2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

SVIn2: An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor

Sharmin Rahman[1], Alberto Quattrini Li[2], and Ioannis Rekleitis[1]

***Abstract*— This paper presents a novel tightly-coupled keyframe-based Simultaneous Localization and Mapping (SLAM) system with loop-closing and relocalization capabilities targeted for the underwater domain.**

Our previous work, SVIn, augmented the state-of-the-art visual-inertial state estimation package OKVIS to accommodate acoustic data from sonar in a non-linear optimization-based framework. This paper addresses drift and loss of localization – one of the main problems affecting other packages in underwater domain – by providing the following main contributions: a robust initialization method to refine scale using depth measurements, a fast preprocessing step to enhance the image quality, and a real-time loop-closing and relocalization method using bag of words (BoW). An additional contribution is the addition of depth measurements from a pressure sensor to the tightly-coupled optimization formulation. Experimental results on datasets collected with a custom-made underwater sensor suite and an autonomous underwater vehicle from challenging underwater environments with poor visibility demonstrate performance never achieved before in terms of accuracy and robustness.

I. INTRODUCTION

Exploring and mapping underwater environments such as caves, bridges, dams, and shipwrecks, are extremely important tasks for the economy, conservation, and scientific discoveries [1]. Currently, most of the efforts are performed by divers that need to take measurements manually using a grid and measuring tape, or using hand-held sensors [2], and data is post-processed afterwards. Autonomous Underwater Vehicles (AUVs) present unique opportunities to automate this process; however, there are several open problems that still need to be addressed for reliable deployments, including robust Simultaneous Localization and Mapping (SLAM), the focus of this paper.

Most of the underwater navigation algorithms [3], [4], [5], [6], [7] are based on acoustic sensors, such as Doppler velocity log (DVL), Ultra-short Baseline (USBL), and sonar. However, data collection with these sensors is expensive and sometimes not suitable due to the highly unstructured underwater environments. In recent years, many vision-based state estimation algorithms have been developed using monocular, stereo, or multi-camera system mostly for indoor and outdoor environments. Vision is often combined with Inertial Measurement Unit (IMU) for improved estimation of pose in challenging environments, termed as *Visual-Inertial Odometry* (VIO) [8], [9], [10], [11], [12]. However, the underwater environment – e.g., see Fig. 1 – presents unique challenges to vision-based state estimation. As shown in a previous study [13], it is not straightforward to deploy the available vision-based state estimation packages underwater. In particular, suspended particulates, blurriness, and light and color attenuation result in features that are not as clearly defined as above water. Consequently results from different vision-based state estimation packages show a significant number of outliers resulting in inaccurate estimate or even complete tracking loss.

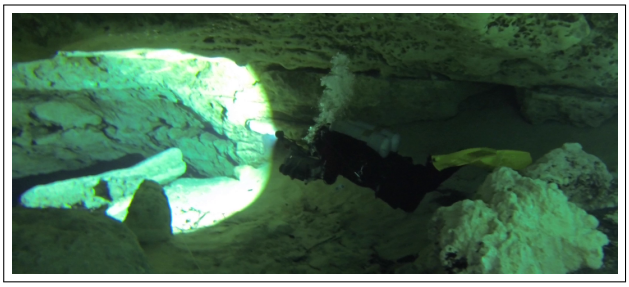


Fig. 1. Underwater cave in Ginnie Springs, FL, where data have been collected using an underwater stereo rig.

In this paper, we propose *SVIn2*, a novel SLAM system specifically targeted for underwater environments – e.g., wrecks and underwater caves – and easily adaptable for different sensor configuration: acoustic (mechanical scanning profiling sonar), visual (stereo camera), inertial (linear accelerations and angular velocities), and depth data. This makes our system versatile and applicable on-board of different sensor suites and underwater vehicles.

In our recent work, *SVIn* [14], acoustic, visual, and inertial data is fused together to map different underwater structures by augmenting the visual-inertial state estimation package OKVIS [9]. This improves the trajectory estimate especially when there is varying visibility underwater, as sonar provides robust information about the presence of obstacles with accurate scale. However, in long trajectories, drifts could accumulate resulting in an erroneous trajectory.

In this paper, we extend our work by including an image enhancement technique targeted to the underwater domain, introducing depth measurements in the optimization process, loop-closure capabilities, and a more robust initialization. These additions enable the proposed approach to robustly and accurately estimate the sensor's trajectory, where every other approach has shown incorrect trajectories or loss of localization.

[1]S. Rahman and I. Rekleitis are with the Computer Science and Engineering Department, University of South Carolina, Columbia, SC, USA srahman@email.sc.edu, yiannisr@cse.sc.edu

[2]A. Quattrini Li is with the Department of Computer Science, Dartmouth College, Hanover, NH, USA alberto.quattrini.li@dartmouth.edu

To validate our proposed approach, first, we assess the performance of the proposed loop-closing method, by comparing it to other state-of-the-art systems on the EuRoC micro-aerial vehicle public dataset [15], disabling the fusion of sonar and depth measurements in our system. Second, we test the proposed full system on several different underwater datasets in a diverse set of conditions. More specifically, underwater data – consisting of visual, inertial, depth, and acoustic measurements – has been collected using a custom made sensor suite [16] from different locales; furthermore, data collected by an Aqua2 underwater vehicle [17] include visual, inertial, and 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.

The paper is structured as follows. The next section discusses related work. Section III presents the mathematical formulation of the proposed system and describes the approach developed for image preprocessing, pose initialization, loop-closure, and relocalization. Section IV presents results from a publicly available aerial dataset and a diverse set of challenging underwater environments. We conclude this paper with a discussion on lessons learned and directions of future work.

II. RELATED WORK

Sonar based underwater SLAM and navigation systems have been exploited for many years. Folkesson et al. [18] used a blazed array sonar for real-time feature tracking. A feature reacquisition system with a low-cost sonar and navigation sensors was described in [19]. More recently, Sunfish [20] – an underwater SLAM system using a multi-beam sonar, an underwater dead-reckoning system based on a fiber-optic gyroscope (FOG) IMU, acoustic DVL, and pressure-depth sensors – has been developed for autonomous cave exploration. Vision and visual-inertial based SLAM systems also developed in [21], [22], [23] for underwater reconstruction and navigation. Corke et al. [24] compared acoustic and visual methods for underwater localization showing the viability of using visual methods underwater in some scenarios.

The literature presents many vision-based state estimation techniques, which use either *monocular* or *stereo* cameras and that are *indirect* (feature-based) or *direct* methods, including, for example, MonoSLAM [25], PTAM [26], ORB-SLAM [27], LSD-SLAM [28], and DSO [29]. In the following, we highlight some of the state estimation systems which use visual-inertial measurements and feature-based method.

To improve the pose estimate, vision-based state estimation techniques have been augmented with IMU sensors, whose data is fused together with visual information. A class of approaches is based on the *Kalman Filter*, e.g., Multi-State Constraint Kalman Filter (MSCKF) [11] and its stereo extension [12]; ROVIO [30]; REBiVO [31]. The other spectrum of methods optimizes the sensor states, possibly within a window, formulating the problem as a *graph optimization* problem. For feature-based visual-inertial systems, as in OKVIS [9] and Visual-Inertial ORB-SLAM [8], the optimization function includes the IMU error term and the reprojection error. The *frontend* tracking mechanism maintains a local map of features in a marginalization window which are never used again once out of the window. VINS-Mono [10] uses a similar approach and maintains a minimum number of features for each image and existing features are tracked by Kanade-Lucas-Tomasi (KLT) sparse optical flow algorithm in local window. Delmerico and Scaramuzza [32] did a comprehensive comparison specifically monitoring resource usage by the different methods. While KLT sparse features allow VINS-Mono running in real-time on low-cost embedded systems, often results into tracking failure in challenging environments, e.g., underwater environments with low visibility. In addition, for loop detection additional features and their descriptors are needed to be computed for keyframes.

Loop closure – the capability of recognizing a place that was seen before – is an important component to mitigate the drift of the state estimate. FAB-MAP [33], [34] is an appearance-based method to recognize places in a probabilistic framework. ORB-SLAM [27] and its extension with IMU [8] use bag-of-words (BoW) for loop closure and relocalization. VINS-Mono also uses a BoW approach.

Note that all visual-inertial state estimation systems require a proper *initialization*. VINS-Mono uses a loosely-coupled sensor fusion method to align monocular vision with inertial measurement for estimator initialization. ORB-SLAM with IMU [8] performs initialization by first running a monocular SLAM to observe the pose first and then, IMU biases are also estimated.

Given the modularity of OKVIS for adding new sensors and robustness in tracking in underwater environment – we fused sonar data in previous work in [14] – we extend OKVIS to include also depth estimate, loop closure capabilities, and a more robust initialization to specifically target underwater environments.

III. PROPOSED METHOD

This section describes the proposed system, SVIn2, depicted in Fig. 2. The full proposed state estimation system can operate with a robot that has stereo camera, IMU, sonar, and depth sensor – the last two can be also disabled to operate as a visual-inertial system.

Due to low visibility and dynamic obstacles, it is hard to find good features to track. In addition to the underwater vision constraints, e.g., light and color attenuation, vision-based systems also suffer from poor contrast. Hence, we augment the pipeline by adding an image preprocessing step, where *contrast adjustment* along with *histogram equalization* is applied to improve feature detection underwater. In particular, we use a *Contrast Limited Adaptive Histogram Equalization* (CLAHE) filter [35] in the *image pre-processing* step.

In the following, after defining the state, we describe the proposed initialization, sensor fusion optimization, loop

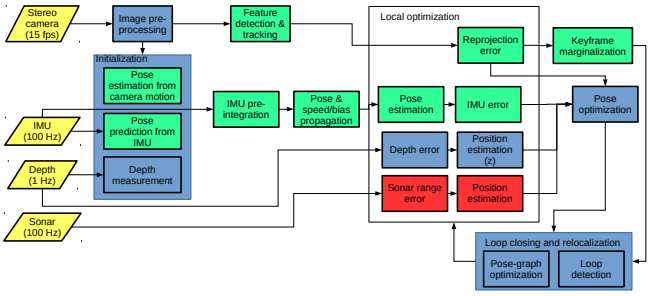


Fig. 2. Block diagram of the proposed system, SVIn2; in yellow the sensor input with frequency from the custom-made sensor suite, in green the components from OKVIS, in red the contribution from our previous work [14], and in blue the new contributions in this paper.

closure and relocalization steps.

A. Notations and States

The full sensor suite is composed of the following coordinate frames: Camera (stereo), IMU, Sonar (acoustic), Depth, and World which are denoted as C, I, S, D, and W respectively. The transformation between two arbitrary coordinate frames X and Y is represented by a homogeneous transformation matrix ${}_X\mathbf{T}_Y = [{}_X\mathbf{R}_Y | {}_X\mathbf{p}_Y]$ where ${}_X\mathbf{R}_Y$ is rotation matrix with corresponding quaternion ${}_X\mathbf{q}_Y$ and ${}_X\mathbf{p}_Y$ is position vector.

Let us now define the robot R state $\mathbf{x}_R$ that the system is estimating as:

$$\mathbf{x}_R = [{}_W\mathbf{p}_I^T, {}_W\mathbf{q}_I^T, {}_W\mathbf{v}_I^T, \mathbf{b}_g{}^T, \mathbf{b}_a{}^T]^T \quad (1)$$

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

The associated error-state vector is defined in minimal coordinates, while 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 [9], [36]:

$$\delta\boldsymbol{\chi}_R = [\delta\mathbf{p}^T, \delta\boldsymbol{\alpha}^T, \delta\mathbf{v}^T, \delta\mathbf{b}_g{}^T, \delta\mathbf{b}_a{}^T]^T \quad (2)$$

which represents the error for each component of the state vector with $\delta\boldsymbol{\alpha} \in \mathbb{R}^3$ being the minimal perturbation for rotation.

B. Tightly-coupled Non-Linear Optimization with Sonar-Visual-Inertial-Depth measurements

For the tightly-coupled non-linear optimization, we use the following cost function $J(\mathbf{x})$, which includes the reprojection error $\mathbf{e}_r$ and the IMU error $\mathbf{e}_s$ with the addition of the sonar error $\mathbf{e}_t$ (see [14]), and the depth error e_u:

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

where i denotes the camera index – i.e., left ($i = 1$) or right ($i = 2$) camera in a stereo camera system with landmark index j observed in the k^{th} camera frame. $\mathbf{P}_r^k$, $\mathbf{P}_s^k$, $\mathbf{P}_t^k$, and P_u^k represent the information matrix of visual landmarks, IMU, sonar range, and depth measurement for the k^{th} frame respectively.

For completeness, we briefly discuss each error term – see [9] and [14] for more details. The reprojection error describes the difference between a keypoint measurement in camera coordinate frame C and the corresponding landmark projection according to the stereo projection model. The IMU error term combines all accelerometer and gyroscope measurements by *IMU pre-integration* [36] between successive camera measurements and represents the *pose*, *speed and bias* error between the prediction based on previous and current states. Both reprojection error and IMU error term follow the formulation by Leutenegger *et al.* [9].

The concept behind calculating the sonar range error, introduced in our previous work [14], is that, if the sonar detects any obstacle at some distance, it is more likely that the visual features would be located on the surface of that obstacle, and thus will be approximately at the same distance. The step involves computing a visual patch detected in close proximity of each sonar point to introduce an extra constraint, using 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 sonar point. As such, given the sonar measurement $\mathbf{z}_t^k$, the error term $\mathbf{e}_t^k({}_W\mathbf{p}_I^k, \mathbf{z}_t^k)$ is based on the difference between those two distances which is used to correct the position ${}_W\mathbf{p}_I^k$. We assume an approximate normal conditional probability density 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:

$$f(\mathbf{e}_t^k|{}_W\mathbf{p}_I^k) \approx \mathcal{N}(\mathbf{0}, \mathbf{W}_t^k) \quad (4)$$

The information matrix is:

$$\mathbf{P}_t^k = \mathbf{W}_t^{k\,-1} = \left(\frac{\partial\mathbf{e}_t^k}{\partial\delta\hat{\mathbf{p}^k}}\mathbf{Q}(\delta\hat{\mathbf{p}^k}|\mathbf{z}_t^k)\frac{\partial\mathbf{e}_t^k}{\partial\delta\hat{\mathbf{p}^k}}^T\right)^{-1} \quad (5)$$

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

$$\frac{\partial\mathbf{e}_t^k}{\partial\delta\hat{\mathbf{p}^k}} = \left[\frac{-l_x + {}_W p_x}{r}, \frac{-l_y + {}_W p_y}{r}, \frac{-l_z + {}_W p_z}{r}\right] \quad (6)$$

where ${}_W\boldsymbol{l} = [l_x, l_y, l_z, 1]$ represents the sonar landmark in homogeneous coordinate and can be calculated by a simple

geometric transformation in world coordinates given *range* r and *head-position* θ from the sonar measurements:

$$_W\boldsymbol{l} = (_W\mathbf{T}_{II}\mathbf{T}_S[\mathbf{I}_3|r\cos(\theta), r\sin(\theta), 0]^T_S) \tag{7}$$

The pressure sensor, introduced in this paper, provides accurate depth measurements based on water pressure. 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[1]:

$$_Wp_{z_D}{}^k = d^k - d^0 \tag{8}$$

With depth measurement z_u^k, the depth error term $e_u^k(_Wp_{z_I}{}^k, z_u^k)$ can be calculated as the difference between the robot position along the z direction and the depth data to correct the position of the robot. The error term can be defined as:

$$e_u^k(_Wp_{z_I}{}^k, z_u^k) = |_Wp_{z_I}^k - _W\,p_{z_D}^k| \tag{9}$$

The information matrix calculation follows a similar approach as the sonar and the Jacobian is straight-forward to derive.

All the error terms are added in the *Ceres Solver* non-linear optimization framework [37] to formulate error-state (Eq. (2)) and estimate the robot state (Eq. (1)).

C. Initialization: Two-step Scale Refinement

A robust and accurate initialization is required for the success of tightly-coupled non-linear systems, as described in [8] and [10]. For underwater deployments, this becomes even more important as vision is often occluded as well as is negatively affected by the lack of features for tracking. Indeed, from our comparative study of visual-inertial based state estimation systems [38], in underwater datasets, most of the state-of-the-art systems either fail to initialize or make wrong initialization resulting into divergence. Hence, we propose a robust initialization method using the sensory information from stereo camera, IMU, and depth for underwater state estimation. The reason behind using all these three sensors is to introduce constraints on *scale* to have a more accurate estimation on initialization. Note that no acoustic measurements have been used because the sonar range and visual features contain a temporal difference, which would not allow to have any match between acoustic and visual features, if the robot is not moving. This is due to the fact that the sonar scans on a plane over 360° *around the robot* and camera detects features *in front of the robot* [14]; see Fig. 3.

The proposed initialization works as follows. First, we make sure that the system only initializes when a minimum number of visual features are present to track (in our experiments 15 worked well). Second, the two-step refinement of the initial scale from the stereo vision takes place.

[1]More precisely, $_Wp_{z_D}{}^k = (d^k - d^0) + \textit{init\_disp\_from\_IMU}$ 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.

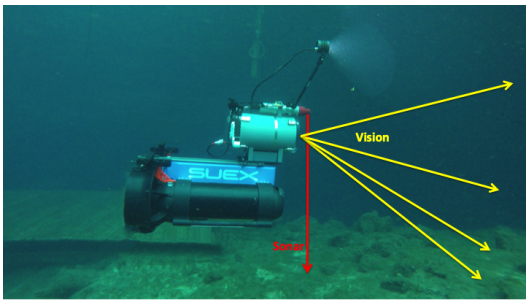


Fig. 3. Custom made sensor suite mounted on a dual DPV. Sonar scans around the sensor while the cameras see in front.

The depth sensor provides accurate depth measurements which are used to refine the initial scale factor from stereo camera. Including a scale factor s_1, the transformation between camera C and depth sensor D can be expressed as

$$_Wp_{z_D} = s_1 * {}_Wp_{z_C} + {}_W\mathbf{R}_{zCC}\mathbf{p}_D \tag{10}$$

For *keyframe* k, solving Eq. (10) for s_1, provides the first refinement r_1 of the initial stereo scale $_W\mathbf{p}_{r1C}$, i.e.,

$$_W\mathbf{p}_{r1C} = s_1 * {}_W\mathbf{p}_C \tag{11}$$

In the second step, the refined measurement from stereo camera in Eq. (11) is aligned with the IMU pre-integral values. Similarly, the transformation between camera C and IMU I with scale factor s_2 can be expressed as:

$$_W\mathbf{p}_I = s_2 * {}_W\mathbf{p}_{r1C} + {}_W\mathbf{R}_{CC}\mathbf{p}_I \tag{12}$$

In addition to refining the scale, we also approximate initial *velocity* and *gravity* vector similar to the method described in [10]. The state prediction from IMU integration $\hat{\mathbf{x}}_R^{i+1}(\mathbf{x}_R^i, \mathbf{z}_I^i)$ with IMU measurements $\mathbf{z}_I^i$ in OKVIS [9] with conditional covariance $\mathbf{Q}(\delta\hat{\mathbf{x}}_R^{i+1}|\mathbf{x}_R^i, \mathbf{z}_I^i)$ can be written as (the details about IMU pre-integration can be found in [36]):

$$\begin{aligned}
_W\hat{\mathbf{p}}_I^{i+1} &= {}_W\mathbf{p}_I^i + {}_W\,\mathbf{v}_I^i\Delta t_i + \frac{1}{2}{}_W\mathbf{g}\Delta t_i{}^2 + {}_W\,\mathbf{R}_I^i\boldsymbol{\alpha}_{I_i}^{i+1} \\
_W\hat{\mathbf{v}}_I^{i+1} &= {}_W\mathbf{v}_I^i + {}_W\,\mathbf{g}\Delta t_i + {}_W\,\mathbf{R}_I^i\boldsymbol{\beta}_{I_i}^{i+1} \\
_W\hat{\mathbf{q}}_I^{i+1} &= \boldsymbol{\gamma}_{I_i}^{i+1}
\end{aligned} \tag{13}$$

where $\boldsymbol{\alpha}_{I_i}^{i+1}$, $\boldsymbol{\beta}_{I_i}^{i+1}$, and $\boldsymbol{\gamma}_{I_i}^{i+1}$ are IMU pre-integration terms defining the motion between two consecutive keyframes *i* and *i+1* in time interval Δt_i and can be obtained only from the IMU measurements. Eq. (13) can be re-arranged with respect to $\boldsymbol{\alpha}_{I_i}^{i+1}$, $\boldsymbol{\beta}_{I_i}^{i+1}$ as follows:

$$\begin{aligned}
\boldsymbol{\alpha}_{I_i}^{i+1} &= {}_I\mathbf{R}_W^i({}_W\hat{\mathbf{p}}_I^{i+1} - {}_W\,\mathbf{p}_I^i - {}_W\,\mathbf{v}_I^i\Delta t_i - \frac{1}{2}{}_W\mathbf{g}\Delta t_i{}^2) \\
\boldsymbol{\beta}_{I_i}^{i+1} &= {}_I\mathbf{R}_W^i({}_W\hat{\mathbf{v}}_I^{i+1} - {}_W\,\mathbf{v}_I^i - {}_W\mathbf{g}\Delta t_i)
\end{aligned} \tag{14}$$

Substituting Eq. (12) into Eq. (14), we can estimate $\boldsymbol{\chi}_S = [\mathbf{v}_I^i, \mathbf{v}_I^{i+1}, {}_W\,\mathbf{g}, s_2]^T$ by solving the linear least square problem in the following form:

$$\min_{\boldsymbol{\chi}_S} \sum_{i \in K} \left\| \hat{\boldsymbol{z}}_{S_i}^{i+1} - \mathbf{H}_{S_i}^{i+1} \boldsymbol{\chi}_S \right\|^2 \tag{15}$$

where $\hat{\boldsymbol{z}}_{S_i}^{i+1} =$

$$\begin{bmatrix} \hat{\boldsymbol{\alpha}}_{I_i}^{i+1} - {}_I\mathbf{R}_W^i \, {}_W\mathbf{R}_C^{i+1} \, {}_C\mathbf{p}_I^{i+1} + {}_I\mathbf{R}_C^i \, {}_C\mathbf{p}_I^i \\ \hat{\boldsymbol{\beta}}_{I_i}^{i+1} \end{bmatrix}$$

and $\mathbf{H}_{S_i}^{i+1} =$

$$\begin{bmatrix} -\mathbf{I}\Delta t_i & \mathbf{0} & -\frac{1}{2} {}_I\mathbf{R}_W^i \Delta {t_i}^2 & {}_I\mathbf{R}_W^i ({}_W\mathbf{p}_{r1C}^{\ i+1} - {}_W\mathbf{p}_{r1C}^{\ i}) \\ -\mathbf{I} & {}_I\mathbf{R}_W^i \, {}_W\mathbf{R}_I^{i+1} & -{}_I\mathbf{R}_W^i \Delta t_i & \mathbf{0} \end{bmatrix}$$

D. Loop-closing and Relocalization

In a sliding window and marginalization based optimization method, drift accumulates over time on the pose estimate. A global optimization and relocalization scheme is necessary to eliminate this drift and to achieve global consistency. We adapt DBoW2 [39], a bag of binary words (BoW) place recognition module, and augment OKVIS for loop detection and relocalization. For each keyframe, only the descriptors of the *keypoints* detected during the local tracking are used to build the BoW database. No new features will be detected in the loop closure step.

A pose-graph is maintained to represent the connection between keyframes. In particular, a node represents a keyframe and an edge between two keyframes exists if the matched keypoints ratio between them is more than 0.75. In practice, this results into a very sparse graph. With each new keyframe in the pose-graph, the loop-closing module searches for candidates in the bag of words database. A query for detecting loops to the BoW database only returns the candidates outside the current marginalization window and having greater than or equal to score than the neighbor keyframes of that node in the pose-graph. If loop is detected, the candidate with the highest score is retained and feature correspondences between the current keyframe in the local window and the loop candidate keyframes are obtained to establish connection between them. The pose-graph is consequently updated with loop information. A 2D-2D descriptor matching and a 3D-2D matching between the known landmark in the current window keyframe and loop candidate with outlier rejection by PnP RANSAC is performed to obtain the geometric validation.

When a loop is detected, the global relocalization module aligns the current keyframe pose in the local window with the pose of the loop keyframe in the pose-graph by sending back the drift in pose to the windowed sonar-visual-inertial-depth optimization thread. Also, an additional optimization step, similar to Eq. (3), is taken only with the matched landmarks with loop candidate for calculating the sonar error term and reprojection error:

$$J(\mathbf{x}) = \sum_{i=1}^{2} \sum_{k=1}^{K} \sum_{j \in \mathit{Loop}(i,k)} \mathbf{e}_r^{i,j,k^T} \mathbf{P}_r^k \mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1} \mathbf{e}_t^{k^T} \mathbf{P}_t^k \mathbf{e}_t^k \tag{16}$$

After loop detection, a 6-DoF (position, $\mathbf{x_p}$ and rotation, $\mathbf{x_q}$) pose-graph optimization takes place to optimize over relative constraints between poses to correct drift. The relative transformation between two poses $\mathbf{T}_i$ and $\mathbf{T}_j$ for current keyframe in the current window i 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, $\mathbf{e}_{\mathbf{x_p},\mathbf{x_q}}^{i,j}$ between keyframes i and j is formulated minimally in the tangent space:

$$\mathbf{e}_{\mathbf{x_p},\mathbf{x_q}}^{i,j} = \Delta\mathbf{T}_{ij} \hat{\mathbf{T}}_i \hat{\mathbf{T}}_j^{-1} \tag{17}$$

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

$$\begin{aligned} J(\mathbf{x_p}, \mathbf{x_q}) &= \sum_{i,j} \mathbf{e}_{\mathbf{x_p},\mathbf{x_q}}^{i,j\ T} \mathbf{P}_{\mathbf{x_p},\mathbf{x_q}}^{i,j} \mathbf{e}_{\mathbf{x_p},\mathbf{x_q}}^{i,j} \\ &+ \sum_{(i,j) \in \mathit{Loop}} \rho(\mathbf{e}_{\mathbf{x_p},\mathbf{x_q}}^{i,j\ T} \mathbf{P}_{\mathbf{x_p},\mathbf{x_q}}^{i,j} \mathbf{e}_{\mathbf{x_p},\mathbf{x_q}}^{i,j}) \end{aligned} \tag{18}$$

where $\mathbf{P}_{\mathbf{x_p},\mathbf{x_q}}^{i,j}$ is the information matrix set to identity, as in [40], and ρ is the Huber loss function to potentially downweigh any incorrect loops.

IV. EXPERIMENTAL RESULTS

The proposed state estimation system, SVIn2, is quantitatively validated first on a standard dataset, to ensure that loop closure and the initialization work also above water. Moreover, it is compared to other state-of-the-art methods, i.e., VINS-Mono [10], the basic OKVIS [9], and the MSCKF [11] implementation from the GRASP lab [41]. Second, we qualitatively test the proposed approach on several different datasets collected utilizing a custom made sensor suite [16] and an Aqua2 AUV [17].

A. Validation on Standard dataset

Here, we present results on the EuRoC dataset [15], one of the benchmark datasets used by many visual-inertial state estimation systems, including OKVIS (Stereo), VINS-Mono, and MSCKF. To compare the performance, we disable depth and sonar integration in our method and only assess the loop-closure scheme.

Following the current benchmarking practices, an alignment is performed between ground truth and estimated trajectory, by minimizing the least mean square errors between estimate/ground-truth locations, which are temporally close, varying rotation and translation, according to the method from [42]. The resulting metric is the Root Mean Square Error (RMSE) for the translation, shown in Table I 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) has been shown. Our method shows reduced RMSE in every sequence from OKVIS, validating the improvement of pose-estimation after loop-closing. SVIn2 has also less RMSE than MSCKF and slightly higher in some sequences, but comparable, to results from VINS-Mono. Fig. 4 shows the trajectories for each method

TABLE I

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

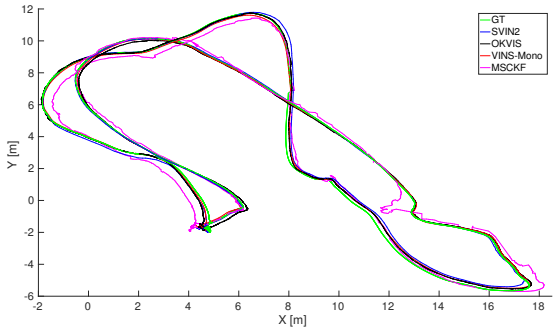


Fig. 4. Trajectories on the MH 04 sequence of the EuRoC dataset.

together with the ground truth for the *Machine Hall 04 Difficult* sequence.

B. Underwater datasets

Our proposed state estimation system – SVIn2 – is targeted for the underwater environment, where sonar and depth can be fused together with the visual-inertial data. The stereo cameras are configured to capture frames at 15 fps, IMU at 100 Hz, Sonar at 100 Hz, and Depth sensor at 1 Hz. Here, we show results from four different datasets in three different underwater environments. First, a sunken bus in Fantasy Lake (NC), where data was collected by a diver with a custom-made underwater sensor suite [16]. The diver started from outside the bus, performed a loop around and entered in it from the back door, exited across and finished at the front-top of the bus. The images are affected by haze and low visibility. Second and third, data from an underwater cavern in Ginnie Springs (FL) is collected again by a diver with the same sensor suite as for the sunken bus. The diver performed several loops, around one spot in the second dataset – Cavern1 – and two spots in the third dataset – Cavern2 – inside the cavern. The environment is affected by complete absence of natural light. Fourth, 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. In the underwater datasets, it is a challenge to get any ground truth, because it is a GPS-denied unstructured environment.

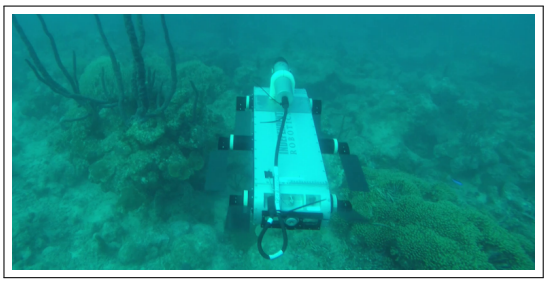


Fig. 5. The Aqua2 AUV [17] equipped with the scanning sonar collecting data over the coral reef.

As such, the evaluation is qualitative, with a rough estimate on the size of the environment measured beforehand by the divers collecting the data.

Figs. 6-9 show the trajectories from SVIn2, OKVIS, and VINS-Mono in the datasets just described. MSCKF was able to keep track only for some small segments in all the datasets, hence excluded from the plots. For a fair comparison, when the trajectories were compared against each other, sonar and depth were disabled in SVIn2. All trajectories are plotted keeping the original scale produced by each package.

Fig. 6 shows the results for the submerged bus dataset. VINS-Mono lost track when the exposure increased for quite some time. It tried to re-initialize, but it was not able to track successfully. Even using *histogram equalization* or a *contrast adjusted histogram equalization* filter, VINS-Mono was not able to track. Even if the scale drifted, OKVIS was able to track using a *contrast adjusted histogram equalization* filter in the image pre-processing step. Without the filter, it lost track at the high exposure location. The proposed method was able to track, detect, and correct the loop, successfully.

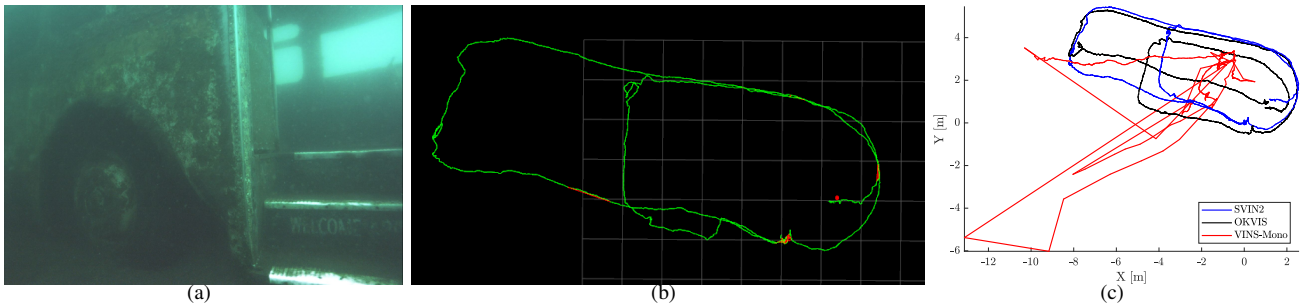
In Cavern1 – see Fig. 7 – VINS-Mono tracked successfully the whole time. However, as can be noticed in Fig. 7(c), the scale was incorrect based on empirical observations during data collection. OKVIS instead produced a good trajectory, and SVIn2 was also able to detect and close the loops.

In Cavern2 (Fig. 8), 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 similar behavior even if the images were pre-processed with different filters. 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.

In the cemetery dataset – Fig. 9 – both VINS-Mono and OKVIS were able to track, but VINS-Mono was not able to reduce the drift in trajectory, while SVIn2 was able to fuse and correct the loops.

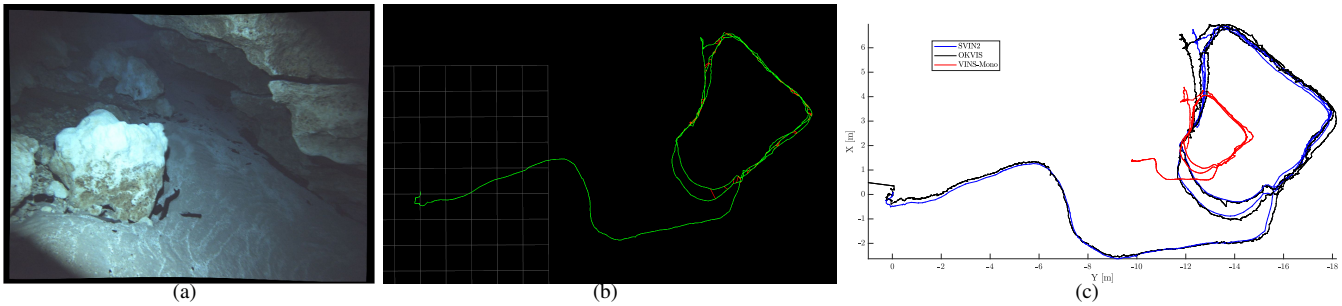
V. CONCLUSIONS

In this paper, we presented SVIn2, a state estimation system with robust initialization, sensor fusion of depth, sonar, visual, and inertial data, and loop closure capabilities. While the proposed system can also work out of the water, by disabling the sensors that are not applicable, our system



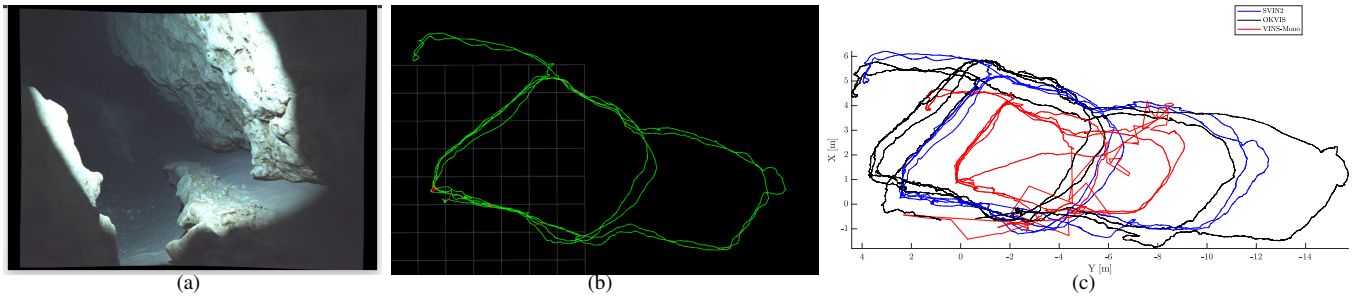
(a) (b) (c)

Fig. 6. (a) Submerged bus, Fantasy Lake, NC, USA with a 53 m trajectory; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.



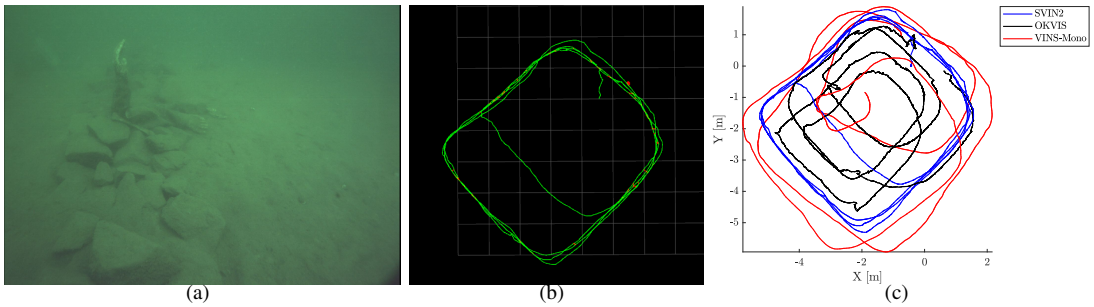
(a) (b) (c)

Fig. 7. (a) Cave environment, Ballroom, Ginnie Springs, FL, USA, with a unique loop covering a 87 m trajectory; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.



(a) (b) (c)

Fig. 8. (a) Cave environment, Ballroom, Ginnie Springs, FL, USA, with two loops in different areas covering a 155 m trajectory; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.



(a) (b) (c)

Fig. 9. (a) Aqua2 in a fake cemetery, Lake Jocassee, SC, USA with a 80 m trajectory; trajectories from SVIn2 with visual, inertial, and depth sensor (no sonar data has been used) shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.

is specifically targeted for underwater environments. Experimental results in a standard benchmark dataset and different underwater datasets demonstrate excellent performance.

Utilizing the insights gained from implementing the proposed approach, an online adaptation of the discussed framework for the limited computational resources of the Aqua2 AUV [17] is currently under consideration; see Fig. 5. It is worth noting that maintaining the proper attitude of the

traversed trajectory and providing an estimate of the distance traveled will greatly enhance the autonomous capabilities of the vehicle [43]. Furthermore, accurately modeling the surrounding structures would enable Aqua2, as well as other vision based underwater vehicles to operate near, and through, a variety of underwater structures, such as caves, shipwrecks, and canyons.

ACKNOWLEDGMENT

The authors would like to thank the National Science Foundation for its support (NSF 1513203, 1637876).

REFERENCES

[1] R. Ballard, "Why we must explore the sea," *Smithsonian Magazine*, 2014.

[2] J. Henderson, O. Pizarro, M. Johnson-Roberson, and I. Mahon, "Mapping submerged archaeological sites using stereo-vision photogrammetry," *International Journal of Nautical Archaeology*, vol. 42, no. 2, pp. 243–256, 2013.

[3] J. J. Leonard and H. F. Durrant-Whyte, *Directed sonar sensing for mobile robot navigation*. Springer Science & Business Media, 2012, vol. 175.

[4] C.-M. Lee *et al.*, "Underwater navigation system based on inertial sensor and doppler velocity log using indirect feedback Kalman filter," *International Journal of Offshore and Polar Engineering*, vol. 15, no. 02, 2005.

[5] J. Snyder, "Doppler Velocity Log (DVL) navigation for observation-class ROVs," in *MTS/IEEE OCEANS, SEATTLE*, 2010, pp. 1–9.

[6] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, "Imaging sonar-aided navigation for autonomous underwater harbor surveillance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 4396–4403.

[7] P. Rigby, O. Pizarro, and S. B. Williams, "Towards geo-referenced AUV navigation through fusion of USBL and DVL measurements," in *OCEANS*, 2006, pp. 1–6.

[8] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, 2017.

[9] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[10] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.

[11] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. ICRA*. IEEE, 2007, pp. 3565–3572.

[12] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, 2018.

[13] A. Quattrini Li, A. Coskun, S. M. Doherty, S. Ghasemlou, A. S. Jagtap, M. Modasshir, S. Rahman, A. Singh, M. Xanthidis, J. M. O'Kane, and I. Rekleitis, "Experimental comparison of open source vision based state estimation algorithms," in *Proc. ISER*, 2016.

[14] S. Rahman, A. Quattrini Li, and I. Rekleitis, "Sonar Visual Inertial SLAM of Underwater Structures," in *Proc. ICRA*, 2018.

[15] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[16] S. Rahman, A. Quattrini Li, and I. Rekleitis, "A modular sensor suite for underwater reconstruction," in *MTS/IEEE Oceans Charleston*, 2018, pp. 1–6.

[17] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguere, A. German, H. Liu, S. Saunderson, A. Ripsman, S. Simhon, L. A. Torres-Mendez, E. Milios, P. Zhang, and I. Rekleitis, "A visually guided swimming robot," in *Proc. IROS*, 2005, pp. 1749–1754.

[18] J. Folkesson, J. Leonard, J. Leederkerken, and R. Williams, "Feature tracking for underwater navigation using sonar," in *Proc. IROS*. IEEE, 2007, pp. 3678–3684.

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

[20] K. Richmond, C. Flesher, L. Lindzey, N. Tanner, and W. C. Stone, "SUNFISH®: A human-portable exploration AUV for complex 3D environments," in *MTS/IEEE OCEANS Charleston*, 2018, pp. 1–9.

[21] J. Salvi, Y. Petillo, S. Thomas, and J. Aulinas, "Visual SLAM for underwater vehicles using video velocity log and natural landmarks," in *MTS/IEEE OCEANS*, 2008, pp. 1–6.

[22] C. Beall, F. Dellaert, I. Mahon, and S. B. Williams, "Bundle adjustment in large-scale 3d reconstructions based on underwater robotic surveys," in *MTS/IEEE OCEANS, Spain*, 2011, pp. 1–6.

[23] F. Shkurti, I. Rekleitis, M. Scaccia, and G. Dudek, "State estimation of an underwater robot using visual and inertial information," in *Proc. IROS*, 2011, pp. 5054–5060.

[24] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu, "Experiments with underwater robot localization and tracking," in *Proc. ICRA*. IEEE, 2007, pp. 4556–4561.

[25] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1Point RANSAC for Extended Kalman Filtering: Application to Real-time Structure from Motion and Visual Odometry," *J. Field Robot.*, vol. 27, no. 5, pp. 609–631, 2010.

[26] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, 2007, pp. 225–234.

[27] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[28] J. Engel, T. Schps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *Proc. ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer, 2014, vol. 8690, pp. 834–849.

[29] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, 2018.

[30] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.*, vol. 36, 2017.

[31] J. J. Tarrio and S. Pedre, "Realtime edge based visual inertial odometry for MAV teleoperation in indoor environments," *J. Intell. Robot. Syst.*, pp. 235–252, 2017.

[32] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *Proc. ICRA*, 2018.

[33] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.

[34] ——, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *Int. J. Robot. Res.*, vol. 30, no. 9, pp. 1100–1123, 2011.

[35] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[36] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, 2017.

[37] S. Agarwal, K. Mierle, and Others, "Ceres Solver," http://ceres-solver.org, 2015.

[38] B. Joshi, S. Rahman, M. Kalaitzakis, B. Cain, J. Johnson, M. Xanthidis, N. Karapetyan, A. Hernandez, A. Quattrini Li, N. Vitzilaios, and I. Rekleitis, "Experimental Comparison of Open Source Visual-Inertial-Based State Estimation Algorithms in the Underwater Domain," in *Proc. IROS*, 2019, (accepted).

[39] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, 2012.

[40] H. Strasdat, "Local accuracy and global consistency for efficient visual slam," Ph.D. dissertation, Citeseer, 2012.

[41] Research group of Prof. Kostas Daniilidis, "Monocular MSCKF ROS node," https://github.com/daniilidis-group/msckf_mono, 2018.

[42] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, 1991.

[43] J. Sattar, G. Dudek, O. Chiu, I. Rekleitis, P. Giguere, A. Mills, N. Plamondon, C. Prahacs, Y. Girdhar, M. Nahon, and J.-P. Lobos, "Enabling autonomous capabilities in underwater robotics," in *Proc. IROS*, 2008, pp. 3628–3634.