SM/VIO: Robust Underwater State Estimation Switching Between Model-based and Visual Inertial Odometry

Bharat Joshi[a*], Hunter Damron[b*], Sharmin Rahman[a], and Ioannis Rekleitis[a]

***Abstract*— This paper addresses the robustness problem of visual-inertial state estimation for underwater operations. Underwater robots operating in a challenging environment are required to know their pose at all times. All vision-based localization schemes are prone to failure due to poor visibility conditions, color loss, and lack of features. The proposed approach utilizes a model of the robot's kinematics together with proprioceptive sensors to maintain the pose estimate during visual-inertial odometry (VIO) failures. Furthermore, the trajectories from successful VIO and the ones from the model-driven odometry are integrated in a coherent set that maintains a consistent pose at all times. Health-monitoring tracks the VIO process ensuring timely switches between the two estimators. Finally, loop closure is implemented on the overall trajectory. The resulting framework is a robust estimator switching between model-based and visual-inertial odometry (SM/VIO). Experimental results from numerous deployments of the Aqua2 vehicle demonstrate the robustness of our approach over coral reefs and a shipwreck.**

I. INTRODUCTION

This paper proposes a novel framework for solving the robustness problem of state estimation underwater. Central to any autonomous operation is the ability of the robot to know where it is with respect to the environment, a task described under the general term of state estimation. Over the years many different approaches have been proposed; however, state estimation underwater is a challenging problem that still remains open. Vision provides rich semantic information and through place recognition results in loop closures. Unfortunately, as demonstrated in recent work on comparing numerous open-source packages of visual and visual/inertial state estimation [1], [2], in an underwater environment there are frequent failures for a variety of reasons. In contrast to above water scenarios, GPS-based localization is impossible. In addition to the traditional difficulties of vision-based localization, the underwater environment is prone to rapid changes in lighting conditions, limited visibility, and loss of contrast and color information with depth. Light scattering from suspended plankton and other matter causes "snow effects" and blurring, while the incident angle at which light rays hit the surface of the water can change the visibility at different times of the day [3]. Finally, as light travels at increasing depths, different parts of its spectrum are absorbed; red is the first color that is seen as black, and eventually orange, yellow, green, and blue follow [4], [5]. In addition to all the above underwater specific challenges, an unknown environment often presents areas where there are no visible landmarks. For example, in Fig. 1 an Aqua2 [6] Autonomous Underwater Vehicle (AUV) mapping the deck of a shipwreck reaches the starboard side where the front cameras see only empty water with no features.

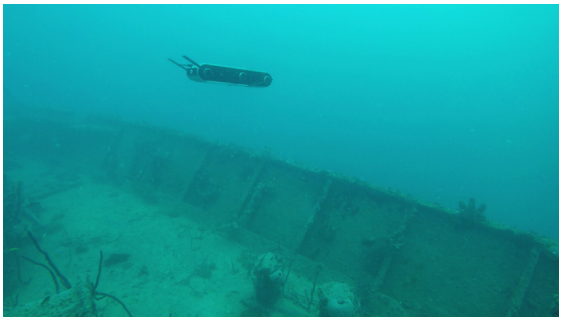


Fig. 1: Aqua2 AUV navigating over the Stavronikita shipwreck, Barbados. The front cameras are only seeing blue water when approaching the side of the wreck.

Visual inertial odometry (VIO) has been used for state estimation in a multitude of environments such as indoor, outdoor and even gained some traction in harsh environments such as underwater [7]. While most VIO research often focuses on improving accuracy, robustness is as critical for autonomous operations. If VIO fails during deployment the results could be catastrophic leading to vehicle loss. From our early investigations [1], [2], many vision-based approaches diverge, or outright fail, sometimes at random; however, deploying a vehicle underwater in autonomous mode requires that it will return to base, or a collection point, during every deployment. It is very important for AUVs to be able to keep track of their pose; even with diminished accuracy; over the whole operation. We propose switching between VIO and a model-based estimator addressing the accuracy and robustness of state estimation by identifying failure modes, generating robust predictors for estimator divergence/failure, always producing a pose estimate.

The core of the proposed approach is a robust switching estimator framework, which always provides a realistic estimate reflecting the true state of the vehicle. First of all the health of VIO [8], [9] is monitored by tracking the number of features detected, their spatial distribution, their quality, and their temporal continuity. By utilizing the measures described above when an estimator starts diverging, before complete failure, an alternative estimator is introduced based on sensor inputs robust to underwater environment changes. For example, there is a model-based estimator [10], [11] used for controlling the Aqua2 vehicles combining the

* The first two authors have contributed equally to the paper.

[a]University of South Carolina, Columbia, SC, USA, yiannisr@cse.sc.edu, {bjoshi,srahman}@email.sc.edu.

[b]Epic Systems Corporation, Madison, WI, USA, 53703, hdamron@epic.com

The authors would like to acknowledge the generous support of the National Science Foundation grants (NSF 2024741, 1943205).

inertial and water depth signals together with the flipper configuration and velocity [12], [13]; when the visual/inertial input deteriorates, the proposed system switches to the model-based estimator until the visual/inertial estimates are valid again. The choice of switching-based loosely coupled fusion of odometry estimates ensures flexibility in choosing both the VIO and the conservative estimator in a modular fashion. The two estimators switch back and forth based on the health status of the VIO estimator. Finally, a loop-closure framework ensures the consistent improvement of the combined estimator. Our main contribution is a robust switching-based state estimation framework termed Robust Switching Model-based/Visual Inertial Odometry (SM/VIO) capable of keeping track of an AUV even when VIO fails. This allows the AUV to carry out underlying tasks such as path planning, coverage, and performing motion patterns maintaining a steady pose and relocalize when visiting previous areas. Extensive experiments over different terrains validate the contribution of the proposed robust switching estimator framework in maintaining a realistic pose of the AUV at all times. In contrast, state-of-the-art VIO algorithms [7], [14]–[17] result in a much higher error or even complete failure.

II. RELATED WORK

In recent years a plethora of open source packages addressing the problem of vision-based state estimation has appeared [8], [9], [16], [18]–[26]. Quattrini Li *et al*. [1] compared several packages on a variety of datasets to measure the performance in different environments. Extending the above comparison with a focus on the underwater domain, Joshi *et al*. [2] investigated the performance of VIO packages. The above comparisons demonstrated that many packages require special motions [19], or only work for a limited number of images [27], [28], or are strictly offline [29]. Furthermore, intermittent failures were observed, the most common explanation being the random nature of the RANSAC technique [30] utilized by most of them. The underwater state estimation approach SVIn2 by Rahman *et al*. [7] demonstrated improved accuracy and robustness; however, it did not provide any assurances for uninterrupted estimates, which is the focus of this paper.

Utilizing an AUV to explore an underwater environment has gained popularity over the years. Sonar and stereo estimation for object modeling has been proposed in [31], [32]. Nornes *et al*. [33] acquired stereo images utilizing an ROV off the coast of Trondheim Harbour, Norway. In [34] a deep-water ROV is adopted to map, survey, sample, and excavate a shipwreck area. Sedlazeck *et al*. [35], reconstructed a shipwreck in 3D by pre-processing images collected by an ROV and applying a Structure from Motion based algorithm. The images used for testing such an algorithm contained some structure and a lot of background, where only water was visible. Submerged structures were reconstructed in 3D [36]. Finally, recent work by Nisar *et al*. [37] proposed the use of a model-based estimator to calculate external forces in addition to the pose of aerial vehicles, ignoring failure modes of VIO. In all previous work, when the state estimation failed there was no recovery. In contrast, the proposed approach of SM/VIO for underwater environments addresses the VIO failure and the AUV can continue operations until reaching another feature-rich area.

The use of switching estimators (also called observers) has not been applied in many mobile robotics applications and not, to our knowledge, to an AUV. Liu [38] presented a generic approach for non-linear systems. Suzuki *et al*. [39] utilized a switching observer to model ground properties together with the robot's kinematics. Manderson *et al*. [40] utilized a model estimator in conjunction with Direct Sparse Odometry [41] without monitoring the health, switching estimators, and merging the two trajectories into one.

III. THE PROPOSED SYSTEM

a) **Overview:** The proposed approach (SM/VIO) utilizes a model-based estimator termed primitive estimator (PE), utilizing the water depth sensor, the IMU, and the motor commands to propagate the state of the AUV forward when the visual-inertial estimator fails; see Fig. 3(a) for an estimate from PE. It is worth noting that the AUV is using the same model to navigate, as such the PE estimate of the lawnmower pattern in Fig. 3(a) follows the exact pattern, however, does not correspond to the actual trajectory which was affected by external forces (e.g. water current). When VIO is consistent it is the preferred estimator having higher accuracy due to the exteroceptive sensors (vision and acoustic). Key to the proposed approach is a health monitor process that tracks the performance of VIO over time and informs a decision for switching between VIO and PE; see Fig. 3(c) for the switching estimator trajectory, where the switch points are marked green. When the VIO restarts tracking successfully, the health monitor informs the switch from the PE to the VIO. Throughout this process a consistent pose is maintained. More specifically, when the VIO fails, the PE is initialized with the last accurate pose from VIO, and when the VIO restarts the last pose of PE is utilized. Finally, during VIO controlled operations, loop closure is performed, also optimizing the PE produced trajectories; the complete framework is outlined in Fig. 2. Following the approach of Joshi *et al*. [42], the stable 3D features are tracked and their position is updated after every loop closure, thus resulting into a consistent point cloud. Next we discuss the individual components of SM/VIO.

The target vehicle is the Aqua2 AUV [6], an amphibious hexapod robot. Underwater, Aqua2 utilizes the motion from six flippers, each actuated independently by an electric motor. The robot's pose is described using the vector $\mathbf{x} = \left[{}_W\mathbf{p}_I^T, \quad {}_W\mathbf{q}_I^T\right]$, ${}_W\mathbf{p}_I^T = [x, y, z]$ represents the position of the robot in the world frame, and ${}_W\mathbf{q}_I^T = [q_w, q_x, q_y, q_z]$ is the quaternion representing the robot's attitude. Aqua2 vehicles are equipped with three cameras, an IMU, and a water pressure sensor.

b) **Primitive Estimator:** The primitive estimator maintains a local copy of the robot's pose $[{}_W\mathbf{p}_I^T, {}_W\mathbf{q}_I^T]$, which is updated at a rate of $100\,\text{Hz}$. The IMU provides an absolute

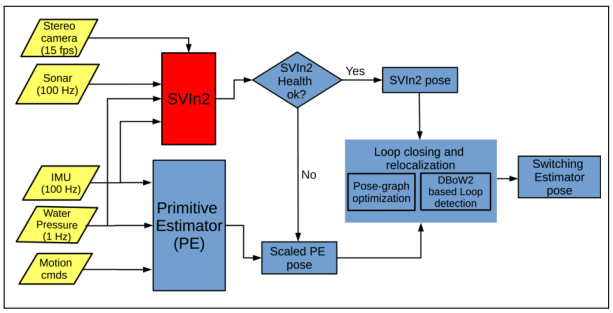


Fig. 2: Overview of the switching estimator.

measurement of ${}_W\mathbf{q}_I^T$. The velocity of the robot is estimated by the forward speed command v_x and the heave (up/down) command v_z sent to the Aqua2 during field trails. These commands are used by Aqua2 to perform motion primitives and control the flipper motion. Since the same commands are used for Aqua2 control and PE predictions, the resulting PE trajectories will look perfectly aligned with the desired motion primitives; this is a drawback of just using PE prediction. At each time step t, the position is updated by

$$ {}_W\mathbf{p}_{I,t+1} \coloneqq {}_W\mathbf{p}_{I,t} + {}_W\mathbf{R}_I[v_x^t, 0, v_z^t]^T \Delta t_{t,t+1} \qquad (1) $$

where ${}_W\mathbf{R}_I$ is the rotation matrix corresponding to ${}_W\mathbf{q}_I$. Because the water pressure sensor provides an absolute measurement of depth, this measurement is used instead of the above estimate for z. Moreover, the forward velocity estimates are correct only up to scale depending on external forces (*e.g.* ocean currents) and acceleration measurements error accumulation. Hence, before integrating the PE trajectory into the robust switching estimator, we scale the PE trajectory using the scaling factor between the PE and the VIO trajectory, as explained later.

c) **SVIn2 Review:** We use a VIO system that fuses information from visual, inertial, water pressure (depth), and acoustic (sonar) sensors presented in Rahman *et al.* [7]–[9], termed SVIn2. More specifically, SVIn2 estimates the state of the robot by minimizing a joint estimate of the reprojection error and the IMU error, with the addition of the sonar error and the water depth error. SVIn2 performs non-linear optimization on sliding-window keyframes using the reprojection error and the IMU error term formulation similar to Leutenegger *et al.* [14]. The depth error term can be calculated as the difference between the AUV's position along the z direction and the water depth measurement provided by a pressure sensor.

Loop-closing and *relocalization* is achieved using the binary bag-of-words place recognition module DBoW2 [43]. The loop closure module maintains a pose graph with odometry edges between successive keyframes and a loop-closure odometry edge is added between the current keyframe and a loop closure candidate when they have enough descriptor matches and pass PnP RANSAC-based geometric verification. For a complete description, please refer to [7].

d) **Health Monitoring:** As described in earlier studies [1], [2], estimators often diverge or outright fail even in conditions where they were working before; intermittent failures are much more challenging in the field. Robustness measures and divergence predictors are crucial in detecting imminent failures. To monitor the health of the vision-based state estimator, we employ the following criteria hierarchically; the most important criterion is checked first. The VIO health is evaluated based on the following conditions hierarchically and considered untrustworthy based on:

1) Keyframe detection. If a keyframe has not been detected after kf_wait_time seconds the VIO has failed. The only exception is when the system is stationary (zero velocity).
2) The number of triangulated 3D keypoints that have feature detections in the current keyframe is less than a specified threshold, *min_kps*. We found that *min_kps* between 10-20 worked well.
3) The number of feature detections per quadrant, in the current keyframe, is less than a specified threshold, *min_kps_per_quadrant*. To account for situations where there are high number of features detected robustly in a small area; see Fig. 3(e-f) where the bottom two quadrants contain all the features. The quadrant criterion is applied only if the total number of feature detections is less than $10 \times min\_kps\_per\_quadrant$.
4) The ratio of new keypoints to the total keypoints is more than 0.75. The newly triangulated points are those that were not observable previously.
5) The ratio of keypoints with feature detector response less than the average feature detector response in the current keyframe to the total keypoints is more than 0.85. The choice of a high threshold for the ratio is motivated by the fact that hierarchically more important criteria have already been satisfied. Hence, this criterion has less importance overall.

Please note, the choice of the above parameters is flexible. For instance, the minimum number of tracked keypoints should be higher than the minimum number of points required for relative camera pose estimation using epipolar geometry. Thus, these parameters should only be taken as reference. During our experiments, we found out that changing the parameters slightly does not change the performance of the switching estimator greatly and the parameters where selected through experimental verification.

e) **Integration of SVIn2 and Primitive Estimator results:** Utilizing the framework described in Rahman *et al.* [8], [9] the graph SLAM formulation, based on the Ceres package [44], is augmented to consider estimates from multiple observers thus maintaining the history of the estimates and enabling loop closures.

We denote the poses SVIn2 and PE as ${}_W\mathbf{T}_{sv}$ and ${}_W\mathbf{T}_{pe}$, respectively, representing them as homogeneous 4×4 transformation matrices. The goal of the integration process is to provide a robust switching estimator pose ${}_W\mathbf{T}_{ro}$ which matches ${}_W\mathbf{T}_{sv}$ locally when SVIn2 is properly running, and matches ${}_W\mathbf{T}_{pe}$ locally when SVIn2 is reporting failure. To find the scaling factor between SVIn2 and PE, we compute the ratio of the two trajectory lengths when both estimators

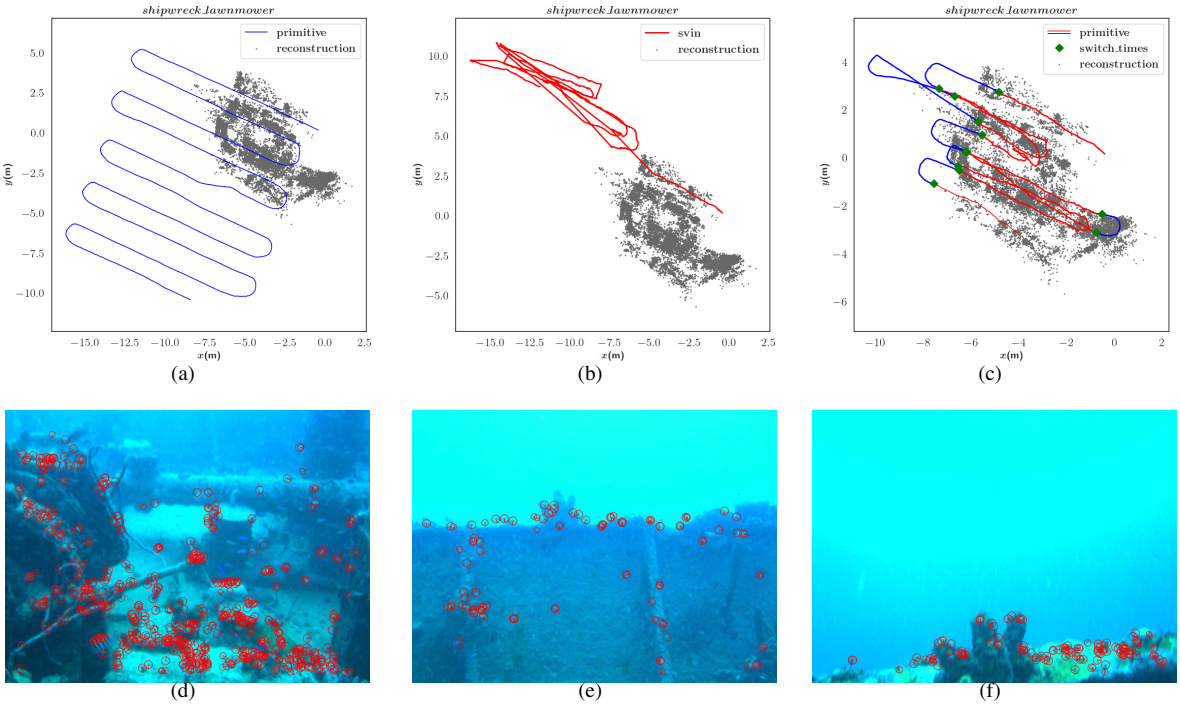


Fig. 3: First row, an overview example: (a) Trajectory according to the primitive estimator; PE believes the AUV performed a near perfect lawnmower pattern. (b) Trajectory according to SVIn2 [9]; due to tracking loss the VIO is way off the actual wreck. (c) Trajectory resulting from the proposed method; the switching estimator utilized the robust parts of VIO (in red) switching to PE when tracking was lost (in blue). The stable 3D features detected are plotted as grey points. Second row, characteristic images of the shipwreck: (d) the AUV is over the wreck seeing the deck; (e) the AUV is approaching the side of the wreck, still able to localize, but the number of features decreases; (f) the AUV is at the edge of the wreck seeing mostly blue water and the estimator switches from VIO to PE.

are tracking well. More specifically, we compute the relative distance travelled as estimated by PE and SVIn2 between successive keyframes at time t and $t+1$ and compute the scaling factor s as:

$$s = \frac{\sum \| {}_W\mathbf{R}_{sv,t}^{-1}({}_W\mathbf{P}_{sv,t+1} - {}_W\mathbf{P}_{sv,t}) \|}{\sum \| {}_W\mathbf{R}_{pr,t}^{-1}({}_W\mathbf{P}_{pr,t+1} - {}_W\mathbf{P}_{pr,t}) \|} \tag{2}$$

The scaling factor keeps updating over time whenever SVIn2 is tracking, to account for any changes in external factors. For the sake of convenience, we assume that the PE pose ${}_W\mathbf{T}_{pe}$ is appropriately scaled by the scaling factor, s, to match the SVIn2 scale. Initially, when SVIn2 starts tracking, ${}_W\mathbf{T}_{sv}$ is equivalent to ${}_W\mathbf{T}_{ro}$. When SVIn2 fails, we keep track of robust estimator pose ${}_W\mathbf{T}_{ro}^{st}$ and primitive estimator pose ${}_W\mathbf{T}_{pe}^{st}$ at switching time, st. When PE is working normally, we compute the relative displacement of the current PE pose with respect to PE pose at the time of switching by ${}_W\mathbf{T}_{pe}^{st^{-1}} \cdot {}_W\mathbf{T}_{pe}$. This local displacement is then applied to the robust estimator pose using Eq. 3 while making sure that the robust estimator pose tracks the PE pose locally during this time.

$${}_W\mathbf{T}_{ro} \coloneqq {}_W\mathbf{T}_{ro}^{st} \cdot {}_W\mathbf{T}_{pe}^{st^{-1}} \cdot {}_W\mathbf{T}_{pe} \tag{3}$$

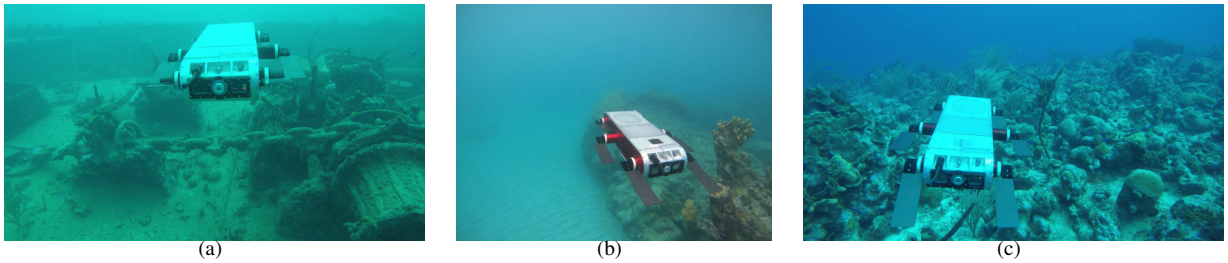
It should be noted that ${}_W\mathbf{T}_{ro}^{st} \cdot {}_W\mathbf{T}_{pe}^{st^{-1}}$ remains constant until SVIn2 starts tracking again.

Similarly, when switching from the primitive estimator back to SVIn2, the robust estimator tracks the local displacement from SVIn2 using Eq. 4 with ${}_W\mathbf{T}_{sv}^{st}$ remaining constant until next switch to PE occurs.

$${}_W\mathbf{T}_{ro} \coloneqq {}_W\mathbf{T}_{ro}^{st} \cdot {}_W\mathbf{T}_{sv}^{st^{-1}} \cdot {}_W\mathbf{T}_{sv} \tag{4}$$

We make sure that the robust estimator tracks PE locally when SVIn2 fails and tracks SVIn2 again when it recovers as VIO is the preferred estimator maintaining robust uninterrupted pose estimate. As SVIn2 is capable of maintaining an accurate estimate in the presence of brief failures of visual tracking by relying on inertial data, it is not desirable to switch between SVIn2 and PE back and forth frequently, as this introduces additional noise. To reduce frequent switching between estimators, we wait for a small number of successive tracking failures to switch from SVIn2 to PE and vice-versa.

When the VIO frontend can not detect or track enough keypoints to initialize a new keyframe, no keyframe information is generated. In this case, we wait for the specified time (set as a parameter) if we do not receive any keyframe information from SVIn2 for kf_wait_time (generally set between 1 to 3 secs), we switch to the primitive estimator. Furthermore, we need to introduce these keyframes into the pose graph differently than regular keyframes as they only contain the odometry information from PE. These keyframes



(a) (b) (c)

Fig. 4: Three environments where the AUV was deployed (Barbados): (a) over a shipwreck performing a lawnmower pattern; (b) over a mixed sand and coral area performing multiple squares; (c) over a coral reef performing a lawnmower pattern.

cannot be used for loop closure as they do not possess the keyframe image, features, and the 3D keypoints (used for geometric verification using PnP RANSAC) associated with them. It is worth noting that even if the SVIn2 health status is bad, we can use the keyframes originated from SVIn2 for loop closure.

IV. EXPERIMENTS

a) **Datasets:** The Aqua2 AUV has been deployed in a variety of challenging environments including shipwrecks, see Fig. 4(a); areas with sand and coral heads, see Fig. 4(b); and coral reefs, see Fig. 4(c). During each deployment, Aqua2 performs predefined trajectory patterns while using the odometry information from the PE. We have tested our approach on the following datasets:

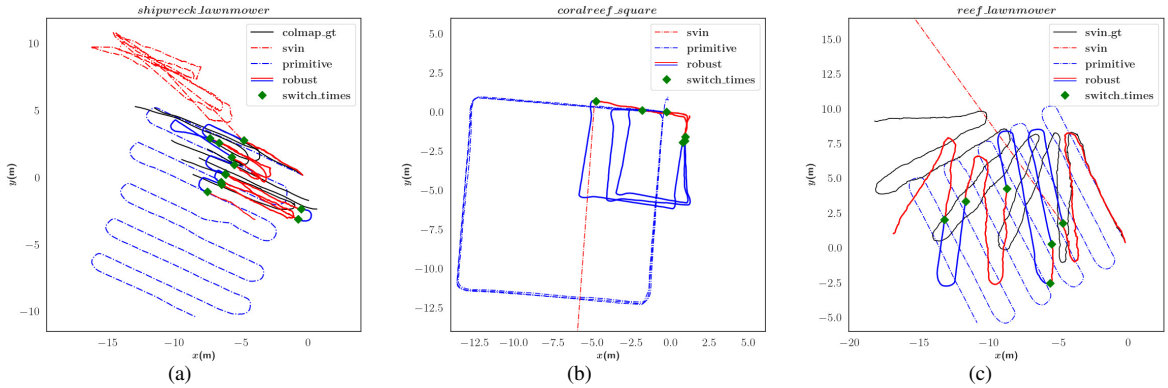
- **lawnmower over shipwreck**: The Aqua2 AUV performing a lawnmower pattern over the Stavronikita shipwreck, Barbados. During operations around shipwrecks, a common challenge is the lack of features when the wreck is out of the field of view; for example, while mapping the superstructure, the AUV can move over the side of the wreck (see Fig. 3(e-f)), thus facing the open water with no reference. Since VIO is not able to track while facing open water, the AUV's pose cannot be estimated correctly without using the PE. We obtain the ground truth trajectory for the section with the shipwreck in view by using COLMAP [45], scale enforced using the rig constraints.
- **squares over coral reef**: The Aqua2 AUV performed square patterns over an area with sand and coral heads, Barbados; see Fig. 4(b). During operations over coral reefs, drop-offs present similar conditions as wrecks, where the vehicle is facing blue water or a sandy patch. In addition, patches of sand present feature-impoverished areas where VIO fails.
- **lawnmower over coral reef**: The Aqua2 performed a lawnmower pattern over a coral reef, Barbados; see Fig. 4(c). During operations, the VIO was able to track successfully the whole trajectory. This dataset was later artificially degraded to simulate loss of visual tracking. Utilizing the consistent track produced by VIO as ground truth, a quantitative study of the switching estimator is presented. It's worth mentioning that COLMAP was not able to register images during strips with fast rotation; hence not used for ground truth.

b) **Trajectory Estimation:** Trajectories were produced with PE, SVIn2, and the proposed SM/VIO estimators. Figure 5 presents the resulting trajectories for the three datasets. In all cases the PE trajectory (blue dash-dotted line) accurately traced the requested pattern as the primitive estimator was also used to guide the robot. The VIO (SVIn2) (red dash-dotted line) diverged when visual tracking failed. Finally the proposed estimator SM/VIO (solid red and blue line with green diamonds marking the switching of estimators) tracked consistently the pose of the AUV.

The shipwreck_lawnmower dataset presents a very challenging scenario, the AUV swims over the deck, VIO tracks consistently the feature-rich clutter (Fig. 3(d)), then the AUV approaches the sides of the wreck, the number of features is reduced (Fig. 3(e)) until it goes over the side (Fig. 3(f)) and faces blue water. As the detected features are drastically reduced the VIO continues forward, moving further away from the true position. It is worth noting that several loop closures kept the VIO estimate close enough to the wreck structure but in the wrong area. The proposed framework switched to the PE upon loss of visual tracking as can be seen from the green diamond signifying the switch in Fig. 5(a). COLMAP was able to register images in sections with shipwreck in view.

In the reef_square dataset the AUV performed three squares over an area with some coral heads and a large sandy patch. As can be seen from Fig. 4(b) and Fig. 5(b) only one side of the square contains enough features for VIO tracking; however, these features enabled repeated loop closures. The primitive estimator over estimated the forward velocity producing squares much larger than the actual trajectory. The VIO upon loss of visual tracking failed (red dash-dotted line). SM/VIO produced accurate trajectories utilizing the loop closures. The top side of the square, where VIO was operational produced consistent trajectories across all squares. The last dataset is discussed next, presenting a quantitative evaluation of the SM/VIO estimator.

c) **Quantitative Analysis:** The third dataset (lawnmower over a coral reef) produced VIO results without any loss of tracking, albeit without any loop closures. The visual input was artificially degraded (Gaussian blur with kernel size 21 and standard deviation 11 was introduced on selected images) randomly in order to generate controlled failures for



(a) (b) (c)

Fig. 5: Resulting trajectories from three datasets. Each plot presents the PE trajectory, the SVIn2 trajectory, and the proposed robust switching estimator (solid line with red the parts of VIO and blue the PE contributions, switching points are marked by green points), approximate ground truth in (a) and (c) is plotted as a solid black line: (a) Stavronikita shipwreck, lawnmower pattern. (b) Mixed sand and coral area, multiple squares. Please note that as the SVIn2 trajectory lost track it moved far away. (c) Coral reef, lawnmower pattern. This dataset has no loop closures, however, SVIn2 maintained track over the complete trajectory. The visual data were artificially degraded at random occasions to trigger the switch to PE.

the VIO. Fig. 5(c) presents such a scenario of three failures of 30 seconds each. For this study, these failures lasted for varying duration and a different number was introduced each time. More specifically, as can be seen in Table I, we introduced one, three, and five failures, in a trajectory of 314 seconds with a total length of 108.13 meters as estimated by the successful SVIn2 estimator. Each scenario was run five times, the average Root Mean Square Error (RMSE) and standard deviation are reported. One failure of 60 seconds was introduced resulting in average RMSE of 3.2 meters. Three failures for 15, 30, and 45 seconds were introduced, resulting on average around 3 meters. Finally, five failures of 20 seconds were introduced resulting on average of RMSE 4.37 meters. It is worth noting that in all cases the pure VIO estimates diverged rapidly upon loss of visual tracking; see Fig. 5(c) red dash-dotted line.

TABLE I: Quantitative analysis of robust switching estimator based on root mean squared translation error. The table shows mean and standard deviation of error over 5 runs.

dataset	length (in meters)	mean rmse (in meters)	s.d. (in meters)
reef_lmw_1_60	108.13	3.21	0.47
reef_lmw_3_15	108.13	3.21	0.61
reef_lmw_3_30	108.13	3.01	0.58
reef_lmw_3_45	108.13	3.56	0.32
reef_lmw_5_20	108.13	4.37	0.90

d) **Comparison with other VIO packages:** The shipwreck_lawnmower dataset was used to compare with well known VIO packages [9], [14]–[16]. The ground truth is obtained using COLMAP [45] which was able to track images with shipwreck in view as it does not require continuous tracking. We compared the performance of various VIO algorithms with COLMAP baseline using root mean squared average translation error (ATE) metric after se3 alignment. As can be seen in II, the proposed estimator maintained a pose estimate and exhibited the least RMSE, in contrast other algorithms deviated after losing track. OpenVINS [16] was not able to recover after losing track when the shipwreck is out of view and has a very high error. It is worth noting that all the VIO algorithms lose track when the robot approaches the side of the wreck facing blue water.

TABLE II: Performance of popular open-source VIO packages on the wreck dataset. The root mean squared ATE compared to COLMAP trajectory after se3 alignment.

VIO Algorithm	Time to first track loss (in sec)	Recovery?	RMSE (in m)
OpenVINS [16]	23.7	No	×
OKVIS [14]	23.4	Partial	5.199
VINS-Fusion [15]	23.6	Partial	53.189
SVIn2 [9]	23.4	Yes	1.438
SM/VIO	N/A	Yes	1.295

V. CONCLUSION

The presented estimator robustly tracked an AUV even when traveling through blue water or over a featureless sandy patch. The proposed system uses an Aqua2 vehicle [6] and the SVIn2 [9] VIO approach; however, any AUV with a well-understood motion model can be utilized together with any accurate VIO package. Recent deep learning based inertial odometry approaches [46]–[48] can also serve as a conservative alternative estimator. An evaluation of visual features for the underwater domain [49]–[51] will contribute additional information to the VIO health monitor.

Future use of the proposed approach will be to combine it with coral classification algorithms [52], [53] in order to extract accurate coral counts over trajectories [54] and models of the underlying reef geometry, and for mapping underwater structures [55]. We are currently working on extending the Aqua2 vehicle operations inside underwater caves. The challenging lighting conditions in conjunction with the extreme environment require the localization abilities of the vehicle to be robust even when one of the sensors fails temporarily.

REFERENCES

[1] 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 *International Symposium of Experimental Robotics (ISER)*, Tokyo, Japan, Mar. 2016.

[2] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, Nov. 2019, pp. 7221–7227.

[3] B. Woźniak and J. Dera, *Light absorption in sea water*. Springer Verlag, 2007.

[4] S. Skaff, J. Clark, and I. Rekleitis, "Estimating surface reflectance spectra for underwater color vision," in *British Machine Vision Conference (BMVC)*, Leeds, U.K., Sep. 2008, pp. 1015–1024.

[5] M. Roznere and A. Quattrini Li, "Real-time model-based image color correction for underwater robots," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[6] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton AB, Canada, Aug. 2005, pp. 1749–1754.

[7] S. Rahman, A. Quattrini Li, and I. Rekleitis, "SVIn2: A Multi-sensor Fusion-based Underwater SLAM System," *International Journal of Robotics Research*, July 2022.

[8] ——, "Sonar Visual Inertial SLAM of Underwater Structures," in *IEEE International Conference on Robotics and Automation*, Brisbane, Australia, May 2018, pp. 5190–5196.

[9] ——, "An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 1861–1868.

[10] D. Meger, F. Shkurti, D. Cortés Poza, P. Giguère, and G. Dudek, "3d trajectory synthesis and control for a legged swimming robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 2257–2264.

[11] D. Meger, J. C. G. Higuera, A. Xu, P. Giguere, and G. Dudek, "Learning legged swimming gaits from experience," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2332–2338.

[12] N. Plamondon and M. Nahon, "Trajectory tracking controller for an underwater hexapod vehicle," in *OCEANS 2008*. IEEE, 2008, pp. 1–8.

[13] P. Giguere, C. Prahacs, and G. Dudek, "Characterization and modeling of rotational responses for an oscillating foil underwater robot," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3000–3005.

[14] 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, no. 3, pp. 314–334, 2015.

[15] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," 2019.

[16] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020.

[17] 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 Trans. on Robotics*, 2021.

[18] F. Shkurti, I. Rekleitis, M. Scaccia, and G. Dudek, "State estimation of an underwater robot using visual and inertial information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, US, Sep. 2011, pp. 5054–5060.

[19] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007, pp. 225–234.

[20] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[21] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, vol. 8690, pp. 834–849.

[22] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Proc. IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 15–22.

[23] D. Ball, S. Heath, J. Wiles, G. Wyeth, P. Corke, and M. Milford, "OpenRatSLAM: an open source brain-based SLAM system," *Autonomous Robots*, vol. 34, no. 3, pp. 149–176, 2013.

[24] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052 –1067, jun. 2007.

[25] R. Mur-Artal, J. Montiel, and J. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[26] R. M.-A. Tardos and Juan, "Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, 2015.

[27] M. A. Lourakis and A. Argyros, "SBA: A Software Package for Generic Sparse Bundle Adjustment," *ACM Transactions Mathematical Software*, vol. 36, no. 1, pp. 1–30, 2009.

[28] L. Zhao, S. Huang, Y. Sun, L. Yan, and G. Dissanayake, "Parallaxba: bundle adjustment using parallax angle feature parametrization," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 493–516, 2015.

[29] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] M. Babaee and S. Negahdaripour, "3-d object modeling from occluding contours in opti-acoustic stereo images," in *MTS/IEEE OCEANS - San Diego*, Sept 2013, pp. 1–8.

[32] M. Pfingsthorn, A. Birk, and H. Bulow, "Uncertainty estimation for a 6-dof spectral registration method as basis for sonar-based underwater 3d slam," in *Proc. IEEE International Conference on Robotics and Automation*, May 2012, pp. 3049–3054.

[33] S. M. Nornes, M. Ludvigsen, Øyvind Ødegard, and A. J. Sørensen, "Underwater photogrammetric mapping of an intact standing steel wreck with ROV," in *Proc. of the International Federation of Automatic Control (IFAC)*, 2015, pp. 206–211.

[34] F. Søreide and M. E. Jasinski, "Ormen Lange: Investigation and excavation of a shipwreck in 170m depth," in *MTS/IEEE OCEANS*, 2005, pp. 2334–2338.

[35] A. Sedlazeck, K. Köser, and R. Koch, "3D reconstruction based on underwater video from ROV Kiel 6000 considering underwater imaging conditions," in *MTS/IEEE OCEANS*, 2009, pp. 1–10.

[36] C. Beall, B. Lawrence, V. Ila, and F. Dellaert, "3d reconstruction of underwater structures," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 4418–4423.

[37] B. Nisar, P. Foehn, D. Falanga, and D. Scaramuzza, "Vimo: Simultaneous visual inertial model-based odometry and force estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2785–2792, 2019.

[38] Y. Liu, "Switching observer design for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 42, no. 12, pp. 1699–1703, 1997.

[39] M. Suzuki, H. Fujimoto, and Y. Hori, "The simultaneous estimation method of terrain parameter and vehicle dynamics variables for agricultural vehicle," in *2019 IEEE International Conference on Mechatronics (ICM)*, vol. 1, 2019, pp. 596–601.

[40] T. Manderson, J. C. Gamboa, S. Wapnick, J.-F. Tremblay, F. Shkurti, D. Meger, and G. Dudek, "Vision-based goal-conditioned policies for underwater navigation in the presence of obstacles," in *Proceedings of Robotics: Science and Systems*, July 2020.

[41] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[42] B. Joshi, M. Xanthidis, S. Rahman, and I. Rekleitis, "High definition, inexpensive, underwater mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, 2022, p. accepted.

[43] 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.
[44] S. Agarwal, K. Mierle, and Others, "Ceres Solver," http://ceres-solver.org, 2015.
[45] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.
[46] S. Herath, D. Caruso, C. Liu, Y. Chen, and Y. Furukawa, "Neural inertial localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 6604–6613.
[47] X. Cao, C. Zhou, D. Zeng, and Y. Wang, "Rio: Rotation-equivariance supervised learning of robust inertial odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 6614–6623.
[48] S. Herath, H. Yan, and Y. Furukawa, "Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3146–3152.
[49] 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, "Vision-based shipwreck mapping: on evaluating features quality and open source state estimation packages," in *MTS/IEEE OCEANS - Monterrey*, Sep. 2016, pp. 1–10.
[50] ——, "On understanding the challenges in vision-based shipwreck mapping," in *ICRA 2016 Workshop on Marine Robot Localization and Navigation*, Stockholm, Sweden, May 2016, pp. 1–3.
[51] F. Shkurti, I. Rekleitis, and G. Dudek, "Feature tracking evaluation for pose estimation in underwater environments," in *Canadian Conference on Computer and Robot Vision (CRV)*, St. John, NF Canada, 2011, pp. 160–167.
[52] M. Modasshir, A. Quattrini Li, and I. Rekleitis, "Mdnet: Multi-patch dense network for coral classification," in *MTS/IEEE OCEANS - Charleston*. IEEE, 2018, pp. 1–6.
[53] O. Beijbom *et al.*, "Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation," *PloS one*, vol. 10, no. 7, 2015.
[54] M. Modasshir, S. Rahman, and I. Rekleitis, "Autonomous 3D Semantic Mapping of Coral Reefs," in *12th Conference on Field and Service Robotics (FSR)*, Tokyo, Japan, Aug. 2019, pp. 365–379.
[55] M. Xanthidis, B. Joshi, M. Roznere, W. Wang, N. Burgdorfer, A. Quattrini Li, P. Mordohai, S. Nelakuditi, and I. Rekleitis, "Mapping of underwater structures by a team of autonomous underwater vehicles," in *International Symposium of Robotics Research*, 2022.