TURTLMap: Real-time Localization and Dense Mapping of Low-texture Underwater Environments with a Low-cost Unmanned Underwater Vehicle

Jingyu Song[*1], Onur Bagoren[*1], Razan Andigani[1], Advaith Sethuraman[1], and Katherine A. Skinner[1]

Abstract— Significant work has been done on advancing localization and mapping in underwater environments. Still, state-of-the-art methods are challenged by low-texture environments, which is common for underwater settings. This makes it difficult to use existing methods in diverse, real-world scenes. In this paper, we present TURTLMap, a novel solution that focuses on textureless underwater environments through a real-time localization and mapping method. We show that this method is low-cost, and capable of tracking the robot accurately, while constructing a dense map of a low-textured environment in real-time. We evaluate the proposed method using real-world data collected in an indoor water tank with a motion capture system and ground truth reference map. Qualitative and quantitative results validate the proposed system achieves accurate and robust localization and precise dense mapping, even when subject to wave conditions. The project page for TURTLMap is https://umfieldrobotics.github.io/TURTLMap.

I. INTRODUCTION

Robotic systems play an important role in improving our understanding of underwater environments [1], [2]. Among them, unmanned underwater vehicles (UUVs) stand out for their role in environmental monitoring and ocean exploration [2], [3]. To safely conduct these tasks, UUVs are required to have accurate and robust localization (i.e., state estimation) capabilities [3]–[7]. However, achieving such accurate localization for UUVs is non-trivial due to the lack of GPS and Wi-Fi while the vehicle is underwater [4], [5].

The past few decades have seen great advances in localization systems, enabling robots on land to achieve real-time state estimation in complex environments [8], [9]. These methods can be categorized into odometry or simultaneous localization and mapping (SLAM) techniques [8], [10]–[15]. With these methods, vision is usually combined with inertial measurements to improve accuracy and robustness [16]. However, underwater environments pose unique challenges to achieving similar performance for marine robotic systems compared to terrestrial systems equipped with visual or visual-inertial sensor configurations [4], [5], [17]–[19]. For vision-based localization algorithms, common challenges include visibility change, light and color attenuation, absence of light, and feature sparsity [20], [21]. Despite these challenges, recent work has been successful in integrating vision into a robust localization system [6], [22]. SVin2 [4], [5] is one of the state-of-the-art (SOTA) underwater SLAM methods fusing camera, sonar, IMU and a barometer for
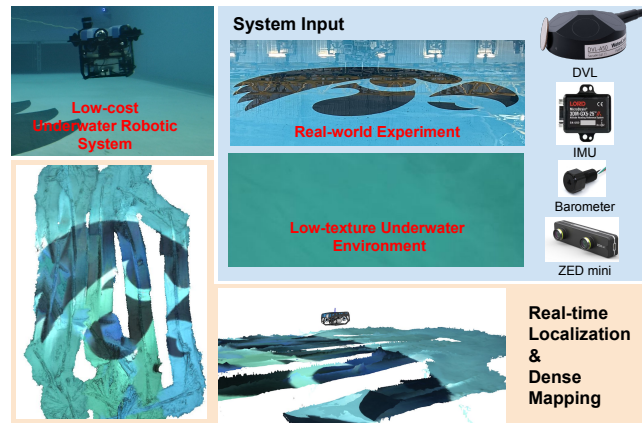
Fig. 1: Overview of TURTLMap. We propose a real-time localization and dense mapping solution for low-texture underwater environments with a low-cost underwater robot.

improved accuracy and robustness. However, it still struggles in low-texture regions, which are often encountered in underwater environments. To tackle these challenges, additional modalities are usually introduced. The Doppler velocity log (DVL) is a popular acoustic sensor supporting underwater relative navigation. Existing methods have seen success of fusing DVL, inertial, and pressure sensors for robust odometry with satisfying performance [7], [23]. Xu et al. [24] fuse camera and DVL in a SLAM framework with improvement in low-texture environments, but they do not address dense mapping. Researchers in [25] propose a real-time dense mapping solution built on [4], [5]. However, the method's reliance on vision-based state estimation poses open challenges in general underwater environments with low-texture scenes.

In this work, we propose and demonstrate **TURTLMap**, a novel framework focusing on **T**extureless **U**nderwater environments through a **R**eal-**T**ime **L**ocalization and **Map**ping method (Fig. 1). In these environments, visual sensors are usually unreliable for navigation. Our main contributions are: (i) We propose a pose graph-based DVL-inertial-barometer localization package that achieves improved localization accuracy in low-texture regions. (ii) With the proposed localization package and a real-time volumetric mapping package, we demonstrate real-time and accurate dense mapping on an onboard embedded computer on a low-cost UUV with a downward-facing stereo camera in a low-texture underwater environment. (iii) We provide extensive quantitative and qualitative evaluation on localization and mapping performance with ground truth localization provided by an underwater motion capture system and reference mapping provided by a 3D CAD model of the experimental site

with and without wave conditions. Our contribution includes detailed description of integrated hardware and we provide open-source software supporting future implementation and extension for the research community, which will be made available here: https://umfieldrobotics.github.io/TURTLMap.

II. RELATED WORKS

A. Localization for Marine Robots

Localization (i.e., state estimation) is one of the key tasks for underwater robots [4]–[6]. SLAM-based methods have been widely adopted to the underwater domain and they typically fuse vision, inertial sensors and/or acoustic sensors [4], [5], [22], [24]. The current SOTA underwater SLAM system, SVin2, presents a multi-sensor fusion approach that leverages digital pipe profiling sonar, visual and inertial sensors, and a barometer [4], [5]. SVin2 relies on feature detection from camera images for its visual front-end [10]. Though it fuses other modalities such as sonar and IMU, the localization module is still vision-oriented, making it hard to scale to featureless (low-texture) underwater environments [25], [26].

To reduce reliance on visual features for localization, it is common to integrate a DVL to provide seafloor-relative velocity. Researchers in [24] fuse vision with DVL odometry to achieve robust localization. Our method instead relies on DVL, IMU, and barometer sensors for localization for enhanced robustness in low-texture environments with wave conditions. We additionally integrate a vision-based mapping module capable of real-time dense reconstruction. The sensor configuration in [6] is similar to ours but in a reversed upward-looking orientation for under ice exploration. This method proposes fused odometry with camera, DVL, IMU and a pressure sensor for a robot navigating in a low-texture under ice environment. It differs from our method as it assumes planar features and slowly-varying altitude to aid visual tracking, which could be valid in its target scenario (i.e., under ice exploration) but does not apply to general underwater low-texture environments. Inspired by the success of recent DVL-based odometry methods [7], [8], [23], we design a pose graph-based state estimation framework fusing DVL, IMU and pressure sensors via GT-SAM [27]. Since our method does not require visual input for localization, there is no need for specific assumptions and aid for visual features, making it robust in low-texture environments with reliable localization accuracy. Another similar work [28] formulates a DVL factor and fuses it with an existing LiDAR-camera SLAM algorithm for a surface vehicle doing maritime infrastructure inspection. Our method takes inspiration from its DVL factor design and further develops a real-time framework supporting additional sensors (i.e., barometer) for UUVs in an underwater setting.

B. Mapping in Underwater Scenes

There has been prominent work on underwater mapping, with works ranging from acoustic-based [7], large-scale mapping [29], to close-range visual mapping [4], [5], [30]–[32]. In this section, we review works relating to vision-based mapping, with an emphasis on real-time or dense mapping methods used in underwater robotics applications. Methods that enable real-time mapping in underwater environments typically utilize SLAM-based formulations [33], where the visual component plays a crucial role in both localization and mapping. In SVin2 [4], [5], a camera-based system is fused with a sonar in order to produce a feature-based map. The produced map is sparse, and relies on a feature-rich environment for denser and more accurate mapping. Wang et al. [25] leverage the localization output from SVin2 to develop a state-of-the-art method for real-time dense 3D mapping in underwater environments. However, experiments on real test sites demonstrate open challenges, showing that both localization and mapping performance suffers in environments that lack rich features, such as when the vehicle moves over a sandy region. Our proposed method overcomes this challenge by leveraging multi-modal sensors for robust localization in low-texture regions, which enables dense mapping with a real-time visual mapping framework. In particular, we leverage Voxblox [34] to achieve real-time dense 3D mapping following its successful application in other robotics fields [35].

III. TECHNICAL APPROACH

Figure 2 shows an overview of the proposed system for localization and mapping in low-texture underwater environments. TURTLMap takes the measurements from all navigation sensors (i.e., DVL, IMU and barometer) as input to the localization stack. The mapping module subscribes to the estimated pose and camera depth to conduct real-time dense mapping. Each component of the system will be discussed in more detail below.

A. Sensor Integration

Our underwater robotic system is equipped with a DVL, an IMU and a barometer for navigation, and a stereo camera for dense mapping. The DVL provides direct measurement of the linear velocity of the vehicle, while the IMU measures the angular velocity and acceleration. The barometer measures the depth of the vehicle to the water surface based on the water pressure. For mapping, the stereo camera provides dense depth as a point cloud. The measurements from the aforementioned sensors are handled in ROS [36]. Detailed hardware setup will be introduced in Section IV-A.

B. Pose Graph Localization

We formulate the state estimation problem as a factor graph with our integrated DVL, barometer, and IMU as navigation sensors. Specifically, we implement a keyframe-based maximum-a-posteriori DVL-inertial-barometer estimator in GTSAM [27]. We follow the state-of-the-art Kimera-VIO [37] for framework design and add customized factors for the DVL and barometer, discussed in more detail below.

The state of the system at time i can be described as:

$$\mathbf{x}_i = [\mathbf{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i^g, \mathbf{b}_i^a, \mathbf{b}_i^v]. \tag{1}$$

We assume that the IMU frame coincides with the body frame, and use the prior extrinsic calibration to handle measurements taken in different sensor frames. We define $\mathbf{R}_i \in \mathrm{SO}(3)$ as the vehicle rotation, $\mathbf{p}_i \in \mathbb{R}^3$ is the vehicle
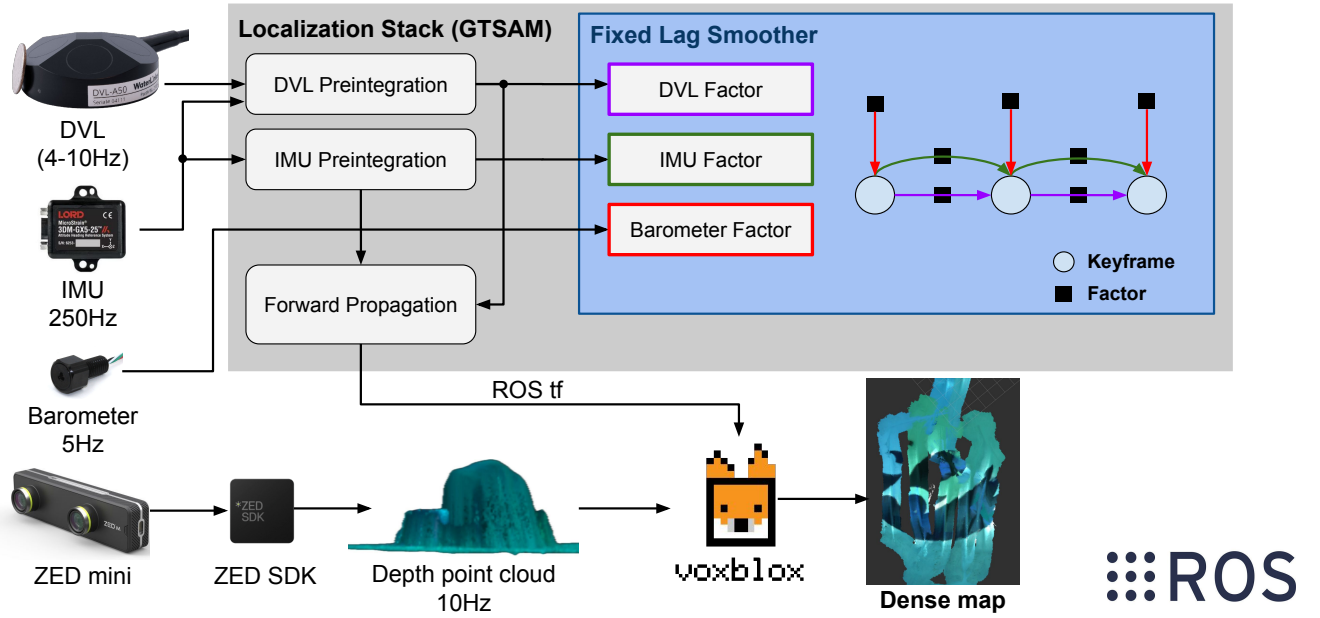
Fig. 2: An overview of TURTLMap for real-time localization and dense mapping. We develop the system using ROS [36]. The localization stack is formulated as a pose graph optimization framework with DVL, IMU and barometer measurements. The estimated pose is published as a transform message using ROS tf tool. Voxblox [34] subscribes to the pose and the depth point cloud from the ZED mini camera to construct a dense map in real-time.

position, $\mathbf{b}_i^g, \mathbf{b}_i^a \in \mathbb{R}^3$ are the biases of IMU gyroscope and accelerometer measurements, and $\mathbf{b}_i^v \in \mathbb{R}^3$ is the linear velocity bias for the DVL. The rotation $\mathbf{R}_i$ and position $\mathbf{p}_i$ are expressed in the world coordinate frame (NED convention), while other state variables are expressed in the body frame. In our approach, we estimate the state of all keyframes:

$$\mathcal{X}_k = \{\mathbf{x}_i\}_{i \in \mathcal{K}_k}, \qquad (2)$$

where we denote $\mathcal{K}_k$ as the set of all keyframes up to time k. This design maintains a sparse pose graph that ensures fast runtime. We follow Kimera-VIO [37] and VILENS [38] to use a fixed-lag smoother with a pre-defined time horizon. The fixed-lag smoother can help to bound the estimation time by marginalizing states that fall out of the smoothing horizon, which is important to ensure consistent real-time performance over long experimental duration.

We set up a ROS node to subscribe to different sensor drivers for their measurements. We assign an individual thread for IMU data since it is publishing at a high frequency (i.e., 250Hz). The IMU measurements are preintegrated using the implementation of [16] in GTSAM [27]. We take inspiration from [38] to design a preintegration function for DVL measurements. We also design a barometer factor to supervise the vehicle depth.

For every keyframe, we add a preintegrated IMU factor, a preintegrated DVL factor and a barometer factor. It is important to handle the sensor synchronization appropriately. Since IMU measurements have a high frame rate, every DVL and barometer measurement has a matched IMU measurement within tolerable time difference. In our sensor setup, the frame rate of the barometer is around 5 Hz, while the DVL frequency can vary between 4 and 15 Hz depending on the altitude. We insert a new keyframe based on the time elapsed from the last keyframe using the barometer message times-

tamp. This strategy ensures an accurate measurement for the vehicle depth for every keyframe, which fully leverages the advantage of the barometer.

Furthermore, as the DVL message frequency can be as low as 4 Hz, the time gap between the latest DVL measurement and the current keyframe is not negligible. By assuming the IMU propagation can estimate an accurate velocity change over a short time horizon (i.e., max 0.25 second), we propose to use the IMU preintegration to obtain the latest velocity for the current keyframe and add that as a pseudo measurement to the DVL preintegration.

1) IMU Factor: We follow the existing works [37], [38] to use the preintegrated IMU factor to constrain the pose and velocity between two consecutive keyframes. This idea is first proposed in [16] to remove the need for recomputation of the integration between two keyframes. We use the same preintegration design implemented in GTSAM [27] to handle the high frequency signals. The parameters (e.g., noise density, noise random walk) used in IMU preintegration are set according to the data sheet of the IMU. As mentioned above, we also set up an additional IMU preintegration for estimating the velocity change since the latest DVL measurement. This additional preintegration is reset every time a new DVL measurement is received. We refer to [16] for more details about preintegrated IMU factors.

2) Barometer Factor: The barometer provides accurate depth measurements based on the water pressure. Since the robot state is defined in the NED frame, the depth of the robot at time i is defined as

$$d_i = {p^z}_i - {p^z}_0, \qquad (3)$$

where d_i is the depth compensated by the initial depth, and p^z is the z axis value of $\mathbf{p}$. For the depth measurement z_i,

we define the residual as

$$r_{d_i, z_i} = d_i - z_i. \tag{4}$$

The barometer is integrated as a unary factor on $\mathbf{p}_i$, and its Jacobian used in our factor implementation is

$$\frac{\partial r_{d_i, z_i}}{\partial \delta \mathbf{p}_i} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \times \mathbf{R}_i, \tag{5}$$

which is a 1×3 vector derived by following the lifting strategy introduced in [16].

3) DVL Factor: The DVL measures the linear velocity of the robot. We follow common practices [7], [28], [38] to use the constant velocity model for velocity integration. Unlike [28], [38], we decouple the IMU preintegration with the DVL preintegration and use the estimated altitude change from the IMU's onboard Extended Kalman Filter (EKF). According to our analysis and comparison with ground truth, the estimated altitude change from the onboard EKF of our IMU is accurate in the short time gap between consecutive DVL measurements. In addition, decoupling the IMU preintegration reduces the system computation load and can leverage the fine-tuned onboard attitude reference system. We also directly leverage the reported DVL velocity measurement uncertainty from the manufacturer driver to inform the DVL factor of the measurement quality. As we observe that the uncertainty of DVL measurements can increase considerably due to multiple reasons (e.g., strong environmental disturbance such as waves, fast change of altitude in scenarios such as driving from beach to sea, out of operating range), this design has advantages of tuning the weight for DVL factors in the pose graph adaptively.

We follow [7], [38] to add a slowly varying bias term $\mathbf{b}^v$ to the velocity:

$$\tilde{\mathbf{v}} = \mathbf{v} + \mathbf{b}^v + \boldsymbol{\eta}^v, \tag{6}$$

where η^v is the linear velocity Gaussian white noise term. We formulate the DVL factor as a translation residual between two consecutive keyframes. The translation measurement is defined as:

$$\Delta \tilde{\mathbf{p}}_{ij} = \sum_{k=i}^{j-1} \Delta \tilde{\mathbf{R}}_{ik} (\tilde{\mathbf{v}}_k - \mathbf{b}_i^v) \Delta t, \tag{7}$$

where k is an intermediary measurement between keyframes i and j. As mentioned above, we decouple the IMU preintegration by directly using the $\Delta \tilde{\mathbf{R}}_{ik}$ measurement from the IMU's onboard EKF. Our proposed DVL factor residual is formulated as

$$\mathbf{r}_{\mathcal{D}_{ij}} = \mathbf{R}_i^\mathsf{T} (\mathbf{p}_j - \mathbf{p}_i) - \Delta \tilde{\mathbf{p}}_{ij}(\mathbf{b}_i^v). \tag{8}$$

The Jacobians for the involved state variables $\mathbf{p}_i$ and $\mathbf{p}_j$ can be derived following [16]. We follow [28], [38] to implement the Jacobian for $\mathbf{b}_i^v$. We compute the covariance of $\mathbf{r}_{\mathcal{D}_{ij}}$ iteratively,

$$\boldsymbol{\Sigma}_{i,k+1}^{\mathcal{D}} = \mathbf{A} \boldsymbol{\Sigma}_{i,k}^{\mathcal{D}} \mathbf{A}^\top + \mathbf{B} \boldsymbol{\Sigma}_\eta^{\mathcal{D}} \mathbf{B}^\top, \tag{9}$$

where the first term $\boldsymbol{\Sigma}_{i,i}^{\mathcal{D}}$ starts from $\mathbf{0}$, and $\boldsymbol{\Sigma}_\eta^{\mathcal{D}}$ is set as the measurement uncertainty from the DVL driver report. We derive $\mathbf{A} = \mathbf{I}_{3\times3}$ and $\mathbf{B} = \Delta \tilde{\mathbf{R}}_{ik} \Delta t$. We refer to [28], [38] for a more detailed derivation process.

C. Real-time Volumetric Mapping

We leverage Voxblox [34], a real-time volumetric mapping library that can run on embedded devices as the mapping algorithm of this work. Benefiting from its tight ROS integration, the integration of Voxblox with the proposed localization package is straight-forward. As shown in Fig. 2, Voxblox directly subscribes to the depth measurements of the ZED mini camera. It leverages the ROS tf tool to query the estimated pose of the camera output by our robust localization system to reconstruct the scene using the mesh representation.

We use the neural mode of the depth estimation in the ZED SDK, which is the most accurate and stable depth sensing mode according to our analysis. In our experiment, we set the camera resolution to VGA and the publishing rate of the depth point cloud to 10 Hz. These settings can be adjusted according to the computation power of the embedded device. One of the important factors affecting the mapping performance is the camera calibration. We re-calibrate the stereo camera for underwater usage. We use the multiple camera calibration tool in Kalibr toolbox [39] and a customized Aprilgrid target to obtain the underwater calibration parameters. We demonstrate a comparison of the depth point cloud between default calibration and the underwater calibration in Fig. 3. Note that the tank bottom should appear to be flat, so this validates the calibration using prior knowledge of the reference structure.

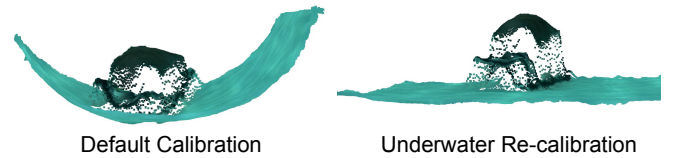Default Calibration   Underwater Re-calibration

Fig. 3: A comparison of the stereo depth point cloud of a rock platform placed on the tank bottom with the default calibration (left) and the underwater calibration (right). The bottom surface of the tank is flat.

IV. EXPERIMENT & RESULTS

A. Vehicle Setup

We set up the proposed TURTLMap package based on a low-cost robotic system with our proposed sensor configuration as discussed below.

1) Robot Platform: We build the proposed solution based on the heavy configuration of the Blue Robotics BlueROV2 UUV, which is an affordable, high-performance ROV with high flexibility of customization. We use the official payload skid to accommodate the added hardware items. To accommodate the components that are not waterproof, we include a customized Sexton aluminum enclosure and mount it onto the payload skid of the BlueROV2. The proposed localization and mapping package is running on the NVIDIA Jetson Orin Nano embedded computer. Figure 4 shows an overview of the robot platform with the customized sensor enclosure.

2) Hardware Configuration: When choosing among various navigation and perception sensors, as well as the embedded computer, the most important consideration factors are the cost, size, and difficulty of integration. We choose
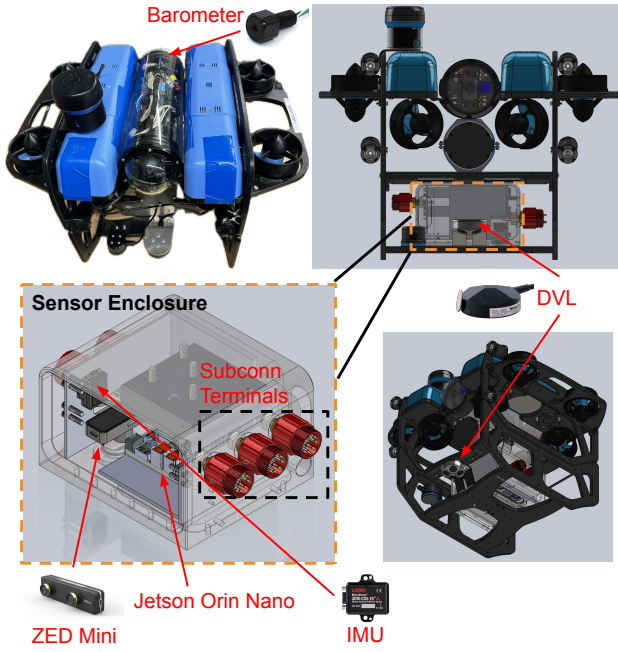
Fig. 4: An overview of the robot platform and the customized sensor enclosure. The Blue Robotics Bar30 barometer is located in the electronics enclosure of the BlueROV2. The Waterlinked DVL A50 is placed on the payload skid with its transducers facing downward. The ZED mini stereo camera is placed inside the customized sensor enclosure with an NVIDIA Jetson Orin Nano embedded computer. We also place a LORD MicroStrain 3DM-GX5-25 IMU inside the sensor enclosure. Additional sensors (e.g., Blue Robotics Ping360 scanning sonar, Blue Robotics Ping1D sonar) are mounted on the robot, but they are not used in this work.

to use the combination of DVL, IMU and barometer to build the localization stack. We choose the Waterlinked DVL A50 model due to it being a low-cost option among existing DVLs on the market, along with its compact size. Another advantage of this DVL is the official integration guide and support for the BlueROV2. Though the DVL is equipped with an internal IMU, its data is inaccessible. We only use the velocity measurements from the DVL to maintain compatibility of the proposed method to various DVL models. Though the BlueROV2 comes with an IMU, its frame rate is limited by MAVLink (10Hz by default), which is a major challenge for robust state estimation. We add an additional LORD MicroStrain 3DM-GX5-25 IMU to estimate the rotation of the vehicle. This IMU is reasonably priced and has an onboard EKF for accurate attitude measurement. The robot also comes with a Blue Robotics Bar30 barometer that has absolute measurement for robot depth. We access its measurement through communication with the BlueROV2 Raspberry Pi via MAVLink for the proposed state estimation package.

Though pose tracking via visual sensors is not reliable in underwater low-texture environments, we value its importance in dense reconstruction of these environments (aided by robust localization). We choose the ZED mini stereo camera due to its reasonable price and small form factor that fit well to our budget and limited payload space. We place the ZED mini camera downward-facing in the custom waterproof enclosure. The ZED mini camera is accompanied by the ZED SDK, providing useful functions for robotic application (e.g., image capturing, depth sensing). In this work, we use the
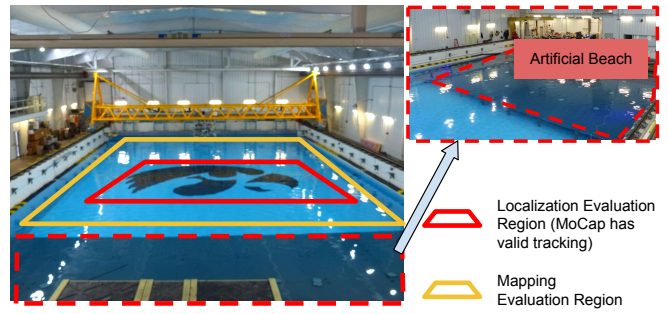

Fig. 5: The towing tank at the University of Iowa. We show different regions for localization evaluation and mapping evaluation. The region where the MoCap system has valid tracking is relatively small. We exclude the artificial beach for map evaluation due to the lack of its reference structure.

TABLE I: Details of the logs used in the evaluation of the proposed method.

Log	Wave	Description	Duration (s)
1	✗	Robot surveys the tank while being maintained about 1.75 m to the bottom	352
2	✓	Wave Height: 0.1 m Wave Frequency: 1 Hz Robot drives two squares and maintains at the water surface	280

depth measurements from the ZED SDK for mapping. It is worth mentioning that the IMU inside the ZED mini camera is not used in this project due to its data quality and low frame rate. The ZED SDK requires an embedded computer with a GPU. We choose to use an NVIDIA Jetson Orin Nano due to the low cost. It also has a compact design that can fit in the sensor enclosure easily and we have tested that it has stable thermal performance when placed in an enclosure.

We maintain proper calibration to ensure accurate localization and mapping. We use the IMU-camera calibration function in Kalibr toolbox [40], [41] to obtain the transformation between the IMU and the ZED mini stereo camera. We leverage the CAD model of the robot platform to obtain the extrinsics for other sensors.

B. Experiment Setup

To demonstrate the capability of the proposed framework, experiments were conducted in the towing tank at the University of Iowa. We use the wave generator to create natural environment disturbance to mimic real-world challenges for
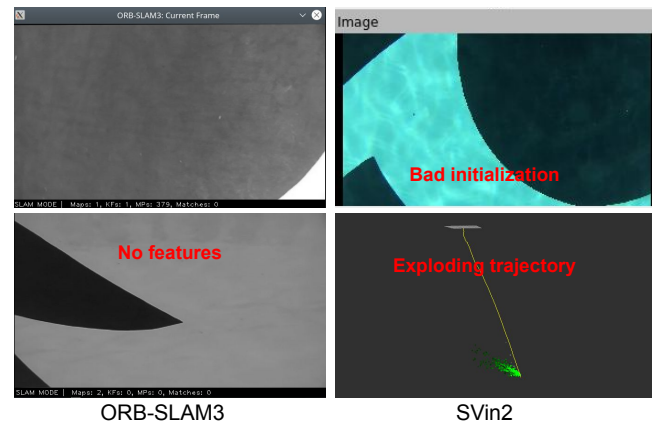

Fig. 6: Low-texture underwater environments are challenging to state-of-the-art visual localization methods (i.e., ORB-SLAM3 [14], Svin2 [4], [5]).

TABLE II: Comparison of the localization accuracy of the proposed method (TURTLMap) and baselines. Abbreviations: DVL (D), IMU (I), Barometer (B). ATE_{pos} is in meters and ATE_{rot} is in degrees.

Method	Modality			Log 1		Log 2	
	D	I	B	ATE_{pos}	ATE_{rot}	ATE_{pos}	ATE_{rot}
DVL Odom.	✓	✓	✗	1.84	7.52	0.99	6.52
UKF [7]	✓	✓	✓	0.48	11.86	0.36	15.8
TURTLMap	✓	✓	✓	**0.18**	**3.72**	**0.26**	**4.22**

TABLE III: Ablation study of the proposed localization module in TURTLMap. ATE_{pos} is in meters and ATE_{rot} is in degrees.

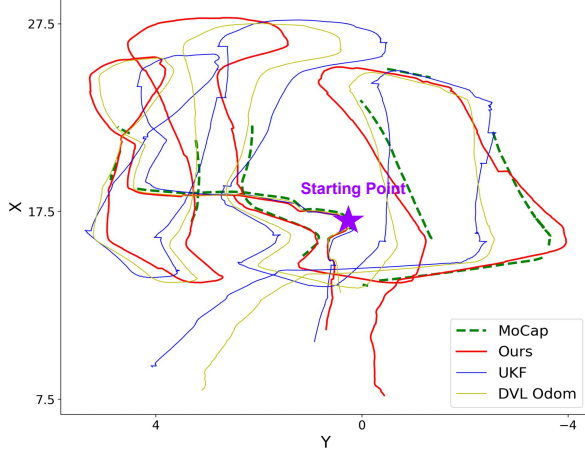| Pseodo Vel.	Sensor Unc.	Log 1		Log 2	
		ATE_{pos}	ATE_{rot}	ATE_{pos})	ATE_{rot}
✗	✓	0.32	5.44	0.40	4.72
✓	✗	0.23	**3.63**	0.28	4.42
✓	✓	**0.18**	3.72	**0.26**	**4.22**

Fig. 7: A qualitative comparison of estimated trajectories for log 1 in the x-y plane shown in meters. The starting point indicates where the robot enters the MoCap's valid tracking zone. Our method maintains the most accurate trajectory with the least amount of drift over time. Best viewed with color and zoomed-in.

localization. The towing tank is an ideal place to evaluate the proposed system due to the equipped underwater motion capture (MoCap) system that can be used as the ground truth reference for vehicle trajectory, and a reference structure of the tank for validating the mapping accuracy. We show an overview of the tank in Fig. 5, which also indicates regions we use for evaluating the localization and mapping modules.

C. Evaluation Setting

We use two logs collected at different wave and driving patterns as listed in Table I. To evaluate localization performance, we use the the open-source tool [42] to evaluate absolute trajectory error (ATE) between the output of our localization algorithm and the MoCap trajectory. As noted in Fig. 5, the MoCap system does not have full coverage of the tank, so we only conduct evaluation in the region where the MoCap system has valid tracking of the robot. As each log is initiated at the artificial beach region, which is outside of the MoCap tracking zone, we align the reference frame of our localization package to the MoCap reference frame according to the first valid MoCap pose. We run state-of-the-art (SOTA) vision-based SLAM methods, ORB-SLAM3 [14] and SVin2 [4], [5], as baselines to show the challenges that vision-based localization methods face in low-texture underwater environments. We further compare our proposed localization method against the onboard DVL dead reckoning results (DVL Odom.), and a baseline Unscented Kalman Filter (UKF) implemented in [7].

We use the ground truth mesh reference from the CAD model to evaluate the generated dense map in Cloud Compare [43]. Though the artificial beach is captured in our logs, due to the lack of the actual reference structure of the beach,

we only evaluate the map with the bottom surface of the tank, as indicated in Fig. 5. We also put an artificial rock platform with known 3D structure in the tank to validate the capability of the proposed solution in mapping small objects. In our experiments, we follow the default configuration of Voxblox [34] setting the voxel size to 0.05 m.

D. Localization Results

The underwater low-texture environments pose a significant challenge to the selected SOTA SLAM methods. ORB-SLAM3 [14] has a hard time passing the initialization stage with its stereo or stereo-inertial mode. Though we try to tune the related parameters and apply CLAHE image preprocessing following [4], [5], due to the limited number of tracked features, the trajectory quickly diverges and the map is reset. A similar issue happens with SVin2 [4], [5], as we also notice the quickly exploding trajectory. We show examples of difficulties with these two baselines in Fig. 6. This demonstrates that SOTA visual SLAM systems are not reliable for navigation in low-texture underwater environments.

We report quantitative comparison of our method and other localization baselines in Table II. From this table, we can see that the proposed method is able to maintain the lowest ATE in both position and rotation. In addition, we plot the trajectories of the proposed state estimation algorithm and other baselines in Fig. 7, which shows the proposed method closely aligns to the ground truth trajectory with the least amount of drift.

In our localization system design, we propose to use IMU preintegration to generate an additional pseudo DVL velocity measurement to mitigate the time gap between the latest DVL measurement and current keyframe. We also design the DVL factor to be aware of the reported DVL velocity measurement uncertainty. We conduct an ablation study of these two design components. As shown in Table III, using the IMU to predict a pseudo velocity measurement for the latest keyframe is effective in improving the localization accuracy. When compared with using a constant uncertainty model for the DVL velocity measurement, we see improvement from the proposed method as it fuses the sensor measurement quality reported from the manufacturer driver to determine the factor weight in the pose graph.

E. Mapping Results

We evaluate the dense maps generated by TURTLMap of the two logs in Table I. We compare it with the reference structure of the tank, which is a simple flat surface. Table IV reports the error between the generated map and the ground truth reference with different localization methods as input. We do not include ORB-SLAM3 [14] and SVin2 [4], [5] as they fail to generate a map. It is demonstrated that the
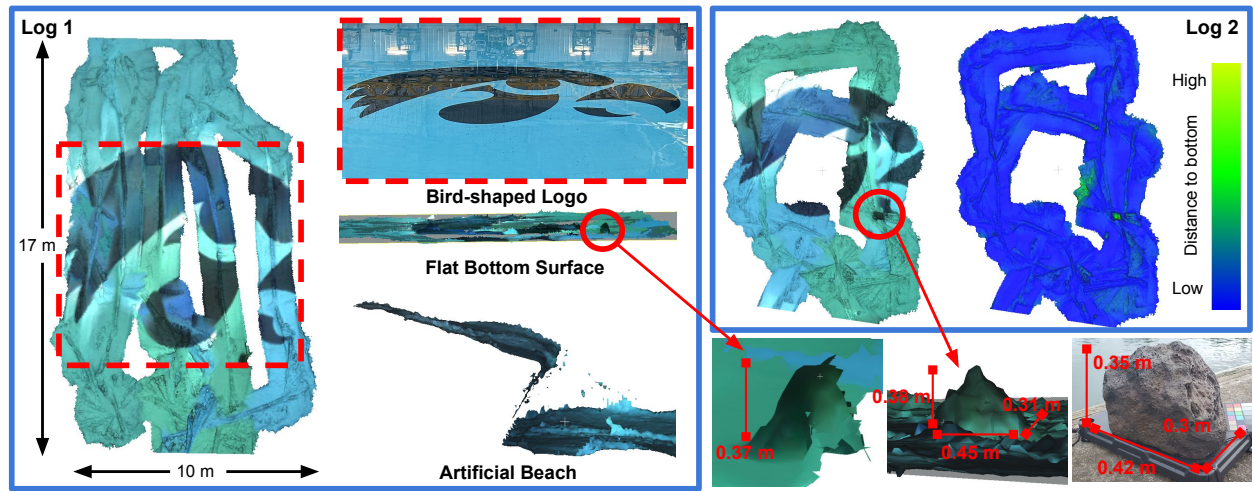
Fig. 8: Qualitative results of the map generated from both logs. In log 1, we highlight the reconstructed bird-shaped logo and the corresponding image. The map matches the flat bottom surface well while including details of the artificial beach. In log 2, we plot a color map of distance to the bottom surface showing the capability of the proposed solution under wave disturbance. We also show a comparison of the rock dimensions captured from both logs.

TABLE IV: Quantitative results of mapping evaluation.

Localization Input	Mapping Error (m)	
	Log 1	Log 2
DVL Odom	1.733	0.806
UKF [7]	0.105	0.112
TURTLMap (Ours)	**0.019**	**0.040**

proposed mapping system with our localization input is able to maintain an accurate map with low error compared to other localization baselines as our method produces more accurate state estimation. The comparison between the two logs also demonstrates that the environmental disturbance caused by waves introduces challenges to the proposed system, but the mapping error only increases to 0.04 m.

We also show a comparison of the generated map from different perspectives with images of the tank structure in Fig. 8. The qualitative comparison highlights the capability of the proposed real-time mapping solution. In log 1, the large-scale bird-shaped logo is reconstructed in detail in the generated map and matches the picture well, demonstrating the effectiveness of the proposed solution. The map generated from log 2 also validates the system's capabilities under wave disturbance, with the rock platform being mapped precisely. We show a color map with distance to the ground truth reference of the bottom surface to highlight the mapping accuracy. We additionally show a qualitative example of the artificial beach in the generated map, demonstrating the capability of the system in maintaining an accurate map in scenarios of significant altitude change. This capability is very important for safe navigation for UUVs.

F. Runtime

We measure the runtime of the localization module and mapping module on the Jetson Orin Nano embedded computer. The time cost of pose graph optimization at each keyframe varies between 0.01s and 0.04s. Due to the applied fixed lag smoother, the time cost is consistent over the complete run. The time cost for each map update is consistently around 0.02s in the longest survey log. As these two modules are running on different threads, we can conclude that the proposed system is capable of running in real-time.

V. CONCLUSION & FUTURE WORK

We have proposed TURTLMap, an effective real-time localization and dense mapping solution for a low-cost UUV. The robustness of TURTLMap is evaluated in challenging low-texture underwater scenes with and without waves. We develop a multi-modal localization stack using a factor graph. We conduct real-time volumetric mapping based on stereo depth information and estimated robot pose from the proposed localization module. We conduct extensive evaluation of the proposed system with real-world data including underwater MoCap and ground truth mapping reference, which shows the effectiveness of the proposed system.

In this work, only the stereo point cloud is used for mapping. Extending this work to fuse the sparse acoustic measurements for mapping is an interesting future direction. Another future direction is to evaluate the proposed system in real field environments with a mix of low-texture and high-texture scenes. In addition, we plan to extend the proposed system to add visual odometry factors and visual loop closure in environments with sufficient visual features, enhancing performance in activities such as shipwreck survey and coral reef monitoring [4], [5], [25], [44]. This will be a natural extension due to our factor graph formulation.

VI. ACKNOWLEDGEMENT

REFERENCES

[1] S. B. Williams, O. R. Pizarro, M. V. Jakuba, C. R. Johnson, N. S. Barrett, R. C. Babcock, G. A. Kendrick, P. D. Steinberg, A. J. Heyward, P. J. Doherty, I. Mahon, M. Johnson-Roberson, D. Steinberg, and A. Friedman, "Monitoring of benthic reference sites: Using an autonomous underwater vehicle," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 73–84, 2012.

[2] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *IFAC conference of manoeuvering and control of marine craft*, vol. 88. Lisbon, 2006, pp. 1–12.

[3] H. Qin, X. Wang, G. Wang, M. Hu, Y. Bian, X. Qin, and R. Ding, "A novel ins/usbl/dvl integrated navigation scheme against complex underwater environment," *Ocean Engineering*, vol. 286, p. 115485, 2023.

[4] S. Rahman, A. Q. Li, and I. Rekleitis, "Svin2: An underwater slam system using sonar, visual, inertial, and depth sensor," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1861–1868.

[5] S. Rahman, A. Quattrini Li, and I. Rekleitis, "Svin2: A multi-sensor fusion-based underwater slam system," *The International Journal of Robotics Research*, vol. 41, no. 11-12, pp. 1022–1042, 2022.

[6] L. Zhao, M. Zhou, and B. Loose, "Tightly-coupled visual- dvl- inertial odometry for robot-based ice-water boundary exploration," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7127–7134.

[7] J. Song, O. Bagoren, and K. A. Skinner, "Uncertainty-aware acoustic localization and mapping for underwater robots," in *OCEANS 2023-Limerick*. IEEE, 2023, pp. 1–9.

[8] T.-Y. Lin, T. Li, W. Tong, and M. Ghaffari, "Proprioceptive invariant robot state estimation," *arXiv preprint arXiv:2311.04320*, 2023.

[9] X. Yu, S. Teng, T. Chakhachiro, W. Tong, T. Li, T.-Y. Lin, S. Koehler, M. Ahumada, J. M. Walls, and M. Ghaffari, "Fully proprioceptive slip-velocity-aware state estimation for mobile robots via invariant kalman filtering and disturbance observer," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8096–8103.

[10] S. Khattak, C. Papachristos, and K. Alexis, "Keyframe-based direct thermal–inertial odometry," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3563–3569.

[11] R. Zhang, T.-Y. Lin, C. E. Lin, S. A. Parkison, W. Clark, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "A new framework for registration of semantic point clouds from stereo and rgb-d cameras," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 214–12 221.

[12] M. G. Jadidi, W. A. Clark, A. M. Bloch, R. Eustice, and J. W. Grizzle, "Continuous direct sparse visual odometry from rgb-d images." in *Robotics: Science and Systems*, 2019.

[13] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[14] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[15] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of field robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[16] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[17] E. Iscar, K. A. Skinner, and M. Johnson-Roberson, "Multi-view 3d reconstruction in underwater environments: Evaluation and benchmark," in *OCEANS 2017-Anchorage*. IEEE, 2017, pp. 1–8.

[18] A. Mallios, E. Vidal, R. Campos, and M. Carreras, "Underwater caves sonar data set," *The International Journal of Robotics Research*, vol. 36, no. 12, pp. 1247–1251, 2017.

[19] B. Joshi, S. Rahman, M. Kalaitzakis, B. Cain, J. Johnson, M. Xanthidis, N. Karapetyan, A. Hernandez, A. Q. Li, N. Vitzilaios *et al.*, "Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7227–7233.

[20] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing underwater imagery using generative adversarial networks," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7159–7165.

[21] K. Oliver, W. Hou, and S. Wang, "Image feature detection and matching in underwater conditions," in *Ocean Sensing and Monitoring II*, vol. 7678. SPIE, 2010, pp. 185–196.

[22] A. Kim and R. M. Eustice, "Real-time visual slam for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.

[23] E. R. Potokar, K. Norman, and J. G. Mangelson, "Invariant extended kalman filtering for underwater navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5792–5799, 2021.

[24] S. Xu, T. Luczynski, J. S. Willners, Z. Hong, K. Zhang, Y. R. Petillot, and S. Wang, "Underwater visual acoustic slam with extrinsic calibration," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7647–7652.

[25] W. Wang, B. Joshi, N. Burgdorfer, K. Batsosc, A. Q. Lid, P. Mordohaia, and I. Rekleitisb, "Real-time dense 3d mapping of underwater environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5184–5191.

[26] B. Joshi, C. Bandara, I. Poulakakis, H. G. Tanner, and I. Rekleitis, "Hybrid visual inertial odometry for robust underwater estimation," in *OCEANS 2023 - MTS/IEEE U.S. Gulf Coast*, 2023, pp. 1–7.

[27] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam)

[28] A. Thoms, G. Earle, N. Charron, and S. Narasimhan, "Tightly coupled, graph-based dvl/imu fusion and decoupled mapping for slam-centric maritime infrastructure inspection," *IEEE Journal of Oceanic Engineering*, vol. 48, no. 3, pp. 663–676, 2023.

[29] I. Torroba, M. Cella, A. Terán, N. Rolleberg, and J. Folkesson, "Online stochastic variational gaussian process mapping for large-scale bathymetric slam in real time," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3150–3157, 2023.

[30] M. Johnson-Roberson, M. Bryson, A. Friedman, O. Pizarro, G. Troni, P. Ozog, and J. C. Henderson, "High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology," *Journal of field robotics*, vol. 34, no. 4, pp. 625–643, 2017.

[31] A. V. Sethuraman, M. S. Ramanagopal, and K. A. Skinner, "Waternerf: Neural radiance fields for underwater scenes," in *OCEANS 2023-MTS/IEEE US Gulf Coast*. IEEE, 2023, pp. 1–7.

[32] T. Zhang and M. Johnson-Roberson, "Beyond nerf underwater: Learning neural reflectance fields for true color correction of marine imagery," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6467–6474, 2023.

[33] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, 02 2014.

[34] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for onboard mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1366–1373.

[35] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, "Graph-based subterranean exploration path planning using aerial and legged robots," *Journal of field robotics*, vol. 37, no. 8, pp. 1363–1388, 2020.

[36] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

[37] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1689–1696.

[38] D. Wisth, M. Camurri, and M. Fallon, "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2022.

[39] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 473–480.

[40] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 1280–1286.

[41] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 2088–2095.

[42] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251.

[43] "CloudCompare (version 2.11) [GPL software]," 2023. [Online]. Available: http://www.cloudcompare.org

[44] A. V. Sethuraman, A. Sheppard, O. Bagoren, C. Pinnow, J. Anderson, T. C. Havens, and K. A. Skinner, "Machine learning for shipwreck segmentation from side scan sonar imagery: Dataset and benchmark," *The International Journal of Robotics Research*, 2024.