Hybrid Visual Inertial Odometry for Robust Underwater Estimation

Bharat Joshi, Chanaka Bandara, Ioannis Poulakakis, Herbert G. Tanner, and Ioannis Rekleitis

***Abstract*— Vision-based state estimation is challenging in underwater environments due to color attenuation, low visibility and floating particulates. All visual-inertial estimators are prone to failure due to degradation in image quality. However, underwater robots are required to keep track of their pose during field deployments. We propose robust estimator fusing the robot's dynamic and kinematic model with proprioceptive sensors to propagate the pose whenever visual-inertial odometry (VIO) fails. To detect the VIO failures, health tracking is used, which enables switching between pose estimates from VIO and a kinematic estimator. Loop closure implemented on weighted posegraph for global trajectory optimization. Experimental results from an Aqua2 Autonomous Underwater Vehicle field deployments demonstrates the robustness of our approach over different underwater environments such as over shipwrecks and coral reefs. The proposed hybrid approach is robust to VIO failures producing consistent trajectories even in harsh conditions.**

I. Introduction

The blue economy, which encompasses all economic activity around the coasts and oceans, contributes more than $373 billion to the US GDP and supports more than two million jobs [1]. While seventy one percent of the Earth is covered by water, only a tiny portion of the underwater environment is adequately mapped and explored [2]. The role of robotics and automation technology is critical in moving this needle. One of the most challenging aspects of autonomous underwater vehicle (AUV) deployment beneath the surface is still cost, with a significant percentage associated to the robotic platform and its sensor payload. And while platform cost is gradually decreasing with advances in manufacturing and electronics, and the ability to integrate commercial off-the-shelf (COTS) components into effective perception, decision-making, and action cyberphysical architectures, autonomous underwater (and thus GPS-denied) navigation, in environments like those of Fig. 1 where ultra-short baseline (USBL) technologies cannot easily be deployed, remains a critical challenge.

Vision-based state estimation has gained traction in recent years as cameras are lightweight, power efficient and provide semantic information easily understandable to humans. Recent work has demonstrated that existing open source packages for visual inertial odometry (VIO) are prone to failure in an underwater environment [3], [4]. Exacerbating the challenges of GPS-denied state estimation and localization, underwater environments often present instances of rapid changes in visibility, lighting conditions and contrast, loss of color, blurring, and "snow effects." In addition, one cannot assume uniform availability of optical features and landmarks [5], [6] over the whole area of deployment that a vision system can exploit. In alternative approaches, when VIO fails, there is no recourse for state estimation functionality recovery.

Fig. 1: Autonomous Underwater Vehicle over the Stavronikita shipwreck, Barbados.

Unable to produce continuous estimate of AUV's pose at best hampers control strategies that rely on robot's state to produce trajectories and at worse can results in vehicle loss. As such, tracking robot's pose at all times is very important albeit with diminished accuracy. As such, in our recent work we proposed using a primitive estimator to track an AUV using proprioceptive sensors and a simple vehicle kinematic model [7]. This paper makes a contribution in the area of VIO-based underwater navigation by further improving the accuracy and the robustness with focus on better modeling of the AUV kinematics. This switching estimator is inherently more robust than alternative solutions because it can switch between visual-inertial estimation and model-based state propagation, whenever the former fails to acquire enough features from the environment to prevent catastrophic estimate divergence.

Compared to earlier work [7] the proposed estimator here incorporates a more advanced and accurate dynamical model for the underwater platform, thus boosting the accuracy of the model-based estimate which is used by the navigation algorithm until the VIO estimator comes back online; which we term here as *kinematic estimator* (KE). In addition, the kinematic estimator uses recent gyroscope bias estimates from VIO for more accurate orientation estimation. In the

Bandara, Poulakakis and Tanner are with the Center for Autonomous and Robotic Systems at the University of Delaware. Joshi and Rekleitis are with the Department of Computer Science and Engineering at the University of South Carolina. This work is supported in part by the National Science Foundation (NSF 1943205, 2024741). The authors would like to thank Halcyon Dive Systems for their support with equipment.

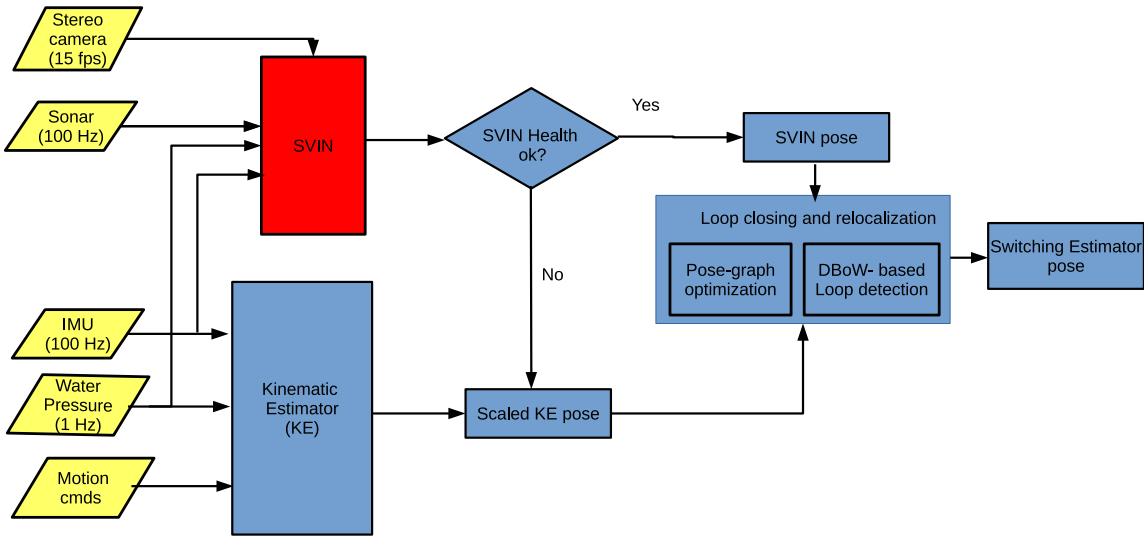


Fig. 2: Overview of the hybrid switching estimator.

event of loop closure, the weighted posegraph optimization process gives less priority to the kinematic estimator edges compared to VIO produced edges. The ability to switch online between the two estimator modalities, with awareness of the evolving accuracy performance of each one, provides significant robustness to changes in the visual underwater environment. This allows the AUV to carryout the underlying tasks of coverage and exploration [8] while maintaining it's ability to localize accurately when visiting same area again. We performed two experiments where an Aqua2 AUV [9], [10] performs lawnmower pattern over a shipwreck and a square pattern over a coral reef. While most VIO algorithms [11]–[13], either diverge or fail drastically, our hybrid inference system is able to keep steady pose estimate with lower error.

II. Related Work

Several open source packages solving the problem of estimating visual and visual/inertial odometry (VO and VIO) have been published [14]–[25]. The VO/VIO methods can be broadly divided into *direct* methods [26], [27] and *indirect* (features-based) [11], [23], [28]–[30] based on how the information from the images is incorporated into the proposed framework. Visual SLAM methods can be classified depending on the backend into non-linear filtering [23], [31]–[33] and least squares optimization [11], [27]–[29], [34] approaches. Comparing the performance of these packages over a variety of datasets demonstrated several challenges; see Quattrini Li *et al*. [3]. More recently, Joshi *et al*. [4] examined the performance of different VIO packages in the underwater domain. The above research showed that many approaches are strictly offline [35], require special motions [15], or are computationally constrained to a small number of images [36], [37]. In addition, intermittent failures appeared, where the randomness of the RANSAC technique [38] was the cause. SVIN by Rahman *et al*. [13], an underwater VIO showed better robustness and accuracy; unfortunately, SVIN did diverged when the health of the vision estimator failed.

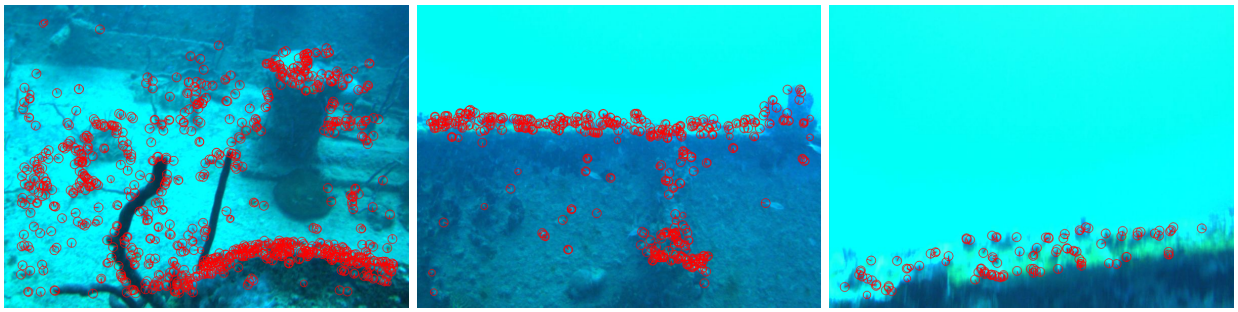
III. Proposed System

The proposed computational architecture for robust underwater state estimation intends to combine a dynamical model-predictive estimator with a VIO [13] module (Fig. 2).

The idea is to rely on VIO as long as the vehicle operates in feature rich environments where the associated estimator is expected to yield accurate estimates, and intermittently switch to a dynamical model-based estimator to propagate the state estimate forward until the VIO algorithm observes images with sufficient features to yield a better estimate of robot's pose. A supervisory logic will switch between the two estimators using an estimate of their expected state accuracy.

The VIO module fuses velocity and acceleration measurements from an inertial measurement unit (IMU) at 100 Hz, depth measurements from a water pressure sensor, and stereo camera images at 15 frames per second (fps). This sensor fusion algorithm leverages earlier work and a codebase that has been introduced under the name SVIN [13], [24], [25]. In our earlier realizations, the model-based predictive estimator has been utilizing a simple *kinematic* model of the form ${}^{\mathcal{W}}\boldsymbol{p}_{\mathcal{I}}(t+1) = {}^{\mathcal{W}}\boldsymbol{p}_{\mathcal{I}}(t) + {}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}}(t)\left[\,\bar{v}_x(t)\;0\;\bar{v}_z(t)\,\right]^{\intercal}\Delta t$, where ${}^{W}\boldsymbol{p}_{t+1}$ represents the position of the robot's frame $\mathcal{I}$ in the fixed world frame $\mathcal{W}$ at time step $t+1$, ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}}$ denotes the rotation matrix from frame $\mathcal{I}$ to frame $\mathcal{W}$, $\bar{v}_x$ and $\bar{v}_z$ are the vehicle commanded speed components along the surge and heave directions, and Δt is the length of the time interval between successive updates.

In this paper, the simple kinematic model is upgraded by utilizing a more complete *kinematic* model for motion prediction, that takes into account the full kinematic state $(\boldsymbol{p}_{\mathcal{I}}, \boldsymbol{\eta}_{\mathcal{I}})$ of the system, where $\boldsymbol{\eta}_{\mathcal{I}} = (\phi, \theta, \psi)$ denotes the vector of the robot's roll, pitch, and yaw angles.



(a) (b) (c)

Fig. 3: (a) representative frame when the AUV has sufficient visual features. (b) Example of a case of barely adequate features to produce estimates from visual odometry. (c) The AUV now has too few features and VIO will fail.

A. VIO Overview

An extension to visual-inertial SLAM was proposed by Rahman *et al*. [13] to incorporate water pressure and sonar measurements. This Sonar Visual Inertial Navigation (SVIN) system is found to be effective in underwater environments and used as our visual inertial estimator.

The visual frontend detects and tracks BRISK [39] features across multiple frames. An IMU preintegration technique similar to [40] is employed to propagate IMU measurements between frames. The backend estimates the robot's pose by jointly minimizing the reprojection error from feature tracking, the IMU error from propagation, the water pressure depth and the sonar error. A sliding window of keyframes along with associated error terms optimized for real-time processing. In addition, the depth error is formulated as the difference in AUV's along the z-direction and the water depth measurement from the pressure sensor. To calculate the sonar error, we accumulate the 3D points in the current optimization window nearby the sonar measurement and calculate the error as the difference of the centroid of the 3D point cluster and the sonar range measurement.

A separate *loop-closure* and *relocalization* module uses the output of VIO and maintains a posegraph of relative transformations between keyframes. We only optimize the posegraph with 4 degrees of freedom (DoF) involving position and yaw as roll and pitch are observable in the VIO system. The loop-closure module implemented using the BRIEF [41] vocabulary with a bag-of-words place recognition module [42]. In the event of a loop closure, an odometry edge is added between the current and the candidate keyframe when they have enough descriptor matches and pass the PnP-RANSAC based geometric verification. The *loop-closure* module is extended to accept odometry information from the kinematic estimator with lower weight assigned to odometry edge from KE compared to their visual counterparts. These KE frames are not used for loop closing as they lack visual information.

B. Health Tracking

In our previous work [3], [4], we compared the performance of various visual SLAM algorithms in underwater domain. These studies found that visual failures are frequent especially when there is no visible structure in front of the camera. For the continuous operation of the robot, it is important to detect divergence in pose estimation. In Kalman-filter based VIO, a set limit on covariance estimate can be used as health indicator. However, estimating the pose covariance is inefficient in optimization-based methods as other states need to be marginalized. Moreover, the covariance in optimization packages such as ceres [43] is estimated using jacobians [1] and does not follow the SLAM intuition which is often time consuming. Thus, we employ a health monitoring mechanism tailored to vision-based state estimation. The health monitoring considers multiple criteria hierarchically with most important criterion considered first. We employ the following criteria hierarchically and health status of the VIO is updated based on:

- **Keyframe detection:** We wait for *kf_wait_time* between two keyframe and if there is no new keyframe during this time, we assume VIO frontend has failed. An exception to this criterion is when the robot is stationary which can be handled using zero velocity updates. We use *kf_wait_time* of around ≈ 2 secs.
- **Number of triangulated points:** The number of 3D keypoints triangulated in the current keyframe should be more than a set threshold, *min_kf_points*. We used *min_kf_points* ranging between 10 and 20 depending on the dataset.
- **Spatial distribution of features:** It is desirable that the features are detected and tracked uniformly across all image regions. Thus, we keep track of feature detections per quadrant in the current keyframe and check if is less than a certain threshold, *min_kps_per_quadrant*. However, there are situations where large number of good features are detected in a small region in an images; for example Fig. 3b, 3c where the bottom half contains all the features. To account for such instances, we only apply quadrant criterion if the total number of feature detections is less than $10 \times$ *min_kps_per_quadrant*.
- **Feature track length:** In an ideal scenario, we want the features to be tracked across multiple keyframes. So, we calculate the ratio of new keypoints to the total keypoints and set the threshold at 0.75. These

[1] `http://ceres-solver.org/nnls_covariance.html`

new keypoints are those not tracked across multiple keyframes.

- **Feature detector response:** Finally, we compute the average corner response of the feature detections in an image. We set a higher threshold of ratio of keypoints less than the average response in an image to 0.85. The higher threshold is used as this is considered the least important criterion.

These threshold values are chosen empirically based on the visual SLAM literature. For instance, at least 6 feature matches are required between two images to calculate the relative transformation based on epipolar geometry. Also, for accurate pose estimation a feature is required to be tracked across multiple keyframes. We found that slight changes in these parameters did not have any significant effect on the performance of hybrid switching estimator. Hence, these parameters are provided as reference and we suggest practitioners to slightly tune these parameters depending on the target environment.

C. Kinematic Estimator

Figure 4 depicts the frames used for the motion analysis of the robot and the rotations involved in the kinematic representation.

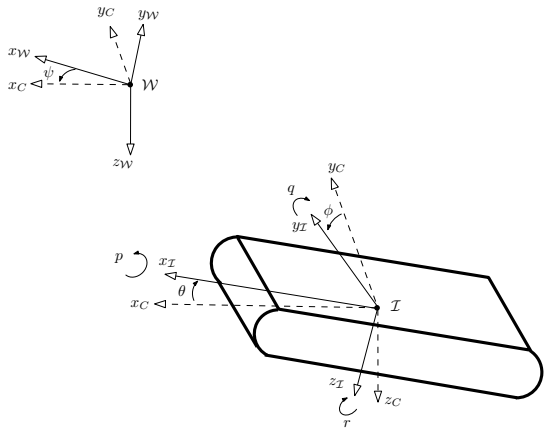


Fig. 4: Frames and rotations associated with the robot kinematics.

Let $w = [p, q, r]$ be the angular velocity measured using gyroscope expressed in the body frame. Whenever the VIO is working, it able able to estimate accelerometer and gyroscope biases correctly. Thus, we use the recent gyroscope bias estimates b_g from VIO to further correct angular velocity estimates as

$$w = w - b_g$$

The kinematic state of the robot is being updated according to the following equations:

$$
\begin{aligned}
{}^{\mathcal{W}}\boldsymbol{p}_{\mathcal{I}}(t+1) &= {}^{\mathcal{W}}\boldsymbol{p}_{\mathcal{I}}(t) + {}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}}(t) \begin{bmatrix} v_x(t) \\ 0 \\ v_z(t) \end{bmatrix} \Delta t \\
\boldsymbol{\eta}_{\mathcal{I}}(t+1) &= \boldsymbol{\eta}_{\mathcal{I}}(t) \\
&\quad + \underbrace{\begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix}}_{S(t)} \begin{bmatrix} p(t) \\ q(t) \\ r(t) \end{bmatrix} \Delta t
\end{aligned}
$$

where ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{I}}(t)$ represents the rotation matrix corresponding to euler angles $\boldsymbol{\eta}_{\mathcal{I}}$.

D. Integration of VIO and Kinematic Estimator

The pose estimates from the VIO and the kinematic estimator are combined in a switching framework proposed in [7]. The VIO framework described in Rahman *et al*. [13] is augmented to use pose estimates from multiple odometry sources into a single posegraph with different weights. The relative pose constraints in posegraph are classified as visual or kinematic; with double weight assigned to relative pose constraints pertaining to VIO. Whenever a loop closure is performed, the relative pose between visual keyframes is retained whereas those between kinematic estimator is relaxed owning to their lower accuracy.

The switching estimator is designed to track either the VIO or the kinematic estimator trajectory locally based on the health tracking status. Let ${}_W\mathbf{T}_{vio}$ and ${}_W\mathbf{T}_{ke}$ be the pose of AUV represented in the world coordinate frame as 4×4 transformation matrices. The switching estimator locally resembles ${}_W\mathbf{T}_{vio}$ when the VIO is working properly. Whenever, the health tracking reports VIO failure, it resembles ${}_W\mathbf{T}_{ke}$. Initially, when VIO starts tracking, the pose of the robust switching estimator ${}_W\mathbf{T}_{ro}$ is equivalent to ${}_W\mathbf{T}_{vio}$. When the VIO health tracker indicates imminent failure, switching to the kinematic estimator occurs. We keep track of the robust pose and the kinematic estimator at switching time *s* as ${}_W\mathbf{T}^s_{ro}$ and ${}_W\mathbf{T}^s_{ke}$ respectively. Now, we calculate to local displacement of the kinematic estimator with respect to the kinematic estimator pose at switching time as ${}_W\mathbf{T}^{s^{-1}}_{ke}$ ${}_W\mathbf{T}_{ke}$. This local displacement estimates robot motion using the kinematic estimator from the time switching occurs and is used to propagate the robust pose using Eq. (1); here $\cdot$ represents matrix multiplication.

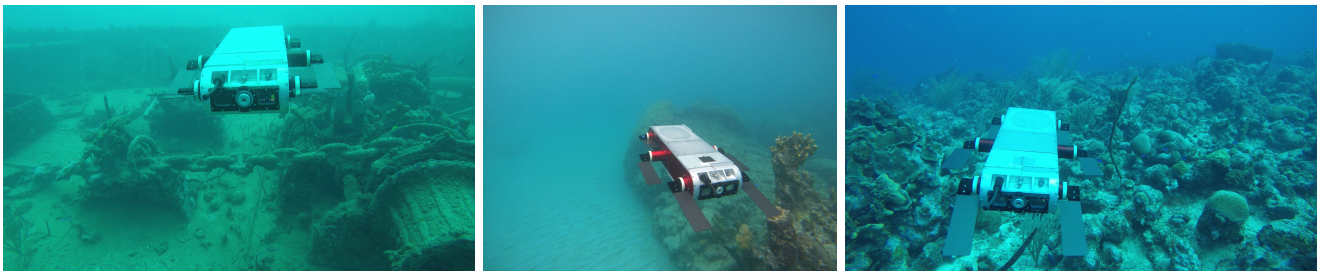
$$ {}_W\mathbf{T}_{ro} \coloneqq {}_W\mathbf{T}^s_{ro} \cdot {}_W\mathbf{T}^{s^{-1}}_{ke} \cdot {}_W\mathbf{T}_{ke} \tag{1} $$

This makes sure that the robust estimator tracks the kinematic estimator propagation locally. Please note that ${}_W\mathbf{T}^s_{ro} \cdot {}_W\mathbf{T}^{s^{-1}}_{ke}$ remains constant until the next switching occurs.

Extending the same analogy, the switching to VIO occurs when health tracking shows that the VIO has recovered. When switching back to the VIO, the robust estimator tracks the local displacement from the VIO pose at switching time ${}_W\mathbf{T}^s_{vio}$ as shown in Eq. (2) which remains constant until the next switching to the kinematic estimator happens.

$$ {}_W\mathbf{T}_{ro} \coloneqq {}_W\mathbf{T}^s_{ro} \cdot {}_W\mathbf{T}^{s^{-1}}_{vio} \cdot {}_W\mathbf{T}_{vio} \tag{2} $$

The accuracy and robustness of the proposed estimator comes from it's capability to track the robot's motion using either the VIO or the kinematic estimator (KE) depending on the health tracking status. Whenever the VIO recovers from failures, it is always the preferred estimator owing to it's better accuracy compared to the KE. As SVIn2 is able to propagate the state using IMU measurements whenever the visual frontend tracking fails for a small amount of time



(a) (b) (c)

Fig. 5: 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.

upto $\approx$ 2 secs VIO is still operational. As such, we do not want frequent switching between VIO and KE. To reduce frequent switching between VIO and KE, we wait for a certain number of successive failures before switching from VIO to KE and the same number of successive good tracking attempts before switching from KE to VIO.

The keyframes from KE are added in the posegraph differently than the regular VIO keyframes as they only contain relative pose constraints without image information. In addition, these KE keyframes can not be used for loop closure as they do not contain information required for loop closure such as feature detections and 3D triangulated keypoints used for PnP RANSAC based geometric verification. Also, the relative pose error between consecutive keyframes is implemented as a 4DoF posegraph optimization owing to the gravity direction observability from IMU. To account for the lower accuracy of KE keyframes, the weights between successive keyframes is multiplied by 0.5 such that the section of final trajectory coming from KE is deformed more after loop closure.

IV. Datasets

We performed multiple experiments using an Aqua2 AUV performing motion patterns over a variety of challenging environments including lawnmower over a shipwreck see Fig. 5a; multiple squares over sand and coral heads, see Fig. 5b; and lawnmower over a coral reef, see Fig. 5c. The Aqua2 AUV performs predefined trajectory patterns while using odometry information from the primitive estimator (PE). We conducted the field trials on the following datasets:

A. Shipwreck Lawnmower

The Aqua2 AUV is performing a lawnmower pattern over the Stavronikita shipwreck, Barbados. The Aqua2 AUV maneuvers over the side of shipwreck facing open water, see Fig. 3b and Fig. 3c. Since the visual frontend can not detect and track features when facing open water, the VIO is not able to track the AUV's pose and diverges. The ground truth is obtained using COLMAP [35] using images registered that have view of the shipwreck. The scale is enforced using known stereo-rig constraints.

B. Coral Square

The Aqua2 AUV performs square patterns over sandy coral reef area in Barbados; see Fig. 5b. During the operation the AUV veers into areas only seeing either sandy patches or open water; thus the VIO diverges. We do not have ground truth for this dataset and only use it for qualitative evaluation.

C. Coral Lawnmower

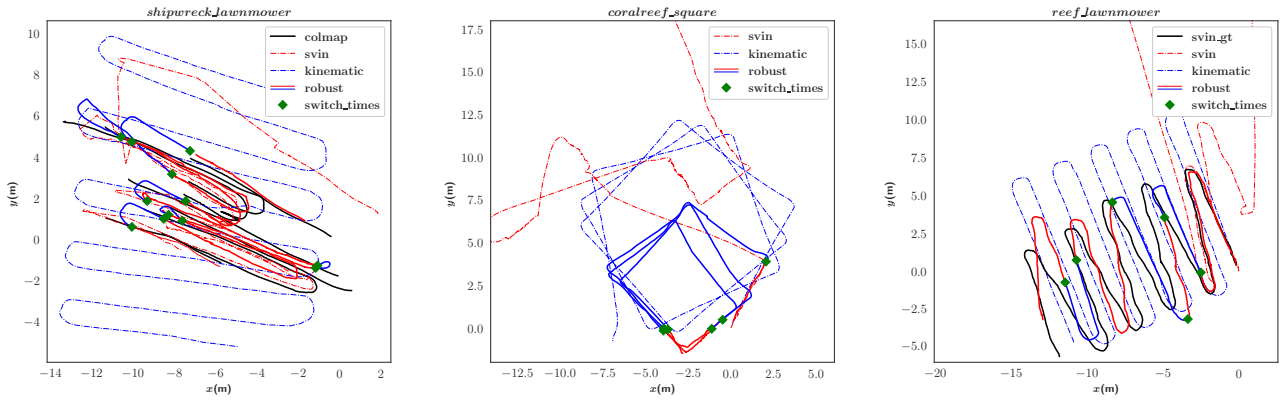
In this dataset, the Aqua2 AUV performs a lawnmower pattern over a coral reef in Barbados; see Fig. 5c. The VIO is able to track the robot's pose over the whole time of operation. This dataset is artificially degraded by applying Gaussian blur on the images for 30 seconds at 3 different times. Thus, we induced VIO failures in sections of the trajectory by using Gaussian blue. For this dataset, we use the VIO trajectory as ground truth. It is worth noting that COLMAP was not able to register all the images due to fast rotation, and thus it was not used as ground-truth.

V. Experimental Results

We tested the proposed hybrid robust estimator in the above described three different datasets and the trajectories estimated using kinematic estimator (KE), SVIN and robust hybrid estimator are shown in Fig. 6. The kinematic estimator tracks the requested pattern almost perfectly as a similar primitive estimator is used to generate control strategies (shown with blue dash-dotted line). The SVIN VIO losses track and diverges in all the trajectories and the VIO trajectory is plotted as dash-dotted red line. The proposed hybrid estimator is able to keep track of AUV's pose during the whole operation and the resulting trajectory is shown as a solid blue and red line with green diamonds marking the places where switching occurred.

A. Shipwreck lawnmower

The shipwreck lawnmower dataset is very challenging for any VIO method as AUV performs lawnmower patter over the Stavronikita shipwreck. Initially the robot starts from the middle of the shipwreck with feature rich areas Fig. 3a, then the AUV slowly maneuvers towards side of the shipwreck with the number of detected features decreasing gradually, see Fig. 3b. Eventually, the AUV reaches the side of the shipwreck facing mostly open water with very few features visible; see Fig. 3c. As the number of detected features are greatly diminished, the VIO loses track and deviates from



(a) (b) (c)

Fig. 6: Trajectories of the evaluated algorithm on lawnmower pattern over shipwreck; squares over a mixture of sand and corals; and lawnmower pattern over the coral reef utilizing an Aqua2 AUV. COLMAP (black) obtained using global bundle adjustment is used as ground truth up to scale for the first dataset. SVIn2 is marked with dashed red line; KE is marked in dashed blue line; and the proposed approach is represented by a solid red and blue line with green diamonds indicating the switching spots.

TABLE I: Performance of popular open-source VIO packages and SM/VIO on the shipwreck dataset taken from [7]. The root mean squared ATE compared to COLMAP trajectory after SE(3) alignment.

VIO Algorithm	Time to first track loss (in sec)	Recovery?	RMSE (in m)
OpenVINS [23]	23.7	No	×
OKVIS [11]	23.4	Partial	5.199
VINS-Fusion [12]	23.6	Partial	53.189
SVIN [25]	23.4	Yes	1.438
SM/VIO [7]	N/A	Yes	1.295
Ours	N/A	Yes	0.878

the true trajectory. The VIO is able to recover and decrease the error after loop closure.

The shipwreck lawnmower dataset contains ground-truth obtained using COLMAP [35] which was able to register images when the robot was moving over the shipwreck and does not require continuous tracking. Thus, this dataset is used to compare the performance of the hybrid switching estimator with other VIO algorithms [11]–[13], [23] and our previous work SM/VIO [7]. Absolute trajectory error (ATE) metric is used to compare the trajectories with COLMAP after SE(3) alignment. As seen in Table I, the hybrid switching estimator is able to maintain consistent pose over the whole trajectory with the least root mean squared (RMSE) error. All other VIO algorithms lose track when the robot reaches the side of shipwreck facing open water. OpenVINS [23] did not recover after losing track the first time and diverges.

B. Coral Square

In the coral reef dataset, the AUV performs three squares over coral heads next to large sandy patches. One section of the square had good features as seen in Fig. 5b and VIO only tracks this section of the square. However, since this section is visited multiple times, there were frequent loop closures as evident in Fig. 6b. The VIO quickly diverges when moving over the sandy area and hybrid switching estimator is able to keep track over the whole duration using pose estimates from kinematic estimator. Since only very small section contains good quality images, this dataset is only used for qualitative evaluation.

C. Coral Lawnmower

In this dataset, the Aqua2 AUV performs lawnmower patterns over the coral reef and VIO does not loose track during the operation. The images in this dataset are artificially degraded using Gaussian blur with a kernel size of 21 and standard deviation 11 on three sections for 30 seconds each. Thus, we induced controlled failures to test the robustness of our approach. COLMAP [35] was not able to register all images due to fast rotation around the corners producing multiple disconnected trajectories. Hence, the VIO trajectories on original dataset was used as ground-truth; see Fig. 6c. It should be noted that pure VIO diverges rapidly upon degradation of images as seen in Fig. 6c red dash-dotted trajectory. Moreover, using the KE we were able to improve the switching estimator producing an RMSE error of 1.50m compared to 3.01m in our previous work [7].

VI. Conclusion

A supervisory logic that enables the AUV's state estimation system to monitor the accuracy of underwater VIO and switch it off by temporarily replacing state estimation with a model-based dead-reckoning system until sufficient visual features are re-acquired, has shown significant promise in making state estimation below the surface more robust and suitable for advanced motion control feedback. This paper advances the state of the art further in this direction by providing additional fidelity to the fail-safe dead-reckoning function of the estimator, through the utilization of a nonlinear rigid-body vehicle dynamical model.

REFERENCES

[1] National Oceanic and Atmospheric Administration, "NOAA blue economy strategic plan 2021–2025," U.S. Department of Commerce, Tech. Rep., January 19, 2021.

[2] U. W. S. School, "How much water is there on, in, and above the Earth?" https://water.usgs.gov/edu/earthhowmuch.html, accessed: 2017-06-20, 2016.

[3] 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.

[4] 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. [Online]. Available: https://arxiv.org/abs/1904.02215

[5] 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.

[6] 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)*, 2011, pp. 160–167.

[7] B. Joshi, H. Damron, S. Rahman, and I. Rekleitis, "SM/VIO: Robust Underwater State Estimation Switching Between Model-based and Visual Inertial Odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[8] B. Joshi, M. Xanthidis, M. Roznere, N. J. Burgdorfer, P. Mordohai, A. Q. Li, and I. Rekleitis, "Underwater exploration and mapping," in *IEEE OES AUV Symposium*, Singapore, Sept. 2022, pp. 1–7.

[9] 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)*, Aug. 2005, pp. 1749–1754.

[10] G. Dudek, P. Giguere, C. Prahacs, S. Saunderson, J. Sattar, L.-a. Torres-Mendez, M. Jenkin, A. German, A. Hogue, A. Ripsman, J. Zacher, E. Milios, H. Liu, P. Zhang, M. Buehler, and C. Georgiades, "AQUA: An Amphibious Autonomous Robot," *Computer*, vol. 40, no. 1, pp. 46–53, 2007.

[11] 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.

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

[13] S. Rahman, A. Quattrini Li, and I. Rekleitis, "SVIn2: A Multi-sensor Fusion-based Underwater SLAM System," *International Journal of Robotics Research*, vol. 41, no. 11-12, pp. 1022–1042, July 2022.

[14] 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.

[15] 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)*, Nov. 2007, pp. 225–234.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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*, 2020.

[24] S. Rahman, A. Quattrini Li, and I. Rekleitis, "Sonar Visual Inertial SLAM of Underwater Structures," in *IEEE International Conference on Robotics and Automation*, May 2018, pp. 5190–5196.

[25] ——, "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.

[26] J. Engel, J. Stueckler, and D. Cremers, "Large-Scale Direct SLAM with Stereo Cameras," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 09 2015.

[27] L. von Stumberg, V. Usenko, and D. Cremers, "Direct Sparse Visual-Inertial Odometry using Dynamic Marginalization," in *International Conference on Robotics and Automation (ICRA)*, 05 2018.

[28] C. Campos, R. Elvira, J. J. Gómez, J. M. 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.

[29] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[30] S. Rahman, A. Q. 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.

[31] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015.

[32] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.

[33] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3565–3572.

[34] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.

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

[36] 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.

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

[38] 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.

[39] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *International Conference on Computer Vision (ICCV)*, 2011, pp. 2548–2555.

[40] 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, 2017.

[41] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision – ECCV 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792.

[42] 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.

[43] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver