi} \boxminus \boldsymbol{\pi}_{gl}\|_{W_{ijkl}}. \quad (3)$$

A weight matrix, W_{ijkl} , is based on the characteristic curvature of the ship hull. It is a function of two vehicle poses from which the planes k and l were observed (that is, i and j , respectively):

$$W_{ijkl} = \text{diag}(\Delta_x^2, \Delta_y^2, \Delta_z^2) + C\Sigma_{ij}C^\top,$$

where Σ_{ij} is the joint covariance between poses i and j . C is the Jacobian of the function $c(\cdot)$ that is a simple curvature model of the surface being observed. Since the two measured planes will not be perfectly coplanar due to surface curvature, it is a simple way to take this into account when computing error from the \boxminus operator described in §3.4. This approach is illustrated in Fig. 5, and the mathematical formulation of the curvature model is

$$\begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_z \end{bmatrix} = c(\mathbf{x}_{gi}, \mathbf{x}_{gj}; \boldsymbol{\pi}_{ik}, \boldsymbol{\pi}_{jl}, r_a, r_e) = \boldsymbol{\pi}_{jl} - \left(\mathbf{x}_{ij} \boxminus \text{trans} \left(\text{dir}(\boldsymbol{\pi}_{ik}) + \begin{bmatrix} t_{ijy}^i/r_a \\ t_{ijz}^i/r_e \\ 0 \end{bmatrix} \right) \right),$$

where $\text{dir}(\cdot)$ converts a 3-dimensional Cartesian vector to one expressed in 3-dimensional spherical coordinates: azimuth, elevation, and distance to the origin. Conversely, $\text{trans}(\cdot)$ converts spherical coordinates to Cartesian. See §A.1 for the full definition of these functions. The user-defined characteristic radii for azimuth and elevation are denoted r_a and r_e , respectively. For the experimental results in this paper, we use $r_a = 322$ m and $r_e = 7$ m. Similar to what is described in Ozog and Eustice (2013b), the results presented in this paper are almost identical so long as r_a and r_e are within the same order of magnitude as these provided values.

3.4.3 Range Constraints between Planes and Poses

In the event that fitting a plane to a point cloud is ill-conditioned, as is the case when all points are colinear, or when there is only a single point, we use a simple ray-tracing method to constrain a plane with these ill-conditioned DVL beams. This is illustrated in Fig. 6.

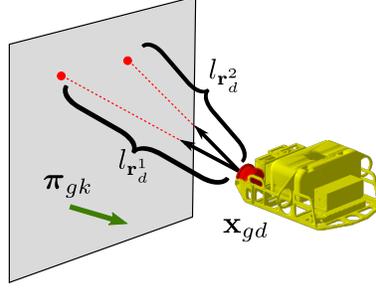


Figure 6: Given plane k and DVL pose d in the global frame, the points of intersection for each ray can be computed so long as the ray origin and direction are known using (4). We can compare the predicted range to the DVL-measured range, and use this as a spatial constraint between the pose and plane. This is particularly useful when the points detected by the plane are ill-conditioned for fitting a plane (such as only two DVL beams that will always lie on a line, as in this figure).

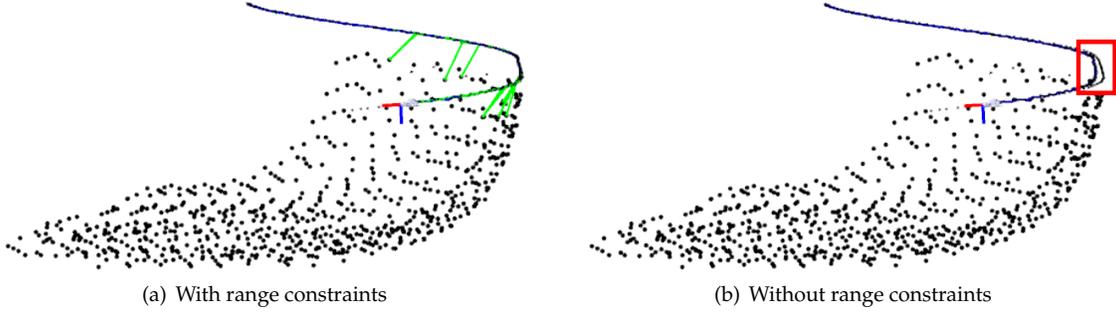


Figure 7: In practice, we find the HAUV may only successfully measure the top two DVL returns — especially when surfaced. Here, we show the benefits of leveraging this information in the context of SLAM. In each figure, the viewpoint is from stern-to-bow, and green lines encode the invocation of a range constraint. In (a), two surface missions are well-aligned, whereas in (b), there is some noticeable drift, highlighted in red. In each figure, black dots denote pose nodes in a sparsified graph using the GLC method discussed in §4.1.

Let \mathbf{r}_n^i be the n -th unit-norm ray expressed in pose frame i . If \mathbf{r}_n^i is known to intersect with plane π_{ik} , then we can compute the predicted ray length, $\hat{l}_{\mathbf{r}_n^i}$, by tracing the ray until it intersects with the surface. Because the surface is planar, we can derive a closed-form expression for the length:

$$\hat{l}_{\mathbf{r}_n^i}(\mathbf{x}_{ij}, \pi_{ik}; \mathbf{r}_n^i) = \frac{\|\mathbf{x}_{ij} \boxminus \pi_{ik}\|}{\mathbf{r}_n^{i\top}(\mathbf{x}_{ij} \boxminus \pi_{ik})}. \quad (4)$$

The resulting factor potential between pose j and plane k is computed by weighting the difference between the observed beam range, $z_{l_{\mathbf{r}_n^i}}$, and the predicted range from above:

$$\Phi\left(\mathbf{x}_{ij}, \pi_{ik}; z_{l_{\mathbf{r}_n^i}}, \sigma_{z_{l_{\mathbf{r}_n^i}}}, \mathbf{r}_n^i\right) = \left(\frac{z_{l_{\mathbf{r}_n^i}} - \hat{l}_{\mathbf{r}_n^i}(\mathbf{x}_{ij}, \pi_{ik}; \mathbf{r}_n^i)}{\sigma_{z_{l_{\mathbf{r}_n^i}}}}\right)^2, \quad (5)$$

where $\sigma_{z_{l_{\mathbf{r}_n^i}}}$ is the standard deviation of the range measurement noise from the n^{th} beam.

Intuitively, this factor potential is less informative than the piecewise planar factor defined in (3). However, in the event that a plane is not observable from the DVL’s point cloud, it still provides a useful constraint for the SLAM trajectory if a local planar patch has been previously established. In practice, we use this constraint in the event that the incident angle for the bottom two DVL beams is too large, which we find can happen when the HAUV is at the surface. An example of the benefit of this information is shown in Fig. 7.

4 Multi-session SLAM for Long-term Hull Inspection

The typical use case for the SLAM system discussed above is for single-session surveys. When the robot begins a new survey minutes or days after the previous survey, the navigation is reset and all the information from the previous map is no longer used in navigation correction. For long-term or multi-robot applications, this is detrimental for two reasons. First, there is less information for the robot to correct its navigation. Second, each session has a different local reference frame. If a particular area for one of these surveys demands attention, it requires a human-in-the-loop to identify where the area of interest lies in a hull-relative reference frame. It is therefore beneficial, though challenging, to have the surveying robot build each map with respect to the same reference frame. In this section, we present: (i) an accurate and efficient global localization system to make the initial alignment, (ii) surface-based SLAM techniques to make the real-time alignment robust and accurate, and (iii) managing the computational complexity of optimizing long-term SLAM graphs.

To illustrate how these techniques are used in practice during real-world operations, the following is a high-level outline of our multi-session SLAM system: (i) load past sessions as a SLAM graph, (ii) sparsify the graph with GLC, (iii) deploy the HAUV and start building a separate, unaligned pose-graph, (iv) localize to the past session with an initial guess of the alignment, and (v) refine alignment using monocular camera measurements and piecewise-planar constraints on anchor nodes. These steps are illustrated in Fig. 8.

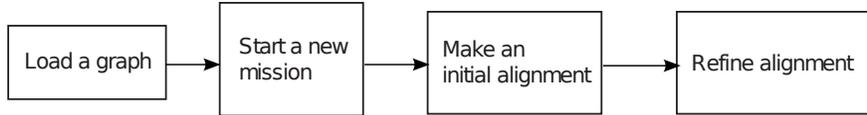
GLC has been thoroughly-evaluated on standard SLAM graphs containing pose nodes and landmark nodes. However, the graphs used in this work are significantly more nonstandard because they contain anchor nodes, planar nodes, and planar observation models. This section will also describe how GLC can be used with these aspects of our SLAM system.

4.1 GLC-based Approximate Node Removal

A detailed derivation and computational performance analysis of GLC node marginalization and removal can be found in Carlevaris-Bianco and Eustice (2013a,b). GLC-based graph sparsification starts with an n -ary factor that captures the information within the elimination clique following marginalization. In short, this n -ary factor is computed by considering all the measurements contained in the Markov blanket of the node that is to be marginalized. To handle low-rank measurements, this method computes eigenvalue-decomposition of the induced target information, Λ_t , and forms a generic linear observation model for the n -ary factor:

$$\mathbf{z}_{glc} = \mathbf{G}\mathbf{x}_c + \mathbf{w}', \quad (6)$$

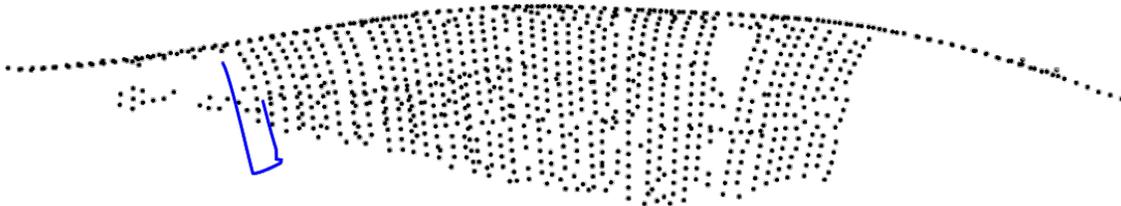
where $\mathbf{w}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{q \times q})$, $\mathbf{G} = \mathbf{D}^{1/2}\mathbf{U}^\top$, $\Lambda_t = \mathbf{U}\mathbf{D}\mathbf{U}^\top$, q is the rank of Λ_t , and \mathbf{x}_c is the current linearization of nodes contained in the elimination clique. $\mathbf{U}\mathbf{D}\mathbf{U}^\top$ is the Eigendecomposition of Λ_t , where \mathbf{U} is a $p \times q$ matrix of Eigenvectors and \mathbf{D} is a $q \times q$ diagonal matrix of Eigenvalues. To preserve sparsity in the graph, the target information Λ_t is approximated using a Chow-Liu Tree (CLT) structure, where the CLT’s unary and binary potentials are represented as GLC factors. These resulting unary and binary factors replace the node’s original surrounding factors.



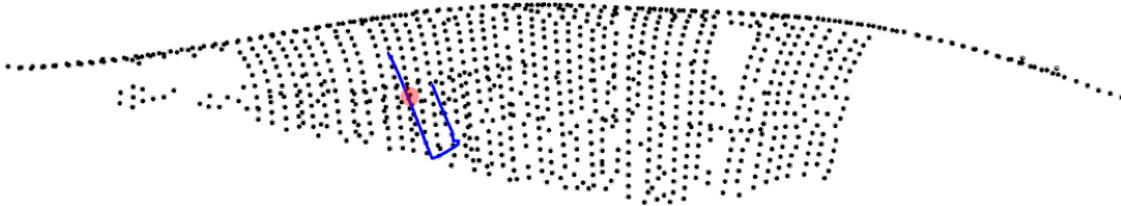
(a) Multi-session SLAM diagram



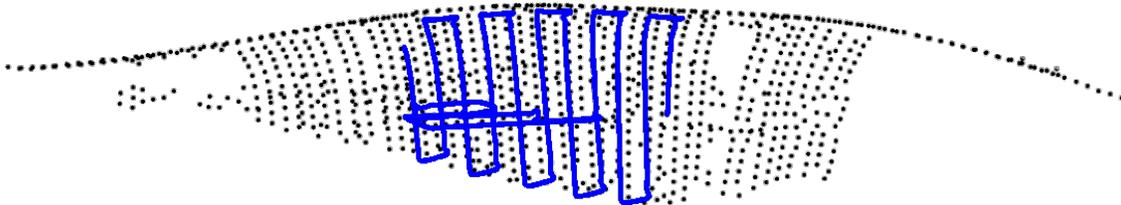
(b) Sparse graph loaded (factors not rendered)



(c) Unaligned pose-graph



(d) Initial alignment to sparse graph



(e) Refine alignment and complete survey

Figure 8: Depiction of multi-session SLAM using the techniques provided in §4. The red region in (d) denotes the point of map reacquisition, where the robot determines a rough initial alignment to the graph from (b). We refine this alignment in real-time using the same techniques as our single-session system from previous work. Once the session from (e) is complete, we sparsify and repeat the process with additional sessions.

GLC optionally supports node reparameterization into a local reference frame around x_c when evaluating the observation model from (6). This avoids committing to a world-frame linearization if the nodes are not well-optimized. Instead, nodes are linearized about a local relative-frame transformation. This relative transformation arbitrarily picks a single pose node in the elimination clique as the common frame. For planar nodes, we use the Ξ operator to express planes with respect to a root node. In the next section, we describe how to support anchor nodes in GLC's reparameterization step.

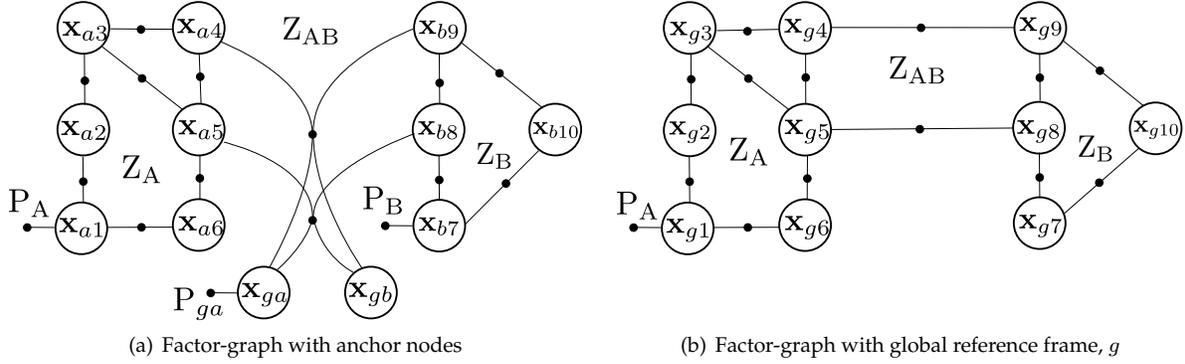


Figure 9: A factor-graph topology for a simple SLAM example involving two sessions, A (left side) and B (right side). In (a), the graph encodes the distribution from (7), where each session has an associated anchor node. This graph can be converted to a global representation in (b) without any loss of information using (9). The only discarded factors are the full-state prior factors in session B and on the anchor node \mathbf{x}_{ga} . These prior factors have little probabilistic importance and function primarily to keep the SLAM information matrix non-singular.

4.2 Global Multi-session SLAM from Anchor Nodes

For a robot or multiple robots performing multi-session SLAM, a pose-graph containing anchor nodes is a popular method to align multiple relative graphs (Kim et al., 2010). In this approach, each robot session has an associated anchor node. An anchor node is a node containing the transformation from a global reference frame to the corresponding session’s reference frame. The important advantages of anchor nodes over global multi-session SLAM, where *all* sessions are expressed in a common frame, are (i) faster convergence of the nonlinear least-squares solver, and (ii) individual sessions can optimize their pose graphs before any constraints between them are observed.

For a two-session case consisting of sessions A and B, the factor-graph containing anchor nodes, from Fig. 9(a), encodes a probability distribution of the form

$$\begin{aligned}
 p(X \mid Z_A, Z_B, Z_{AB}, P_A, P_B, P_{ga}) = \\
 p(\mathbf{x}_{ga}, \mathbf{x}_{a1_A}, \mathbf{x}_{a2_A}, \dots, \mathbf{x}_{aN_A}, \\
 \mathbf{x}_{gb}, \mathbf{x}_{b1_B}, \mathbf{x}_{b2_B}, \dots, \mathbf{x}_{bM_B} \mid Z_A, Z_B, Z_{AB}, P_A, P_B, P_{ga}),
 \end{aligned} \tag{7}$$

where \mathbf{x}_{ga} is an anchor node representing the 6-DOF transformation from the global-frame, g , to the reference frame a of session A. \mathbf{x}_{ai_A} is the 6-DOF relative-pose from frame a to frame i of session A. X denotes the set of variable nodes in the factor graph (i.e., the unknowns). Z_A and Z_B denote the sensor and odometry measurements contained in sessions A and B, respectively, and Z_{AB} denotes the sensor measurements between nodes in sessions A and B. P_A and P_B denote full-state priors in sessions A and B, and P_{ga} denotes the full-state prior on the anchor node, \mathbf{x}_{ga} . Finally, N and M are the number of nodes in sessions A and B, respectively, not including the anchor node (Fig. 9).

The parameterization from (7) is somewhat inconvenient because in order to place the nodes into a common frame (when visualizing the distribution, for instance, or when computing the expected information gain of a measurement between two nodes), a sparse nonlinear function, f , must be applied to the distribution:

$$\mu_{f(X)} = \begin{bmatrix} \mathbf{x}_{ga} \oplus \mathbf{x}_{a1_A} \\ \vdots \\ \mathbf{x}_{ga} \oplus \mathbf{x}_{aN_A} \\ \mathbf{x}_{gb} \oplus \mathbf{x}_{b1_B} \\ \vdots \\ \mathbf{x}_{gb} \oplus \mathbf{x}_{m_B} \end{bmatrix}, \quad (8)$$

where \oplus is defined in Smith et al. (1986) as the compounding operation. The covariance of the resulting distribution is computed to first order as

$$\Sigma_{f(X)} = J_f \Sigma_X J_f^\top,$$

where J_f is the Jacobian of $\mu_{f(X)}$.

Extending this analysis to more than two sessions is trivial. It is also straightforward when X contains plane nodes. In this case, the sparse nonlinear function f contains the \boxplus operator discussed in §3.4.

For convenience, we alter the distribution so that all nodes are in a common frame. Along with simplifying path planning, visualization, and hypothesizing informative measurements, doing so allows for trivial application of the GLC reparameterization suggested by Carlevaris-Bianco and Eustice (2013b). This alteration is illustrated in Fig. 9. First, we reduce the state variables using (8), effectively removing the anchor nodes from the factor-graph. Next, we remove the full-state priors for nodes in session B and the anchor node in session A, leaving us with the distribution

$$\begin{aligned} p(X' \mid Z_A, Z_B, Z_{AB}, P_A) = \\ p(\mathbf{x}_{g1_A}, \mathbf{x}_{g2_A} \dots, \mathbf{x}_{gN_A}, \\ \mathbf{x}_{g1_B}, \mathbf{x}_{g2_B} \dots, \mathbf{x}_{gM_B} \mid Z_A, Z_B, Z_{AB}, P_A), \end{aligned} \quad (9)$$

In this way, none of the sensor measurements are discarded when converting the relative pose-graphs to the global-frame. In practice, we perform this operation immediately after the robot completes a survey, and before performing offline GLC-sparsification.

4.3 Reacquisition to Sparsified Graph by Particle Filtering

This section describes how we accomplish the initial alignment step illustrated in Fig. 8(d). Once a graph has been sparsified using GLC, we use a particle filter to estimate a distribution of likely poses in the reference frame of the past session. Our particle filter is based on a classic Monte-Carlo localization framework. We use odometry, depth, pitch, and roll measurements derived from sensors to propagate particles to their next state. Our particle filter approach is summarized in Fig. 10.

4.3.1 Weighting particles from planar measurements

To weight the particles as new plane measurements are observed, we use a method similar to the computation of potentials used in our factor-graph, described in §3.4. When a plane-fitting measurement, $\mathbf{z}_{\pi_{ik}}$ from (2), is received by the particle filter, it finds the nearest neighboring pose node i according to

$$i = \underset{i \in I_{\Pi}}{\operatorname{argmin}} \|\mathbf{t}_{gp_i}^g - \mathbf{t}_{gi}^g\|,$$

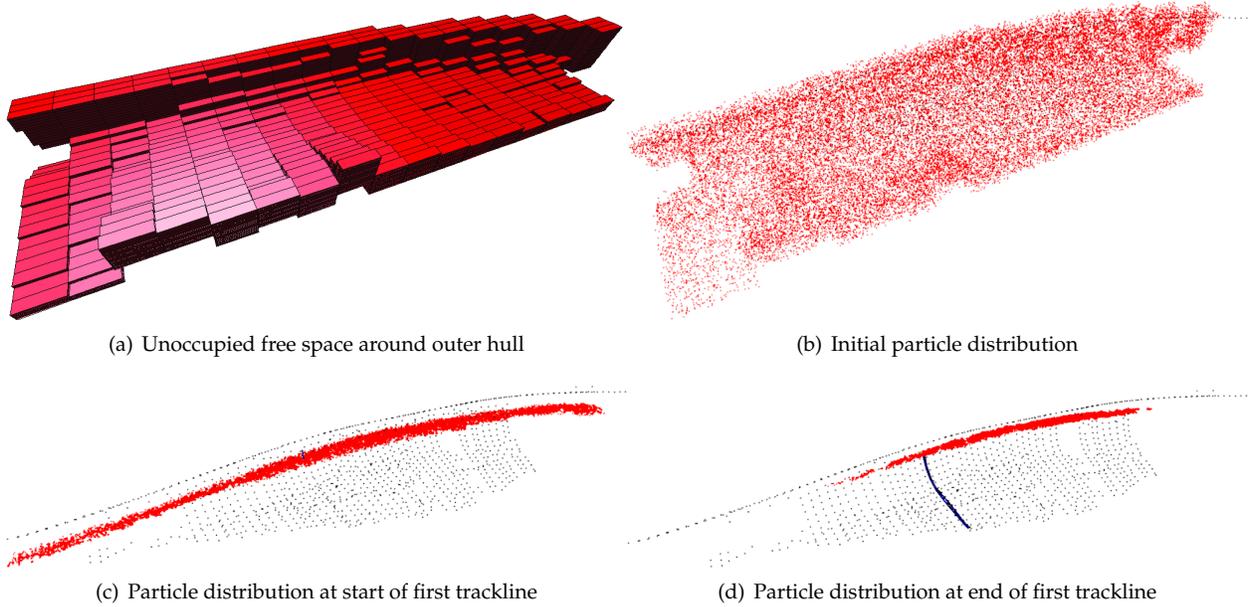


Figure 10: Particle filter reacquisition into a previous session SLAM graph. We improve the efficiency of our particle filter by computing a simple occupancy grid, shown in (a), based upon the planar features of the sparsified graph. Only the grid cells that lie less than 3 m from of surface of the nearest plane may contain particles from the initial distribution. We uniformly distribute particles, shown as red dots in (b), over the cells from (a). When the new survey begins, the particles are weighted based on their compatibility with planar measurements. In (c) and (d), particles are overlaid on the pose-graph where the black dots are graph nodes. Over time, the distribution of particles will tend toward a smaller set of candidate keyframes to search.

where I_{Π} is the set of pose indices in the GLC-sparsified graph that have a corresponding plane. Finding the minimum over all I_{Π} is quite slow for large graphs, so this operation is approximated using a k -dimensional (KD)-tree from the FLANN library (Muja and Lowe, 2009).

Next, we take i' to be the index of the plane that is observed from pose i . Finally, we set the weight, w_{p_i} , by computing the Mahalanobis distance between the observed and expected planar measurement:

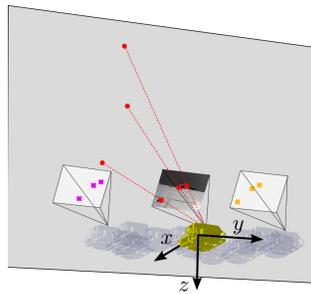
$$w_{p_i} = \|\mathbf{z}_{\pi_{p_i k}} - (\mathbf{x}_{g_i} \boxminus \mathbf{x}_{g_{i'}})\|_{\Sigma_{i i'}}^2.$$

To get the initial distribution of particles, we compute a simple binary 3D occupancy grid from the GLC-sparsified graph, and remove any particles that are assigned to occupied cells. For each cell, if it lies outside the nearest pose-plane pair, that cell is marked as “not occupied.” Otherwise, the cell is marked as “occupied.”

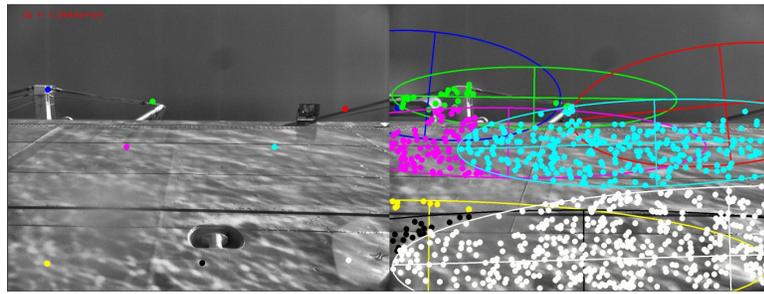
The particle filtering approach is summarized in Fig. 10. We find that the particle distribution tends toward a narrow band at the proper depth. With the addition of planar measurements, this distribution can cull out ends of the ship that do not agree with the structural information observed by the robot. This behavior is shown in Fig. 10(c) and Fig. 10(d).

Algorithm 1 Match current keyframe to candidate keyframes based on particle distribution

- 1: **Input:** Particles $p_1 \dots p_N$, current keyframe k , set of all pose indices in GLC-sparsified graph with corresponding keyframe I_K
 - 2: $S \leftarrow \emptyset$
 - 3: **for** $p_i \in \{p_1 \dots p_N\}$ **do**
 - 4: $S \leftarrow S \cup \text{NEIGHTESTNEIGHBOR}(p_i, I_K)$ ▷ Uses kd-tree
 - 5: **end for**
 - 6: **Output:** $\text{FINDBESTMATCH}(k, S)$ ▷ Uses GPU
-



(a) Planar scene and relative-pose prior for upward-looking periscope



(b) $3\text{-}\sigma$ confidence regions for 2D point transfer from the past keyframe (left) to the candidate keyframe (right).

Figure 11: When searching for putative correspondence, the geometry of the periscope images can be exploited since the variation within relative-poses from the same place is mostly in the y -axis (“side-to-side”). First, we can back-project the points from the past camera pose assuming a planar scene, as shown in (a). Second, we model the relative-pose uncertainty as a Gaussian ellipsoid and project this uncertainty as 2D ellipses in the candidate keyframe’s image plane, shown in (b). We can search for putative SIFT correspondences within these ellipses, rather than searching the entire image.

4.3.2 Planar Scene Prior for Putative Correspondence in Periscope Images

The uncertainty between two matching poses in the initial alignment step can be modeled as a covariance ellipsoid with high variation in the side-to-side axis. The other dimensions are well-instrumented with depth and IMU sensors. This can be probabilistically modeled by assuming a zero-mean transformation between the two candidate poses with an appropriately-constructed transformation covariance matrix.

If we further approximate the periscope scene as purely planar, we can back-project points from the first candidate camera pose onto a known plane in 3D space. We can then re-project these points into the image of the second candidate camera, and propagate the covariance terms using an Unscented Transform (UT) (Julier, 2002; Ozog and Eustice, 2013a). This process is illustrated in Fig. 11, and is a special case of the PCCS described by Eustice et al. (2008). Note that this technique is only useful for the periscope images, where the scene is large as compared to the underwater images. Effectively, this constrains corresponding features in periscope images to share similar pixel row indices. By contrast, correspondences in underwater images could have very different row indices. Therefore, for the underwater case, we revert to a purely putative correspondence search because it runs faster on a graphics processing unit (GPU).

4.3.3 Keyframe Matching with SIFT and RANSAC

After resampling, if the distribution of particles is sufficiently small in the x, y plane, we find the set of all images contained in the particle distribution. For our experiments, we build this set when the square-root of the determinant of the marginal x, y covariance is less than 250 m. This threshold tends to produce a keyframe set that contains about 100 images and takes roughly three to four seconds to perform a

Algorithm 2 Sparsification algorithm

Input: Alignment of two SLAM sessions, $A+B$. A is the pre-sparsified prior session and B is the current session.

Output: Set of all nodes to maginalize, S . Updates exemplar views.

```
1:  $N = \text{SPATIALLYREDUNDANTNODES}(B)$ 
2: for node in  $B$  and not in  $N$  do
3:   views  $\leftarrow \text{NEARESTEXEMPLARCLUSTERORNEW}(A, \text{node})$ 
4:   if  $\text{ANYVIEWCLOSEINTIME}(\text{views}, \text{node})$  or not  $\text{VISUALLYSALIENT}(\text{node})$  then
5:      $S \leftarrow S \cup \text{node}$ 
6:   else
7:      $S \leftarrow S \cup \text{PUSH}(\text{views}, \text{node})$   $\triangleright$  "views" is fixed-size FIFO. If views is full, PUSH returns popped node.
8:   end if
9: end for
10:  $\text{GLCREMOVE}(A+B, S \cup N)$ 
```

brute-force search. This step, denoted as `FINDBESTMATCH` in Algorithm 1, uses a GPU for SIFT descriptor extraction and matching, and finally rejects outliers by using an eight-point RANSAC algorithm to fit a fundamental matrix between the current keyframe and a candidate image extracted from the set. The keyframe with the most inliers above a threshold is selected to be the best match. For our experiments, we choose a threshold of 12 inliers to avoid false positives. If no matches are found, the search is repeated when the robot moves approximately one meter from the point at which the previous search was attempted. Doing so prevents redundant searching.

4.3.4 Maintaining Keyframe Diversity in Merged Graph

Maintaining diversity in the keyframes associated with vehicle pose nodes in the SLAM graph is critically important for the particle filter to successfully localize. To maintain diversity of the images associated with nodes in our merged SLAM graph, we use a simple algorithm that is inspired from the work by Konolige and Bowman (2009). In their approach, a robot maintains a maximum number of recent exemplar views using a simple image closeness measure and a least-recently used cache algorithm. By contrast, our algorithm only adds additional exemplar images if the keyframe is visually salient and sufficiently distant in time to all others from the same neighborhood (lines 3 and 4 in Algorithm 2). This approach produces graphs whose sizes are closely bound to the size of the ship hull and maintains nodes with high utility for multi-session SLAM.

In practice, Algorithm 2 tends to preserve nodes that are both visually salient (for potential camera measurements), and useful for their locally planar structure (for piecewise planar or planar range measurements). Intuitively, this algorithm simply keeps the most recently-used views in a fixed-size neighborhood. An example of the how we maintain image diversity for the graphs used in the experimental results section is shown in Fig. 12.

This algorithm has various user-defined parameters. For the HAUV datasets, we chose a threshold of 1.5 m to decide if a node is spatially redundant (line 1 in Algorithm 2), and the maximum number of views per neighborhood as three (line 7 in Algorithm 2). We used the local saliency metric described by Kim and Eustice (2013) as the method for determining if an image is visually salient.

5 Experimental Trials

This section describes experimental trials with the HAUV performing automated inspections on the *USS Saratoga* and *SS Curtiss* vessels depicted in Fig. 13. We present an extensive multi-session SLAM graph merged on these two vessels over three years. Further analysis on the proposed method is followed in terms of complexity, comparison to other vision-based loop-closure methods, and map accuracy in sparsification.

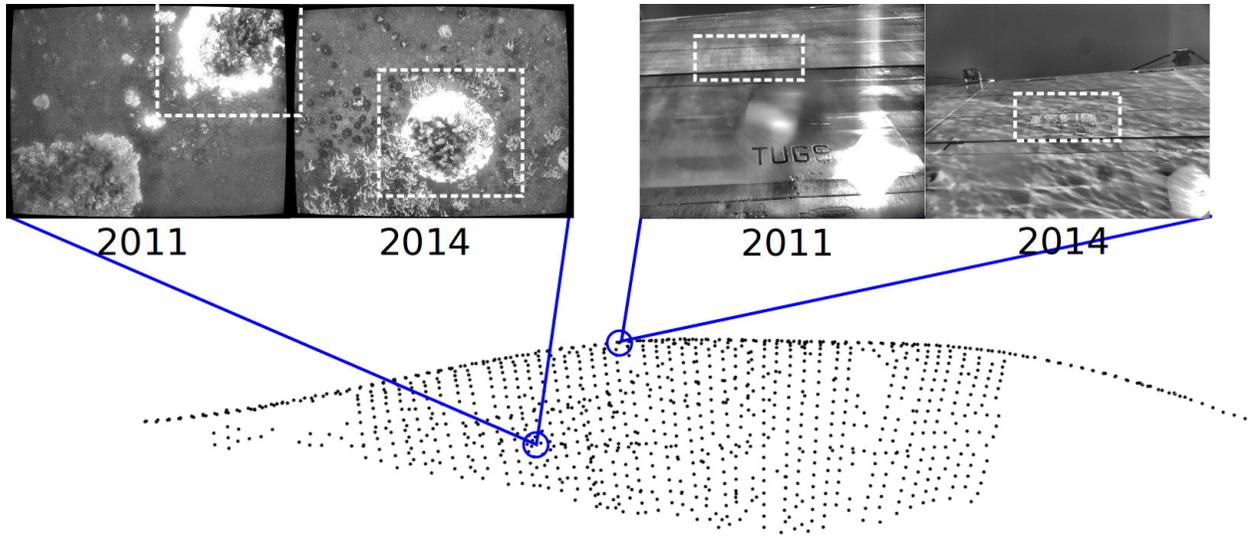


Figure 12: Illustration of exemplar views for the GLC *SS Curtiss* graphs. Two example view neighborhoods are circled in blue. Exemplar views within these neighborhoods (shown above), are noticeably different between 2011 and 2014. We manually outlined common regions in white for illustration. In the left example, different bio-foul patterns are growing on the same circular structure on the hull bottom. In the right example, SIFT features are detected on the pitting from the 2014 data, however, this pitting is not present in the 2011 data.



(a) *USS Saratoga*

	Dim [m]
Length	324
Beam (waterline)	39.6
(extreme)	76.8
Draft	11.3

(b) General characteristics



(c) *SS Curtiss*

	Dim [m]
Length	183
Beam	27
Draft	9.1

(d) General characteristics

Figure 13: The two vessels reported in this work are (a) the *USS Saratoga* and (c) the *SS Curtiss*. General characteristics of the vessels are given in the respective tables.

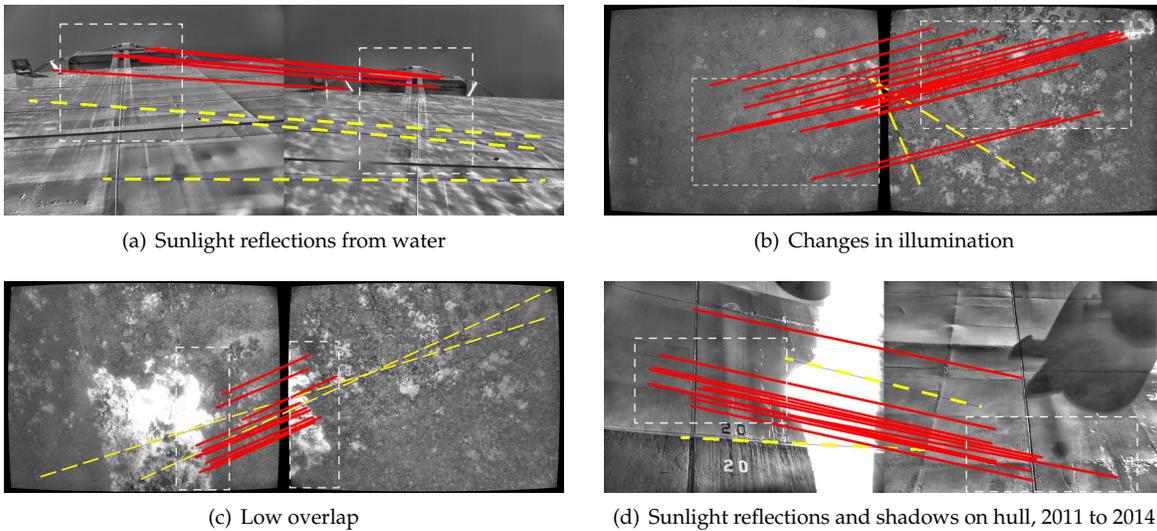


Figure 14: These examples show our hull-relative place recognition system correctly localizing in challenging conditions. The scenarios in (b), (c), and (d) are especially difficult for a human to identify from a set of candidate keyframes. The feature correspondences shown are taken directly from RANSAC. Because purely epipolar-based feature matching across two views is prone to outliers (even with RANSAC), we manually annotated each figure with true inliers (solid red lines) and false inliers (dotted yellow lines). Common regions are denoted with dotted white boxes.

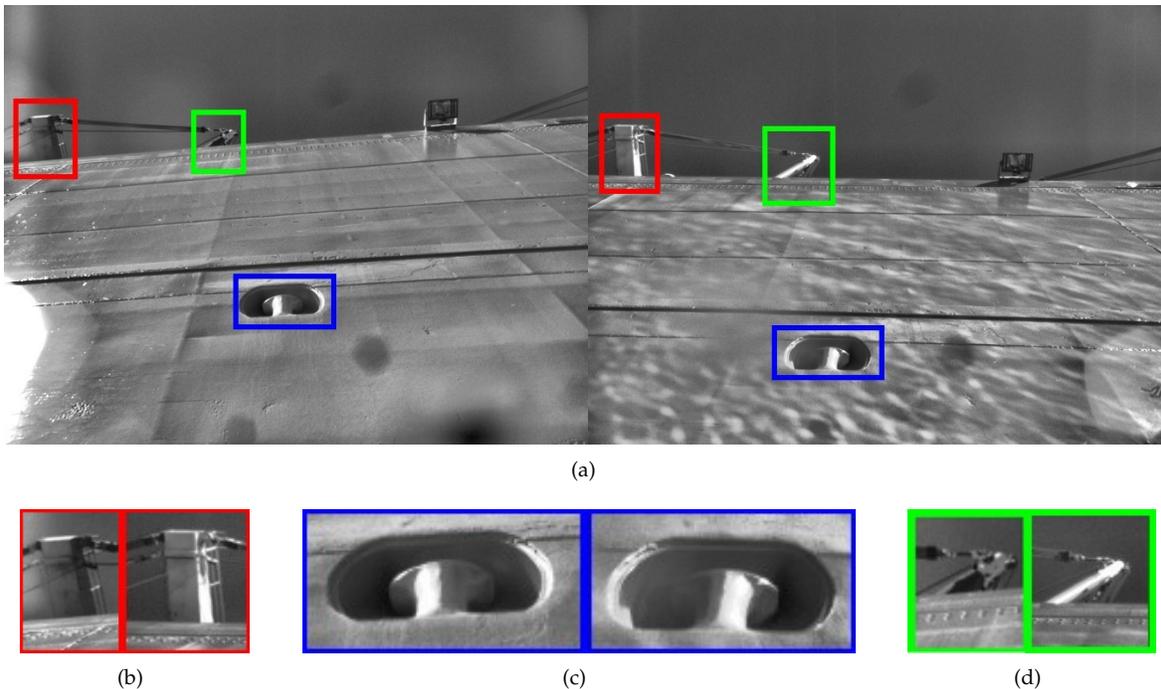


Figure 15: Example of a matching keyframe that our place recognition system fails to detect. We manually colored boxes around key areas to emphasize that the pixel intensity gradients in each patch are very different. The local features within these patches do not match because the SIFT descriptor is not invariant to strong changes in pixel gradient.

Table 1: Summary of HAUV field trials

Session ID	Date	Cameras Used	Survey Type	Trajectory Length (m)
<i>USS Saratoga</i>				
1	May 21, 2013	Periscope	Surface	239
2	May 21, 2013	Periscope	Surface	136
3	May 21, 2013	Periscope + Underwater	Underwater + Surface	334
4	May 22, 2013	Periscope + Underwater	Underwater + Surface	490
5	May 22, 2013	Periscope + Underwater	Underwater + Surface	528
6	May 23, 2013	Periscope + Underwater	Underwater + Surface	493
7	May 23, 2013	Periscope + Underwater	Underwater + Surface	256
8	Aug. 08, 2013	Periscope + Underwater	Underwater + Surface	143
				2619 Total
<i>SS Curtiss</i>				
1	Mar. 08, 2014	Periscope	Surface	341
2	Mar. 08, 2014	Periscope + Underwater	Underwater + Surface	367
3	Mar. 08, 2014	Periscope + Underwater	Underwater + Surface	768
4	Mar. 08, 2014	Periscope + Underwater	Underwater + Surface	719
5	Mar. 08, 2014	Periscope + Underwater	Underwater + Surface	497
6	Mar. 09, 2014	Periscope + Underwater	Underwater + Surface	587
7	Mar. 10, 2014	Periscope + Underwater	Underwater + Surface	449
8	Mar. 11, 2014	Periscope + Underwater	Underwater + Surface	934
9	Feb. 09, 2011	Periscope	Surface	148
10	Feb. 09, 2011	Periscope	Underwater + Surface	1125
11	Feb. 05, 2011	Underwater	Underwater	426
12	Feb. 05, 2011	Underwater	Underwater	1179
				7540 Total

5.1 Experimental Setup

The vehicle has two cameras: an underwater camera, which is actuated to always point nadir to the ship hull, and the periscope camera, which is rigidly attached to the top of the vehicle so as to capture images of the ship superstructure when the vehicle is at the surface. Table 1 gives an overview of how these cameras were used in each dataset, along with the survey date. Note that our system does not require that the time sequence of surveys be causal. For instance, we can build a map of the ship hull from 2014 before co-registering surveys from 2011.

The HAUV executes a survey using one or both of the underwater and periscope cameras. When both are used, the vehicle switches between the two as the vehicle approaches the surface, and as it submerges again for an additional trackline.

5.2 Multi-session SLAM Results

We performed the proposed multi-session SLAM on two vessels, *USS Saratoga* and *SS Curtiss*. In particular, the *SS Curtiss* was surveyed over a three year period as shown in Table 1. While the current mission executes, it is localized to a GLC-sparsified graph using Algorithm 1. Examples of matches using this approach are shown in Fig. 14. Though the algorithm shows some resilience to lighting, sunlight, and shadows, this is not always the case (see Fig. 15 for one such example). In total, eight missions on *USS Saratoga* and twelve missions on *SS Curtiss* are merged as in Fig. 16. These missions are accumulated into a common hull-relative reference frame, and sparsified after each session. The end result is a far more extensive map of each ship hull, as compared to the single session case.

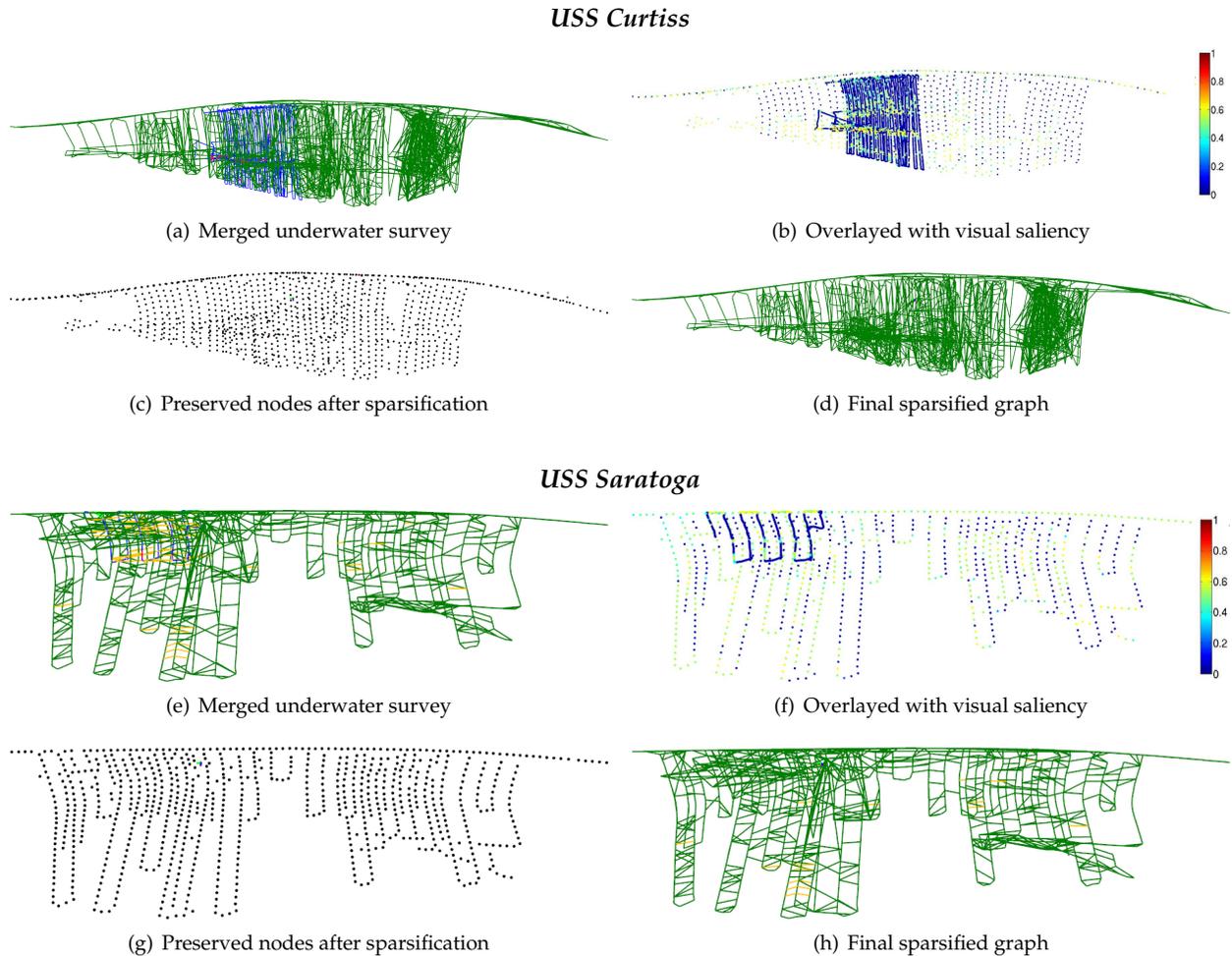


Figure 16: Example of underwater surveys that are automatically aligned with a GLC-sparsified graph. The top and bottom halves show examples from the *SS Curtiss* and the *USS Saratoga*, respectively. In these figures, we disable rendering the planar patches and factors for the sake of visual clarity. In (b) and (f), we overlay the nodes with their local saliency score to show that low scores (in blue) will be marginalized while nodes with high scores (shown in yellow), will be kept so as to provide a more visually informative set of nodes for future localization (see Algorithm 2). Despite the increased spatial extent of the green GLCs, the graphs in (d) and (h) have significantly fewer edges than the graphs in (a) and (e), as also shown in Fig. 17.

5.3 Graph Complexity Over Time

For our evaluation, we take the graph complexity to be the total number of variable nodes and factor nodes in the SLAM graph. The graph complexities over time using our framework for each vessel are shown in Fig. 17 over each successive session. By the end of the third session for each vessel, we lose the ability for real-time performance unless we apply GLC between sessions. Conversely, the use of our method for graph sparsification preserves real-time performance over the course of many more sessions.

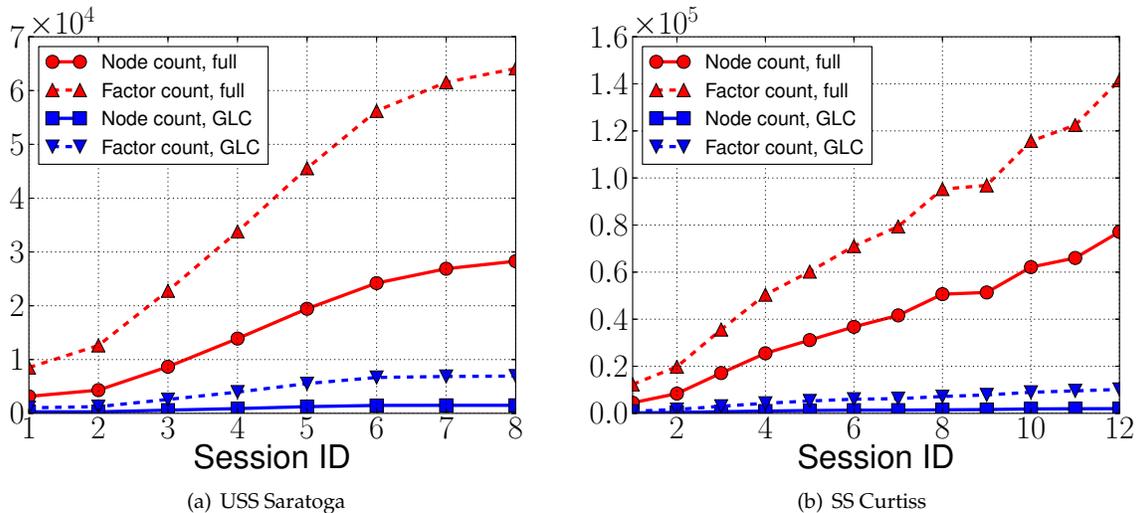


Figure 17: Graph complexity over multiple sessions for the *USS Saratoga*, in (a), and the *SS Curtiss*, in (b). Without managing the graph complexity, the number of nodes and factors grows unbounded with time. However, the size of the graph using the GLC framework is much more closely bounded to the size of the ship hull. For the *USS Saratoga*, sessions 1 through 7 occur within a week of each other, but session 8 occurs four months later. For the *SS Curtiss*, sessions 5 through 12 occur approximately three years after sessions 1 through 4.

Algorithm 3 Match current keyframe to feasible keyframes from visual place-recognition system

- 1: **Input:** Set of all keyframes, K , in GLC-sparsified graph, current keyframe k , belief threshold τ , ship-specific vocabulary, V
 - 2: $S \leftarrow \text{FAB-MAPV2}(k, K, V, \tau)$
 - 3: **Output:** $\text{FINDBESTMATCH}(k, S)$ ▷ Uses GPU
-

5.4 Comparison to Bag-of-Words Place Recognition

To baseline the performance of our particle filter, we use an open-source implementation of FAB-MAP version 2.0 (Glover et al., 2012) as an appearance-only method for place recognition, which represents each keyframe as a visual BoW using a vocabulary that is learned offline. This algorithm is summarized in Algorithm 3, and can be used in place of Algorithm 1 for initial alignment. FAB-MAP provides the belief that the live image is taken from the same location as a place in a prior map. If a match is detected with significant probability, a threshold is used to determine if the SLAM front-end should accept the match. This threshold is typically set very high to avoid false-positives, but we use a very low threshold of 0.0001 for our application to ensure that FAB-MAP returns as many candidates as possible. These candidates are geometrically verified using RANSAC in the final step to reject outliers.

For FAB-MAP, we learn a separate SIFT vocabulary and CLT over the distribution of codewords for each vessel. In practice, this is impractical because it requires a training stage before multi-session SLAM can be attempted. Even so, based upon our experiments, FAB-MAP’s performance is acceptable when matching images of the ship’s above-water superstructure, like the ones shown in Fig. 14(a) or Fig. 18, top row. However, for the underwater images, FAB-MAP performs poorly, and we were not able to successfully identify an underwater loop-closure in any of our experiments, even with a very low loop-closure probability threshold. For underwater images, FAB-MAP consistently identifies the live underwater image as being taken from an unseen place. Two typical cases for the periscope and underwater cameras are shown visually in Fig. 18.

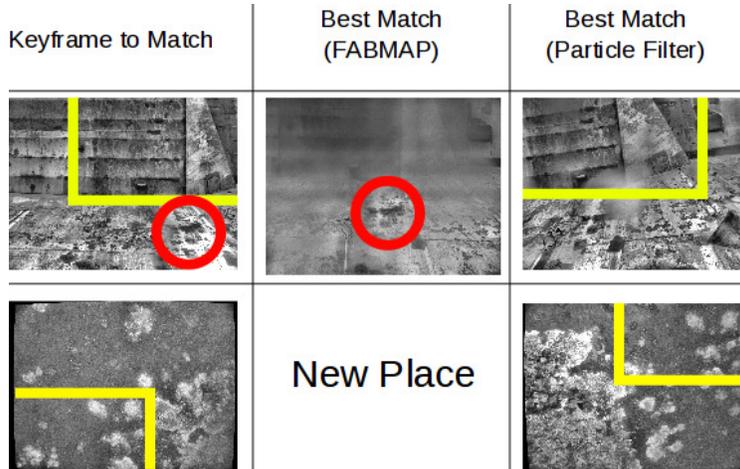


Figure 18: Two representative attempts at aligning the current survey into past sessions using FAB-MAP and our particle filter. Corresponding image regions, where identified, are highlighted. We allow FAB-MAP to return as many candidates as possible by setting the loop-closure probability threshold low. By contrast, the number of candidate matches identified by our particle filter depends on how discriminative the planar measurements are. Because the image matching algorithm is done on the GPU, each candidate match takes roughly 70 ms when using a consumer-grade GPU. FAB-MAP performs adequately for surveys consisting of periscope images (top row) but fails using underwater images (bottom row), where it consistently assigns the “new place” label.

Table 2: Time to localize to past GLC graph

Session	Time until alignment (sec)	
	FAB-MAP	Particle Filter Search
<i>USS Saratoga</i> 2013 Session 7, starting from surface	7.1	5.2
<i>USS Saratoga</i> 2013 Session 7, starting underwater	143.2	20.6
<i>SS Curtiss</i> 2011 Session 12, underwater-only	N/A	15.3

Where our method excels over FAB-MAP is the ability re-localize into a previous session while underwater. We consider three scenarios: *i*) starting the survey with the robot at the surface, *ii*) starting submerged, and *iii*) a survey conducted entirely underwater. The results summarized in Table 2 are as follows: both methods are comparable when at the surface (first row), but FAB-MAP is unable to re-localize while starting underwater, and only when the robot surfaces can it find a good match (second row). For an entirely underwater survey (third row), FAB-MAP is unable to localize. As a whole, our system, which is tailored for the sensor payload of the HAUV, can more quickly and reliably localize to our long-term SLAM graphs.

Using either the particle filter or FAB-MAP, we strive to keep the recall of candidate images as high as possible. Reasonably low precision is not a primary concern because we can geometrically verify candidate matches relatively quickly. If we compute the precision and recall of the set S from particle filtering (Algorithm 1) and FAB-MAP (Algorithm 3), we find that the particle filter produces a set of candidate images with a lower average precision of 3.0%, but high average recall of 100.0%. FAB-MAP, on the other hand, produces a higher average precision of 12.1%, but a much lower average recall of only 27.7%. In other words, the particle filter always contains the correct place within the candidate set, but requires geometrically verifying more keyframes. With FAB-MAP, the candidate set is smaller and therefore faster to process, however, localization will often fail because a true match is not contained in the set of candidate images.

5.5 Comparison to Full Graph

To assess the accuracy of the GLC-based marginalization tool, we tracked the graph produced by the repeated application of sparse-approximate GLC after each session is completed. This graph was directly compared to a synthetically-created full graph, with no marginalized nodes, that contains the exact set of measurements that the HAUV accumulated over the course of each session. For the *SS Curtiss*, we show the full graph over all twelve sessions reported in this paper in Fig. 19.

These comparisons are shown in Fig. 20 for the *USS Saratoga* and in Fig. 21 for the *SS Curtiss*. Each measure is computed only for the vehicle pose nodes for sake of clarity. For each session, we compute the following: (i) marginal Kullback-Leibler Divergence (KLD) over each node (a measure of the similarity between two probability distributions), (ii) absolute positional distance between each corresponding node, (iii) absolute attitude difference between each node, defined as the l_2 -norm of the difference of the poses' Euler angles, and (iv) ratio of the determinants of marginal covariances of each node. For the KLD calculation, we use the well-known closed-form expression for KLD between two Gaussian distributions (Kullback and Leibler, 1951).

As a whole, the accuracy of our system is quite comparable to the full graph, showing typical errors well within a meter despite a total path length of multiple kilometers. These errors are absolute, however, and they do not take into account the strong correlations between neighborhoods of nodes. In addition, as shown in Fig. 22, the marginal ellipses of the approximated poses are consistent with the full graph. Though repeated sparsification may eventually result in inconsistent estimates, recent works include Carlevaris-Bianco and Eustice (2013b, 2014), which can guarantee that the GLC approximation is conservative to counter these inconsistencies. Seeing how these methods affect the success rate of data association is an interesting topic that remains to be explored. For the long-term datasets covered here, inconsistent approximations were not an issue; we were able to successfully establish camera measurements across multiple sessions throughout the field trials.

5.6 Importance of Planar Constraints and Comparison to CAD Model

We provide some qualitative results in Fig. 23 to show that the effectiveness of the method depends strongly on the use of planar constraints. In this figure, we disabled both the piecewise-planar factor potential from (3) and the planar range factors from (4). To keep the comparison fair, we provided the same particle filter localizations used in Fig. 19.

Additionally, we used the ground truth CAD model to assess the quality of the SLAM trajectory, with and without planar constraints. We converted the DVL range returns into a global-frame point cloud using the estimated SLAM poses and down-sampled them with voxel grid filter. Then, we rigidly aligned these points to the CAD mesh with generalized iterative closest point (GICP) (Segal et al., 2009). Finally, for each point in the DVL point cloud we found the nearest vertex in the CAD mesh and computed the Euclidean distance.

This method serves as a sufficient proxy for the structural consistency of the SLAM estimate as compared to ground truth. The results of this comparison are shown in Fig. 24. For the SLAM estimate with no planar constraints, the error distribution had a mean of 1.31 m and standard deviation of 1.38 m. Furthermore, 20% of the DVL points had an error of more than 1.5 m.

Conversely, the use of planar constraints brought the mean and standard deviation to 0.45 m and 0.19 m, respectfully. In addition, there were no points that had an error of more than 1.5 m. With these results, we conclude that camera constraints alone are not sufficient for long-term hull inspection, and we have shown that the results are greatly improved if the ship hull surface itself is included in the SLAM pipeline.

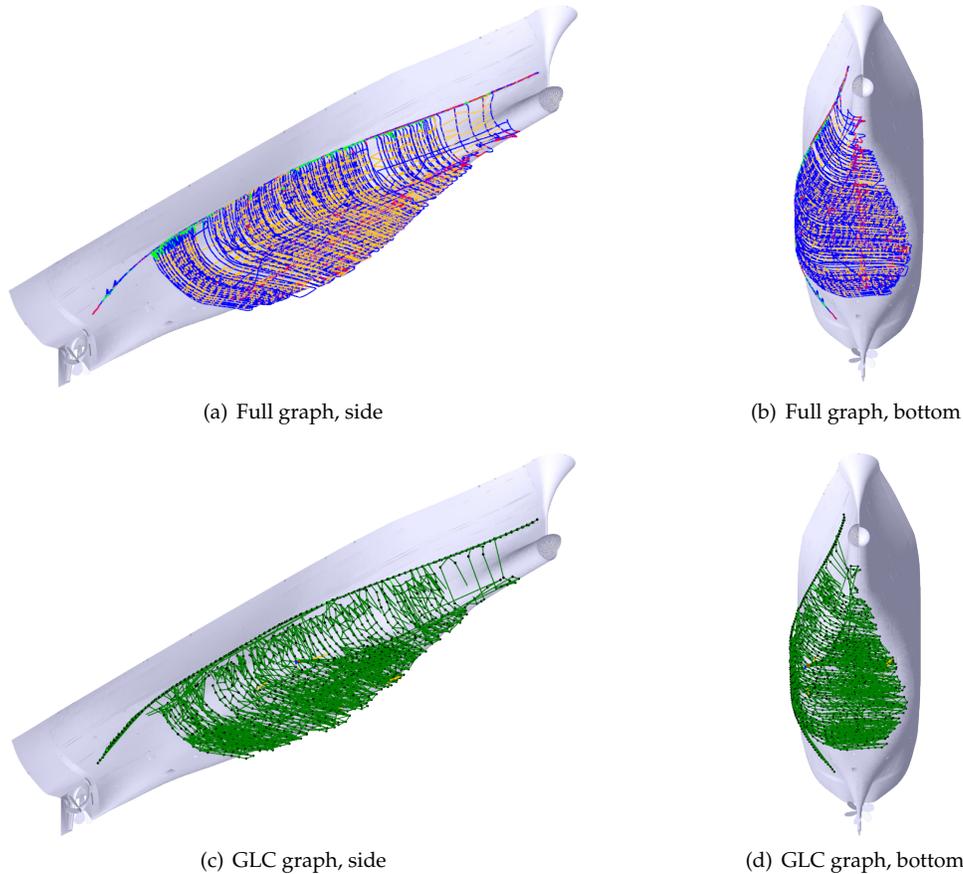


Figure 19: The full, unsparsified graph of the *SS Curtiss* after twelve automatically-aligned sessions that spans from February, 2011 to March, 2014. The CAD model is aligned to provide visual clarity and size reference — it is not used in the SLAM system. The non-sparsified factor graphs in (a) and (b) consist of odometry factors (blue), camera links (red), piecewise-planar factors from (3) (yellow), and planar range factors from (5) (lime green). The sparsified graphs in (c) and (d) consist mostly of GLC factors (green). These graphs serve as the comparison for the results in Fig. 21 for `SessionID = 12`.

6 Conclusion

We provided an overview of how to adapt our visual SLAM algorithm for long-term use on large ship hulls. We use the GLC framework to remove redundant or unneeded nodes from a factor graph. Doing so involves supporting a node reparameterization (root-shift) operation to avoid unnecessary error induced by world-frame linearization. Furthermore, we described a particle filtering algorithm that can use planar surface measurements to narrow a search space over past images that match the current image.

We showed results from our localization algorithm automatically aligning SLAM sessions separated in time by days, months, and years. Once sessions were aligned to a past graph, the result was sparsified and the process was repeated. Using simple sparsification criteria, we show that the complexity of our factor graphs remain more closely bounded to the ship hull area, rather than growing unbounded with time. Furthermore, despite the repeated applications of graph sparsification with GLC, the errors between the sparsified and non-sparsified graphs are reasonably small.

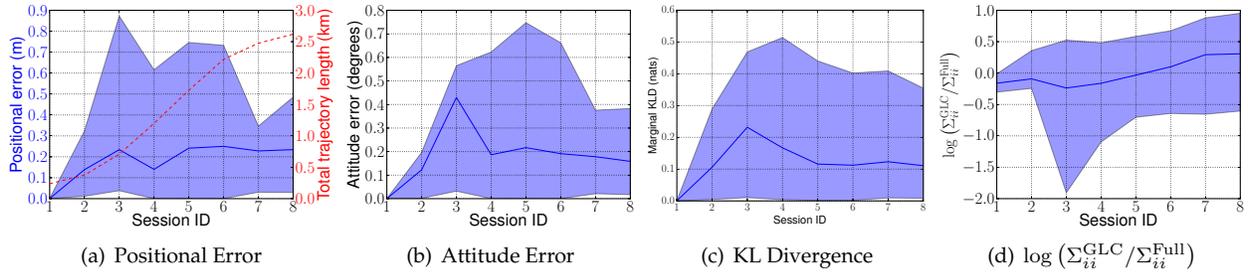


Figure 20: Positional error and trajectory length (a), attitude error (b), average KLD (c), and log-ratio of marginal covariances (d) as a timeseries for the 2014 *USS Saratoga* datasets. The average value is shown as solid blue line. Percentile bounds are shown in the shaded region, from 5% to 95%.

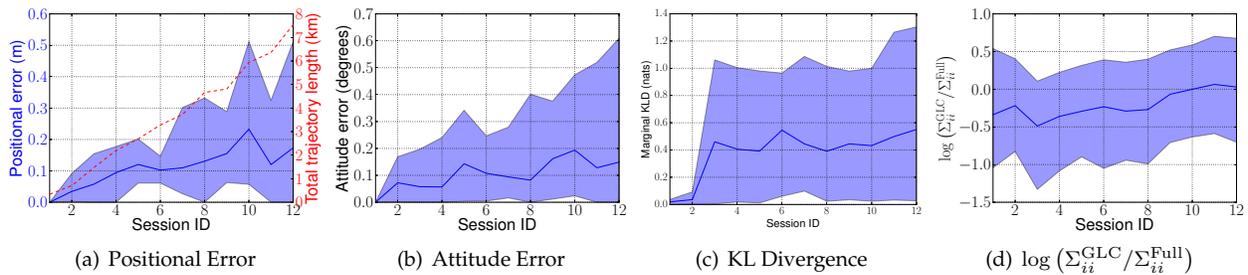


Figure 21: Comparisons to the full graph for each session for the *SS Curtiss* datasets. We show positional error (a), attitude error (b), average KLD (c), and log-ratio of marginal covariances (d) as a timeseries. The average value is shown as solid blue line. Percentile bounds are shown in the shaded region, from 5% to 95%.

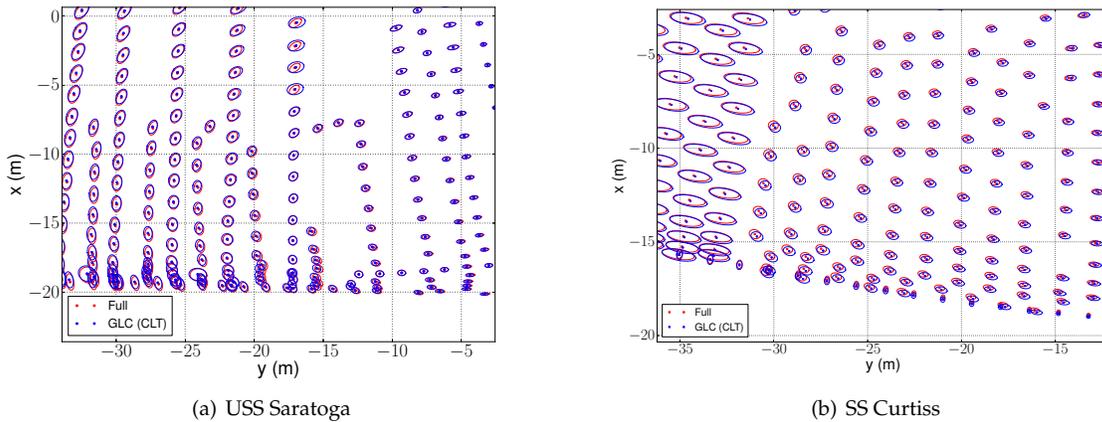


Figure 22: Visualizations of $3\text{-}\sigma$ positional covariances for the full graph are shown in red, and the corresponding ellipses from the GLC-sparsified graph are shown in blue. Our system marginalizes approximately 95% of nodes from each session so that the system can perform in real-time, yet the probabilistic deviation from the full graph is small. The results for the *USS Saratoga* and *SS Curtiss* are shown in (a) and (b), respectively.

Acknowledgments

This work was supported by the Office of Naval Research under award N00014-12-1-0092.

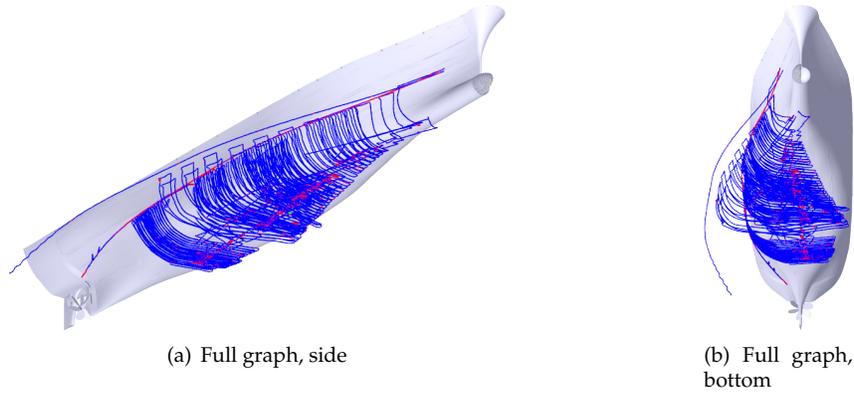


Figure 23: The quality of the SLAM estimate suffers significantly if the planar-based constraints from §3.4.2 and §3.4.3 are left out of the factor graph. These constraints are encoded as yellow and green lines in Fig. 19(a). The CAD model is once again provided for visual reference, and to show that there are obvious discrepancies between it and the SLAM estimate. These discrepancies are quantified in Fig. 24.

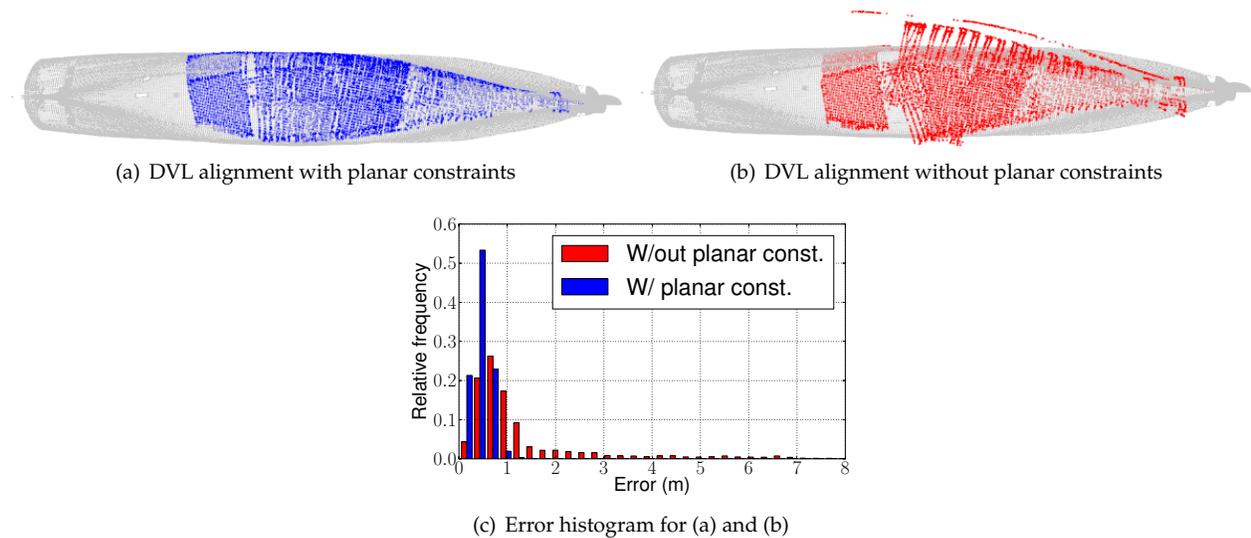


Figure 24: For the *SS Curtiss*, we were able to quantify the error with respect to the ground truth CAD model by registering the DVL returns (red and blue point clouds) to the CAD mesh vertices (gray point cloud). The DVL point cloud in (a) was derived from the SLAM estimate in Fig. 19 (with planar constraints), while the point cloud in (b) was derived from Fig. 23 (without planar constraints). The Euclidean error between corresponding vertices in the CAD model mesh is summarized in (c), where the planar constraints provide a much tighter distribution of error.

A Appendix

A.1 Spherical to Cartesian Coordinates

The characteristic curvature model used in this paper lends itself to surface normals expressed in spherical coordinates. To convert the xyz normal vector to a stacked vector of normal azimuth, a , elevation, e , and magnitude, m , we use the $\text{dir}(\cdot)$ function:

$$\text{dir} \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) = \begin{bmatrix} \text{atan2}(y, x) \\ \text{atan2}\left(z, \sqrt{x^2 + y^2}\right) \\ \sqrt{x^2 + y^2 + z^2} \end{bmatrix}.$$

Similarly, to convert from spherical coordinates back to Cartesian, we use the $\text{trans}(\cdot)$ function, defined as:

$$\text{trans} \left(\begin{bmatrix} a \\ e \\ m \end{bmatrix} \right) = m \begin{bmatrix} \cos(e) \cos(a) \\ \cos(e) \sin(a) \\ \sin(e) \end{bmatrix}.$$

References

- Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., and Burgard, W. (2013). Robust map optimization using dynamic covariance scaling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 62–69, Karlsruhe, Germany.
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, pages 404–417, Graz, Austria. Springer.
- Belcher, E., Matsuyama, B., and Trimble, G. (2001). Object identification with acoustic lenses. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, volume 1, pages 6–11, Kona, HI, USA.
- Bonnín-Pascual, F. and Ortiz, A. (2010). Detection of cracks and corrosion for automated vessels visual inspection. In *Proceedings of the International Conference of the Catalan Association for Artificial Intelligence*, pages 111–120, Tarragona, Spain.
- Boon, B., Brennan, F., Garbatov, Y., Ji, C., Parunov, J., Rahman, T., Rizzo, C., Rouhan, A., Shin, C., and Yamamoto, N. (2009). Condition assessment of aged ships and offshore structures. In *International Ship and Offshore Structures Congress*, volume 2, pages 313–365, Seoul, Korea.
- Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *International Journal of Robotics Research*, 23(12):1113–1139.
- Bosse, M. and Zlot, R. (2008). Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *International Journal of Robotics Research*, 27(6):667–691.
- Bowen, A. D., Yoerger, D. R., Taylor, C., McCabe, R., Howland, J., Gomez-Ibanez, D., Kinsey, J. C., Heintz, M., McDonald, G., Peters, D. B., Bailey, J., Bors, E., Shank, T., Whitcomb, L. L., Martin, S. C., Webster, S. E., Jakuba, M. V., Fletcher, B., Young, C., Buescher, J., Fryer, P., and Hulme, S. (2009). Field trials of the Nereus hybrid underwater robotic vehicle in the challenger deep of the Mariana Trench. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–10, Biloxi, MS, USA.
- Carlevaris-Bianco, N. and Eustice, R. M. (2011). Multi-view registration for feature-poor underwater imagery. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 423–430, Shanghai, China.
- Carlevaris-Bianco, N. and Eustice, R. M. (2013a). Generic factor-based node marginalization and edge sparsification for pose-graph SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5728–5735, Karlsruhe, Germany.

- Carlevaris-Bianco, N. and Eustice, R. M. (2013b). Long-term simultaneous localization and mapping with generic linear constraint node removal. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1034–1041, Tokyo, Japan.
- Carlevaris-Bianco, N. and Eustice, R. M. (2014). Conservative edge sparsification for graph SLAM node removal. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 854–860, Hong Kong, China.
- Carvalho, A., Sagrilo, L., Silva, I., Rebello, J., and Carneval, R. (2003). On the reliability of an automated ultrasonic system for hull inspection in ship-based oil production units. *Applied Ocean Research*, 25(5):235–241.
- Cummins, M. and Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6):647–665.
- Dellaert, F. and Kaess, M. (2006). Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181–1203.
- Eade, E., Fong, P., and Munich, M. (2010). Monocular graph SLAM with complexity reduction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3017–3024, Taipei, Taiwan.
- Eustice, R. M., Pizarro, O., and Singh, H. (2008). Visually augmented navigation for autonomous underwater vehicles. *IEEE Journal of Ocean Engineering*, 33(2):103–122.
- Glover, A., Maddern, W., Warren, M., Reid, S., Milford, M., and Wyeth, G. (2012). OpenFABMAP: An open source toolbox for appearance-based loop closure detection. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4730–4735, St. Paul, USA.
- Grisetti, G., Stachniss, C., Grzonka, S., and Burgard, W. (2007). A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of the Robotics: Science & Systems Conference*, pages 65–72, Atlanta, GA, USA.
- Gustafson, E., Jalving, B., ystein Engelhardttsen, and Burchill, N. (2011). HUGIN 1000 Arctic class AUV. In *The Polar Petroleum Potential Conference & Exhibition*, pages 714–721, Halifax, Nova Scotia.
- Harris, S. and Slate, E. (1999). Lamp ray: Ship hull assessment for value, safety and readiness. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, volume 1, pages 493–500, Seattle, WA, USA.
- Hover, F., Vaganay, J., Elkins, M., Willcox, S., Polidoro, V., Morash, J., Damus, R., and Desset, S. (2007). A vehicle system for autonomous relative survey of in-water ships. *Marine Technology Society Journal*, 41(2):44–55.
- Hover, F. S., Eustice, R. M., Kim, A., Englot, B., Johannsson, H., Kaess, M., and Leonard, J. J. (2012). Advanced perception, navigation and planning for autonomous in-water ship hull inspection. *International Journal of Robotics Research*, 31(12):1445–1464.
- Ishizu, K., Sakagami, N., Ishimaru, K., Shibata, M., Onishi, H., Murakami, S., and Kawamura, S. (2012). Ship hull inspection using a small underwater robot with a mechanical contact mechanism. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–6, Hampton Roads, VA, USA.
- Johannsson, H., Kaess, M., Englot, B., Hover, F., and Leonard, J. J. (2010). Imaging sonar-aided navigation for autonomous underwater harbor surveillance. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4396–4403, Taipei, Taiwan.
- Julier, S. (2002). The scaled unscented transformation. In *Proceedings of the American Control Conference*, volume 6, pages 4555–4559, Anchorage, AK, USA.
- Kaess, M. and Dellaert, F. (2009). Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems*, 57(12):1198–1210.
- Kaess, M., Johannsson, H., Rosen, D., Carlevaris-Bianco, N., and Leonard, J. (2010). Open source implementation of iSAM. <http://people.csail.mit.edu/kaess/isam>.
- Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378.

- Kim, A. and Eustice, R. M. (2009). Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1559–1565, St. Louis, MO, USA.
- Kim, A. and Eustice, R. M. (2013). Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Transactions on Robotics*, 29(3):719–733.
- Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N., and Teller, S. (2010). Multiple relative pose graphs for robust cooperative mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3185–3192, Anchorage, AK, USA.
- Konolige, K. and Bowman, J. (2009). Towards lifelong visual maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1156–1163, St. Louis, MO, USA.
- Kretzschmar, H. and Stachniss, C. (2012). Information-theoretic compression of pose graphs for laser-based SLAM. *International Journal of Robotics Research*, 31(11):1219–1230.
- Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.
- Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g^2o : A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3607–3613, Shanghai, China.
- Kunz, C., Murphy, C., Singh, H., Pontbriand, C., Sohn, R. A., Singh, S., Sato, T., Roman, C., Nakamura, K.-i., Jakuba, M., Eustice, R., Camilli, R., and Bailey, J. (2009). Toward extraplanetary under-ice exploration: Robotic steps in the arctic. *Journal of Field Robotics*, 26(4):411–429.
- Leonard, J. and Feder, H. (1999). A computationally efficient method for large-scale concurrent mapping and localization. In J., H. and Koditschek, D., editors, *Proceedings of the International Symposium on Robotics Research*, Salt Lake City, Utah, USA.
- Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- Menegaldo, L. L., Ferreira, G., Santos, M. F., and Guerato, R. S. (2009). Development and navigation of a mobile robot for floating production storage and offloading ship hull inspection. *IEEE Transactions on Industrial Electronics*, 56(9):3717–3722.
- Menegaldo, L. L., Santos, M., Ferreira, G. A. N., Siqueira, R. G., and Moscato, L. (2008). SIRUS: A mobile robot for floating production storage and offloading (FPSO) ship hull inspection. In *Proceedings of the IEEE International Workshop on Advanced Motion Control*, pages 27–32, Trento, Italy.
- Milne, P. (1983). *Underwater acoustic positioning systems*. Gulf Publishing Company, Houston.
- Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application*, pages 331–340, Lisboa, Portugal.
- Negahdaripour, S. and Firoozfam, P. (2006). An ROV stereovision system for ship-hull inspection. *IEEE Journal of Oceanic Engineering*, 31(3):551–564.
- Neira, J. and Tardos, J. (2001). Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897.
- Ni, K., Steedly, D., and Dellaert, F. (2007). Tectonic SAM: Exact, out-of-core, submap-based SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1678–1685, Roma, Italy.
- Nistér, D. and Stewénius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168.
- Ozog, P. and Eustice, R. M. (2013a). On the importance of modeling camera calibration uncertainty in visual SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3762–3769, Karlsruhe, Germany.
- Ozog, P. and Eustice, R. M. (2013b). Real-time SLAM with piecewise-planar surface models and sparse 3D point clouds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1042–1049, Tokyo, Japan.

- Ozog, P. and Eustice, R. M. (2014). Toward long-term, automated ship hull inspection with visual SLAM, explicit surface optimization, and generic graph-sparsification. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3832–3839, Hong Kong, China.
- Pizarro, O., Eustice, R. M., and Singh, H. (2009). Large area 3-D reconstructions from underwater optical surveys. *IEEE Journal of Oceanic Engineering*, 34(2):150–169.
- Ridao, P., Carreras, M., Ribas, D., and Garcia, R. (2010). Visual inspection of hydroelectric dams using an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):759–778.
- Roman, C. and Singh, H. (2005). Improved vehicle based multibeam bathymetry using sub-maps and SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2422–2429, Edmonton, Alberta, Canada.
- Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-ICP. In *Proceedings of the Robotics: Science & Systems Conference*, Seattle, WA, USA.
- Singh, H., Armstrong, R., Gilbes, F., Eustice, R., Roman, C., Pizarro, O., and Torres, J. (2004). Imaging coral I: Imaging coral habitats with the SeaBED AUV. *The Journal for Subsurface Sensing Technologies and Applications*, 5(1):25–42.
- Smith, R., Self, M., and Cheeseman, P. (1986). Estimating uncertain spatial relationships in robotics. In *Proceedings of Uncertainty in AI*, pages 267–288, Philadelphia, PA, USA. Elsevier.
- Sunderhauf, N. and Protzel, P. (2012). Switchable constraints for robust pose graph SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1879–1884, Algarve, Portugal.
- Thrun, S. and Montemerlo, M. (2006). The graph SLAM algorithm with applications to large-scale mapping of urban structures. *International Journal of Robotics Research*, 25(5-6):403–429.
- Trevor, A. J. B., Rogers, J. G., and Christensen, H. I. (2012). Planar surface SLAM with 3D and 2D sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3041–3048, St. Paul, MN, USA.
- Trimble, G. and Belcher, E. (2002). Ship berthing and hull inspection using the CetusII AUV and MIRIS high-resolution sonar. In *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, volume 2, pages 1172–1175, Biloxi, MS, USA.
- VanMiddlesworth, M., Kaess, M., Hover, F., and Leonard, J. (2013). Mapping 3D underwater environments with smoothed submaps. In *Proceedings of the International Conference on Field and Service Robotics*, pages 17–30, Brisbane, Australia.
- Walter, M., Hover, F., and Leonard, J. (2008). SLAM for ship hull inspection using exactly sparse extended information filters. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1463–1470, Pasadena, CA, USA.
- Weingarten, J. and Siegwart, R. (2006). 3D SLAM using planar segments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3062–3067, Beijing, China.