

SVIn2: A multi-sensor fusion-based underwater SLAM system

The International Journal of
Robotics Research
2022, Vol. 41(11-12) 1022–1042
© The Author(s) 2022
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/02783649221110259
journals.sagepub.com/home/ijrr



Sharmin Rahman¹ , Alberto Quattrini Li²  and Ioannis Rekleitis¹

Abstract

This paper presents SVIn2, a novel tightly-coupled keyframe-based Simultaneous Localization and Mapping (SLAM) system, which fuses Scanning Profiling Sonar, Visual, Inertial, and water-pressure information in a non-linear optimization framework for small and large scale challenging underwater environments. The developed real-time system features robust initialization, loop-closing, and relocalization capabilities, which make the system reliable in the presence of haze, blurriness, low light, and lighting variations, typically observed in underwater scenarios. Over the last decade, Visual-Inertial Odometry and SLAM systems have shown excellent performance for mobile robots in indoor and outdoor environments, but often fail underwater due to the inherent difficulties in such environments. Our approach combats the weaknesses of previous approaches by utilizing additional sensors and exploiting their complementary characteristics. In particular, we use (1) acoustic range information for improved reconstruction and localization, thanks to the reliable distance measurement; (2) depth information from water-pressure sensor for robust initialization, refining the scale, and assisting to limit the drift in the tightly-coupled integration. The developed software—made open source—has been successfully used to test and validate the proposed system in both benchmark datasets and numerous real world underwater scenarios, including datasets collected with a custom-made underwater sensor suite and an autonomous underwater vehicle Aqua2. SVIn2 demonstrated outstanding performance in terms of accuracy and robustness on those datasets and enabled other robotic tasks, for example, planning for underwater robots in presence of obstacles.

Keywords

Underwater SLAM, sensor fusion, marine robotics

1. Introduction

We propose a Simultaneous Localization and Mapping (SLAM) system, which exploits the complementary characteristics of cameras, Inertial Measurement Unit (IMU), sonar, and pressure sensor for robust autonomy of underwater vehicles in challenging and highly unstructured underwater environments.

Autonomous Underwater Vehicles (AUVs) present unique opportunities to automate the exploration and mapping of underwater environments—such as caves, bridges, dams, oil rigs, and shipwrecks—which are extremely important tasks for the economy, conservation, and scientific discoveries. Currently, the autonomy of underwater vehicles is hindered by a number of challenges. As localization infrastructures (e.g., GPS) present on land are not available underwater, one of the challenges relates to state estimation. In recent years, many visual odometry and SLAM systems (Davison et al., 2007; Engel et al., 2014, 2018; Forster et al., 2017b; Klein and Murray, 2007; Mur-Artal et al., 2015) have been developed using monocular, stereo, or multi-camera rigs mostly for indoor and outdoor environments that can be classified as *indirect* (feature-based), *direct*, or *semi-direct* methods. Vision is often

combined with inertial measurements for improved accuracy and robustness in pose estimation, termed *Visual-Inertial Odometry (VIO)* or *Visual-Inertial SLAM (VI-SLAM)* (Leutenegger et al., 2015; Mourikis and Roumeliotis, 2007; Mur-Artal and Tardós, 2017; Qin et al., 2018; Sun et al., 2018). As shown in previous studies (Joshi et al., 2019; Quattrini Li, et al., 2016a), the available visual or visual-inertial state estimation packages produce inaccurate trajectories or even complete tracking loss in underwater due to the inherent challenges in such environments. In particular, vision-based state estimation underwater has many open challenges, including visibility, light and color attenuation (Fabbri

¹Computer Science and Engineering Department, University of South Carolina, Columbia, SC, USA

²Department of Computer Science, Dartmouth College, Hanover, NH, USA

Corresponding author:

Alberto Quattrini Li, Department of Computer Science, Dartmouth College, 15 Thayer Drive, Dartmouth Reality and Robotics Lab, Hanover, NH 03755, USA.

Email: alberto.quattrini.li@dartmouth.edu



Figure 1. Underwater cave in Quintana Roo, Mexico, where data have been collected using an underwater stereo rig.

et al., 2018; Roznere and Quattrini Li, 2019; Skaff et al., 2008) or complete absence of natural light, floating particulates, blurriness, varying illumination, and lack of features (Oliver et al., 2010)—for example, see Figure 1—leaving an interesting gap to be investigated in the current state-of-the-art.

In this work, we present a novel SLAM system, *SVIn2*, targeted for underwater environments and easily adaptable for other domains—for example, indoor and outdoor—by choosing a subset of different sensor configurations including: visual (monocular, stereo camera, or multi-camera), inertial (linear accelerations and angular velocities), Digital Pipe Profiling Sonar (DPP-sonar) (Imagenex Technology Corp, 2022)—that is, a mechanical scanning profiling sonar—and/or water-depth data. This makes our system versatile and applicable onboard of different sensor suites and underwater vehicles. We augmented the state-of-the-art visual-inertial state estimation package OKVIS (Leutenegger et al., 2015) to accommodate acoustic range data from a DPP-sonar in a tightly-coupled non-linear optimization-based framework. This augmentation improves the trajectory estimate, especially when there is varying visibility underwater, as the DPP-sonar provides robust information about the presence of obstacles with accurate scale. However, in long trajectories, drifts could accumulate resulting in an erroneous trajectory. To account for drifts, we introduced depth measurements from a water-pressure sensor in the optimization process, loop-closing and relocalization capabilities using the bag-of-words (BoW) framework, and a more robust initialization process to refine scale using water-depth measurements. These additions enable the proposed approach to robustly and accurately estimate the robot’s trajectory, where other approaches have shown incorrect trajectories or complete loss of localization.

To validate our proposed approach, first, we assessed the performance of the proposed loop-closing method by comparing it to other state-of-the-art systems on the EuRoC micro-aerial vehicle (MAV) public dataset (Burri et al., 2016), disabling the fusion of DPP-sonar and water-pressure measurements in our system. Second, we tested the proposed full system on several underwater datasets obtained under a diverse set of conditions. More

specifically, underwater data—consisting of visual, inertial, water depth, and acoustic range measurements—have been collected using a custom-made sensor suite (Rahman et al., 2018a) from different locales; furthermore, data were also collected by an Aqua2 underwater vehicle (Dudek et al., 2005), including visual, inertial, and water-depth measurements. The results on the underwater datasets illustrate the loss of tracking and/or failure to maintain consistent scale for other state-of-the-art systems, while our proposed method maintains correct scale without diverging. In the absence of ground truth trajectories in underwater, we used COLMAP (Schönberger et al., 2016)—an open-source Structure from Motion (SfM) library—as a baseline for comparing the performance of the proposed algorithm, *SVIn2* (with sonar and pressure sensor disabled) with other state-of-the-art visual-inertial odometry/SLAM packages. Third, we performed a 3D landmark-based validation to show the estimation accuracy of *SVIn2* using fiducial markers (Fiala, 2005). Fourth, we tested our proposed system using a recent public underwater dataset, named AQUALOC (Ferrera et al., 2019). Fifth, we performed an ablation study to observe the contribution of each sensor to localization accuracy.

The contributions of this paper significantly extend the preliminary results we presented in (Rahman et al., 2018b, 2019) with a more complete system description and additional experimental analysis, including an ablation study, landmark-based validation, comparison with the SfM package, and experiments on a public underwater dataset, as briefly introduced above. The code is released open-source at (Rahman, 2020).

The paper is structured as follows: the next section discusses the current state-of-the-art on state estimation underwater and above water. Section 3 provides an overview of the proposed pipeline along with the approach developed for the image preprocessing step and the notations used. Section 4 describes the mathematical formulation and derivation of the tightly-coupled DPP-sonar, stereo camera, inertial, and water-depth sensor integration. Section 5 and Section 6 present the pose initialization, and the loop-closure/relocalization step, respectively. Experimental results from a publicly available aerial dataset and a diverse set of challenging underwater environments are presented in Section 7. We then conclude this paper and discuss directions of future work.

2. Related work

Researchers have studied the robot state estimation problem for decades. Here, we highlight those specifically tailored for underwater environments and the most recent ones on visual-inertial state estimation. For a more complete overview, the reader is encouraged to look at recent surveys by Cadena et al. (2016) and Huang (2019).

2.1. Acoustic sensor based underwater navigation

Paull et al. (2013) presented a review of the commonly used sensors and general methods for AUV navigation and localization. More recently, Maurelli et al. (2021) discussed active and passive localization techniques for AUVs. Sonar (e.g., imaging sonar, scanning profiling sonar, and multi-beam sonar) and/or camera are used to bound the odometry drift from dead-reckoning system, that is, IMU or Doppler Velocity Log (DVL).

Most of the underwater navigation algorithms (Johannsson et al., 2010; Lee et al., 2005; Leonard and Durrant-Whyte, 2012; Rigby et al., 2006; Snyder, 2010) are based on acoustic sensors such as DVL and Ultra-Short Baseline (USBL). DEPTHX (DEep Phreatic Thermal eXplorer) (Stone, 2007) designed by Stone Aerospace was equipped with a number of sensors for mapping a cenote—a vertical shaft filled with water (Gary et al., 2008)—including an IMU, two depth sensors, a DVL, and 54 narrow single-beam echosounders. Sunfish (Richmond et al., 2018)—an underwater SLAM system using a multibeam sonar, an underwater dead-reckoning system based on a fiber-optic gyroscope (FOG) IMU, DVL, and pressure-depth sensors—has been developed for autonomous underwater cave exploration.

A number of works have tried to reduce the cost of the underwater robot without requiring expensive DVL or Inertial Navigation System (INS). Williams et al. (2000) presented an EKF-based SLAM for an underwater robot with a mechanical scanning sonar positioned to map the horizontal plane. White et al. (2010) presented field experiments for mapping and localization in ancient underwater cisterns using a mechanical scanning sonar also positioned to map the horizontal plane, using a particle filter. Folkesson et al. (2007) used a blazed array forward-looking sonar—composed of two sonar heads, one vertical and one horizontal, giving a position in 3D for features detected by both—for real-time feature tracking with a relatively lower-cost AUV, without requiring DVL and expensive Inertial Navigation System (INS). Building on top of the latter work, Fallon et al. (2013) expanded the system to feature re-acquisition. Mallios et al. (2016) demonstrated the first results of an AUV performing limited penetration inside a cave using a horizontally mounted scanning sonar as the main sensor.

An imaging sonar based SLAM and 3D photomosaicing algorithm has been proposed by Westman et al. (2018) and Ozog et al. (2015), respectively. Teixeira et al. (2019) presented dense reconstruction of underwater scenes using a multibeam sonar. McConnell et al. (2020) fused two multibeam sonars, one placed horizontally and one vertically, to address the uncertainty over the elevation angle. Similarly, Joe et al. (2021) proposed a mapping sensor configuration of multibeam sonar and profiling sonar exploiting the larger field-of-view covered by the first and the narrow beam of the latter.

Early attempts for underwater localization with camera were shown by Carreras et al. (2003), who presented a landmark-based navigation. SLAM systems have been studied with a downward-looking stereo camera (Eustice et al., 2005, 2006) to map the Titanic. Corke et al. (2007) compared underwater localization methods based on a network of acoustic sensor nodes which are able to estimate range between each other and based on vision, showing the viability of using visual methods underwater in some scenarios. Navigation and planning algorithms have been proposed using cameras and imaging sonar for ship hull inspection application with a Hovering Autonomous Underwater Vehicle (HAUV) (Hong et al., 2019; Hover et al., 2012). The same group also developed vision-only SLAM system for the same application (Kim and Eustice, 2013; Ozog and Eustice, 2014; Ozog et al., 2016). Recent work fused stereo camera and DVL for underwater SLAM (Xu et al., 2021).

Our work is in the direction of reducing the sensors necessary for underwater SLAM, without requiring DVL or expensive INS, as some of the works presented above. We consider a different sensor configuration where the mechanical scanning sonar is placed to map the vertical plane, parallel to the image plane, so that cave structures can be mapped, as will be discussed later in the paper.

2.2. Visual-inertial state estimation

Vision is often combined with IMU for their complementary characteristics: while cameras are exteroceptive sensors capturing the external world, IMU provides information about self-motion. In addition, combining vision with an IMU can solve the *scale* issue in monocular vision-based SLAM, as it can be used to estimate the motion between camera frames. Gravity can also be estimated, which makes two rotational degrees of freedom (DoF)—that is, absolute pitch and roll—observable, providing another advantage of integrating vision with IMU. In the following, we highlight some of the state-of-the-art visual-inertial VIO and SLAM methods. A class of state estimation approaches is based on the *Kalman Filter*. Examples include the Multi-State Constraint Kalman Filter (MSCKF) (Mourikis and Roumeliotis, 2007)—which has been deployed in the underwater domain (Shkurti et al., 2011b)—and its stereo extension (Sun et al., 2018); ROVIO (Bloesch et al., 2017); and REBIVO (Tarrío and Pedre, 2017). Another family of methods optimizes the sensor states—typically within a sliding window—formulating the problem as a *graph optimization* problem. Feature-based visual-inertial systems—such as OKVIS (Leutenegger et al., 2015), Visual-Inertial ORB-SLAM (Mur-Artal and Tardós, 2017), and ORB-SLAM3 (Campos et al., 2021)—have an optimization function that includes the IMU error term and the re-projection error. The *frontend* tracking mechanism maintains a bounded window of *keyframes* and marginalizes states and features which are never used again once out of

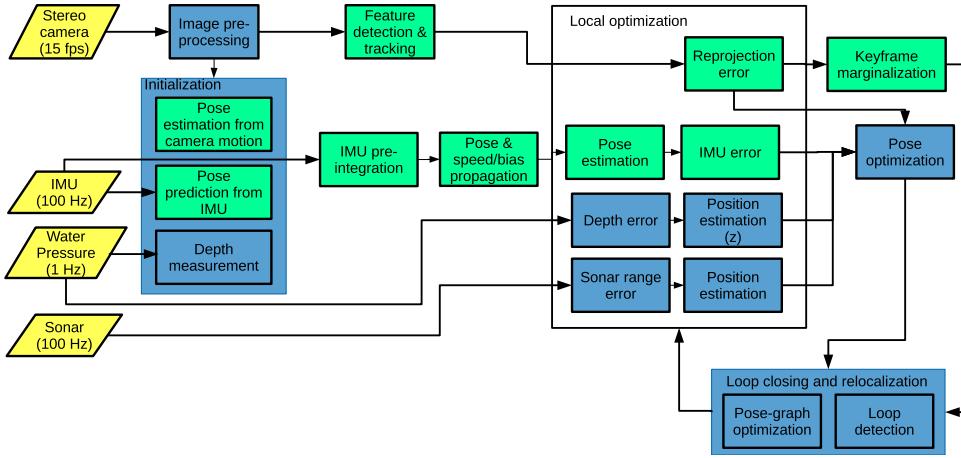


Figure 2. Overview of the proposed approach, SVIn2; in yellow are the sensor feeds and their frequency; in green are the OKVIS (Leutenegger et al., 2015) components; in blue are the components we introduced to handle acoustic range and water-depth data, underwater visual effects, initialization, and loop closure and relocalization (Rahman et al., 2018b, 2019).

the window, to limit the computation required by the optimization. VINS-Mono (Qin et al., 2018) uses a similar approach and maintains a minimum number of features for each image. Existing features are tracked by Kanade–Lucas–Tomasi (KLT) sparse optical flow algorithm in a local window. While KLT sparse features allow VINS-Mono to run in real-time on low-cost embedded systems, this approach often results into tracking failure in challenging environments, for example, underwater environments with low visibility. In addition, for loop detection additional features and their descriptors need to be computed for keyframes. Delmerico and Scaramuzza (2018) did a comprehensive comparison specifically monitoring resource usage by the different methods. A performance evaluation of features for the underwater domain was presented by Shkurti et al. (2011a) and by Quattrini Li et al. (2016b), with a focus on shipwreck environments.

2.3. Why combining vision with acoustic sensor for underwater environments

Vision-based underwater navigation is a difficult task due to its highly unstructured nature. At the same time, camera is one of the cheapest, small, light-weight, and energy-efficient sensors, which provides rich and versatile information about the surroundings. In our recent work (Joshi et al., 2019), we compared the performance of open-source visual-inertial systems in underwater datasets. The results suggest that VO/VIO for both direct methods and feature-based methods is challenging due to light and color attenuation, blurriness, and floating particulates. Specifically, for direct methods, the brightness constancy assumption is often violated due to the frequent lighting variations. For indirect methods, low contrast and particulates lead to spurious features. Without the help of dead-reckoning, pure VO frequently tends to lose track without motion prediction, as it becomes difficult to track features reliably across

subsequent frames. Contrary to vision, sonar range measurements are not affected by turbidity or light and color attenuation, hence complementing the camera. DPP-sonar improves the quality of 3D reconstruction by providing features with scale which in turn helps in localization.

For robust tracking, visual-inertial state estimation systems require proper *initialization*. ORB-SLAM with IMU (Mur-Artal and Tardós, 2017) performs initialization by first running a monocular SLAM to observe the pose and then, IMU biases are estimated. VINS-Mono uses a loosely-coupled sensor fusion method to align monocular vision with inertial measurement for estimator initialization. In addition to a good initialization, to mitigate the drift in sliding window and marginalization-based state estimate, *loop closure*—the capability of recognizing a place that was seen before—is an important scheme. Currently ORB-SLAM (Mur-Artal et al., 2015) and its extension with IMU (Mur-Artal and Tardós, 2017) is one of the most reliable feature-based SLAM systems that uses the bag-of-words (BoW) approach for loop closure and relocalization. VINS-Mono also uses the same technique. Another BoW-based approach clustered a set of relevant regions, showing robustness to illumination change in areas that present similarities (e.g., coral reef) (Maldonado-Ramírez et al., 2016). Given the modularity of OKVIS in adding new sensors and robustness in tracking in underwater environment, we decided to extend OKVIS to include DPP-sonar data, water-pressure measurements, loop closure capabilities, and a more robust initialization with 2-step scale refinement using water-depth, to specifically target underwater environments.

3. System overview and preliminaries

SVIn2 pipeline is depicted in Figure 2. The robot sensor configuration can include camera (mono, stereo, or multi), IMU, DPP-sonar, and a water-pressure sensor. The latter

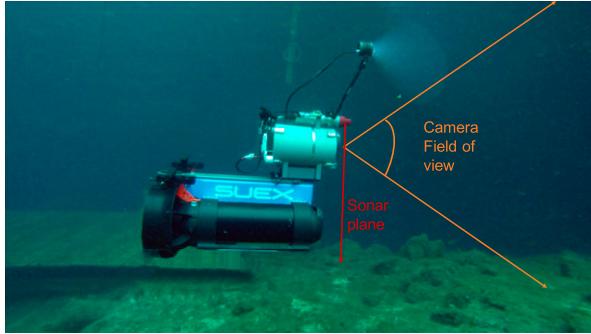


Figure 3. Custom-made sensor suite mounted on a dual DPV. DPP-sonar scans around the sensor while the cameras see in front. Please note, the setup is neutrally buoyant and balanced to hover in place and upright without any support.

two can be disabled to operate as a standard visual-inertial system.

To cope with the challenges of underwater environments described above, we augment the pipeline by adding an optional image pre-processing step to improve feature detection underwater. In particular, we use a *contrast limited adaptive histogram equalization* (CLAHE) filter (Pisano et al., 1998) in the *image pre-processing* step.

After the initialization of the SLAM system with IMU, camera, and water-depth, the SLAM system uses images to detect features and track them over time; IMU for motion; and depth and DPP-sonar sensors—all of them fed into a local optimization framework that minimizes the error terms defined for each sensor. A bag-of-words based loop closure mechanism corrects the drift accumulated over time.

A sensor suite composed of a stereo camera, IMU, pressure sensor, and a DPP-sonar with a field of view covering 360° over a plane parallel to the image plane is shown in Figure 3. The rationale to use the DPP-sonar in that configuration is for mapping underwater cave structures. The following sections, after formally defining the notation, symbols, and state representations used in the subsequent parts of the paper, describe in detail the proposed initialization, sensor fusion optimization, loop closure and relocalization steps.

3.1. Notations and states

We define the following coordinate frames for the complete sensor suite: the i -th Camera C_i , IMU I , Depth (pressure) D , DPP-sonar (acoustic range) S , and World W . A homogeneous transformation matrix $X\mathbf{T}_Y = [X\mathbf{R}_Y \mid X\mathbf{p}_Y]$ represents the transformation between two arbitrary coordinate frames X and Y , with rotation matrix $X\mathbf{R}_Y$ —the corresponding quaternion is $X\mathbf{q}_Y$ —and position vector $X\mathbf{p}_Y$. For example, X and Y could be W and I , respectively, thus $W\mathbf{T}_I$ identifies the transformation matrix from IMU to World. $X\mathbf{C}_Y(\mathbf{q})$ is a function that converts quaternion $X\mathbf{q}_Y$ to its equivalent

rotation matrix, $X\mathbf{R}_Y$. The robot (R) state \mathbf{x}_R that the system is estimating is defined as

$$\mathbf{x}_R = \left[{}_W\mathbf{p}_I^T, {}_W\mathbf{q}_I^T, {}_W\mathbf{v}_I^T, \mathbf{b}_g^T, \mathbf{b}_a^T \right]^T \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^9 \quad (1)$$

where ${}_W\mathbf{p}_I$ is the position, ${}_W\mathbf{q}_I$ is the attitude represented as a quaternion, ${}_W\mathbf{v}_I$ is the linear velocity, all expressed in the IMU reference frame I with respect to the world coordinate W . The state vector also contains the gyroscopes and accelerometers bias \mathbf{b}_g and \mathbf{b}_a .

For solving the state estimation problem, we define the associated error-state vector in minimal coordinates. The perturbation takes place in the tangent space of the state manifold. The transformation from minimal coordinates to tangent space can be done using a bijective mapping (Blanco, 2010; Forster et al., 2017a; Leutenegger et al., 2015) and the error for each component of the state vector \mathbf{x}_R is

$$\delta\mathbf{x}_R = \left[\delta\mathbf{p}^T, \delta\theta^T, \delta\mathbf{v}^T, \delta\mathbf{b}_g^T, \delta\mathbf{b}_a^T \right]^T \in \mathbb{R}^{15} \quad (2)$$

where $\delta\theta \in \mathbb{R}^3$ is the minimal perturbation for rotation (can be converted to its quaternion equivalent via exponential mapping).

4. Tightly-coupled non-linear optimization with sonar-visual-inertial-pressure measurements

The proposed system fuses vision, acoustic range, inertial, and water-pressure measurements within a tightly-coupled non-linear optimization, where we define the cost function as $J(\mathbf{x})$, including the reprojection error \mathbf{e}_r and the IMU error \mathbf{e}_s with the addition of the DPP-sonar error \mathbf{e}_t , and the water-depth error e_u

$$\begin{aligned} J(\mathbf{x}) = & \sum_{i=1}^n \sum_{k=1}^K \sum_{j \in \mathcal{J}(i, k)} \mathbf{e}_r^{i,j,k^T} \mathbf{P}_r^k \mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1} \mathbf{e}_s^{k^T} \mathbf{P}_s^k \mathbf{e}_s^k \\ & + \sum_{k=1}^{K-1} \mathbf{e}_t^{k^T} \mathbf{P}_t^k \mathbf{e}_t^k + \sum_{k=1}^{K-1} P_u^k \| e_u^k \|^2 \end{aligned} \quad (3)$$

where i denotes the camera index with landmark index j observed in the k^{th} camera frame. For example, in a stereo camera system, $n = 2$, where left $i = 1$ and right $i = 2$ camera; note that SVIn2 supports multi-camera systems with arbitrary n , starting from 1. \mathbf{P}_r^k , \mathbf{P}_s^k , \mathbf{P}_t^k , and P_u^k represent the information matrix (weights) of visual landmarks, IMU, sonar range, and water-depth measurement for the k^{th} frame, respectively. Please note, visual landmarks, IMU, and sonar measurements are vectors, while the depth measurement is a scalar.

Intuitively, the IMU error term combines all accelerometer and gyroscope measurements in between camera measurements and represents the *pose, speed, and bias*

error. The reprojection error captures the difference in pixels between a keypoint measurement in camera coordinate frame C and the corresponding landmark projection onto the imaging plane according to the camera projection model. Both reprojection and IMU error terms follow the formulation of Leutenegger et al. (2015). The DPP-sonar error describes the error between the acoustic range measurement and the corresponding visual feature patch. Note that sonar measurements provide additional points for a denser 3D reconstruction. The depth error limits the error of the robot position state in the gravity direction. The Google's *Ceres Solver* (Agarwal et al., 2015)—the non-linear optimization framework—minimizes the cost function $J(\mathbf{x})$ containing such error terms to estimate the robot state \mathbf{x}_R in real-time.

For completeness, the next subsections discuss in detail each error term.

4.1. IMU error term formulation

An IMU provides accelerometer and gyroscope readings. Integration of these readings leads to a dead-reckoning positioning system. The estimate from such an integration drifts quickly over time. Fusing dead-reckoning with absolute positioning readings (for example, vision) limits drifts. Below, we present the formulation of the non-linear IMU kinematics and bias model; more specifically, the formulation of IMU kinematic model, the linearized error-state model, and IMU measurement error.

4.1.1. IMU kinematic model. We employ an IMU kinematic model relating the raw gyroscope measurements, ω_m and raw accelerometer measurements, \mathbf{a}_m from IMU to the real angular velocity ω and the real linear acceleration \mathbf{a} , at time t , respectively, as

$$\begin{aligned}\omega_m(t) &= \omega(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \\ \mathbf{a}_m(t) &= \mathbf{a}(t) + \mathbf{b}_a(t) + \mathbf{R}_W(t)\mathbf{g} + \mathbf{n}_a(t)\end{aligned}\quad (4)$$

In the above equation, the IMU measurements are taken in its local frame, that is, I , which accounts for the gravity $w\mathbf{g}$ transformed with the rotation matrix \mathbf{R}_W in the IMU reference frame, gyroscope bias \mathbf{b}_g , acceleration bias \mathbf{b}_a , and additive noise. The additive noise both in acceleration and gyroscope readings is assumed to be Gaussian white noise with characteristics $\mathbf{n}_a \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \sigma_a^2 \mathbf{I}_{3 \times 3})$, $\mathbf{n}_g \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \sigma_g^2 \mathbf{I}_{3 \times 3})$, respectively. Similarly to the work by Trawny and Roumeliotis (2005), we assume that the noise is equal in all three spatial directions and that the gyro and accelerometer biases are non-static and simulated as a random walk process. The biases characteristics are: $\mathbf{n}_{bg} \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \sigma_{bg}^2 \mathbf{I}_{3 \times 3})$, $\mathbf{n}_{ba} \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \sigma_{ba}^2 \mathbf{I}_{3 \times 3})$. Following the formulation from Leutenegger et al. (2015), the accelerometer bias is modeled as a bounded random walk with time constant $\tau > 0$, whereas the gyro bias is modeled as random walk. The bias driving noise, that is, \mathbf{n}_{bg} and \mathbf{n}_{ba} , corresponds to the process noise, whereas the

rate noise, that is, \mathbf{n}_b and \mathbf{n}_a , corresponds to the measurement noise.

The differential equations that describe the continuous-time IMU kinematics combined with bias models are

$$\begin{aligned}\dot{\mathbf{q}}_W(t) &= \frac{1}{2} \boldsymbol{\Omega} (\omega_m(t) - \mathbf{b}_g(t) - \mathbf{n}_g(t))_I \mathbf{q}_W(t) \\ \dot{\mathbf{b}}_g(t) &= \mathbf{n}_{bg}(t) \\ w\dot{\mathbf{v}}_I(t) &= w\mathbf{R}_I(t)(\mathbf{a}_m(t) - \mathbf{b}_a(t) - \mathbf{n}_a(t)) - w\mathbf{g} \\ \dot{\mathbf{b}}_a(t) &= -\frac{1}{\tau} \mathbf{b}_a(t) + \mathbf{n}_{ba}(t) \\ w\dot{\mathbf{p}}_I(t) &= w\mathbf{v}_I(t)\end{aligned}\quad (5)$$

where the matrix $\boldsymbol{\Omega}$ is defined as

$$\boldsymbol{\Omega}(\omega) = \begin{bmatrix} -[\omega]_\times & \omega \\ -\omega^T & 0 \end{bmatrix}, [\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \in \mathfrak{so}(3)$$

4.1.2. Linearized error-state model. The continuous-time linearized model of the error state takes the form of

$$\begin{aligned}\dot{\delta\chi}_R &= \begin{bmatrix} \delta\dot{\mathbf{p}} \\ \delta\dot{\theta} \\ \delta\dot{\mathbf{v}} \\ \delta\dot{\mathbf{b}}_g \\ \delta\dot{\mathbf{b}}_a \end{bmatrix} \approx \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -[\omega_m - \hat{\mathbf{b}}_g]_\times & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\hat{\mathbf{R}} \left[\mathbf{a}_m - \hat{\mathbf{b}}_a \right]_\times & \mathbf{0} & -\hat{\mathbf{R}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{1}{\tau} \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\mathbf{p} \\ \delta\theta \\ \delta\mathbf{v} \\ \delta\mathbf{b}_g \\ \delta\mathbf{b}_a \end{bmatrix} \\ &\quad + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\hat{\mathbf{R}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{n}_g \\ \mathbf{n}_a \\ \mathbf{n}_{bg} \\ \mathbf{n}_{ba} \end{bmatrix} \\ &= \mathbf{F}_c(\mathbf{x}_R) \delta\chi_R + \mathbf{G}_c(\mathbf{x}_R) \mathbf{n}\end{aligned}\quad (6)$$

where (\cdot) represents prediction and $[\cdot]_\times$ corresponds to the skew-symmetric matrix associated with a vector.

Since the continuous-time system matrix \mathbf{F}_c is constant over the integration period, discrete-time linearized error state transition matrix can be obtained by

$$\mathbf{F}_d(\mathbf{x}_R, \Delta t) = \exp(\mathbf{F}_c(\mathbf{x}_R) \Delta t) \approx \mathbf{I}_{15} + \mathbf{F}_c(\mathbf{x}_R) \Delta t \quad (7)$$

where Δt is the integration time step. The covariance propagation equation can be computed recursively by a

first-order discrete-time covariance update, that is, with the covariance \mathbf{W}_R^p for the p^{th} IMU measurement, it takes the following form

$$\mathbf{W}_R^{p+1} = \mathbf{F}_d(\hat{\mathbf{x}}_R^p, \Delta t) \mathbf{W}_R^p \mathbf{F}_d(\hat{\mathbf{x}}_R^p, \Delta t)^T + \mathbf{G}_d(\hat{\mathbf{x}}_R^p) \mathbf{Q}_d \mathbf{G}_d(\hat{\mathbf{x}}_R^p)^T \Delta t \quad (8)$$

where $\mathbf{Q}_d = \text{diag}(\sigma_g^2 \cdot \mathbf{I}_3, \sigma_a^2 \cdot \mathbf{I}_3, \sigma_{bg}^2 \cdot \mathbf{I}_3, \sigma_{ba}^2 \cdot \mathbf{I}_3)$ is the diagonal matrix containing all the noise densities of the respective processes.

4.1.3. IMU Measurement error formulation. We express the IMU error term $\mathbf{e}_s^k(\mathbf{x}_R^k, \mathbf{x}_R^{k+1}, \mathbf{z}_s^k)$ as a function of robot states at time steps k and $k+1$ (when the images are taken), and all the IMU measurements \mathbf{z}_s^k , containing gyro and accelerometer data in-between these time instances. We assume an approximate normal conditional probability density function f with zero mean and variance \mathbf{W}_s^k , and the associated conditional covariance $\mathbf{Q}(\delta\hat{\mathbf{x}}_R^{k+1} | \mathbf{x}_R^k, \mathbf{z}_s^k)$ for given robot states at camera measurements k and $k+1$

$$f(\mathbf{e}_s^k | \mathbf{x}_R^k, \mathbf{x}_R^{k+1}) \approx \mathcal{N}(\mathbf{0}, \mathbf{W}_s^k) \quad (9)$$

Using the prediction equations, we can now formulate the IMU error term as follows which is simply the difference between the *prediction* based on the previous state and the *actual* state

$$\mathbf{e}_s^k(\mathbf{x}_R^k, \mathbf{x}_R^{k+1}, \mathbf{z}_s^k) = \begin{bmatrix} {}_I\mathbf{R}_W^k \left({}_W\hat{\mathbf{p}}_I^{k+1} - {}_W\mathbf{p}_I^{k+1} \right) \\ 2 \left[{}_I\mathbf{q}_W^k \otimes {}_W\hat{\mathbf{q}}_I^{k+1} \otimes {}_W\mathbf{q}_I^{k+1} \right]_{1:3} \\ {}_I\mathbf{R}_W^k \left({}_W\hat{\mathbf{v}}_I^{k+1} - {}_W\mathbf{v}_I^{k+1} \right) \\ \hat{\mathbf{b}}_g^{k+1} - \mathbf{b}_g^{k+1} \\ \hat{\mathbf{b}}_a^{k+1} - \mathbf{b}_a^{k+1} \end{bmatrix} \in \mathbb{R}^{15} \quad (10)$$

By applying the error propagation law, the associated information matrix \mathbf{P}_s^k is obtained by

$$\mathbf{P}_s^k = \mathbf{W}_s^{k-1} = \left(\frac{\partial \mathbf{e}_s^k}{\partial \delta\hat{\mathbf{x}}_R^{k+1}} \mathbf{Q}(\delta\hat{\mathbf{x}}_R^{k+1} | \mathbf{x}_R^k, \mathbf{z}_s^k) \frac{\partial \mathbf{e}_s^k}{\partial \delta\hat{\mathbf{x}}_R^{k+1}}^T \right)^{-1} \quad (11)$$

4.1.2. Reprojection error formulation. The camera observes the visual features, which are used to update the motion estimate of the robot. As in (Leutenegger et al., 2015), these visual features are stereo-triangulated to create the *local map*. With a window of a sparse set of the latest camera frames/keyframes and their landmarks in the local map, at first a 3D-2D matching is performed using pose prediction from IMU to limit the search-space and then a

2D-2D matching takes place. In both matching steps, outliers are rejected applying the chi-square test (3D-2D matching) using IMU pose prediction and RANSAC.

The reprojection error is formulated as the difference between the feature observation $\mathbf{z}^{i,j,k}$ in image coordinates and the projection of the corresponding 3D point ${}_C_i\mathbf{p}^j$ on to the image plane, where i is the camera index, and j is the 3D landmark index which is visible in the k^{th} image frame

$$\mathbf{e}_r^{i,j,k} = \mathbf{z}^{i,j,k} - \mathbf{h}_i({}_C_i\mathbf{p}^j) \quad (12)$$

here $\mathbf{h}_i(\cdot)$ denotes the camera projection model.

Assuming a perspective camera model, the feature measurement with zero-mean, and Gaussian white noise, $\mathbf{n}^{i,j,k}$ is defined as

$$\mathbf{z}^{i,j,k} = \frac{1}{z_{i,j,k}} \begin{bmatrix} x^{i,j,k} \\ y^{i,j,k} \end{bmatrix} + \mathbf{n}^{i,j,k} \quad (13)$$

$$\begin{bmatrix} x^{i,j,k} \\ y^{i,j,k} \\ z^{i,j,k} \end{bmatrix} = {}_C_i\mathbf{p}^j = {}_C_i\mathbf{C}_I(\mathbf{q})_I \mathbf{C}_W(\mathbf{q}^k) ({}_W\mathbf{p}^j - {}_W\mathbf{p}_I^k) + {}_C_i\mathbf{p}_I \quad (14)$$

The measurement Jacobian \mathbf{H}^k is calculated as

$$\mathbf{H}^k = \mathbf{H}_{\text{proj}} {}_C_i\mathbf{C}_I(\mathbf{q}) \begin{bmatrix} \mathbf{H}_\theta^k & \mathbf{0}_{3 \times 9} & \mathbf{H}_\mathbf{p}^k \end{bmatrix} \quad (15)$$

\mathbf{H}_{proj} , \mathbf{H}_θ^k , and $\mathbf{H}_\mathbf{p}^k$ are the Jacobian of the projection $\mathbf{h}_i(\cdot)$ into the i^{th} camera with respect to the landmark in the homogeneous coordinates, orientation, and translation, respectively

$$\mathbf{H}_{\text{proj}} = \frac{1}{\hat{z}^{i,j,k}} \begin{bmatrix} 1 & 0 & -\frac{\hat{x}^{i,j,k}}{\hat{z}^{i,j,k}} \\ 0 & 1 & -\frac{\hat{y}^{i,j,k}}{\hat{z}^{i,j,k}} \end{bmatrix} \quad (16)$$

$$\mathbf{H}_\theta^k = \left[\left({}_I\mathbf{C}_W(\hat{\mathbf{q}}^k) \left({}_W\mathbf{p}^j - {}_W\mathbf{p}_I^k \right) \right) \right]_\times \delta\theta \quad (17)$$

$$\mathbf{H}_\mathbf{p}^k = -{}_I\mathbf{C}_W(\hat{\mathbf{q}}^k) \quad (18)$$

4.3. DPP-sonar error term formulation

Acoustic range data, though sparser, provide robust information about the presence of obstacles, where visual features reside; thus DPP-sonar helps to correct the robot pose estimate as well as to optimize the use of landmarks coming from both vision and sonar. Due to the low visibility of underwater environments, when it is hard to find visual features, DPP-sonar provides features with accurate scale.

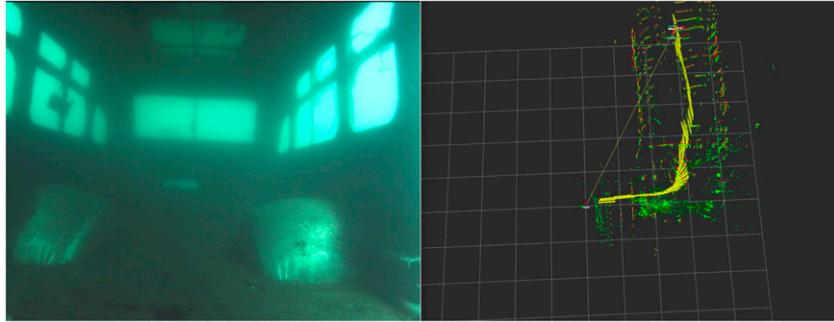


Figure 4. Sunken bus, Fantasy Lake Scuba Park, NC, USA. (left) Sample image of the data collected from inside the bus. (right) Top view of the reconstruction. Yellow arrows represent the pose of the robot; green and red points derive from the visual and DPP-sonar features, respectively.

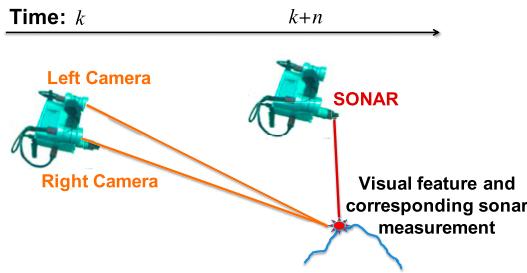


Figure 5. The relationship between DPP-sonar range measurement and stereo camera features. A visual feature detected at time k is only detected by the DPP-sonar with a delay, at time $k + n$, where n depends on the speed the sensor is moving.

Figure 4 shows the visual-acoustic reconstruction using the proposed approach.

The sonar range error follows the intuition that, when the DPP-sonar detects any obstacle at some distance, the visual features corresponding to the same obstacle will be approximately at the same distance. Given the sensor configuration shown in [Figure 3](#), a particular challenge arises: the DPP-sonar features are matched with the visual features after some time, due to the different field of view covered by the two sensors—see [Figure 5](#), where at time k some features are detected by the camera; it takes some time (until $k + n$) for the DPP-sonar to pass by these visual features and thus obtain a related measurement. To address this challenge, visual features detected in close proximity to the DPP-sonar return are grouped together to construct a patch. Then, to fuse range data from DPP-sonar into the traditional VIO framework, the detected visual patches in close proximity of each sonar point introduce extra constraints: the distance of the sonar point to the patch. Here, we assume that the visual-feature based patch is small enough and approximately coplanar with the DPP-sonar point. [Figure 6](#) illustrates the relation between DPP-sonar and visual features in the formulation of the sonar error term.

Algorithm 1 shows how to calculate the *range error* \mathbf{e}_t^k given the robot position ${}_w\mathbf{p}_I^k$ and the DPP-sonar measurement \mathbf{z}_t^k at time k . The DPP-sonar returns *range* r and *head position* θ measurements, which are used to obtain each

sonar landmark ${}_w\mathbf{l} = [l_x, l_y, l_z, 1]$ in homogeneous coordinates by a geometric transformation in world coordinates

$${}_w\mathbf{l} = ({}_w\mathbf{T}_{II}\mathbf{T}_S[\mathbf{l}_3|r \cos(\theta), r \sin(\theta), 0]_S^T) \quad (19)$$

Algorithm 1. DPP-sonar Range Error Algorithm

Input: Estimation of robot position ${}_w\mathbf{p}_I^k$ at time k
 Sonar measurement $\mathbf{z}_t^k = [r, \theta]$ at time k
 List of current visual landmarks, \mathcal{L}_v
 Distance threshold, T_d

Output: Range error e_t^k at time k

/*Compute sonar landmark in world coordinates*/

- 1: ${}_w\mathbf{l} = ({}_w\mathbf{T}_{II}\mathbf{T}_S[\mathbf{l}_3|r \cos(\theta), r \sin(\theta), 0]_S^T)$
- /*Create list of visual landmarks around sonar landmark*/
- 2: $\mathcal{L}_S = \emptyset$
- 3: **for** (every \mathbf{l}_i in \mathcal{L}_v) **do**
- /*Compute Euclidean distance from visual landmark to sonar landmark*/
- 4: $d_S = \|{}_w\mathbf{l} - \mathbf{l}_i\|$
- 5: **if** ($d_S < T_d$) **then**
- 6: $\mathcal{L}_S = \mathcal{L}_S \cup \mathbf{l}_i$
- 7: **end if**
- 8: **end for**
- 9: $\hat{r} = \|{}_w\hat{\mathbf{p}}_I^k - \text{mean}(\mathcal{L}_S)\|$
- 10: **return** $r - \hat{r}$

The sonar range prediction is calculated using the lines 2–9 of Algorithm 1

$$\hat{r} = \left\| {}_w\hat{\mathbf{p}}_I^k - \text{mean}(\mathcal{L}_S) \right\| \quad (20)$$

where \mathcal{L}_S is the subset of visual landmarks around the sonar landmark and the *range error* term is formulated as the difference between the two distances. Note that we approximate the visual patch with the centroid ($\text{mean}(\mathcal{L}_S)$), to filter out noise on the visual landmarks.

Given the sonar measurement \mathbf{z}_t^k , the error term $e_t^k({}_w\mathbf{p}_I^k, \mathbf{z}_t^k)$ is used to correct the position ${}_w\mathbf{p}_I^k$. We assume an approximate normal conditional probability density

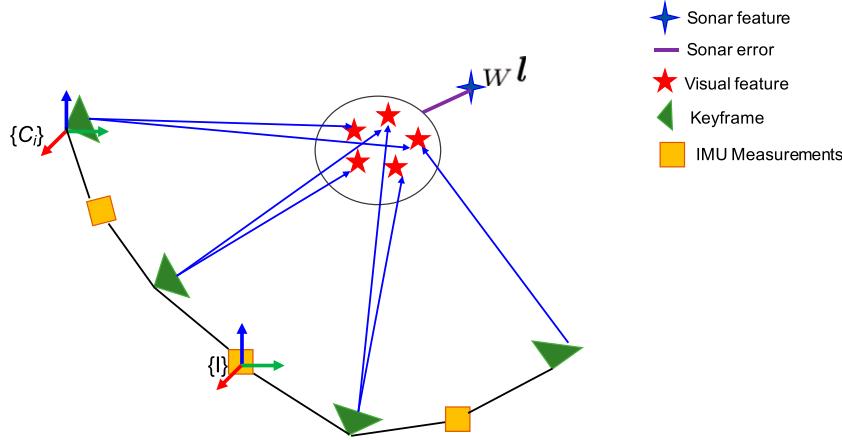


Figure 6. DPP-sonar error formulation.

function f with zero *mean* and \mathbf{W}_t^k *variance*, and the conditional covariance $\mathbf{Q}(\delta\hat{\mathbf{p}}^k | \mathbf{z}_t^k)$, updated iteratively as new sensor measurements are integrated

$$f(\mathbf{e}_t^k | {}_W \mathbf{p}_I^k) \approx \mathcal{N}(\mathbf{0}, \mathbf{W}_t^k) \quad (21)$$

The information matrix is

$$\mathbf{P}_t^k = \mathbf{W}_t^{k-1} = \left(\frac{\partial \mathbf{e}_t^k}{\partial \delta\hat{\mathbf{p}}^k} \mathbf{Q}(\delta\hat{\mathbf{p}}^k | \mathbf{z}_t^k) \frac{\partial \mathbf{e}_t^k}{\partial \delta\hat{\mathbf{p}}^k} \right)^{-1} \quad (22)$$

The Jacobian can be derived by differentiating the expected *range* r measurement with respect to the robot position

$$\frac{\partial \mathbf{e}_t^k}{\partial \delta\hat{\mathbf{p}}^k} = \left[\frac{-l_x + {}_W p_x}{r}, \frac{-l_y + {}_W p_y}{r}, \frac{-l_z + {}_W p_z}{r} \right] \quad (23)$$

Water-depth error term formulation

The pressure sensor provides accurate depth measurements based on the water pressure. Water-depth values are extracted along the *gravity* direction which is aligned with the z of the world W —observable due to the tightly-coupled IMU integration. The depth data at time k is given by

$${}_W p_{zD}^k = d^k - d^0 \quad (24)$$

More precisely, ${}_W p_{zD}^k = (d^k - d^0) + \text{init_disp_from_IMU}$ to account for the initial displacement along z axis from IMU, which is the main reference frame used by visual SLAM to track the sensor suite/robot.

With depth measurement z_u^k , the depth error term $e_u^k({}_W p_{zI}^k, z_u^k)$ can be calculated as the difference between the robot position along the z direction and the depth data. The error term can be defined as

$$e_u^k({}_W p_{zI}^k, z_u^k) = |{}_W p_{zI}^k - {}_W p_{zD}^k| \quad (25)$$

The weight P_u^k is calculated using the noise variance of the sensor following a similar approach as the sonar, and the Jacobian is straight-forward to derive.

5. Initialization: Two-step scale refinement

Tightly-coupled non-linear systems require a robust and accurate initialization for a successful state estimation, as described in (Mur-Artal and Tardós, 2017; Qin et al., 2018). Our comparative study of visual-inertial based state estimation systems in (Joshi et al., 2019), for underwater environments, shows that most of the state-of-the-art systems fail to initialize or make a wrong initialization leading to divergence of the state estimation process. In this work, we propose a method for robust initialization, which uses camera, IMU, and depth estimate from the water-pressure sensor. Using all these three sensors introduces constraints on *scale*, allowing a more accurate estimation during initialization. While acoustic range measurements are used by the tightly-coupled optimization, they have not been used for initialization, because of the data association challenge described in the previous section: if the robot is not moving, there is no match between acoustic range and visual features—see Figure 3. As such, if DPP-sonar were used for initialization, considering the sensor configuration, there would be a significant delay to initialize and an area would be un-mapped until DPP-sonar and camera acquire a common field-of-view.

The proposed initialization works as follows. First, the system requires a minimum number of visual features to track (in our experiments 15 worked well). This requirement avoids the initialization in a featureless scenario, for example, water with a few floating particulates as features. Second, the initial scale from the stereo vision is refined in two steps.

The first step uses the accurate depth measurement provided by the pressure sensor to correct the initial scale factor estimated from the camera. In particular, including a

scale factor s_1 , the transformation between camera C_i and depth sensor D can be expressed as

$${}_W p_{zD} = s_1 * {}_W p_{zC_i} + {}_W \mathbf{R}_{zC_i C_i} \mathbf{p}_D \quad (26)$$

For frame k , solving the above equation for s_1 provides the first refinement r_1 of the initial camera scale ${}_W \mathbf{p}_{r1C_i}$, that is

$${}_W \mathbf{p}_{r1C_i} = s_1 * {}_W \mathbf{p}_{C_i} \quad (27)$$

The second step aligns the refined measurement from camera in equation (27) with the IMU pre-integral values. Similar to the first step, the transformation between camera C_i and IMU I can be expressed as

$${}_W \mathbf{p}_I = s_2 * {}_W \mathbf{p}_{r1C_i} + {}_W \mathbf{R}_{C_i C_i} \mathbf{p}_I \quad (28)$$

with scale factor s_2 .

On top of the two-step scale refinement, our method approximates initial *velocity* and *gravity* vector similarly to (Qin et al., 2018). From the continuous formulation in Section, the discrete prediction of the state from IMU measurements between two consecutive frames k and $k+1$, considering a time interval $\Delta t_{k,k+1} \in [t_k, t_{k+1}]$ can be written as

$$\begin{aligned} {}_W \hat{\mathbf{p}}_I^{k+1} &= {}_W \mathbf{p}_I^k + {}_W \mathbf{v}_I^k \Delta t_{k,k+1} - \frac{1}{2} {}_W \mathbf{g} \Delta t_{k,k+1}^2 \\ &+ {}_W \mathbf{R}_I^k \boldsymbol{\alpha}_{I_k}^{k+1} \\ {}_W \hat{\mathbf{v}}_I^{k+1} &= {}_W \mathbf{v}_I^k - {}_W \mathbf{g} \Delta t_{k,k+1} + {}_W \mathbf{R}_I^k \boldsymbol{\beta}_{I_k}^{k+1} \end{aligned} \quad (29)$$

Re-arranging equation (29) with respect to $\boldsymbol{\alpha}_{I_k}^{k+1}$, $\boldsymbol{\beta}_{I_k}^{k+1}$ which are IMU pre-integration terms representing the motion between k and $k+1$ within $\Delta t_{k,k+1}$, results in

$$\begin{aligned} \boldsymbol{\alpha}_{I_k}^{k+1} &= {}_I \mathbf{R}_W^k \left({}_W \hat{\mathbf{p}}_I^{k+1} - {}_W \mathbf{p}_I^k - {}_W \mathbf{v}_I^k \Delta t_{k,k+1} \right. \\ &\quad \left. + \frac{1}{2} {}_W \mathbf{g} \Delta t_{k,k+1}^2 \right) \\ \boldsymbol{\beta}_{I_k}^{k+1} &= {}_I \mathbf{R}_W^k \left({}_W \hat{\mathbf{v}}_I^{k+1} - {}_W \mathbf{v}_I^k + {}_W \mathbf{g} \Delta t_{k,k+1} \right) \end{aligned} \quad (30)$$

In equation (30), substituting ${}_W \hat{\mathbf{p}}_I^{k+1}$ and ${}_W \mathbf{p}_I^k$ by equation (28), we can estimate $\boldsymbol{\chi}_S = [\mathbf{v}_I^k, \dots, \mathbf{v}_I^{k+n}, {}_W \mathbf{g}, s_2]$ by solving the linear least square problem in the following form

$$\min_{\boldsymbol{\chi}_S} \sum_{k \in K} \left\| \hat{\mathbf{z}}_{S_k}^{k+1} - \mathbf{H}_{S_k}^{k+1} \boldsymbol{\chi}_S \right\|^2 \quad (31)$$

$$\text{where } \hat{\mathbf{z}}_{S_k}^{k+1} = \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{I_k}^{k+1} - {}_I \mathbf{R}_W^k {}_W \mathbf{R}_{C_i C_i}^{k+1} \mathbf{p}_I^{k+1} + {}_I \mathbf{R}_{C_i C_i}^k \mathbf{p}_I^k \\ \hat{\boldsymbol{\beta}}_{I_k}^{k+1} \end{bmatrix}$$

and $\mathbf{H}_{S_k}^{k+1} =$

$$\begin{bmatrix} -\mathbf{I} \Delta t_{k,k+1} & \mathbf{0} & \frac{1}{2} {}_I \mathbf{R}_W^k \Delta t_{k,k+1}^2 & {}_I \mathbf{R}_W^k ({}_W \mathbf{p}_{r1C_i}^{k+1} - {}_W \mathbf{p}_{r1C_i}^k) \\ -\mathbf{I} & {}_I \mathbf{R}_W^k {}_W \mathbf{R}_I^{k+1} & {}_I \mathbf{R}_W^k \Delta t_{k,k+1} & \mathbf{0} \end{bmatrix}$$

6. Loop-closing and relocalization

Any sliding window and marginalization based optimization method suffers from drift on the pose estimate, which accumulates over time. To eliminate this drift and to achieve global consistency, we add a global optimization and relocalization scheme. We adapt DBoW2 (Gálvez-López and Tardós, 2012), a bag of binary words (BoW) place recognition module, and augment OKVIS for loop detection and relocalization. The BoW database contains the descriptors of the keypoints detected in each keyframe during the local tracking. The loop closure step will use the existing features detected during the tracking step.

Our method maintains a pose-graph representing the connection between keyframes, where a node represents a keyframe and an edge between two keyframes exists if the matched keypoints ratio between them is more than 0.75. Our experiments show that the resulting pose graph is very sparse. Using such a graph, with each new keyframe, the loop-closing module looks only for candidates in the BoW database, which are outside the current marginalization window and have a score greater than or equal to that of the neighbor keyframes of the node checked in the pose-graph. If a loop is detected, the method retains the candidate with the highest score and adds a connection between the current keyframe in the local window and the loop candidate keyframe, with their feature correspondences. Accordingly, the pose-graph is updated with loop information. A 2D-2D descriptor matching and then a geometric validation is performed via a 3D-2D matching between the known landmark in the current window keyframe and the loop candidate with outlier rejection by PnP RANSAC.

The loop detection triggers the global relocalization module, which performs an alignment between the current keyframe pose in the local window with the pose of the loop keyframe in the pose-graph. This alignment is communicated to the windowed sonar-visual-inertial-pressure optimization thread, as a drift correction. The estimate is further improved with an additional optimization step—similar to equation (3): the DPP-sonar and reprojection error terms are calculated considering the matched landmarks with loop candidate; see equation (32)

$$J(\mathbf{x}) = \sum_{i=1}^n \sum_{k=1}^K \sum_{j \in \text{Loop}(i,k)} \mathbf{e}_r^{i,j,k^T} \mathbf{P}_r^k \mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1} \mathbf{e}_t^{k^T} \mathbf{P}_t^k \mathbf{e}_t^k \quad (32)$$

After loop detection, a 6-DoF (position and rotation, $\mathbf{x}_T = [\mathbf{x}_p, \mathbf{x}_q]$) pose-graph optimization takes place over relative constraints between poses to correct the drift. The relative transformation between two poses \mathbf{T}_i and \mathbf{T}_j for current keyframe i in the current window and keyframe j (either loop candidate keyframe or connected keyframe) can be calculated from $\Delta \mathbf{T}_{ij} = \mathbf{T}_j \mathbf{T}_i^{-1}$. The error term, $\mathbf{e}_{\mathbf{x}_T}^{i,j}$ between keyframes i and j is formulated minimally in the tangent space

$$\mathbf{e}_{\mathbf{x}_T}^{i,j} = \Delta \mathbf{T}_{ij} \hat{\mathbf{T}}_i \hat{\mathbf{T}}_j^{-1} \quad (33)$$

where (\cdot) denotes the estimated values obtained from local sonar-visual-inertial-depth optimization. The cost function to minimize is given by

$$J(\mathbf{x}_T) = \sum_{i,j} \mathbf{e}_{\mathbf{x}_T}^{i,j T} \mathbf{P}_{\mathbf{x}_T}^{i,j} \mathbf{e}_{\mathbf{x}_T}^{i,j} + \sum_{(i,j) \in \text{Loop}} \rho\left(\mathbf{e}_{\mathbf{x}_T}^{i,j T} \mathbf{P}_{\mathbf{x}_T}^{i,j} \mathbf{e}_{\mathbf{x}_T}^{i,j}\right) \quad (34)$$

where $\mathbf{P}_{\mathbf{x}_T}^{i,j}$ is the information matrix set to identity, as in (Strasdat, 2012), and ρ is the Huber loss function to down-weight any incorrect loops.

7. Experimental results

We validate SVIn2, the proposed state estimation system, first on a standard dataset, to ensure that loop closure and the initialization work also above water. During this validation, SVIn2 is also compared to other state-of-the-art methods—that is, VINS-Mono (Qin et al., 2018), the basic OKVIS (Leutenegger et al., 2015), and the MSCKF (Mourikis and Roumeliotis, 2007) implementation from the GRASP lab (Research group of Prof. Kostas Daniilidis, 2018). Second, we test the proposed approach on several different underwater datasets collected with a custom-made sensor suite (Rahman et al., 2018a) and an Aqua2 AUV (Dudek et al., 2005). Third, we perform an ablation study to assess the contribution of each sensor. Finally, we validate the proposed approach using a publicly available underwater dataset.

The experiments were run on a desktop computer with an Intel i7-7700 CPU @ 3.6 GHz, 32 GB RAM, running Ubuntu 16.04 and ROS Kinetic, and on an Intel NUC that is on-board of the robots, with an Intel i3-6100U CPU @ 2.3 GHz and 16 GB RAM.

7.1. Validation on visual-inertial benchmark

As standard benchmark dataset used by many visual-inertial state estimation systems—including OKVIS (Stereo), VINS-Mono, and MSCKF—we consider EuRoC dataset (Burri et al., 2016), composed of sensor data collected with an aerial drone. We disable water-depth and DPP-sonar integration in our method and only assess the loop-closure scheme, as such sensors are not available in EuRoC.

Following the current benchmarking practices, ground truth and estimated trajectory are aligned, by minimizing the least mean square errors between estimate/ground-truth locations, which are temporally close, varying rotation and translation (Umeyama, 1991). After such an alignment, the error for each pair of ground truth/estimated pose is calculated—the *Absolute Trajectory Error* (ATE). The Root Mean Square Error (RMSE) of the ATE is calculated for the translation—shown in Table 1 for several Machine Hall sequences in the EuRoC dataset. For each package, every sequence has been run 5 times and the best run (according to RMSE) is taken. Our method shows reduced RMSE in every sequence compared to OKVIS. This validates the improvement of pose-estimation after loop-closing. SVIn2 has also lower RMSE than MSCKF and comparable results to VINS-Mono. Figure 7 shows the trajectories for each method together with the ground truth for one of the *Machine Hall* sequences.

More recently, ORB-SLAM3 (Campos et al., 2021), Kimera (Rosinol et al., 2020), VI-DSO (Von Stumberg et al., 2018) report even lower RMSE, with ORB-SLAM3 showing the lowest in EuRoC dataset. As such, we compared performance of SVIn2 with ORB-SLAM3 in the underwater datasets.

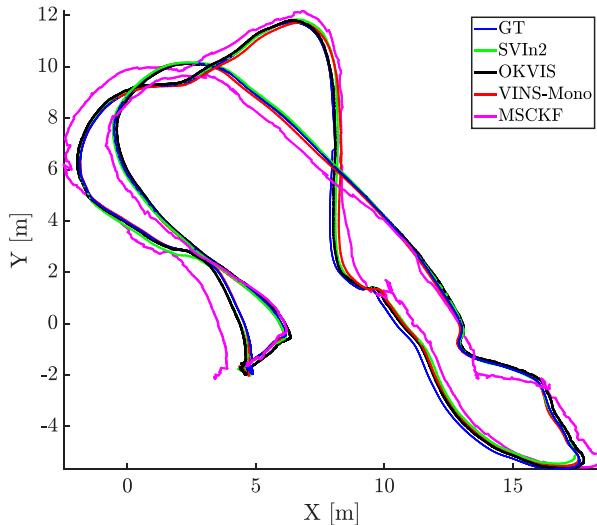


Figure 7. Trajectories on the MH 04 sequence of the EuRoC dataset.

Table 1. The best Absolute Trajectory Error (RMSE) in meters for each Machine Hall EuRoC sequence.

	SVIn2	OKVIS(stereo)	VINS-Mono	MSCKF
MH 01	0.13	0.15	0.07	0.21
MH 02	0.08	0.14	0.08	0.24
MH 03	0.07	0.12	0.05	0.24
MH 04	0.13	0.18	0.15	0.46
MH 05	0.15	0.24	0.11	0.54

Table 2. The ATE RMSE in meters and the tracking duration as a percentage of the total trajectory for each underwater dataset compared to the COLMAP estimated trajectory (the lowest RMSE and longest percentage tracking duration are shown in **bold**). SVIn2, OKVIS, and ORB-SLAM3 all use stereo-inertial data; VINS-Mono uses monocular-inertial data. COLMAP only tracked the complete path for Cavern1 and Cavern2, while tracked partially for Bus and Cemetery. All packages are tested both in the presence and absence of CLAHE filter.

	With CLAHE				Without CLAHE			
	SVIn2	OKVIS	VINS-Mono	ORB-SLAM3	SVIn2	OKVIS	VINS-Mono	ORB-SLAM3
Bus (partial)	0.2092	0.5109	0.0742(part.)	1.6672(part.)	0.6822(part.)	0.7775(part.)	0.0747(part.)	-
89%	100%	100%	21%	79%	70%	70%	20.8%	0%
Cavern1	0.1243	0.2089	1.0155(part.)	0.2523(part.)	0.1096	0.1154	0.8670	0.1553(part.)
100%	100%	100%	99%	85%	100%	100%	100%	94%
Cavern2	0.1722	0.3814	0.3464(part.)	2.4199(part.)	0.1237	0.3725	0.7839(part.)	0.3813(part.)
100%	100%	100%	26.7%	28.7%	100%	100%	93%	88%
Cemetery (partial)	0.2421	1.0868	1.2291	0.8143(part.)	0.6996(part.)	1.1026(part.)	-	0.2165(part.)
97%	100%	100%	100%	86%	65.5%	65.5%	0%	30%

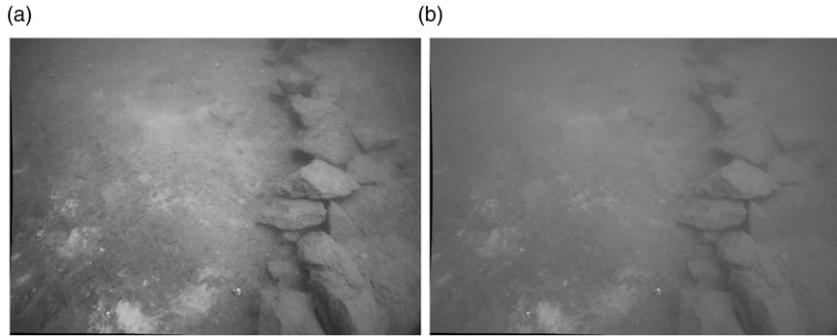


Figure 8. (a) The pre-processing result with CLAHE filter and (b) the corresponding raw image on fake cemetery dataset.

7.2. Validation on underwater datasets

SVIn2 is tested on four different underwater datasets, where DPP-sonar and water-depth sensors are available and can be fused together with the visual-inertial data, to fully exploit our system. After describing the datasets and the experimental setup, we evaluate the trajectories. Since there is no ground truth in unstructured underwater environments, we evaluate the performance first in comparison with a global bundle adjustment system, COLMAP (Schönberger et al., 2016); and second using fiducial tags placed in the environment. The tags are placed securely in the environment and are observed multiple times during the experiments.

7.2.1. Dataset description. The experimental data were collected using a custom-made sensor suite (Rahman et al., 2018a) (see Figure 3) and an Aqua2 robot (Dudek et al., 2005). Both are equipped with a stereo camera, an IMU, and a pressure sensor. The custom-made sensor suite additionally contains a DPP-sonar. More specifically, two USB-3 uEye cameras in a stereo configuration provide data at 15 Hz; a MicroStrain 3DM-GX4-15 IMU generates inertial data at 100 Hz; the Bluerobotics Bar30 pressure sensor provides pressure data at 1 Hz; and an IMAGENEX 831L mechanical scanning profiling sonar sensor acquires a full

360° scan every 4 s. An Intel NUC running Linux and ROS consolidates all the data. A video light is attached to the sensor suite unit to provide artificial illumination of the scene. The DPP-sonar is mounted on top of the main unit which contains the remaining electronics.

The datasets have been collected in three environments with different characteristics:

- *Bus*: a sunken bus in Fantasy Lake (NC), where data was collected by a diver with the custom-made underwater sensor suite (Rahman et al., 2018a). The diver started from outside the bus, performed a loop around, entered in it from the back door, exited across, and finished at the front-top of the bus. The images are affected by haze, strong lighting variations, and low visibility.
- *Cavern1* and *Cavern2*: a diver collected data with the same underwater sensor suite from an underwater cavern in Ginnie Springs (FL). The datasets contain several loops, around one spot in Cavern1 and two spots in Cavern2. The environment is characterized by complete absence of natural light and is illuminated by the video light attached to the sensor suite.
- *Cemetery*: an AUV—Aqua2 robot—collected data over a fake underwater cemetery in Lake Jocassee

(SC) and performed several loops around the tombstones in a square pattern. The visibility, as well as brightness and contrast, was very low.

7.2.2. Trajectory evaluation using COLMAP as a comparative baseline. Given the absence of GPS in underwater environments and of a motion capture system that can work in unstructured environments, we use COLMAP (Schönberger et al., 2016), an open-source Structure-from-Motion library, to generate comparative baseline trajectories for each underwater dataset. Loop detection was enabled via vocabulary tree search. COLMAP performs best among the conventional state-of-the-art multi-view stereo algorithms, as it uses tight integration of multiple techniques—for example, robust neighbor view selection and incorporation of visibility constraints. COLMAP provides an estimation on the shape of the trajectories; however, such trajectories cannot be considered as absolute ground truth. Even after introducing the stereo baseline constraint, we observed that the global optimization reduced the reprojection error, but did not converge. In addition, COLMAP could produce only partial trajectories for some of the datasets due to the water turbidity, low visibility, and lack of good features to track for a longer period in the scene—an indication that vision-only state estimation for underwater environments is not reliable. Therefore, we aligned the estimated trajectories with *scale* for our system as well as the other open-source visual-inertial packages with respect to COLMAP and calculated ATE for each of them.

Table 2 shows the RMSE of the ATE for the different underwater datasets both in the presence and absence of CLAHE filter. MSCKF has been excluded from the table as it failed to track in all of them. VINS-Mono and ORB-SLAM3 could track only partially in most of the datasets and the RMSE is reported only for that part of the trajectory. SVIn2 has the lowest RMSE and tracks successfully in each of the datasets. All four systems—SVIn2, OKVIS, VINS-Mono, and ORB-SLAM3—have improved performance (i.e., lower ATE RMSE and longer tracking duration) when CLAHE is used for *Bus* and *Cemetery* datasets, while CLAHE causes degradation of performance for *Cavern1* and *Cavern2* datasets. We observed that, CLAHE filter improves the performance by helping in feature detection and tracking (see **Figure 8**) in the presence of haze and low contrast. Such datasets can be identified by checking if the *image histogram* lies within a narrow region. We advise to use CLAHE only for those datasets.

Figures 9–12 show the trajectories from SVIn2, OKVIS, ORB-SLAM3, and VINS-Mono together with the trajectories generated by COLMAP in the datasets just described. For a fair comparison, when the trajectories were compared against each other, DPP-sonar and pressure data were disabled in SVIn2.

Figure 9 shows the results for the submerged bus dataset. In particular, even using a *histogram equalization* or a

CLAHE filter, VINS-Mono lost track when the exposure increased for quite some time and tracked only 21% of the total duration—the reported RMSE is calculated for this tracked part only. Without CLAHE, VINS-Mono produces similar result as above. ORB-SLAM3 trajectory showed high drift when the exposure increased and lost track just after entering the bus from the back door, resulting in 79% of tracking duration with CLAHE. It tried to re-initialize, but it was not able to track successfully. Without CLAHE, ORB-SLAM3 cannot track at all and loses track immediately after initialization. Even if the scale drifted, OKVIS was able to track using a CLAHE filter in the image pre-processing step. Without the filter, it lost track at the high exposure location. Our proposed method was able to track, detect, and correct the loop, successfully with CLAHE. Without CLAHE, it tracked partially with a 70% duration.

In Cavern1—see **Figure 10**—VINS-Mono tracked successfully the whole time. However, as can be noticed in **Figure 10(c)**, the scale was incorrect based on empirical observations during data collection. ORB-SLAM3 lost track two times, each for about 10–20 s, but was able to relocalize. OKVIS instead produced a good trajectory, and SVIn2 was also able to detect and close the loops. CLAHE has not been used for any of the SLAM systems mentioned above, including SVIn2: the water was clear and CLAHE produces worse results.

In Cavern2—see **Figure 11**—VINS-Mono lost track at the beginning, reinitialized, was able to track for some time, and detected a loop, before losing track again. VINS-Mono had even worse behavior if the images were pre-processed with different filters. ORB-SLAM3 lost track while taking a turn, and recovered using the relocalization module—leading to a 88% total tracking duration. OKVIS tracked well, but as drifts accumulated over time, it was not able to join the current pose with a previous pose where a loop was expected. SVIn2 was able to track and reduce the drift in the trajectory with successful loop closure. CLAHE has not been used in any systems for the same reason as for Cavern1.

In the Cemetery dataset—see **Figure 12**—both VINS-Mono and OKVIS were able to track with CLAHE, but VINS-Mono was not able to reduce the drift in trajectory, while SVIn2 (with CLAHE) was able to correct the loops. Without the filter, none of the above systems works well. ORB-SLAM3 was able to track partially, 86% of total duration with CLAHE. Without the filter, ORB-SLAM3’s tracking duration is 30%.

We also recorded the mean processing time per frame of each SLAM system and reported the corresponding runtime analysis in **Table 3**. The results show that SVIn2 has comparable processing time with other SLAM systems, despite the additional sensors. Note that OKVIS has a lower processing time compared to others because it does not perform any pose graph optimization or loop closure, resulting in higher drift than the other SLAM systems.

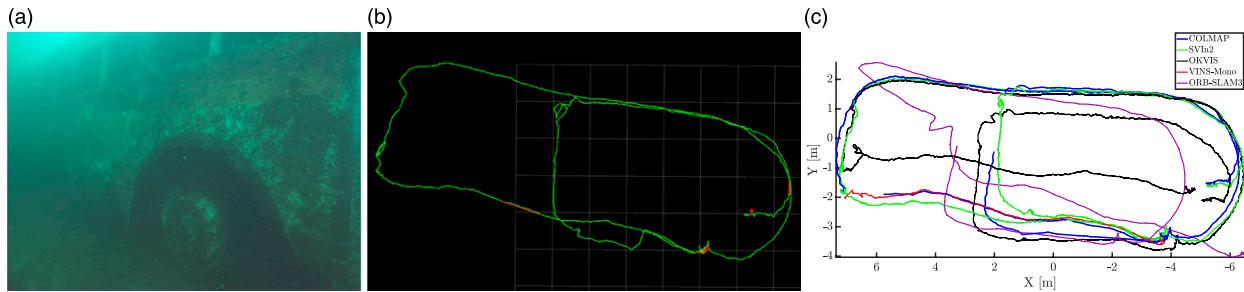


Figure 9. (a) Submerged bus, Fantasy Lake, NC, USA with a 53 m trajectory; (b) trajectories from SVIn2 with all sensors enabled shown in rviz; and (c) *scale* aligned trajectories with COLMAP (comparative baseline), SVIn2 with DPP-sonar and water-depth disabled, OKVIS, VINS-Mono, and ORB-SLAM3 (CLAHE has been used in all systems for improved visibility) are displayed.

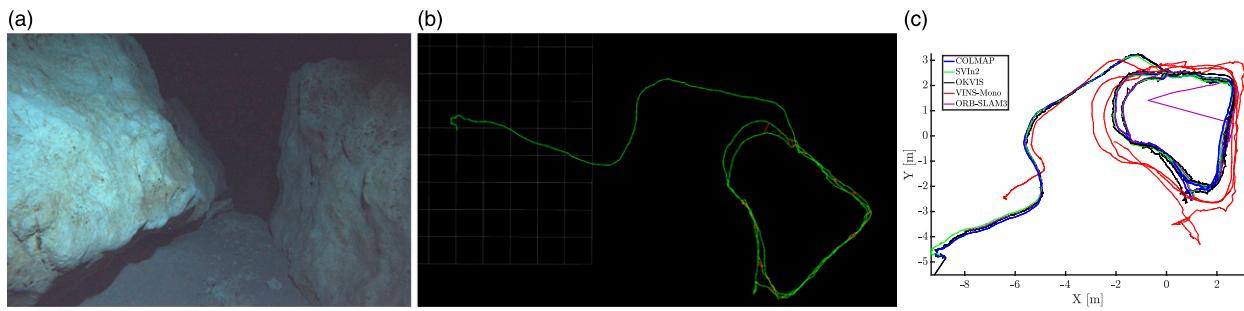


Figure 10. (a) Cave environment, Ballroom, Ginnie Springs, FL, USA, with a unique loop covering a 87 m trajectory; (b) trajectories from SVIn2 with all sensors enabled shown in rviz; and (c) *scale* aligned trajectories with COLMAP (comparative baseline), SVIn2 with DPP-sonar and water-depth disabled, OKVIS, VINS-Mono, and ORB-SLAM3 (CLAHE has not been used in any of the systems) are displayed.

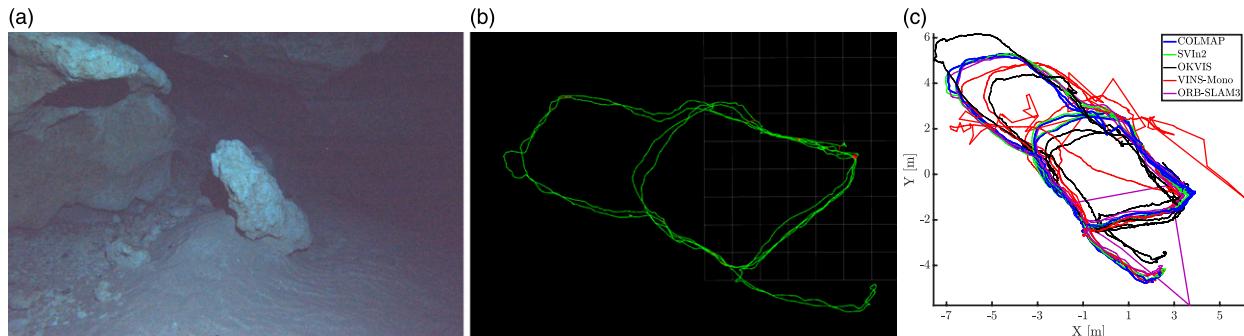


Figure 11. (a) Cave environment, Ballroom, Ginnie Springs, FL, USA, with two loops in different areas covering a 155 m trajectory; (b) trajectories from SVIn2 with all sensors enabled shown in rviz; and (c) *scale* aligned trajectories with COLMAP (comparative baseline), SVIn2 with DPP-sonar and water-depth disabled, OKVIS, VINS-Mono, and ORB-SLAM3 (CLAHE has not been used in any of the systems) are displayed.

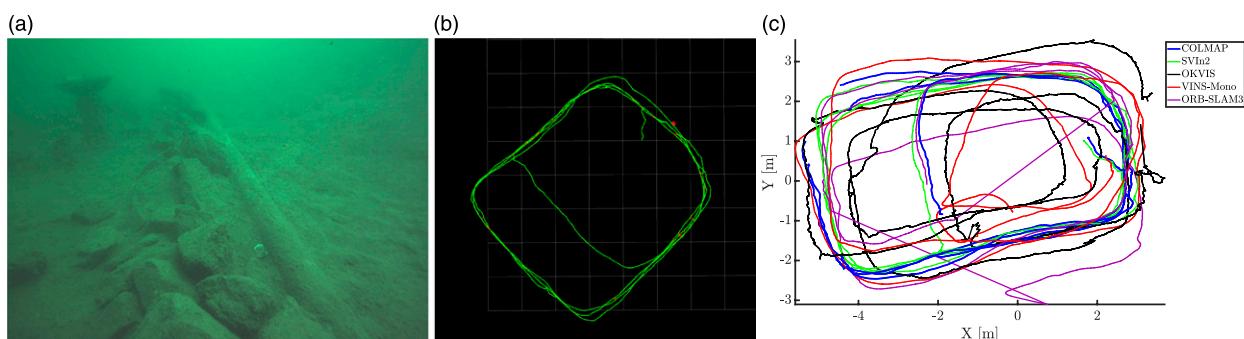


Figure 12. (a) Aqua2 in a fake cemetery, Lake Jocassee, SC, USA with a 80 m trajectory; (b) trajectories from SVIn2 with visual, inertial, and water-depth sensor (no DPP-sonar data has been used) shown in rviz; and (c) *scale* aligned trajectories with COLMAP (comparative baseline), SVIn2 with DPP-sonar and water-depth disabled, OKVIS, VINS-Mono, and ORB-SLAM3 (CLAHE has been used in all systems for improved visibility) are displayed.

Table 3. Run-time comparison of SVIn2 with other SLAM methods on a desktop computer with an Intel i7-7700 CPU @ 3.6 GHz, 32 GB RAM.

	Mean processing time (ms)
SVIn2 (all sensors)	118
ORB-SLAM3 (stereo-in)	115
OKVIS (stereo)	57
VINS-Mono	198

7.2.3. AR-tag based validation. In the absence of absolute ground-truth, we used 3D landmark-based validation with AR-tags (fiducial markers) to quantify the accuracy of our SLAM method. More specifically, we observe how much the pose of the AR-tags deviates from their mean over the whole length of the trajectory: if the drift in trajectory is corrected properly, we should observe the marker at the same location in multiple visits during the whole experiment. Note that the accuracy of pose of the marker is also subject to the accuracy of relative pose estimation between the marker and camera.

As for the experimental setup, only Cavern-2 dataset had a set of six AR-tags printed on waterproof paper placed at a fixed location in the cavern. The dataset contains five loops, where the tags can be observed. To determine the relative

pose between camera and the tags, we used ROS *ar_track_alvar*¹ package with a very low error in marker detection (five of the six tags were used due to a discoloration on the sixth tag). Once the relative pose information from *ar_track_alvar* is obtained, the pose of the marker can be obtained by a simple geometric transformation. Formally, ${}_W\mathbf{T}_M^k = {}_W\mathbf{T}_{C_i C_i}^k \mathbf{T}_M^k$ where ${}_W\mathbf{T}_M^k$ is the marker pose in World coordinate frame W at time k , ${}_W\mathbf{T}_{C_i}^k$ is the pose of the camera C_i in W at time k (produced by SLAM/odometry system), and ${}_{C_i}\mathbf{T}_M^k$ is the relative transformation between camera C_i and marker M at time k (produced by *ar_track_alvar*).

Figure 13 shows the displacement from the mean position of the markers over the whole length of the trajectory for each package plotted over time. The tag was detected at five distinct instances. Table 4 shows the summary of the standard deviation (SD) for translation and orientation components. SVIn2 is the one with the lowest standard deviation. This result indicates that SVIn2 produces the most consistent estimation, sometimes even having a better estimate compared to COLMAP.

In Figure 14, the poses from where the markers were observed in SVIn2, OKVIS, VINS-Mono, and ORB-SLAM3 together with the location of the markers (in magenta) are shown. Figure 14(g) shows that, during the five loops the tags appear together in SVIn2—indicating very low drift in the trajectory. On the other hand, OKVIS

Table 4. Standard deviation in translation and rotation for the detected tags and average distance error (lowest standard deviation is marked in **bold**). CLAHE has not been used for any of the systems.

	t_x (m)	t_y (m)	t_z (m)	Avg dist. error (m)	Yaw, ψ (deg)	Pitch, θ (deg)	Roll, ϕ (deg)
COLMAP	0.069	0.022	0.065	0.048	1.61	0.67	1.08
SVIn2	0.036	0.032	0.038	0.026	5.12	1.00	5.79
OKVIS (Stereo)	0.498	0.578	0.120	0.422	23.08	12.48	20.52
VINS-Mono	0.316	0.165	0.155	0.265	58.55	6.70	31.05
ORB-SLAM3 (Stereo-in)	0.464	1.365	0.693	0.837	54.52	34.26	11.57

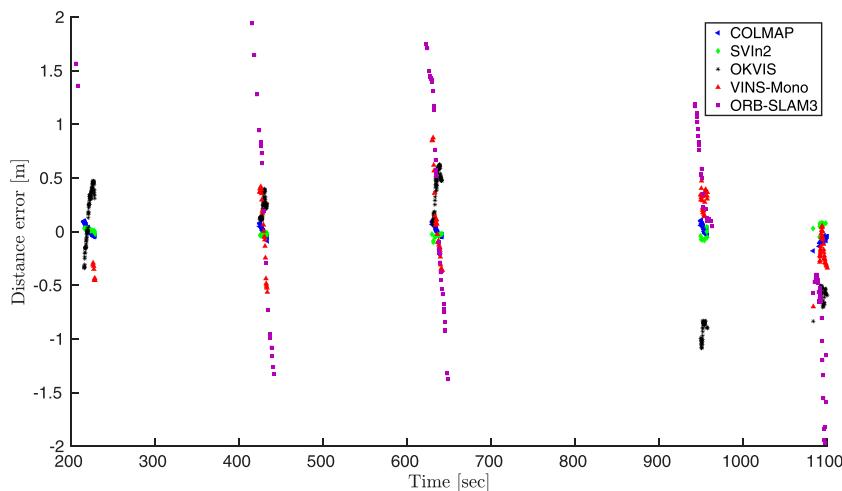


Figure 13. Time versus displacement error of tags.

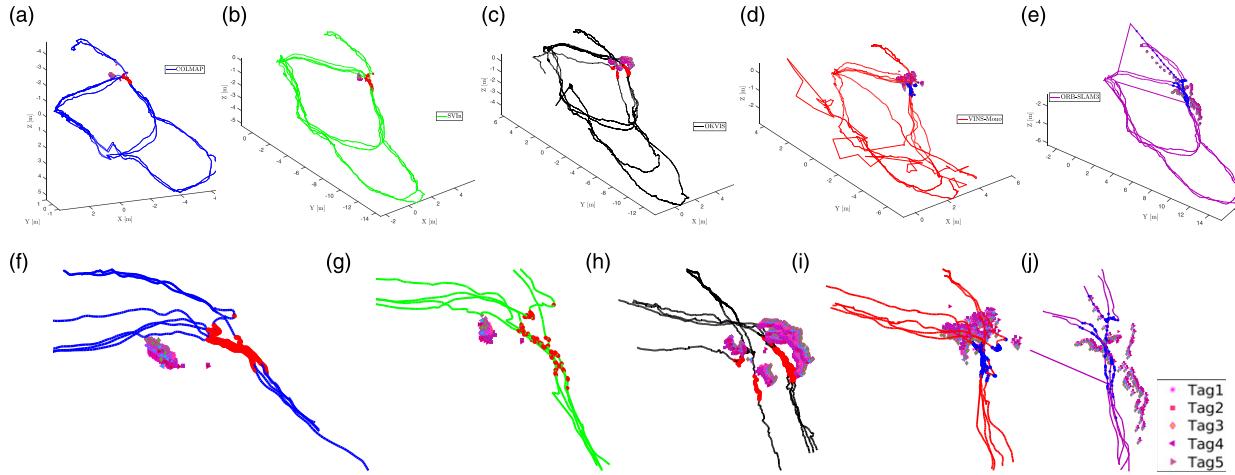


Figure 14. (a)–(e) trajectories from COLMAP, SVIn2, OKVIS, VINS-Mono, and ORB-SLAM3, respectively, with poses (solid circles) on the trajectories) from where the tags are observed and poses of the tags. The corresponding zoomed-in version (f)–(j).

(Figure 14(h)), VINS-Mono (Figure 14(i)), and ORB-SLAM3 (Figure 14(j)) show higher error, with tags that are spread around.

7.3. Ablation study

We performed two ablation studies on (1) the initialization and (2) the SLAM system as a whole, looking at the impact of the different sensors.

7.3.1. Initialization with different sensor configuration. We study how different combinations of sensors affect the pose estimation accuracy with our proposed initialization

method. To isolate the immediate initialization effect from loop closure, in this analysis, we calculate the ATE RMSE of SVIn2 compared to COLMAP within a small area rather than the whole trajectory. In particular, we considered the portion of the Cavern2 dataset, where two tags are placed at a manually measured distance of approx. 7 m. **Table 5** shows the *Stereo* setup to have the most impact in improving the accuracy, while the pressure sensor provides a slight improvement. The *Stereo + Pressure* combination has the lowest RMSE.

7.3.2. SVIn2 with different combination of sensors and loop closure. This section studies how the presence/absence of different sensors and the loop-closure component affects the pose estimation accuracy. As we don't have an absolute ground-truth trajectory underwater, we use the displacement of the 3D landmarks (AR-tags) in the Cavern2 dataset as a metric to evaluate the accuracy of state estimation. **Table 6** shows the average distance error and standard deviation in each translation and rotation component of the AR-tags with/without loop-closure component, with/without DPP-sonar, and with/without pressure sensor in SVIn2.

Table 5. Ablation study for the initialization method using COLMAP (lowest RMSE is marked in **bold**).

SVIn2 config	ATE RMSE (m)
Mono	0.0401
Mono-pressure	0.0385
Stereo	0.0224
Stereo-pressure	0.0205

Table 6. Ablation Study using displacement of 3D landmarks as evaluation metric (lowest standard deviations are shown in **bold**).

Loop closure	DPP-sonar	Pressure sensor	SD t_x (m)	SD t_y (m)	SD t_z (m)	Avg dist error (m)	SD yaw, ψ (deg)	Pitch, θ (deg)	Roll, ϕ (deg)
✓	✗	✗	0.0364	0.0325	0.0380	0.0261	5.1289	1.0040	5.7990
✓	✓	✗	0.0277	0.0243	0.0257	0.0198	5.9044	1.3722	5.6155
✓	✗	✓	0.0315	0.0307	0.0129	0.0313	6.0991	0.9450	6.4602
✓	✓	✓	0.0312	0.0192	0.0223	0.0153	4.8941	1.3268	4.4823
✗	✗	✗	0.4983	0.5788	0.1203	0.4221	23.0878	12.4848	20.5298
✗	✓	✗	0.2778	0.5675	0.2454	0.2373	22.2291	12.4886	20.5340
✗	✗	✓	0.3951	0.6424	0.1580	0.2162	24.6572	12.4809	20.5273
✗	✓	✓	0.3091	0.5726	0.1198	0.1248	22.4677	12.4797	20.5280

Loop-closure (first 4 rows) has the most contribution towards producing a drift-free trajectory. Nevertheless, *loop-closure + DPP-sonar + pressure sensor* produces the least average distance error. As the error terms by DPP-sonar and pressure sensor create constraints only for the translation components of the robot, the rotational components (roll, pitch, yaw) are similar regardless of the use of the DPP-sonar and/or the pressure sensor.

Without loop-closure (last 4 rows), the contributions of DPP-sonar and pressure sensor become more visible. Using only the pressure sensor (second row from the bottom) shows reduced error along Z-axis. *DPP-sonar + pressure sensor* gives the least average distance error in absence of loop-closing. The rotational components are also similar

regardless of using DPP-sonar and/or pressure sensor for the same reason described above.

7.4. Validation on public underwater datasets

Ferrera et al. (2019) provided a set of underwater datasets obtained close to the seabed, named AQUALOC, collected by a Remotely Operated Vehicle (ROV) equipped with a monocular monochromatic camera, a MEMS-IMU, and a pressure sensor. Note that, while additional sensors could improve SVIn2 performance as shown in the ablation study, SVIn2 can work with different sensor configuration up to the minimal requirement which is a monocular camera and an IMU. Thus, SVIn2 is applicable in AQUALOC.

The datasets are characterized by turbidity, backscattering effect, and clouds of sediment stirred up by the ROV—Figure 15 shows a representative picture of the sites. COLMAP trajectories were also provided as “ground truth” to compare and evaluate the performance of SLAM systems. Note that in a few sequences—for example, sequences 4, 6, and 7—the “ground truth” trajectories produced by COLMAP are not continuous, providing partial information on the camera poses. As the generated data were the result of continuous motions of the ROV, the discontinuities represent failures of the COLMAP state estimation process. We ran SVIn2 on the archaeological sites located at a depth of approximately 270 m and 380 m. SVIn2 was able to generate complete trajectories for all of the sequences without losing track. The RMSE error was typically around 2% of its length, with the lowest in sequence 8 with an error

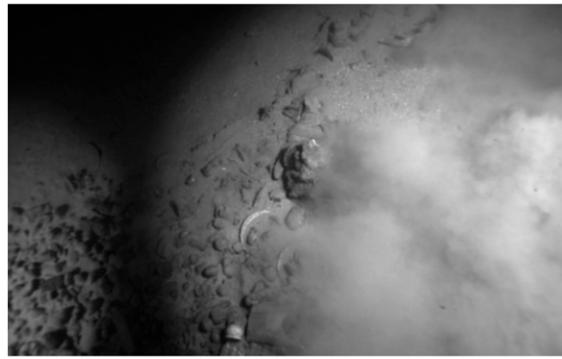


Figure 15. A sample image from AQUALOC Archaeological sites sequences (Ferrera et al., 2019), affected by sandy cloud, low and repetitive texture, and lack of light and features.

Table 7. The RMSE in meters and the error percentage over the full trajectory length for each AQUALOC Archeological sites sequences. Note, the camera pose estimate from COLMAP in Sequence 4 is highly discontinuous and does seem to track a very small portion of the trajectory, thus the RMSE is not calculated.

Sequence #	1	2	3	4	5
SVIn2 RMSE(m)	0.2311	2.4403	0.2801	—	2.7213
Error %	2.0	3.79	2.617	—	6.48
Sequence #	6	7	8	9	10
SVIn2 RMSE(m)	0.6085	1.0526	0.2465	1.5092	2.3710
Error %	1.91	0.86	0.59	2.30	2.83

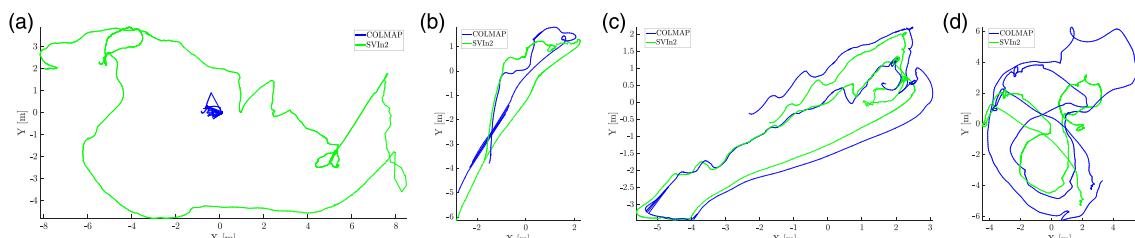


Figure 16. SVIn2 trajectories and provided COLMAP-produced “ground truth” trajectories alignment for Archaeological sequences 4 and 6 (a)-(b), respectively, showing discontinuity in the provided GT. (c) Sequence 8, SVIn2 shows low RMSE, (d) Sequence 10, SVIn2 shows high RMSE.



Figure 17. (a) Aqua 2 vehicle inside a cavern, Ginnie Springs, FL; (b) Aqua 2 vehicle over the Stavronikita shipwreck, Barbados; (c) BlueROV2 deployed inside the Ginnie Spring cavern.

of 0.5% and the highest in sequence 5 with 6.4%, as shown in Table 7.

With COLMAP producing up to scale trajectories, we *scale* align the estimated trajectories from SVIn2 and provided ground truth for all the sequences in the archaeological sites datasets except for sequence 4, as the provided “ground truth” for this dataset is highly discontinuous and hence has been plotted with their own scale without any RMSE calculation; see Figure 16.

In general, there is a need of robust public datasets to validate state estimation systems in underwater environments.

8. Conclusions and future work

This paper investigated the problem of Simultaneous Localization and Mapping in underwater environments, combining visual, inertial, acoustic, and water-pressure information. We focused on the design and development of a robust and accurate system that exploits the complementarity of different sensors, so that robots can operate autonomously in very harsh environments with robustness, safety, and reliability to accomplish a task in real-time with limited computational resources. The result is SVIn2—a tightly-coupled keyframe-based SLAM system which integrates all the above sensors and includes a robust initialization method by two-step scale refinement, loop-closure, and relocalization capabilities as a failure recovery mechanism. We have released the code of our system, which can work in different configurations, so that other researchers can use it according to the available sensors. Experimental results in challenging underwater environments including both publicly available datasets and collected underwater datasets prove the effectiveness of our system. The VIO part of the proposed approach provided improved performance on datasets collected with an inexpensive action camera (Joshi et al., 2022) over shipwrecks and inside underwater caves.

Future extensions include, but are not limited to, integration of other sensors typically used underwater—for example, DVL, USBL—cooperative localization and mapping, and the ability to relocalize when tracking loss happens. In the long term, SVIn2 will be used on AUVs jointly together with planning—as done in a preliminary path planner for AUVs (Xanthidis et al., 2020)—to enable

the safe operation of underwater vehicles, as those depicted in Figure 17, in a variety of underwater environments.

Acknowledgements

The authors would also like to acknowledge the help of the Woodville Karst Plain Project (WKPP) and El Centro Investigador del Sistema Acuífero de Quintana Roo A.C. (CINDAQ) in collecting data, providing access to challenging underwater caves, and mentoring us in underwater cave exploration. Thanks to Nare Karapetyan, Marios Xanthidis, Hunter Damron, John Rose, Steve Cox, and Casey McKinlay for their valuable support on data collection. Last but not least, we would like to thank Halcyon Dive Systems for their support with equipment.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by National Science Foundation (NSF 1943205, 1919647, 2024741, 2024541).

Note

1. https://wiki.ros.org/ar_track_alvar

ORCID iDs

Sharmin Rahman <https://orcid.org/0000-0002-4343-9561>
Alberto Quattrini Li <https://orcid.org/0000-0002-4094-9793>

References

- Agarwal S and Mierle K, Others (2015). Ceres Solver. <http://ceres-solver.org>
- Blanco JL (2010) A tutorial on SE (3) transformation parameterizations and on-manifold optimization. *University of Malaga, Technical Report* 3: 6.
- Bloesch M, Burri M, Omari S, et al. (2017) Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research (IJRR)* 36: 1053–1072.

- Burri M, Nikolic J, Gohl P, et al. (2016) The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research (IJRR)* 35(10): 1157–1163.
- Cadena C, Carlone L, Carrillo H, et al. (2016) Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Transactions on Robotics* 32(6): 1309–1332.
- Campos C, Elvira R, Rodríguez JJG, et al. (2021) ORB-SLAM3: an accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics* 37: 1874–1890.
- Carreras M, Ridao P, García R, et al. (2003) Vision-based localization of an underwater robot in a structured environment. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. Piscataway, NJ: IEEE, volume 1, 971–976.
- Corke P, Detweiler C, Dunbabin M, et al. (2007) Experiments with underwater robot localization and tracking. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE, 4556–4561.
- Davison AJ, Reid ID, Molton ND, et al. (2007) Monoslam: real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(6): 1052–1067.
- Delmerico J and Scaramuzza D (2018) A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE.
- Dudek G, Jenkin M, Prahacs C, et al. (2005) A visually guided swimming robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE, 1749–1754.
- Engel J, Koltun V and Cremers D (2018) Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(3): 611–625.
- Engel J, Schöps T and Cremers D (2014) LSD-SLAM: Large-scale direct monocular SLAM. Berlin, Germany: Springer, 834–849. *European Conference on Computer Vision (ECCV)*
- Eustice R, Singh H, Leonard JJ, et al. (2005) Visually navigating the RMS Titanic with SLAM information filters. In: *Robotics: Science and Systems*. Cambridge, MA: MIT Press, volume 2005, 57–64.
- Eustice RM, Singh H and Leonard JJ (2006) Exactly sparse delayed-state filters for view-based slam. *IEEE Transactions on Robotics* 22(6): 1100–1114.
- Fabbri C, Islam MJ and Sattar J (2018) Enhancing underwater imagery using generative adversarial networks. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE, 7159–7165.
- Fallon MF, Folkesson J, McClelland H, et al. (2013) Relocating underwater features autonomously using sonar-based SLAM. *IEEE Journal of Oceanic Engineering* 38(3): 500–513.
- Ferrera M, Creuze V, Moras J, et al. (2019) AQUALOC: an underwater dataset for visual–inertial–pressure localization. *The International Journal of Robotics Research* 38(14): 1549–1559.
- Fiala M (2005) ARTag, a fiducial marker system using digital techniques. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, volume 2, 590–596.
- Folkesson J, Leonard J, Leederkerken J, et al. (2007) Feature tracking for underwater navigation using sonar. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE, 3678–3684.
- Forster C, Carlone L, Dellaert F, et al. (2017a) On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics* 33(1): 1–21.
- Forster C, Zhang Z, Gassner M, et al. (2017b) SVO: semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics* 33(2): 249–265.
- Gálvez-López D and Tardós JD (2012) Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics* 28(5): 1188–1197.
- Gary M, Fairfield N, Stone WC, et al. (2008) 3D mapping and characterization of sistema Zacatón from DEPTHX (DEep Phreatic Thermal eXplorer). In: Proc. of KARST: Sinkhole Conference ASCE, Tallahassee, FL, 22–26 September 2008.
- Hong S, Chung D, Kim J, et al. (2019) In-water visual ship hull inspection using a hover-capable underwater vehicle with stereo vision. *Journal of Field Robotics* 36(3): 531–546.
- Hover FS, Eustice RM, Kim A, et al. (2012) Advanced perception, navigation and planning for autonomous in-water ship hull inspection. *The International Journal of Robotics Research (IJRR)* 31(12): 1445–1464.
- Huang G (2019) Visual-inertial navigation: a concise review. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE, 9572–9582.
- Imagenex Technology Corp (2022) 831L Digital pipe profiling sonar. Available at: <https://Imagenex.com/products/831l-pipe-profiling>
- Joe H, Cho H, Sung M, et al. (2021) Sensor fusion of two sonar devices for underwater 3d mapping with an AUV. *Autonomous Robots* 45: 543–560.
- Johannsson H, Kaess M, Englot B, et al. (2010) Imaging sonar-aided navigation for autonomous underwater harbor surveillance. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE, 4396–4403.
- Joshi B, Rahman S, Kalaitzakis M, et al. (2019) Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, 7221–7227.
- Joshi B, Xanthidis M, Rahman S, et al. (2022) High definition, inexpensive, underwater mapping. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, 23–27 May 2022.
- Kim A and Eustice RM (2013) Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Transactions on Robotics* 29(3): 719–733.
- Klein G and Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: *IEEE and ACM Int. Symp. on Mixed and Augmented Reality*. Piscataway, NJ: IEEE, 225–234.

- Lee CM, Lee PM, Hong SW, et al. (2005) Underwater navigation system based on inertial sensor and doppler velocity log using indirect feedback Kalman filter. *International Journal of Offshore and Polar Engineering* 15(02).
- Leonard JJ and Durrant-Whyte HF (2012) *Directed Sonar Sensing for Mobile Robot Navigation*. Berlin, Germany: Springer Science & Business Media, volume 175.
- Leutenegger S, Lynen S, Bosse M, et al. (2015) Keyframe-based visual-inertial odometry using nonlinear optimization. *The International Journal of Robotics Research (IJRR)* 34(3): 314–334.
- Maldonado-Ramírez A, Torres-Méndez LA and Castelán M (2016) A bag of relevant regions for visual place recognition in challenging environments. In: *International Conference on Pattern Recognition (ICPR)*. Piscataway, NJ: IEEE, 1358–1363.
- Mallios A, Rida P, Ribas D, et al. (2016) Toward autonomous exploration in confined underwater environments. *Journal Field Robotics* 33(7): 994–1012.
- Maurelli F, Krupiński S, Xiang X, et al. (2021) AUV localisation: a review of passive and active techniques. *International Journal of Intelligent Robotics and Applications* 6: 1–24.
- McConnell J, Martin JD and Englot B (2020) Fusing concurrent orthogonal wide-aperture sonar images for dense underwater 3d reconstruction. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE, 1653–1660.
- Mourikis AI and Roumeliotis SI (2007) A multi-state constraint Kalman filter for vision-aided inertial navigation. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE, 3565–3572.
- Mur-Artal R, Montiel JMM and Tardós JD (2015) ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics* 31(5): 1147–1163.
- Mur-Artal R and Tardós JD (2017) Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters* 2(2): 796–803.
- Oliver K, Hou W and Wang S (2010) Image feature detection and matching in underwater conditions. In: *Ocean Sensing and Monitoring II*. Washington, DC: International Society for Optics and Photonics, volume 7678, 76780N.
- Ozog P, Carlevaris-Bianco N, Kim A, et al. (2016) Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle. *Journal Field Robotics* 33(3): 265–289.
- Ozog P and Eustice RM (2014) Toward long-term, automated ship hull inspection with visual SLAM, explicit surface optimization, and generic graph-sparsification. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE, 3832–3839.
- Ozog P, Troni G, Kaess M, et al. (2015) Building 3d mosaics from an autonomous underwater vehicle, doppler velocity log, and 2d imaging sonar. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE, 1137–1143.
- Paull L, Saeedi S, Seto M, et al. (2013) AUV navigation and localization: a review. *IEEE Journal of Oceanic Engineering* 39(1): 131–149.
- Pisano ED, Zong S, Hemminger BM, et al. (1998) Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. *Journal of Digital Imaging* 11(4): 193–200.
- Qin T, Li P and Shen S (2018) VINS-Mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics* 34(4): 1004–1020.
- Quattrini Li A, Coskun A, Doherty SM, et al. (2016a) Experimental comparison of open source vision based state estimation algorithms. In: *International Symposium on Experimental Robotics (ISER)*, Tokyo, Japan, 3–6 October 2016.
- Quattrini Li A, Coskun A, Doherty SM, et al. (2016b) Vision-based shipwreck mapping: on evaluating features quality and open source state estimation packages. In: *MTS/IEEE OCEANS - Monterey*. Piscataway, NJ: IEEE, 1–10.
- Rahman S (2020) SVIn2 code repository. Available at: <https://github.com/sharminrahman/SVIn2>
- Rahman S, Karapetyan N, Quattrini Li A, et al. (2018a) A modular sensor suite for underwater reconstruction. In: *MTS/IEEE OCEANS - Charleston*. Piscataway, NJ: IEEE, 1–6.
- Rahman S, Quattrini Li A and Rekleitis I (2018b) Sonar Visual Inertial SLAM of Underwater Structures. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE.
- Rahman S, Quattrini Li A and Rekleitis I (2019) SVIn2: An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE.
- Research group of Prof Kostas Daniilidis (2018) Monocular MSCKF ROS node. https://github.com/daniilidis-group/msckf_mono
- Richmond K, Flesher C, Lindzey L, et al. (2018) SUNFISH®: a human-portable exploration AUV for complex 3D environments, In: *MTS/IEEE OCEANS Charleston*. Piscataway, NJ: IEEE, 1–9.
- Rigby P, Pizarro O and Williams SB (2006) Towards geo-referenced AUV navigation through fusion of USBL and DVL measurements. In: *OCEANS*. Piscataway, NJ: IEEE, 1–6.
- Rosinol A, Abate M, Chang Y, et al. (2020) Kimera: an open-source library for real-time metric-semantic localization and mapping. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE.
- Roznere M and Quattrini Li A (2019) Real-time model-based image color correction for underwater robots. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE, 7191–7196.
- Schönberger JL, Zheng E, Pollefeys M, et al. (2016) Pixelwise view selection for unstructured multi-view stereo. In: *European Conference on Computer Vision (ECCV)*. Piscataway, NJ: IEEE.
- Shkurti F, Rekleitis I and Dudek G (2011a) Feature tracking evaluation for pose estimation in underwater environments. In: *Canadian Conference on Computer and Robot Vision (CRV)*. St. John, Canada: CRV, 160–167.
- Shkurti F, Rekleitis I, Scaccia M, et al. (2011b) State estimation of an underwater robot using visual and inertial information.

- In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE, 5054–5060.
- Skaff S, Clark J and Rekleitis I (2008) Estimating surface reflectance spectra for underwater color vision. In: *British Machine Vision Conference (BMVC)*. Leeds, UK: British Machine Vision Association, 1015–1024.
- Snyder J (2010) Doppler Velocity Log (DVL) navigation for observation-class ROVs. In: *MTS/IEEE OCEANS*. Seattle, WA: IEEE, 1–9.
- Stone WC (2007) Design and deployment of a 3-D autonomous subterranean submarine exploration Vehicle. In: *International Symposium on Unmanned Untethered Submersible Technologies*. Umhlanga, South Africa: UUST, 512.
- Strasdat H (2012) Local accuracy and global consistency for efficient visual SLAM. PhD Thesis, Imperial College London.
- Sun K, Mohta K, Pfleiderer B, et al. (2018) Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters* 3(2): 965–972.
- Tarrio JJ and Pedre S (2017) Realtime edge based visual inertial odometry for MAV teleoperation in indoor environments. *Journal of Intelligent & Robotic Systems* 90: 235–252.
- Teixeira PV, Fourie D, Kaess M, et al. (2019) Dense, sonar-based reconstruction of underwater scenes. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE.
- Trawny N and Roumeliotis SI (2005) Indirect Kalman filter for 3D attitude estimation. *University of Minnesota, Department of Computer Science and Engineering, Technical Report* 2.
- Umeyama S (1991) Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(4): 376–380.
- Von Stumberg L, Usenko V and Cremers D (2018) Direct sparse visual-inertial odometry using dynamic marginalization. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE.
- Westman E, Hinduja A and Kaess M (2018) Feature-based SLAM for imaging sonar with under-constrained landmarks. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia: IEEE, 3629–3636.
- White C, Hiranandani D, Olstad CS, et al. (2010) The Malta cistern mapping project: Underwater robot mapping and localization within ancient tunnel systems. *Journal of Field Robotics* 27(4): 399–411.
- Williams SB, Newman P, Dissanayake G, et al. (2000) Autonomous underwater simultaneous localisation and map building. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Piscataway, NJ: IEEE, volume 2, 1793–1798.
- Xanthidis M, Karapetyan N, Damron H, et al. (2020) Navigation in the presence of obstacles for an agile autonomous underwater vehicle. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE, 892–899.
- Xu S, Luczynski T, Willners JS, et al. (2021) Underwater visual acoustic SLAM with extrinsic calibration. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Piscataway, NJ: IEEE, 7647–7652.