Pose-Graph SLAM for Underwater Navigation

Stephen M. Chaves, Enric Galceran, Paul Ozog, Jeffrey M. Walls and Ryan M. Eustice

Abstract This chapter reviews the concept of pose-graph simultaneous localization and mapping (SLAM) for underwater navigation. We show that pose-graph SLAM is a generalized framework that can be applied to many diverse underwater navigation problems in marine robotics. We highlight three specific examples as applied in the areas of autonomous ship hull inspection and multi-vehicle cooperative navigation.

1 Introduction

Simultaneous localization and mapping (SLAM) is a fundamental problem in mobile robotics whereby a robot uses its noisy sensors to collect observations of its surroundings in order to estimate a map of the environment while simultaneously localizing itself within that same map. This is a coupled chicken-and-egg state estimation problem, and remarkable progress has been made over the last two decades in the formulation and solution to SLAM [1].

One of the resulting key innovations in the modeling of the SLAM problem has been the use of pose-graphs [2, 3], which provide a useful probabilistic representation of the problem that allows for efficient solutions via nonlinear optimization methods. This chapter provides an introduction to pose-graph SLAM as a unifying framework for underwater navigation. We first present an introduction to the general SLAM problem. Then, we show how challenging SLAM problems stemming from representative marine

Stephen M. Chaves · Paul Ozog · Jeffrey M. Walls · Ryan M. Eustice
University of Michigan, 2600 Draper Dr, Ann Arbor, MI 48105, USA
e-mail: {schaves,paulozog,jmwalls,eustice}@umich.edu

Enric Galceran
ETH Zurich, Leonhardstrasse 21, LEEJ304, 8092 Zürich, Switzerland
e-mail: enricg@ethz.ch

robotics applications can be modeled and solved using these tools. In particular, we present three SLAM systems for underwater navigation: a visual SLAM system using underwater cameras, a system that exploits planarity in ship-hull inspection using sparse Doppler velocity log (DVL) measurements, and a cooperative multi-vehicle localization system. All of these examples showcase the pose-graph as a foundational tool for enabling autonomous underwater robotics.

2 Simultaneous Localization and Mapping (SLAM)

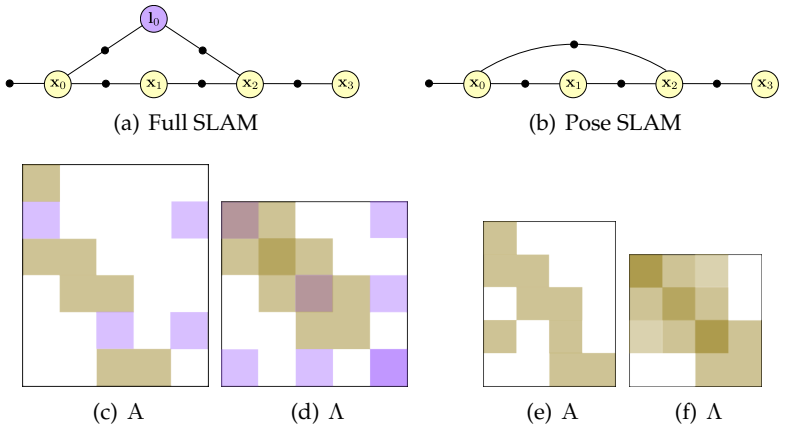
Over the last several decades, robotics researchers have developed probabilistic tools for fusing uncertain sensor data in order to localize within an *a priori* unknown map—the SLAM problem. These tools have reached a level of maturity where they are now widely available [4, 5, 2].

Early approaches to the SLAM problem tracked the most recent robot pose and landmarks throughout the environment in an extended Kalman filter (EKF) [6, 7]. Here, the SLAM estimate is represented as a multivariate Gaussian with mean vector and fully dense covariance matrix. Complexity of the Kalman filter, however, grows with the size of the map, as the measurement updates and memory requirements are quadratic in the state dimension. Thrun et al. [8] observed that the information matrix (inverse of the covariance matrix) of the estimate is approximately sparse, leading to more efficient solutions using an *information* filtering approach that forced sparsity. The information filtering approach features constant time measurement updates and linear memory requirements. Extending the seminal work of Lu and Milios [9], Eustice et al. [10] showed that by considering a delayed-state information filter, the information matrix of the SLAM problem is *exactly* sparse, leveraging the benefits of the information parameterization without sparse approximation errors. Most SLAM systems today formulate the problem in the exactly-sparse sense by optimizing over the entire robot trajectory.

2.1 SLAM Formulation

The *full* SLAM formulation considers optimizing over the entire history of robot poses and landmarks. This problem can be solved using the maximum *a posteriori* (MAP) estimate, given the prior observations of the robot motion and landmarks in the environment:

$$X^*, \mathcal{L}^* = \underset{X,\mathcal{L}}{\arg\max}\, p(X, \mathcal{L}|\mathcal{U}, \mathcal{Z}), \tag{1}$$



(a) Full SLAM (b) Pose SLAM

(c) A (d) Λ (e) A (f) Λ

Fig. 1 Factor graph representations of the full SLAM (a) and pose SLAM (b) formulations. The corresponding measurement Jacobian (A) and information matrix ($\Lambda = A^\top A$) for each system are shown below the factor graphs, with matrix block entries corresponding to poses in yellow and landmarks in purple. In the full SLAM system (a)(c)(d), loop-closures include measurements to landmarks. The columns of the measurement Jacobian A correspond to the following ordering: $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{l}_0\}$. In the pose SLAM system (b)(e)(f), the columns of A correspond to an ordering of $\{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$.

where $\mathbf{x}_i \in X$ are the robot poses, $\mathbf{l}_k \in \mathcal{L}$ are the landmark poses, $\mathbf{u}_i \in \mathcal{U}$ are the control inputs (or motion observations), and $\mathbf{z}_j \in \mathcal{Z}$ are the perceptual observations of map features. The full SLAM formulation is shown in Fig. 1(a) in the form of a factor graph.

Often, when building a SLAM system, the SLAM problem is divided into two sub-problems: (*i*) a "front-end" system that parses sensor measurements to build an optimization problem, and (*ii*) a "back-end" solver that optimizes over robot poses and map features.

The formulation considered in this work is *pose* SLAM (Fig. 1(b)), where there is no explicit representation of landmarks, but rather features observed in the environment are used to construct a relative measurement between robot poses [9]. In this case, the MAP estimate becomes

$$X^* = \underset{X}{\operatorname{argmax}}\, p(X|\mathcal{U}, \mathcal{Z}), \tag{2}$$

and the model of the environment is derived from the robot trajectory itself. This formulation is especially beneficial when the main perceptual sensors are cameras or laser scanners and the environment features are difficult to repeatedly detect or are too numerous to track.

Assuming measurement models with additive Gaussian noise, the optimization of (1) or (2) leads to the following nonlinear least-squares problem:

$$
\begin{aligned}
X^*, \mathcal{L}^* &= \underset{X,\mathcal{L}}{\operatorname{argmax}}\, p(X, \mathcal{L}|\mathcal{U}, \mathcal{Z}) \\
&= \underset{X,\mathcal{L}}{\operatorname{argmin}}\; -\log p(X, \mathcal{L}|\mathcal{U}, \mathcal{Z}) \\
&= \underset{X,\mathcal{L}}{\operatorname{argmin}} \Big\{ \sum_i \|\mathbf{x}_i - f_i(\mathbf{x}_{i-1}, \mathbf{u}_{i-1})\|^2_{\Sigma^i_w} + \sum_j \|\mathbf{z}_j - h_j(\mathbf{x}_{i_j}, \mathbf{l}_{k_j})\|^2_{\Sigma^j_v} \Big],
\end{aligned}
\tag{3}
$$

where f_i and h_j are the measurement models with zero-mean additive Gaussian noise with covariances Σ^i_w and Σ^j_v, and we define $\|\mathbf{e}\|^2_\Sigma = \mathbf{e}^\top \Sigma^{-1} \mathbf{e}$. Linearizing about the current estimate, the problem (3) collapses into a linear least-squares form for the state update vector, solved with the normal equations:

$$
\begin{aligned}
&\underset{\Delta\boldsymbol{\Theta}}{\arg\min} \left\| \mathrm{A}\Delta\boldsymbol{\Theta} - \mathbf{b} \right\|^2, \\
&\Delta\boldsymbol{\Theta} = \left(\mathrm{A}^\top \mathrm{A}\right)^{-1} \mathrm{A}^\top \mathbf{b},
\end{aligned}
\tag{4}
$$

where the vector $\boldsymbol{\Theta}$ includes the poses and landmarks, A is the stacked whitened measurement Jacobian, and $\mathbf{b}$ is the corresponding residual vector. Under the assumption of independent measurements, this formulation leads to an information matrix ($\Lambda = \mathrm{A}^\top \mathrm{A}$) that is inherently sparse, as each observation model depends on only a small subset of poses and landmarks. Thus, modern back-end solvers leverage sparsity patterns to efficiently find solutions.

We solve the nonlinear problem by re-linearizing about the new solution and solving again, repeating until convergence (with Gauss-Newton, for instance). Each linear problem is most commonly solved by direct methods such as Cholesky decomposition of the information matrix or QR factorization of the measurement Jacobian [2, 3]. Aside from direct methods, iterative methods, e.g., relaxation-based techniques [11] and conjugate gradients [12, 13] have also been applied to solve large linear systems in a more memory-efficient and parallelizable way.

2.2 Graphical Representations of SLAM

The SLAM problem introduced above can also be viewed as a probabilistic graphical model known as a *factor graph* (or *pose-graph* in the case of pose SLAM, that is, with no explicit representation of landmarks). A factor graph is a bipartite graph with two types of components: nodes that represent variables to be estimated (poses along the robot's trajectory) and factors that represent constraints over the variables (noisy sensor measurements), as seen in Fig. 1. If each measurement is encoded in a factor, $\Psi_i(\mathbf{x}_i, \mathbf{l}_i)$, where $\mathbf{x}_i$ and $\mathbf{l}_i$ are the robot and landmark poses corresponding to measurement i (and we assume all measurement noise terms are independent), the nonlinear

least-squares problem can be written as

$$X^*, \mathcal{L}^* = \arg\min_{X,\mathcal{L}} \sum_i \Psi_i(\mathbf{x}_i, \mathbf{l}_i), \tag{5}$$

such that the optimization minimizes the sum of squared errors of all the factor potentials. This graphical model view of SLAM is equivalent to the optimization view presented above.

2.3 *Advantages of Graph-based SLAM Methods*

Indeed, recent research in SLAM has turned to graph-based solutions in order to avoid drawbacks associated with filtering-based methods [2, 3]. Notably, EKF-SLAM has quadratic complexity per update, but graph-based methods that parameterize the entire robot trajectory in the information form feature constant-time updates and linear memory requirements. Hence, they are faster on large-scale problems. In addition, unlike filtering-based methods, these optimization-based solutions avoid the commitment to a static linearization point and take advantage of relinearization to better handle nonlinearities in the SLAM problem.

Despite their advantages, nonlinear least-squares SLAM methods present some important challenges. First, since they operate on the information matrix, it is expensive to recover the joint covariances of the estimated variables. Nonetheless, some methods have been developed to improve the speed of joint covariance recovery [14]. Second, since these methods smooth the entire trajectory of the robot, the complexity of the problem grows unbounded over time and peformance degrades as the robot explores. However, the examples presented in this chapter are made possible by online *incremental* graph-based solvers like Incremental Smoothing and Mapping (iSAM) [3] and iSAM2 [15] that leverage smart variable ordering and selective relinearization, and only update the solutions to the parts of the pose-graph that have changed. As we will see in section 3.2.3, the generic linear constraints (GLC) method [16] can additionally be used to compress the representation of the problem and enable tractable operation in large environments over long durations.

Several open source factor graph libraries are available to the community including: Ceres solver [17], iSAM [3], GTSAM [18], and g2o [19].

3 Underwater Pose-Graph Applications

In this section, we outline several representative applications for underwater navigation where the use of pose-graphs has extended the state-of-the-art. Underwater SLAM can take on many forms depending on the sensors available, the operating environment, and the autonomous task to be executed. As we will show, the pose-graph formulation is applicable to many of these forms.

For the remainder of the chapter, let $\mathbf{x}_{ij} = [x_{ij}, y_{ij}, z_{ij}, \phi_{ij}, \theta_{ij}, \psi_{ij}]^\top$ be the 6-degree-of-freedom (DOF) relative-pose of frame j as expressed in frame i, where x, y, z are the Cartesian translation components, and ϕ_{ij}, θ_{ij}, and ψ_{ij} denote the roll (x-axis), pitch (y-axis), and yaw (z-axis) Euler angles, respectively. A pose with a single subscript (e.g., $\mathbf{x}_i$) is expressed with respect to a common local frame.

One foundational sensor that enables underwater SLAM is the Doppler velocity log (DVL), central to all applications presented below. As the robot explores the environment, pose nodes are added to the graph and the dead-reckoned navigation estimate from the DVL is constructed into odometry constraints between consecutive robot poses. In this way, the DVL provides an odometric backbone for various pose SLAM formulations. When available and applicable, absolute prior measurements from, for example, pressure depth, inertial measurement unit (IMU), gyroscope, compass, or GPS can be added to the pose-graph as unary factors.

In the sections that follow, we highlight other factor types derived for specific underwater applications, as well as describe methods centered around pose SLAM that are state-of-the-art in marine autonomy.

3.1 Visual SLAM with Underwater Cameras

Cameras are prevalent perceptual sensors in robotics research because of their low cost but also highly accurate and rich data. Their popularity has led to research in visual SLAM, where measurements derived from the camera are included in the inference[1]. Within the visual pose SLAM formulation, the robot poses in the pose-graph represent discrete image capture events during the underwater mission, and feature-based registrations between overlapping images [22] produce pairwise constraints between the poses. These camera-derived constraints often occur between sequential poses in the graph; however, they can also serve as loop-closure constraints between non-sequential poses when the robot revisits a portion of the environment that it has previously seen, enabling large reductions in its navigation uncertainty.

[1] More background on visual SLAM can be found in [20, 21].

The visual registration process searches for overlapping images within the pose-graph, proposes a camera registration hypothesis given two image candidates, and adds the camera-derived constraint to the graph upon a successful registration. A typical pairwise registration pipeline is shown in Fig. 2 and is described as follows:

1. Given two overlapping images collected by the robot, first undistort each image and enhance with contrast-limited adaptive histogram search (CLAHS) [23].
2. Extract features such as scale-invariant feature transform (SIFT) [24] or speeded up robust features (SURF) [25] from each image.
3. Match features between the images using a nearest-neighbors search in the high-dimensional feature space assisted by pose-constrained correspondence search (PCCS) [20].
4. Fit a projective model among feature matching inliers using a geometric consensus algorithm such as random sample consensus (RANSAC) [26].
5. Perform a two-view bundle adjustment problem to solve for the 5-DOF bearing-only transformation between camera poses and its first-order covariance estimate [27].

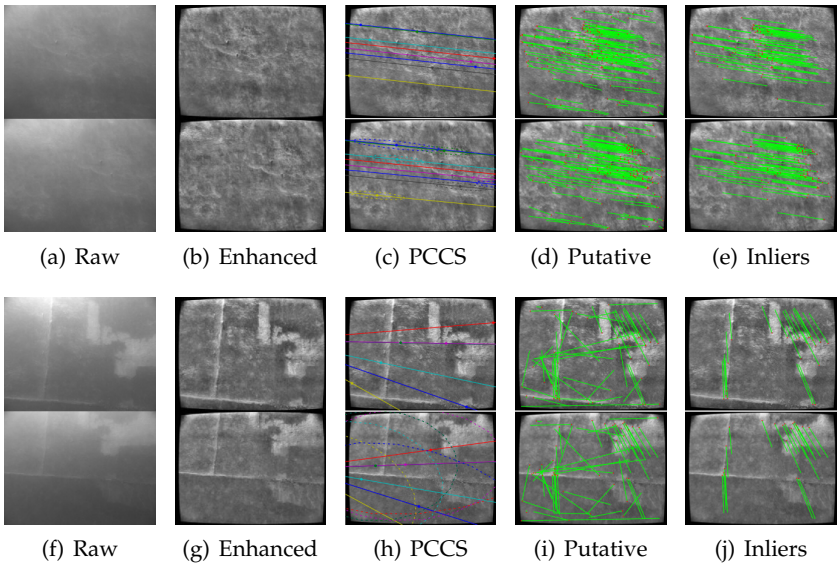
The camera measurement produces a low-rank (modulo scale) relative-pose constraint between two robot poses i and j in the SLAM graph. This measurement, $h^{5\text{dof}}$, therefore has five DOFs: three rotations, and a vector representing the direction of translation that is parameterized by azimuth and elevation angles. We denote the camera measurement as

$$h^{5\text{dof}}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left[\alpha_{ij}, \beta_{ij}, \phi_{ij}, \theta_{ij}, \psi_{ij}\right]^\top, \tag{6}$$

consisting of the baseline direction of motion azimuth α_{ij}, elevation β_{ij}, and the relative Euler angles ϕ_{ij}, θ_{ij}, ψ_{ij}.

3.1.1 Saliency-informed Visual SLAM

The underwater environment is particularly challenging for visual SLAM because it does not always contain visually useful features for camera-derived measurements. In the case of featureless images, the registration pipeline spends much time attempting registrations that are very likely to fail, despite overlap between the image candidates. For autonomous ship hull inspection, Kim and Eustice [21] discovered that the success of camera registrations was correlated to the texture richness, or *visual saliency*, of the corresponding images. In response, they developed two bag-of-words (BoW)-based measures of image registrability, *local saliency* and *global saliency*, to better inform the visual SLAM process.



(a) Raw (b) Enhanced (c) PCCS (d) Putative (e) Inliers

(f) Raw (g) Enhanced (h) PCCS (i) Putative (j) Inliers

Fig. 2 Figures courtesy of Kim and Eustice [21]. Underwater visual SLAM: The pairwise image registration pipeline is shown for two registration hypotheses. The top row shows a feature-poor image set that registers successfully because of strong relative constraints between poses that guide feature-matching via PCCS. The bottom row is also a successful registration, but largely due to the strong features in the images. Steps in the registration pipeline are shown from left to right: (a)(f) Raw overlapping images. (b)(g) Undistorted and enhanced, before extracting features. (c)(h) Feature matching is guided by PCCS. (d)(i) Putative correspondences. (e)(j) Geometric consensus is used to identify inliers. Finally, a two-view bundle adjustment solves for the 5-DOF relative-pose constraint.

In a framework known as *saliency-informed visual SLAM*, Kim and Eustice [21] augmented the SLAM system with the knowledge of visual saliency in order to design a more efficient and robust loop-closure registration process. This system first limits the number of poses added to the graph by only adding poses corresponding to images that pass a local saliency threshold. This thresholding ensures that the graph predominantly contains poses expected to be useful for camera-derived measurements and eliminates poses with a low likelihood of registration. Second, the system orders and proposes loop-closing camera measurement hypotheses according to a measure of saliency-weighted geometric information gain:

$$\mathcal{I}_L^{ij} = \begin{cases} \frac{S_{L_j}}{2} \ln \frac{|\mathrm{R}+\mathrm{H}^{5\mathrm{dof}}\Sigma_{ii,jj}\mathrm{H}^{5\mathrm{dof}\top}|}{|\mathrm{R}|}, & \text{if } S_{L_j} > S_L^{\min} \\ 0, & \text{otherwise} \end{cases}, \tag{7}$$

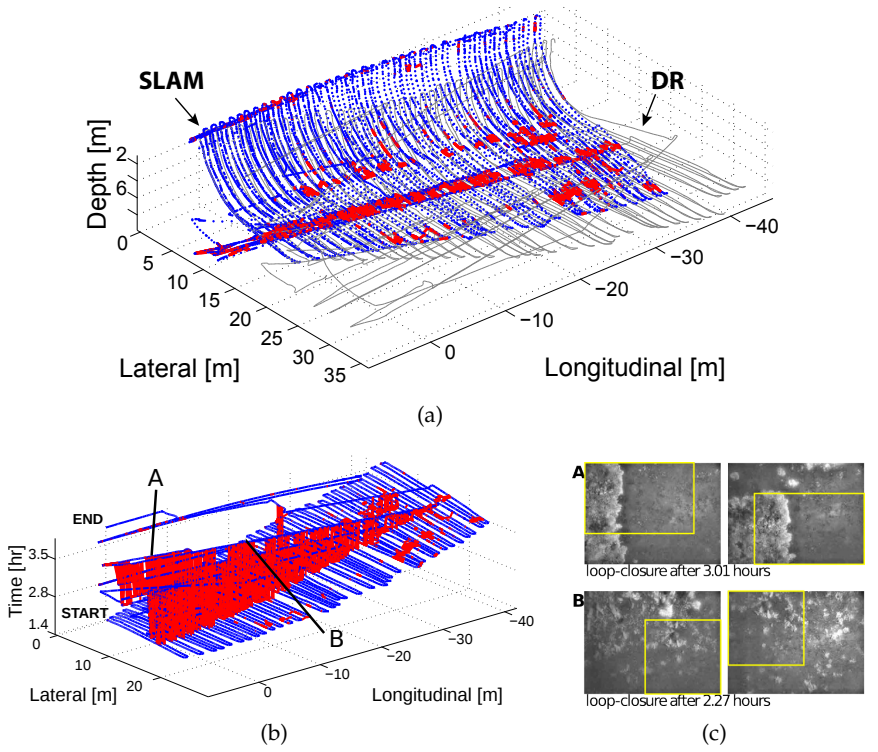
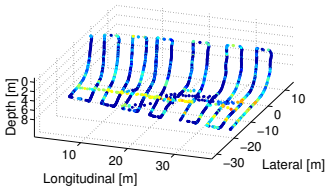


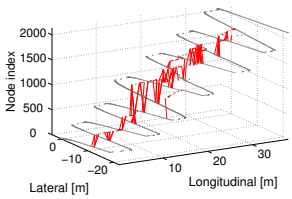
Fig. 3 Figures courtesy of Kim and Eustice [21]. Real-time saliency-informed visual SLAM with the HAUV on the *SS Curtiss*. The pose-graph resulting from SLAM is shown in (a) in blue, with red links representing camera-derived constraints between poses. For comparison, the trajectory estimate from dead-reckoned navigation based on the DVL alone is displayed in gray. The SLAM pose-graph is shown again in (b) with the z-axis scaled by time. In this view, larger red links correspond to larger loop-closures in time. Two loop-closure image pairs are shown in (c).

where R is the five-DOF camera measurement covariance, $\mathrm{H}^{5\mathrm{dof}}$ is the measurement Jacobian of (6), $\Sigma_{ii,jj}$ is the joint marginal covariance of current pose i and target pose j from the current SLAM estimate, S_{L_j} is the local saliency of image j, and $S_L^{\min}$ is the minimum local saliency threshold.

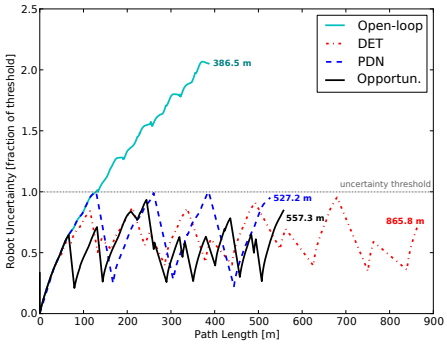
Proposing loop-closures in this way leads to registration hypotheses that both induce significant information in the pose-graph and are likely to be successfully registered, thereby focusing computational resources during SLAM to the most worthwhile loop-closure candidates. The saliency-informed visual SLAM process is shown in Fig. 3 for autonomous ship hull inspection with the HAUV. This result features a total mission time of $3.40\ \mathrm{h}$ and 8,728 poses. The image registration component totalled $0.79\ \mathrm{h}$ and the (cumulative) optimization within iSAM [3] totalled $0.52\ \mathrm{h}$. Thus, the cumulative processing time for this system was $1.31\ \mathrm{h}$, which is 2.6 times faster than real time.



(a) Trajectory – Chaves et al. [29]



(b) Time Elevation



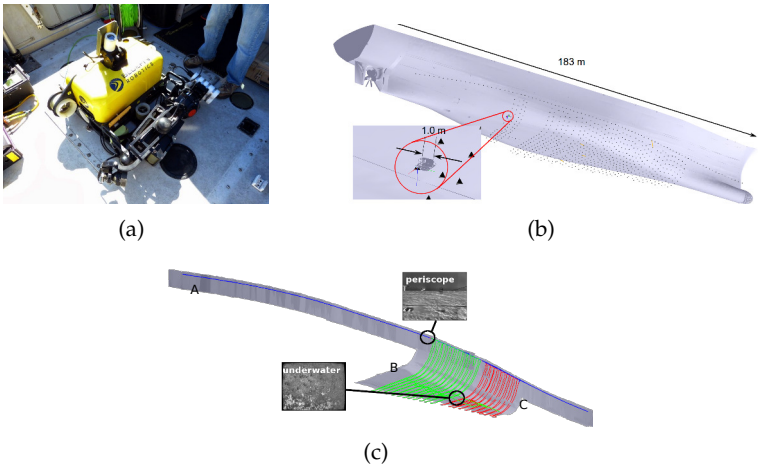
(c) Uncertainty vs. Path Length

Fig. 4 Active visual SLAM results with a hybrid simulation of autonomous ship hull inspection. Shown are four inspection strategies: open-loop coverage survey, a deterministic pre-planned strategy (DET), the PDN method [28], and the opportunistic active SLAM algorithm [29]. The trajectory resulting from the opportunistic approach [29] is shown in (a), with poses color-coded by visual saliency (red=salient, blue=non-salient). A time-elevation plot is shown in (b) with camera-derived constraints in the pose-graph displayed by red links. The uncertainty vs. path length plot of the four strategies in shown in (c). Built on the saliency-informed visual SLAM framework, both the PDN and opportunistic active SLAM methods perform favorably in bounding navigation uncertainty while maintaining efficient area coverage for the inspection task.

3.1.2 Active Visual SLAM

The benefit of saliency-informed visual SLAM can be extended to the *active* SLAM paradigm, where the robot makes decisions about which actions to execute in order to improve the performance of SLAM. Recent works [28, 29] performed belief-space planning for active SLAM with the saliency-informed pose SLAM formulation outlined above.

We can view the active SLAM framework through the lens of the pose-graph by treating each candidate trajectory (or action) as a set of predicted virtual poses and factors that are added to the existing pose-graph built by the robot up to planning time. The robot can then evaluate the solution of this simulated SLAM system within an objective function that quantifies some information-theoretic measure, like navigation uncertainty as described by the covariance matrix. Results from active SLAM methods for autonomous ship hull inspection are given in Fig. 4.



(a) (b)

(c)

Fig. 5 Size comparison of he HAUV and a typically-sized surveyed vessel ((a) and (b)). Using factor graph SLAM, the surface of the ship hull can be estimated as a collection of locally planar patches, shown as gray patches in (c).

3.2 Planar SLAM from Sparse DVL Points

One of the main benefits of factor graphs is the ease of including additional sources of information. In the case of visual SLAM, the vehicle can constrain the estimate of its trajectory with non-visual perceptual data, such as sonar data or acoustic ranges. One interesting source of information is the extraction of coarse perceptual cues using the DVL. In this section, we describe how the DVL can model locally planar environments using a factor graph SLAM back-end[2].

In addition to measuring the velocity of an underwater vehicle, the raw DVL sensor data contains the range of each of the four beams. These three-dimensional (3D) points provide sparse perceptual information that a few researchers have leveraged in prior work, with a particular focus on terrain-aided localization and bathymetric SLAM. Underwater terrain-aided techniques are typically performed with a multibeam sonar, which is much more dense than a DVL. Despite this trend, Eustice et al. [32] and Meduna et al. [33] proposed methods for a vehicle equipped with a DVL to localize with respect to a prior bathymetric map derived from a large surface vessel equipped with a multibeam sonar with high spatial resolution.

More recently, Ozog et al. [31] leveraged this information in the context of automated underwater ship hull inspection with the HAUV, which establishes hull-relative navigation using a DVL pointed nadir to the ship hull surface [34]. In particular, they used the sparse DVL range returns to model a

[2] A more detailed description can be found in [30, 31].

large ship hull as a collection of locally planar features, greatly improving the robustness of long-term underwater visual SLAM across multiple sessions. In this section, we briefly summarize this approach and describe factors necessary for inclusion into a factor graph SLAM back-end.

3.2.1 Pose-to-plane factors

As the HAUV inspects a ship hull, it fits a least-squares 3D plane to a collection of DVL points in the vehicle frame. Suppose this plane, indexed by k and denoted $\boldsymbol{\pi}_k$, is observed with respect to the vehicle at time i. The corresponding observation model for this measurement is

$$\mathbf{z}_{\boldsymbol{\pi}_{ik}} = \mathbf{x}_i \boxminus \boldsymbol{\pi}_k + \mathbf{w}_{ik}, \tag{8}$$

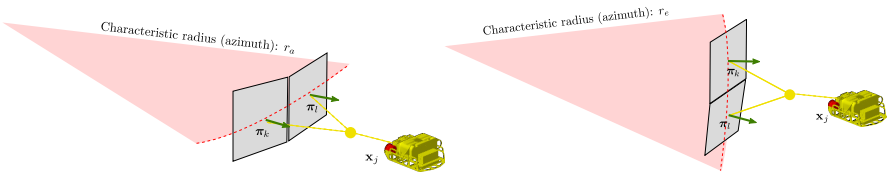
where $\mathbf{x}_i$ is a pose indexed by i, $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{w}_{ik}})$, and $\boxminus$ is a nonlinear function that expresses plane $\boldsymbol{\pi}_k$ with respect to pose frame i. For this section, a plane $\boldsymbol{\pi}^\top = [n_x, n_y, n_z]^\top$ is a 3D column vector consisting of the surface normal of the plane in Euclidean coordinates scaled by the distance of the plane to the local origin.

3.2.2 Piecewise-planar factors

Ship hulls inspected by the HAUV typically exhibit curvature, both in the bow-to-stern and side-to-centerline directions. Therefore, Ozog et al. noted that the HAUV will observe neighboring planar patches that are themselves not co-planar. To address this, they adapted a ternary factor that can constrain two neighboring planes that do not necessarily overlap. The corresponding observation model for a piecewise planar ("pwp") measurement, $\mathbf{z}^{\text{pwp}}_{\boldsymbol{\pi}_{ik}}$, of two neighboring planes, k and l are as follows:

$$\mathbf{z}^{\text{pwp}}_{\boldsymbol{\pi}_{kl}} = (\mathbf{x}_i \boxminus \boldsymbol{\pi}_k) - (\mathbf{x}_i \boxminus \boldsymbol{\pi}_l) + \mathbf{w}^{\text{pwp}}_{kl}, \tag{9}$$

where $\mathbf{w}^{\text{pwp}}_{kl} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{w}^{\text{pwp}}_{kl}})$ is an error term that accounts for the curvature of the ship hull being inspected. By introducing this term, the difference between planes k and l are weighted to give account for them being non-coplanar. In addition, the measurement is conditionally Gaussian by construction and so can be easily incorporated into the factor graph. The curvature model is based on two characteristic radii that are briefly described in Fig. 6.



(a) (b)

Fig. 6 Characteristic radii overview for side-to-side curvature (a) and top-to-bottom curvature (b). These radii account for allowable variations in the surface normals of two neighboring planar patches.

3.2.3 Multi-session SLAM

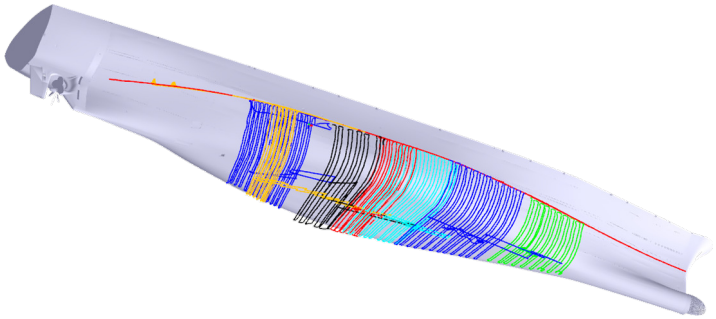
The planar factors described in this section are particularly useful in the context of multi-session SLAM. Ozog et al. showed that these observation models can be incorporated to a visual localization pipeline using a combination of particle filtering and visual SLAM techniques described in Section 3.1. Once localized, the HAUV further adds factors into the SLAM graph using the method inspired from [35]. With this process, multiple sessions can be automatically aligned into a common reference frame in real-time. This pipeline is illustrated in Fig. 7, along with the keyframes used for the visual re-acquisition of the hull.

The HAUV can maintain real-time performance of multi-session SLAM by marginalizing redundant nodes in the pose-graph. Once a pose node is marginalized, however, it induces dense connectivity to other nodes. The GLC framework alleviates this by replacing the target information, Λ_t with a n-ary factor:

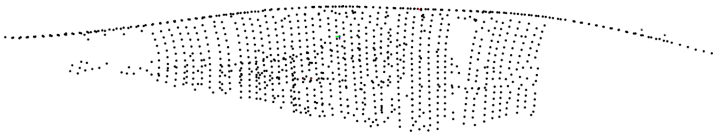
$$\mathbf{z}_{glc} = \mathrm{G}\mathbf{x}_c + \mathbf{w}',$$

where $\mathbf{w}' \sim \mathcal{N}(\mathbf{0}, \mathrm{I}_{q\times q})$, $\mathrm{G} = \mathrm{D}^{1/2}\mathrm{U}^\top$, $\Lambda_t = \mathrm{UDU}^\top$, q is the rank of Λ_t, and $\mathbf{x}_c$ is the current linearization of nodes contained in the elimination clique. $\mathrm{UDU}^\top$ is the Eigendecomposition of Λ_t, where U is a $p \times q$ matrix of Eigenvectors and 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[3]. Thus, GLC serves as an approximate marginalization method for reducing computational complexity of the pose-graph. In the example of Fig. 7, the multi-session pose-graph is reduced from 50,624 to 1,486 nodes.

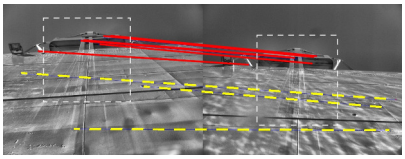
[3] A more detailed description of GLC can be found in [16].



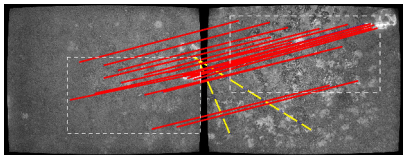
(a) Eight sessions aligned to a common hull-relative frame (each session is shown with a different color). Node count: 50,624.



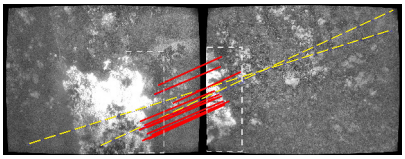
(b) Preserved nodes after GLC sparsification. Node count: 1,486.



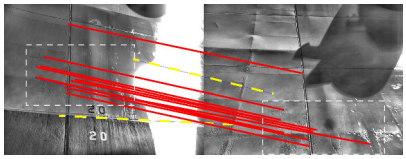
(c) Sunlight reflections from water



(d) Changes in illumination



(e) Low overlap



(f) Sunlight reflections and shadows on hull, 2011 (left) to 2014 (right)

Fig. 7 Planar-based factor potentials and GLC graph sparsification play a key part in the HAUV localization system. This method works in conjunction with the visual SLAM techniques from Section 3.1 to allow for long-term automated ship hull surveillance. Successful instances of localization in particularly challenging hull regions are shown in (c) through (f), with visual feature correspondences shown in red.

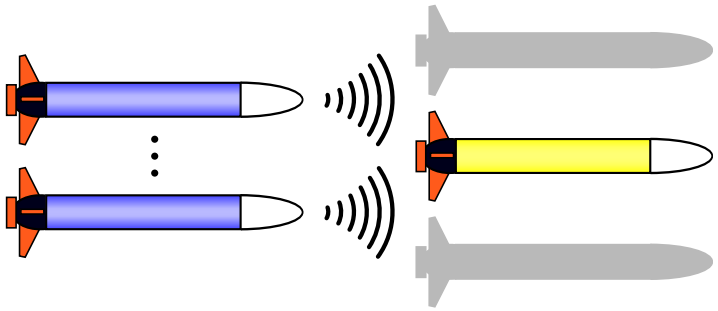


Fig. 8 Cooperative multiple vehicle network. Yellow vehicles (right) benefit from the information shared from blue vehicles (left). Note that communication may be bidirectional.

3.3 Cooperative Localization

Underwater localization with autonomous underwater vehicles (AUVs) in the mid-depth zone is notoriously difficult [36]. For example, both terrain-aided and visually-aided navigation assume that vehicles are within sensing range of the seafloor. Underwater vehicles typically employ acoustic beacon networks, such as narrowband long-baseline (LBL) and ultra-short-baseline (USBL), to obtain accurate bounded-error navigation in this regime. Acoustic beacon methods, however, generally require additional infrastructure and limit vehicle operations to the acoustic footprint of the beacons.

Acoustic modems enable vehicles to both share data and observe their relative range; however, the underwater acoustic channel is unreliable, exhibits low bandwidth, and suffers from high latency (sound is orders of magnitude slower than light) [37]. Despite these challenges, cooperative localization has been effectively implemented among teams of underwater vehicles (Fig. 8). Each vehicle is treated as a mobile acoustic navigation beacon, which requires no additional external infrastructure and is not limited in the range of operations by static beacons. In this section, we show that an effective cooperative localization framework can be built by exploiting the structure of the underlying factor graph[4].

[4] A more detailed description of cooperative localization with factor graph based algorithms appears in [38, 39].

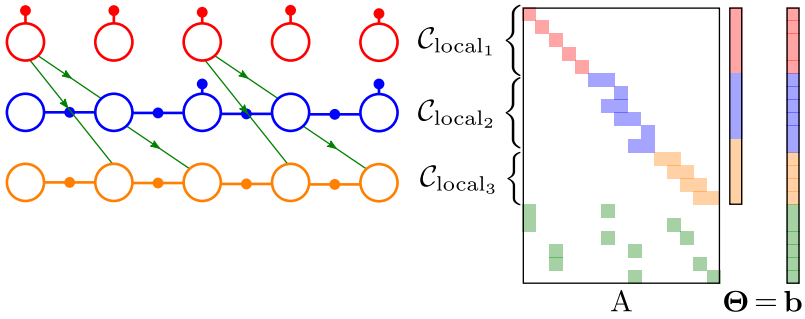


Fig. 9 Example cooperative localization factor graph. Empty circles represent variable pose nodes, solid dots are odometry and prior factors, and arrows illustrate range-only factors and the direction of communication. In this example, red represents a topside ship with access only to GPS, while blue and orange represent AUVs.

3.3.1 Acoustic Range Observation Model

The use of synchronous-clock hardware enables a team of vehicles to observe their relative range via the one-way-travel-time (OWTT) of narrowband acoustic broadcasts [40]. The OWTT relative range is measured between the transmitting vehicle at the time-of-launch (TOL) and the receiving vehicle at the time-of-arrival (TOA). Since ranging is passive—all receiving platforms observe relative range from a single broadcast unlike a two-way ping—OWTT networks scale well.

The OWTT relative range is modeled as the Euclidean distance between the transmitting and receiving vehicles

$$\begin{aligned} z_r &= h_r(\mathbf{x}_{\mathsf{TOL}}, \mathbf{x}_{\mathsf{TOA}}) + w_r \\ &= \left\| x_{\mathsf{TOL}} - x_{\mathsf{TOA}} \right\|_2 + w_r, \end{aligned}$$

where $w_r \sim \mathcal{N}\left(0, \sigma_r^2\right)$ is an additive noise perturbation. Since attitude and depth are typically instrumented with small bounded error, we often project the 3D range measurement into the horizontal plane.

3.3.2 Multiple Vehicle Factor Graph

Representing correlation that develops between individual vehicle estimates as a result of relative observations has been a challenge for cooperative localization algorithms [41, 42]. Factor graphs explicitly represent this correlation by maintaining a distribution over the trajectories of *all* vehicles.

Earlier, we showed the pose SLAM formulation citing a single vehicle (2). We can expand this formulation to represent the posterior distribution of a network of vehicles given relative constraints. Consider, for example, an M vehicle network. The posterior can be factored

$$p(X_1, \ldots, X_M | Z_1, \ldots, Z_M, Z_r) \propto \prod_{i=1}^{M} \underbrace{p(X_i|Z_i)}_{\mathcal{C}_{\text{local}_i}} \prod_k \underbrace{p(\mathbf{z}_k|\mathbf{x}_{i_k}, \mathbf{x}_{j_k})}_{\text{range factors}}, \tag{10}$$

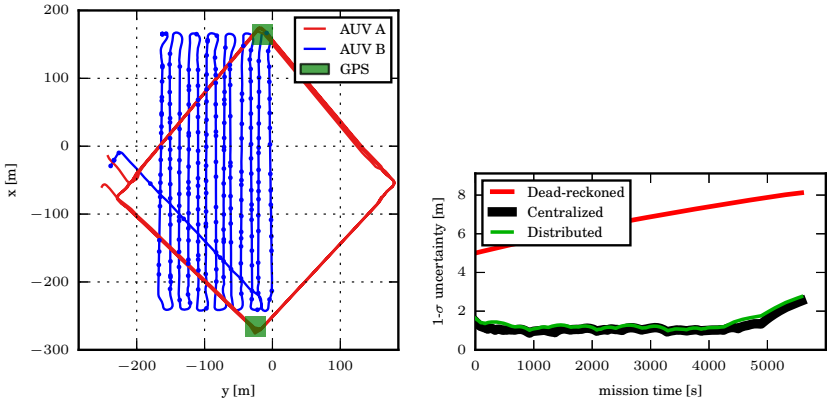
where X_i is the set of ith vehicle poses, Z_i is the set of ith vehicle observations, and Z_r is the set of relative vehicle OWTT range constraints and each $\mathbf{z}_k$ represents a constraint between poses on vehicles i_k and j_k. We see that the posterior factors as a product of each vehicle's local information ($\mathcal{C}_i$) and the set of all relative range observations. Therefore, in order to construct (and perform inference on) the full factor graph, the ith vehicle must have access to the set of local factors from all other vehicles, $\{\mathcal{C}_{\text{local}_j}\}_{j \neq i}$, and the set of all relative vehicle factors. Distributed estimation algorithms can leverage the sparse factor graph structure in order to broadcast information across the vehicle network. The factor graph for a three vehicle network is illustrated in Fig. 9.

Recently, several authors have proposed real time implementations that exploit this property [43, 44, 39]. Fig. 10 illustrates an example of a three vehicle network consisting of two AUVs and a topside ship implementing the algorithm proposed in [39]. AUV-A had intermittent access to GPS during brief surface intervals (highlighted in green). AUV-B remained subsea during the duration of the trial. Fig. 10(b) shows the resulting position uncertainty for AUV-B is bounded and nearly identical to that of the post-process centralized estimator.

Cooperative localization provides a means to improve navigation accuracy by exploiting relative range constraints within networks of vehicles. An architecture based around factor graphs addresses challenges endemic to cooperative localization, provides a sparse representation ideal for low-bandwidth communication networks, and is extensible to new factor types.

4 Conclusion

This chapter has shown how state-of-the-art knowledge about the SLAM problem can enable key marine robotics applications like ship-hull inspection or cooperative navigation. We reviewed the general concept of pose SLAM and its associated mathematical framework based on factor graphs and nonlinear least-squares optimization. We then presented three diverse underwater SLAM systems applied to autonomous inspection and cooperative navigation tasks.



(a) Relative vehicle trajectories (topside not shown).

(b) AUV-B's estimate uncertainty.

Fig. 10 Summary of field trial and performance comparison. ((a)) An XY view of the vehicle trajectories. Blue dots indicate where AUV-B received range observations. ((b)) The smoothed uncertainty in each AUV-B pose as the fourth root of the determinant of the pose marginal covariance.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

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

[4] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[5] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[6] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. Springer-Verlag, 1990, pp. 167–193.

[7] J. J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan, Nov. 1991, pp. 1442–1447.

[8] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *International Journal of Robotics Research*, vol. 23, no. 7–8, pp. 693–716, 2004.

[9] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[10] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, 2006.

[11] T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," *Autonomous Robots*, vol. 12, no. 3, pp. 287–300, 2002.
[12] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with application to large-scale mapping of urban structures," *International Journal of Robotics Research*, vol. 25, no. 5–6, pp. 403–429, 2006.
[13] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C. E. Thorpe, "Subgraph-preconditioned conjugate gradients for large scale SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 2566–2571.
[14] V. Ila, L. Polok, M. Solony, P. Smrz, and P. Zemcik, "Fast covariance recovery in incremental nonlinear least square solvers," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seattle, WA, USA, May 2015, pp. 4636–4643.
[15] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
[16] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, "Generic node removal for factor-graph SLAM," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1371–1385, 2014.
[17] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.
[18] F. Dellaert and Others, "Gtsam," https://borg.cc.gatech.edu/download.
[19] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 3607–3613.
[20] R. M. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation for autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 103–122, 2008.
[21] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.
[22] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
[23] R. Eustice, O. Pizarro, H. Singh, and J. Howland, "UWIT: Underwater image toolbox for optical image processing and mosaicking in Matlab," in *Proceedings of the International Symposium on Underwater Technology*, Tokyo, Japan, Apr. 2002, pp. 141–145.
[24] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[25] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
[27] R. M. Haralick, "Propagating covariance in computer vision," in *Proceedings of the International Conference Pattern Recognition*, vol. 1, Jerusalem, Israel, Oct. 1994, pp. 493–498.
[28] A. Kim and R. M. Eustice, "Active visual SLAM for robotic area coverage: Theory and experiment," *International Journal of Robotics Research*, vol. 34, no. 4–5, pp. 457–475, 2015.
[29] S. M. Chaves, A. Kim, E. Galceran, and R. M. Eustice, "Opportunistic sampling-based active visual SLAM for underwater inspection," *Autonomous Robots*, vol. 40, no. 7, pp. 1245–1265, 2016.
[30] P. Ozog and R. M. Eustice, "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*, Tokyo, Japan, Nov. 2013, pp. 1042–1049.
[31] P. Ozog, N. Carlevaris-Bianco, A. Kim, and R. M. Eustice, "Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle," *Journal of Field Robotics*, vol. 33, no. 3, pp. 265–289, 2016.

[32] R. Eustice, R. Camilli, and H. Singh, "Towards bathymetry-optimized Doppler renavigation for AUVs," in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, Washington, DC, USA, Sep. 2005, pp. 1430–1436.

[33] D. Meduna, S. Rock, and R. McEwen, "Low-cost terrain relative navigation for long-range AUVs," in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, Woods Hole, MA, USA, Oct. 2008, pp. 1–7.

[34] F. S. Hover, J. Vaganay, M. Elkins, S. Willcox, V. Polidoro, J. Morash, R. Damus, and S. Desset, "A vehicle system for autonomous relative survey of in-water ships," *Marine Technology Society Journal*, vol. 41, no. 2, pp. 44–55, 2007.

[35] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010, pp. 3185–3192.

[36] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "Underwater vehicle navigation: Recent advances and new challenges," in *IFAC Conference on Manoeuvring and Control of Marine Craft*, Lisbon, Portugal, Sep. 2006.

[37] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 23–33, 2007.

[38] J. M. Walls and R. M. Eustice, "An origin state method for communication constrained cooperative localization with robustness to packet loss," *International Journal of Robotics Research*, vol. 33, no. 9, pp. 1191–1208, 2014.

[39] J. M. Walls, A. G. Cunningham, and R. M. Eustice, "Cooperative localization by factor composition over faulty low-bandwidth communication channels," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seattle, WA, May 2015.

[40] R. M. Eustice, H. Singh, and L. L. Whitcomb, "Synchronous-clock one-way-travel-time acoustic navigation for underwater vehicles," *Journal of Field Robotics*, vol. 28, no. 1, pp. 121–136, 2011.

[41] S. I. Roumeliotis and G. A. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[42] A. Bahr, M. R. Walter, and J. J. Leonard, "Consistent cooperative localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 3415–3422.

[43] M. F. Fallon, G. Papadopoulis, and J. J. Leonard, "A measurement distribution framework for cooperative navigation using multiple AUVs," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, AK, May 2010, pp. 4256–4263.

[44] L. Paull, M. Seto, and J. J. Leonard, "Decentralized cooperative trajectory estimation for autonomous underwater vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, Sep. 2014, pp. 184–191.