Robust Underwater Visual SLAM Fusing Acoustic Sensing

Elizabeth Vargas*[1], Raluca Scona*[1], Jonatan Scharff Willners[1], Tomasz Luczynski[1], Yu Cao[2], Sen Wang[1], Yvan R. Petillot[1]

***Abstract*— In this paper, we propose an approach for robust visual Simultaneous Localisation and Mapping (SLAM) in underwater environments leveraging acoustic, inertial and altimeter/depth sensors. Underwater visual SLAM is challenging due to factors including poor visibility caused by suspended particles in water, a lack of light and insufficient texture in the scene. Because of this, many state-of-the-art approaches rely on acoustic sensing instead of vision for underwater navigation.**

Building on the sparse visual SLAM system ORB-SLAM2, this paper proposes to improve the robustness of camera pose estimation in underwater environments by leveraging *acoustic odometry*, which derives a drifting estimate of the 6-DoF robot pose from fusion of a Doppler Velocity Log (DVL), a gyroscope and an altimeter or depth sensor. *Acoustic odometry* estimates are used as motion priors and we formulate pose residuals that are integrated within the camera pose tracking, local and global bundle adjustment procedures of ORB-SLAM2.

The original design of ORB-SLAM2 supports a single map and it enters relocalisation when tracking is lost. This is a significant problem for scenarios where a robot does a continuous scanning motion without returning to a previously visited location. One of our main contributions is to enable the system to create a new map whenever it encounters a new scene where visual odometry can work. This new map is connected with its predecessor in a common graph using estimates from the proposed *acoustic odometry*. Experimental results on two underwater vehicles demonstrate the increased robustness of our approach compared to baseline ORB-SLAM2 in both controlled, uncontrolled and field environments.

I. Introduction

Visual inspection of underwater structures is an important application across many industries. In the energy sector, ensuring the safety of offshore operations requires a continuous and accurate assessment of subsea infrastructure. This task could be enormously facilitated through the use of Remotely Operated Underwater Vehicles (ROVs), equipped with visual sensors such as stereo cameras.

Traditionally, ROVs have used acoustic sensors for navigation and state estimation, e.g. sonar [1], complemented by depth sensors [2], Inertial Measurement Unit (IMU)s, magnetometers or a DVL [3]. However, acoustic sensors such as sonars tend to be inadequate for detailed visual inspection. Stereo cameras are better suited to this problem as they provide the necessary visual information, however their integration within a general navigation and state estimation system is non-trivial due to poor underwater visibility conditions and complex distortions [4].

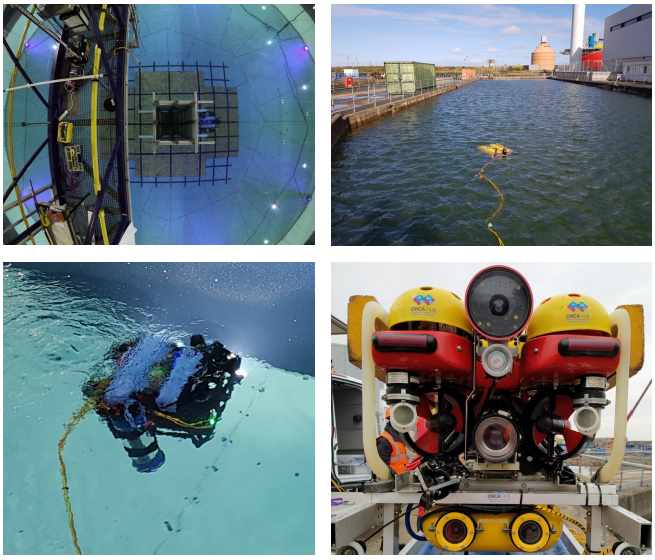

Fig. 1: Experiments were performed using BlueROV (bottom-left) and Falcon ROV (bottom-right). Both robots are equipped with a DVL, a gyroscope, an altimeter (Falcon ROV) / depth sensor (BlueROV) and a stereo camera. BlueROV was used in controlled conditions in an indoor tank (top-left) that contained an underwater motion capture system providing ground truth trajectory estimates. Falcon ROV was used to collect data in an indoor tank (in uncontrolled conditions) as well as during field trials in Blyth, UK (top-right).

In recent years, state-of-the-art visual SLAM systems have demonstrated impressive performance, which however has been mostly limited to scenarios where there is sufficient lighting and texture in the scene. In our underwater scenario, however, visual data is challenging to use due to low visibility, lack of texture or current disturbances causing the camera to point away from the asset to be inspected. Under these conditions, most state-of-the-art visual odometry systems would fail.

We propose to improve the robustness of visual SLAM in underwater environments through sensor fusion of a DVL, gyroscope and an altimeter/depth sensor. Our research is a significant step towards achieving the required reliability levels of visual SLAM for subsea industry applications. Our main contributions are:

- Extension of the state-of-the-art ORB-SLAM2 stereo camera SLAM system using motion priors calculated via *acoustic odometry*. *Acoustic odometry* refers to 6-

*These authors contributed equally.

R. Scona contributed to this work while affiliated to Heriot-Watt University as a Research Associate.

This research is supported by the ORCA Robotics Hub (EP/R026173/1).

[1] The authors are with the School of Engineering & Physical Sciences, Heriot-Watt University, UK.

[2] The author is with the School of Engineering, University of Edinburgh, UK.

DoF pose estimates computed by fusing data from a DVL, a gyroscope and an altimeter/depth sensor.
- Estimation of disjoint maps in scenarios where visual tracking is temporarily lost, allowing for continuous inspection and avoiding the need to revisit areas previously explored.
- An extensive evaluation of our approach in controlled, uncontrolled and challenging field experiments, demonstrating the robustness of our proposed system.

While our paper presents an ORB-SLAM2 specific fusion method, this general formulation could be adapted to different pose-graph SLAM frameworks.

II. Related Works

Research in underwater navigation usually relies on sensor fusion, in particular using acoustic [5] and inertial sensors. Johannsson *et al*. [1] present an approach for underwater vehicle navigation for harbour surveillance which fuses Forward-Looking Sonar (FLS) imaging and DVL in a pose graph formulation. Fallon *et al*. [6] integrate long-range acoustic constraints with GPS data from a surface vehicle and relative pose constraints from targets detected in side-scan sonar images, using Incremental Smoothing and Mapping (iSAM). Likewise He *et al*. in [3] present a modified Fast-SLAM algorithm for Autonomous Underwater Vehicle (AUV) navigation by fusing compass and DVL with a FLS.

Despite the fact that acoustic sensors have traditionally been the preferred choice for underwater environments, the use of visual sensors has increased due to the advances of visual SLAM. In particular, there are works that employ ORB-SLAM [7] in underwater scenarios, such as Hidalgo *et al*. [8] who performed an experimental evaluation of monocular ORB-SLAM in varying weather conditions. Weidner *et al*. [9] use ORB-SLAM2 [10] with stereo cameras to demonstrate dense 3-D mapping of underwater caves, producing maps that are of higher resolution than those obtained using acoustic sensors.

More recent approaches combine visual sensors with acoustic and inertial information using different fusion strategies. Kim *et al*. [11] propose a method that uses a monocular visual SLAM system as a black box motion estimator to provide pose constraints. These pose constraints are integrated into a separate state estimation system fusing DVL, IMU and depth observations within a factor graph formulation. Manzanilla *et al*. [12] fuse an IMU and Parallel Tracking and Mapping (PTAM) constraints within an Extended Kalman filter (EKF). A two-stage navigation approach is presented in [13], based on initially generating an occupancy map of the work space using markers. This map is then used to decide which odometry strategy to use, either plane extraction or feature extraction originating from a sensor fusion system integrating ORB-SLAM2, DVL and IMU within an EKF.

Tightly coupled Visual Inertial Odometry (VIO) approaches have also been considered in underwater scenarios [14]. A comparison of visual-inertial and other state estimation algorithms is presented in [15], finding that OKVIS [16], ROVIO [17] and ORB-SLAM are feasible for underwater environments. Rahman *et al*. [18] propose an extension of the visual-inertial odometry system OKVIS using information from a sonar which is further extended in [2] by introducing a robust initialisation method, real-time loop-closing and relocalisation. This approach was validated with impressive demonstrations in cave mapping applications using a diver.

In our paper, we investigate the use of a DVL, which unlike a sonar, produces explicit linear velocity estimates. We propose a method that fuses vision, DVL, gyroscope and altimeter/depth sensing. To the best of our knowledge, this is the first time that these sensors have been integrated within the graph-based formulation of a visual SLAM system and deployed in realistic underwater scenarios using a ROV for data collection.

III. Methodology

In the following section, we describe our approach for fusing *acoustic odometry* within ORB-SLAM2 [10], a sparse stereo feature-based SLAM algorithm. During the preparation of this manuscript, ORB-SLAM3 [19], an extension of ORB-SLAM2 which fuses IMU data to handle low visibility, was also published. However, a comparison between our system and ORB-SLAM3 is left as future work.

A. Notation

We represent 3D poses using transformation matrices $\mathbf{T} \in \mathbb{SE}(3)$ which are composed of a rotation matrix $\mathbf{R} \in \mathbb{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. The transformation ${}^{(e)}\mathbf{T}_{M_i \to N_j}$ denotes the pose of coordinate frame N at time j relative to coordinate frame M at time i as estimated by estimator e, following the notation introduced in [20]. We make use of v and a to denote the vision and acoustic motion estimators respectively. We refer to the camera and *acoustic odometry* coordinate frames as C and A respectively and the relative pose between these two frames is $\mathbf{T}_{A \to C}$.

B. Motion Priors From Acoustic Odometry

Acoustic odometry refers to the drifting 6-DoF pose estimate obtained by fusing information from a DVL, an altimeter/depth sensor and a gyroscope by means of an EKF. Next we briefly describe the estimates produced by each of these sensors:

- A DVL transmits an acoustic signal and measures the Doppler shift when it returns after being reflected from the bottom. The Doppler shift can be converted to a linear velocity estimate in the horizontal X-Y plane.
- Altimeters estimate the distance from the bottom, while depth sensors estimate the distance from the surface; either of these sensors can be used to estimate the vertical Z position.
- Gyroscopes estimate 3D angular velocities. These can integrated to obtain 3D rotations.

These sensors are fused in an EKF formulation to produce a continuously drifting estimate of the 6-DoF pose of the robot.

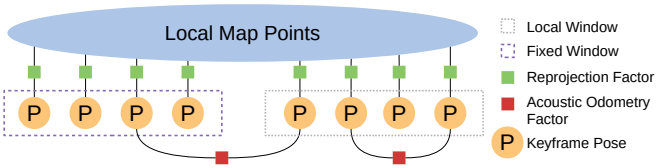

Fig. 2: Local BA factor graph. We introduce pair-wise pose constraints using *acoustic odometry*. The poses within the fixed window are not optimised but act as anchors for the pose constraints and the poses in the local window.

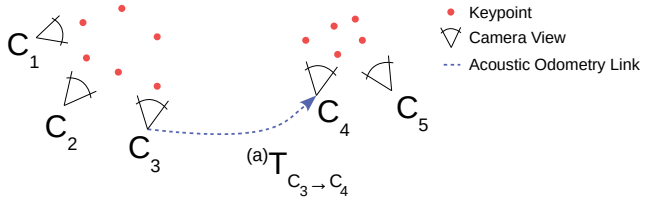

Fig. 3: If pose tracking is lost, we enable the system to initialise a new map as soon as we reach an area that can be tracked. We link the two maps using pose constraints from *acoustic odometry*.

Regarding the Falcon ROV[1], we make use of the proprietary SeeByte Copilot[2] software package which provides this EKF solution. For the BlueROV[3], we implement a custom EKF [21].

During initialisation, all degrees of freedom of this pose are zero, with the exception of the vertical coordinate which is initialised to the distance to the bottom/surface as provided by the altimeter/depth sensor. The DVL provides data at a lower frequency compared to a camera which means that acoustic information is only available for a subset of the camera frames.

Underwater environments often have little visual texture and can be problematic for achieving accurate visual odometry. It is beneficial to incorporate motion priors that can constrain visual odometry optimisations during sequences with little texture. We use the incremental pose estimates from the *acoustic odometry* to formulate $\mathbb{SE}(3)$ relative pose constraints which we implement within the ORB-SLAM2 graph optimisation procedures. The formulation of a pose constraint between poses i and j is:

$$\mathbf{r}_{i,j} = \log_{\mathbb{SE}(3)} \left({}^{(v)}\mathbf{T}_{C_i \to C_j} \, {}^{(a)}\mathbf{T}^{-1}_{C_i \to C_j} \right) \tag{1}$$

where $\log : \mathbb{SE}(3) \to \boldsymbol{se}(3)$ is the matrix logarithm which maps a transformation matrix to an element on the tangent space [22] and the ${}^{(a)}\mathbf{T}_{C_i \to C_j}$ is computed by:

$${}^{(a)}\mathbf{T}_{C_i \to C_j} = \mathbf{T}_{C \to A} \left({}^{(a)}\mathbf{T}^{-1}_{A_0 \to A_i} \, {}^{(a)}\mathbf{T}_{A_0 \to A_j} \right) \mathbf{T}_{A \to C} \tag{2}$$

The corresponding energy term is:

$$E_a(i,j) = \mathbf{r}^T_{i,j} \mathbf{\Lambda}_{i,j} \mathbf{r}_{i,j} \tag{3}$$

where $\mathbf{\Lambda}_{i,j}$ is the information matrix.

[1]https://www.saabseaeye.com/solutions/underwater-vehicles/falcon
[2]https://www.seebyte.com/products/copilot/
[3]https://bluerobotics.com/store/rov/bluerov2/

C. ORB-SLAM2 Integration

ORB-SLAM2 uses ORB features for tracking, mapping and place recognition. The map consists of 3D points with associated ORB feature descriptors and each keyframe stores its 6-DoF pose and the 2D locations of the ORB features visible within it. It maintains different types of graph structures used for different purposes:

- The covisibility graph connects all keyframes that observe a minimum number of common points. This graph is used to determine the local window, which is a small-scale graph structure containing (1) the latest keyframe, (2) those keyframes connected to it and (3) the points observed by these keyframes. The local window is used for real-time camera pose tracking and local mapping.
- The essential graph consists of a minimum spanning tree connecting all keyframes, as well as additional connections between keyframes that observe a large number of common points. This graph is augmented with loop closure connections and is used for global mapping and optimisation.

The system implements three threads for camera pose tracking, local mapping and global mapping, which all leverage bundle adjustment (BA):

- The tracking estimates the pose of the camera relative to the reference keyframe.
- Local mapping runs when a new keyframe is added to the map and it implements BA using a local window of keyframes retrieved from the covisibility graph.
- Global mapping runs when a new loop closure is detected and it involves a global pose graph optimisation of the essential graph followed by global BA to increase accuracy.

Next, we describe the modifications we made to ORB-SLAM2 to integrate *acoustic odometry* estimates.

1) Data Structures: ORB-SLAM2 maintains internal data structures to represent the 3D map points and keyframes. When available, we annotate the keyframe data structures with corresponding pose estimates from *acoustic odometry* and timestamps of these estimates. As previously mentioned, the DVL provides data at relatively low frequency compared to the camera, so this information is only added for a subset of the keyframes.

Next, we describe when the additional pose constraints are included within the estimation process.

2) Threads: ORB-SLAM2 is composed of three parallel threads that perform Tracking, Local Mapping and Loop Closing. We summarise how the algorithm was modified within these threads to integrate the pose obtained from the *acoustic odometry* step.

a) Tracking: This thread tracks the pose of the camera at frame-rate. It performs feature matching between the current frame and the reference keyframe. A motion-only BA procedure then optimises the camera pose using these correspondences.

If both the current frame and the reference keyframe have an associated *acoustic odometry* pose, we formulate

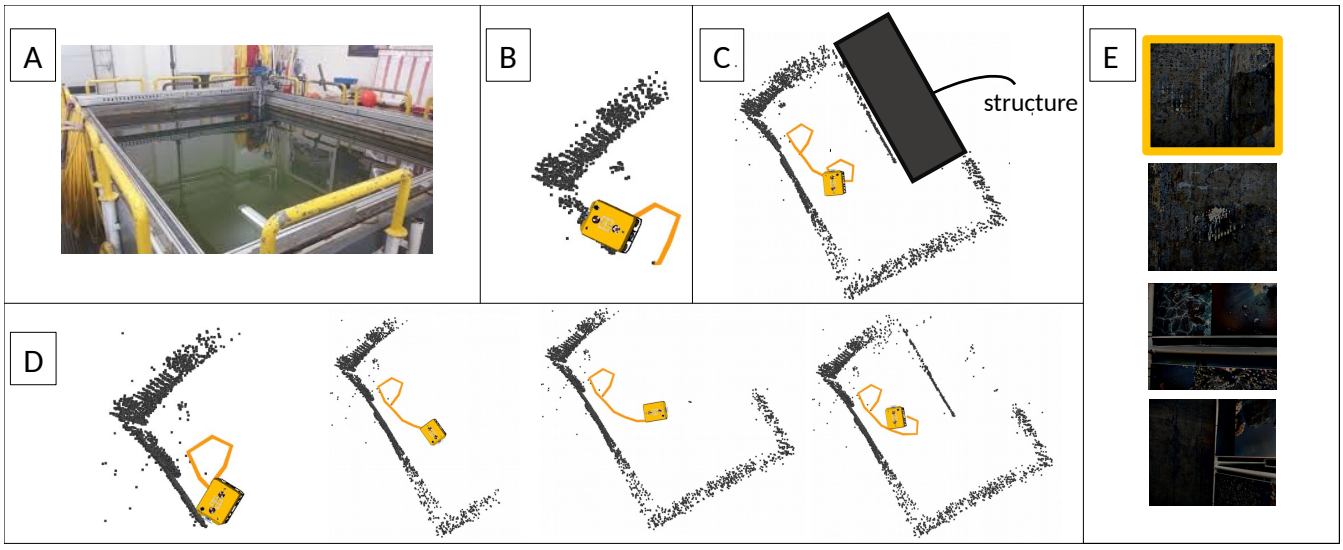

Fig. 4: The proposed algorithm reconstructs the walls of the tank in OSL-LOOP , while ORB-SLAM2 loses track after facing a featureless area. A illustrates the OSL-tank, in which a structure was submerged. B is the result obtained with ORB-SLAM2, which loses track when facing one of the walls in the tank. C presents the obtained sparse reconstruction using the proposed approach. The area in which the structure was placed was highlighted in gray. D Illustrates snapshots of the trajectory and map estimated using our proposed approach, while E shows samples of images from this sequence. The image with the orange rectangle is the one in which ORB-SLAM loses track.

an additional pose constraint and include this in the motion-only BA. The pose of the reference keyframe is kept fixed and only the pose of the current camera frame is optimised.

b) Local Mapping: This component performs local BA after a new keyframe is inserted in the map. It optimises the poses of the last keyframe and all keyframes connected to it in the covisibility graph, as well as all points observed by these keyframes. Other keyframes which observe these points but are not connected to the latest keyframe form the fixed window – they contribute to the cost function but their poses are fixed.

We modify this procedure to add pair-wise pose constraints using *acoustic odometry* information. The pairs chosen are those keyframes whose associated *acoustic odometry* poses are the closest in time. For all pairs, only the poses of the keyframes from the local window are optimised, while the poses from the fixed window stay fixed. The structure of this optimisation problem is shown in Fig. 2.

c) Loop Closing: When a loop is detected, global optimisation is performed to eliminate the drift accumulated in the trajectory. Similarly as within the local mapping procedure, we include pair-wise pose constraints between keyframes whose associated *acoustic odometry* poses are close in time. These additional constraints are included both in the pose graph optimisation as well as within global BA.

3) Joining Sub-Maps: The current ORB-SLAM2 implementation supports a single map. When tracking is lost, the system enters relocalisation mode until the robot returns to a previously visited scene and tracking can be resumed. This can be limiting in practice because a robot may perform a continuous scanning motion without returning to a previously visited scene.

To enable this use case, we perform a reinitialisation-type procedure whenever the camera encounters a new area that is sufficiently textured and can be used for tracking (assuming that tracking was lost beforehand). We use an *acoustic odometry* pose to connect the first keyframe from the new map with the keyframe from the previous map that had the temporally closest *acoustic odometry* pose. The structure of this map graph is shown in Fig 3.

With this strategy we no longer use the relocalisation procedure anymore, but instead rely on the loop closure detection system to observe if we return to a previously visited scene. While this is not a general multi-map support system (such as the one proposed in ORB-SLAM3 [19]), we find this feature useful in practice as it enables the robot to complete scanning missions where some portions cannot be tracked visually, as is demonstrated in Section IV-D.

IV. Evaluation

This section describes the quantitative and qualitative evaluation of our system.

A. Hardware

We use two ROVs to test our system – a BlueROV for testing in controlled environments and the Falcon ROV for tests in uncontrolled and field conditions. Both the Falcon ROV and BlueROV are equipped with custom underwater stereo cameras designed following [23] to work at close range (1-2m) in case of BlueROV and medium range (2-3m) in case of the Falcon ROV. Regarding the DVL, BlueROV is equipped with the Teledyne Explorer and Falcon ROV contains the Workhorse Navigator. Both robots contain MEMS gyroscopes. Finally, Falcon ROV is equipped with an altimeter while BlueROV carries a depth sensor. Table I states the update rates of these sensors. As previously

mentioned, for the Falcon ROV we make use of a proprietary EKF solution which does not provide an interface to the internal gyroscope and altimeter.

Both the EKF and ORB-SLAM2 systems work in real-time and our modifications contribute negligible extra computational load on the system which still works in real-time.

Robot	Sensors	Update Rates
Falcon ROV	camera	12 Hz
	DVL	2 Hz
	EKF	2 Hz
BlueROV	camera	30 Hz
	DVL	7 Hz
	gyroscope	30 Hz
	depth sensor	30 Hz
	EKF	7 Hz

TABLE I: The update rates of the sensors used by Falcon ROV and BlueROV.

B. Datasets

We recorded two different sets of sequences, using the ROVs described in Section IV-A, in order to evaluate our proposed algorithm both quantitatively and qualitatively.

1) BlueROV Sequences: These sequences are designed to test the algorithm and evaluate its quantitative performance in controlled conditions that aim to mimic the challenges found in field data. They were recorded in FloWave TT[4], an ocean energy research facility equipped with an underwater motion capture system that provides ground truth of the robot location. We created a structure to be used for inspection. A top-view of this setup is depicted in Fig. 1 (top-left). We collected the following sequences:

- **EASY-LOOP**: Slow circular movement around the structure using manual control and with the robot's lights turned on.
- **HARD-LOOP**: Fast movement around the structure using manual control and with the robot's lights off.
- **LIGHTS**: Slow movement around the structure with the robot's lights turned off initially and then turned on halfway through the inspection.

2) Falcon Sequences: These sequences are a mix of two test sequences in a fresh water tank in uncontrolled conditions and one field sequence:

- **OSL-LOOP**: Test sequence recorded in the Ocean Systems Lab (OSL) at Heriot-Watt University, by placing the ROV inside a small fresh water tank *(4m x 3m x 2m)*, containing a submerged structure. The sequence consist of a 360 degree loop, where the robot views the outer walls of the tanks first and then navigates around the structure at the end.
- **TUBE**: Test sequence recorded in the wave tank at Heriot-Watt University of the robot inspecting a submerged tube. This sequence is challenging as the robot occasionally loses view of the structure.
- **BLYTH**: Field data recorded in the Offshore Renewable Energy Catapult facilities in Blyth, United Kingdom, where the robot is executing a slow sideways motion to inspect a submerged asset. This sequence includes extremely hard visibility conditions with poor lighting and floating particles in the water.

[4]https://www.flowavett.co.uk/

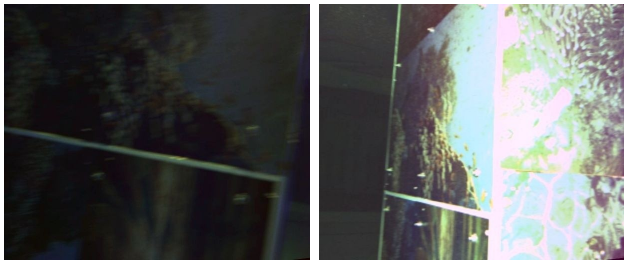

Fig. 5: Challenging sample images from the HARD-LOOP (left) and LIGHTS (right) sequences. Motion blur and overexposed images cause the camera pose tracking of ORB-SLAM2 to fail.

Sequence	Distance (m)	Time (s)	ORB-SLAM2 Error (m)	Proposed Error (m)
EASY-LOOP	9.52	116.4	0.14	0.14
HARD-LOOP	11.49	113.26	FAIL	**0.33**
LIGHTS	10.13	69.68	FAIL	**0.34**

TABLE II: Our proposed algorithm improves over the ORB-SLAM2 baseline. The reported error corresponds to the Absolute Trajectory RMSE. Each sequence was evaluated three times and the best achieved result is reported.

C. Quantitative Results

We evaluate the performance of our system using data collected with the BlueROV which contains ground truth trajectory estimates from a motion capture system. We evaluate the trajectories estimated by our system against this ground truth data and report the Absolute Trajectory RMSE using [24].

The results are presented in Table II, which illustrates the performance of ORB-SLAM2 and the proposed approach in 3 sequences. EASY-LOOP has good visual conditions and both algorithms run successfully in this sequence. In the case of HARD-LOOP, ORB-SLAM2 fails due to strong image blur caused by the fast motion of the robot (Fig. 5 left). In the LIGHTS sequence, ORB-SLAM2 fails to track due to the sudden change in illumination (Fig. 5 right). In contrast, the robust visual odometry of our system enables it to complete these missions.

D. Qualitative Results

The qualitative evaluation was performed using the sequences recorded with Falcon ROV. Due to the absence of ground truth information we focus the discussion on the consistency of the generated maps.

The results for OSL-LOOP are presented in Fig. 4, in which a comparison with ORB-SLAM2 results is also included. As soon as ORB-SLAM2 faces a featureless region, such as the tank's walls, the algorithm loses track and never recovers. On the other hand, our approach completes the loops and reconstructs the tank according to its known dimensions, as well as the submerged structure. A similar

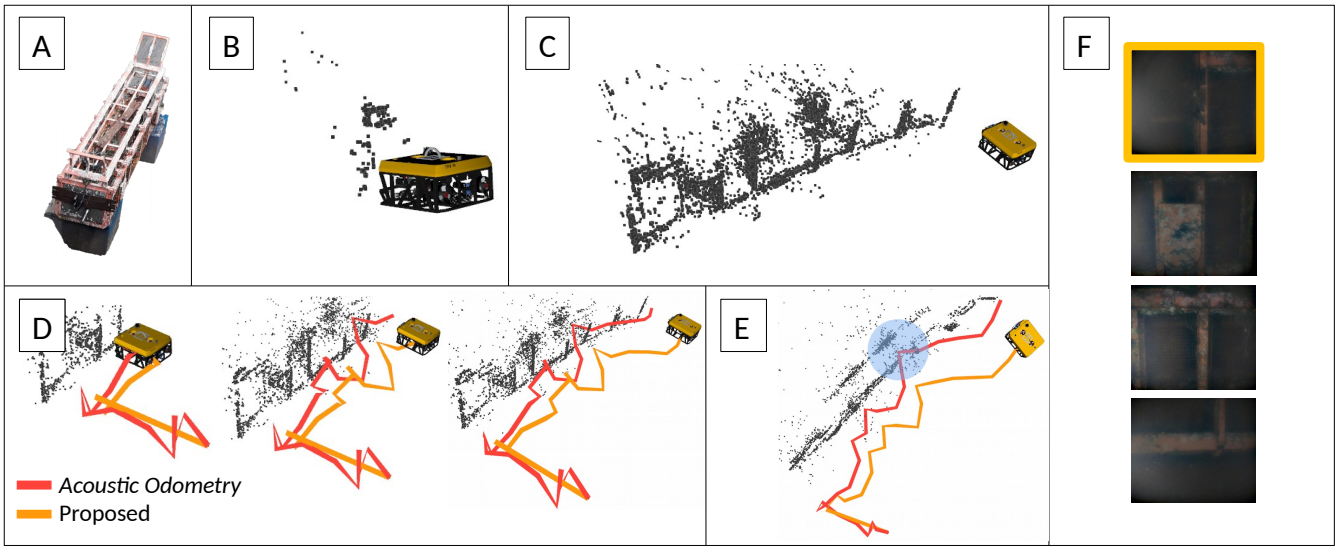

Fig. 6: In challenging visibility conditions, as the ones observed in BLYTH, the proposed approach outperforms ORB-SLAM2. A illustrates the submerged structure that was used for inspection. B is the result obtained with ORB-SLAM2, which loses track seconds after starting the inspection. C presents the obtained sparse reconstruction using the proposed approach. D shows snapshots of the trajectory estimated using our proposed approach (orange) and the one generated by *acoustic odometry* (red). E shows that the drift in *acoustic odometry* leads to an estimate which collides with the reconstruction (blue), while our method corrects for this drift using vision. F shows samples of images from this sequence. The image with the orange rectangle is the one in which ORB-SLAM2 loses track.

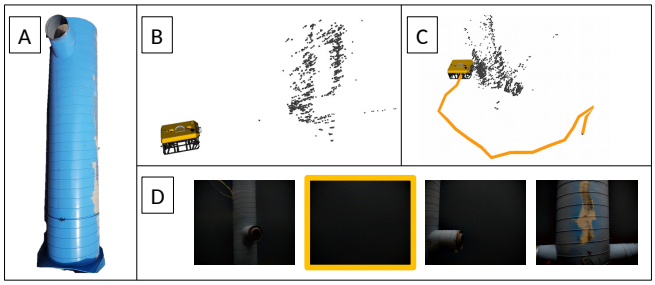

Fig. 7: In cases in which the robot loses view of the structure, the proposed approach outperforms ORB-SLAM2. A illustrates the submerged structure that was used for inspection. B is the result obtained with ORB-SLAM2, where the system loses track shortly after mapping began. C presents the obtained sparse reconstruction using the proposed approach, as well as the estimated trajectory. D shows the sequence of images. The image with the orange rectangle is the one in which ORB-SLAM2 loses track.

situation occurs during the TUBE sequence (Fig. 7), where ORB-SLAM2 fails as soon as the robot loses view of the structure. Our system generates a new map when the structure is in view again and is able to finalise the mission. Finally, in the BLYTH field sequence (Fig. 6) the visibility conditions are very poor and ORB-SLAM2 fails immediately after starting the experiment. Our system manages to reconstruct macro features such as the door and other frame elements of the structure. Our system also reduces the drift compared to *acoustic odometry* by using visual data.

Thanks to the proposed visual odometry and sub-map generation components (as described in Section III-C.3) of our approach, the robot can perform robustly in these challenging conditions, allowing it to successfully complete the missions.

V. Discussion

We presented a visual-acoustic SLAM system that fuses data from a DVL, a gyroscope and an altimeter/depth sensor which was successfully deployed in two different ROVs and in controlled, uncontrolled and field experiments. We showed that our approach outperforms a state-of-the-art visual SLAM baseline, ORB-SLAM2, both quantitatively and qualitatively. Our novel approach prevails where ORB-SLAM2 tends to fail: under adverse but common underwater scenarios, such as low lighting, fast motions and feature-less environments. We showed that advantages are realised for tracking and localisation robustness and also reconstruction, enabled by our sub-map generation procedure. The result is a robust system that can perform continuous mapping of a given area.

As we focus on an inspection application, we always assume the DVL has bottom lock and have not evaluated our approach in sea mode. We also did not evaluate our system in the case of severe DVL outliers. Our system implements a large covariance for *acoustic odometry* and the main benefit from this integration is in the multi-map support. Qualitatively, we observe that integrating *acoustic odometry* in scenarios where visual odometry already works well does not lead to further improvement.

Future work includes the integration of dense mapping into our approach, in order to produce dense reconstructions that can be more useful for visual inspection and which could also be used for collision-free path planning.

Acknowledgements The authors thank Joshua Roe and Roman Gabl for help during data collection at FloWave. We also thank Leonard McLean and Leonard Newbrook for their support in the set-up of the Falcon ROV.

REFERENCES

[1] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, "Imaging sonar-aided navigation for autonomous underwater harbor surveillance," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[2] S. Rahman, A. Q. Li, and I. Rekleitis, "SVIn2: An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[3] B. He, Y. Liang, X. Feng, R. Nian, T. Yan, M. Li, and S. Zhang, "AUV SLAM and experiments using a mechanical scanning forward-looking sonar," *Sensors*, 2012.

[4] T. Łuczyński, M. Pfingsthorn, and A. Birk, "The pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings," *Ocean Engineering*, 2017.

[5] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard, "Relocating underwater features autonomously using sonar-based SLAM," *IEEE Journal of Oceanic Engineering*, 2013.

[6] M. F. Fallon, M. Kaess, H. Johannsson, and J. J. Leonard, "Efficient AUV navigation fusing acoustic ranging and side-scan sonar," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[8] F. Hidalgo, C. Kahlefendt, and T. Braunl, "Monocular ORB-SLAM application in underwater scenarios," *OCEANS*, vol. 1, 2018.

[9] N. Weidner, S. Rahman, A. Q. Li, and I. Rekleitis, "Underwater cave mapping using stereo vision," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2017.

[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robotics*, 2017.

[11] A. Kim and R. M. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Trans. Robotics*, 2013.

[12] A. Manzanilla, S. Reyes, M. Garcia, D. Mercado, and R. Lozano, "Autonomous navigation for unmanned underwater vehicles: Real-time experiments using computer vision," *IEEE Robotics and Automation Letters*, 2019.

[13] A. G. Chavez, Q. Xu, C. A. Mueller, S. Schwertfeger, and A. Birk, "Adaptive navigation scheme for optimal deep-sea localization using multimodal perception cues," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS) Workshop*, 2019.

[14] F. Shkurti, I. Rekleitis, M. Scaccia, and G. Dudek, "State estimation of an underwater robot using visual and inertial information," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[15] B. Joshi, S. Rahman, M. Kalaitzakis, B. Cain, J. Johnson, M. Xanthidis, N. Karapetyan, A. Hernandez, A. Q. Li, N. Vitzilaios, and I. Rekleitis, "Experimental comparison of open source visual-inertial-based state estimation algorithms in the underwater domain," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[16] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial SLAM using nonlinear optimization," *Robotics: Science and Systems (RSS)*, 2013.

[17] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 298–304.

[18] S. Rahman, A. Q. Li, and I. Rekleitis, "Sonar visual inertial SLAM of underwater structures," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2018.

[19] 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," *arXiv preprint arXiv:2007.11898*, 2020.

[20] R. Scona, S. Nobili, Y. R. Petillot, and M. Fallon, "Direct visual SLAM fusing proprioception for a humanoid robot," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[21] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the Robot Operating System," *Advances in Intelligent Systems and Computing*, 2016.

[22] J.-L. Blanco, "A tutorial on SE(3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, vol. 3, 2010.

[23] T. Łuczyński, P. Łuczyński, L. Pehle, M. Wirsum, and A. Birk, "Model based design of a stereo vision system for intelligent deep-sea operations," *Measurement*, 2019.

[24] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM." https://github.com/MichaelGrupp/evo, 2017.