High Definition, Inexpensive, Underwater Mapping

Bharat Joshi, Marios Xanthidis, Sharmin Rahman, Ioannis Rekleitis

Abstract— In this paper we present a complete framework for Underwater SLAM utilizing a single inexpensive sensor. Over the recent years, imaging technology of action cameras is producing stunning results even under the challenging conditions of the underwater domain. The GoPro 9 camera provides high definition video in synchronization with an Inertial Measurement Unit (IMU) data stream encoded in a single mp4 file. The visual inertial SLAM framework is augmented to adjust the map after each loop closure. Data collected at an artificial wreck of the coast of South Carolina and in caverns and caves in Florida demonstrate the robustness of the proposed approach in a variety of conditions.

I. INTRODUCTION

The underwater domain presents a special allure since the early days of exploration [1]; coral reefs, shipwrecks, and underwater caves all present unique views like nothing most people see above water. Underwater exploration using acoustic sensors is well studied, however, the resulting representations convey only limited information; in contrast vision based mapping presents the most familiar representations [2]–[5]. Underwater is a very challenging environment for cameras. The visibility is limited, sometimes objects after a few meters disappear; color attenuation, colors disappear with depth starting with red [6], [7]; floating particulates generate blurriness; there is varying illumination resulting from caustic patterns due to waves up to complete lack of ambient light inside caves; and the reduced number of features makes localization challenging.

Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs) range in cost from a few thousand to hundreds of thousand of dollars. Furthermore, camera technologies for these vehicles, unless at the higher end of the spectrum provide images not of the highest quality; one major challenge is the light has to pass from water to the AUVs window, through air, then the lens of the camera generating additional distortions. In the recent years so-called action cameras and in particular the GoPro cameras have produced exceptional imagery for a fraction of the cost. The improvements in image quality though, were limited by the single camera view which made estimation of scale

The authors are with the Computer Science and Engineering Department, University of South Carolina, Columbia, SC, USA, 29208, {bjoshi,mariosx,srahman}@email.sc.edu, yiannisr@cse.sc.edu.

Fig. 1: Collecting data in an underwater cavern, Ginnie Spring, FL, USA. GoPro 9 camera is attached to the stereo-rig [15], lighting from two Keldan lights [16].

near impossible. As shown in Joshi *et al*. [8] and Quattrini Li *et al*. [9] monocular vision without inertial data has very low accuracy. From the GoPro 5 black, the video contains embedded inertial data at a rate of 200 Hz, without synchronization information. Starting from GoPro 8, the video contains inertial information along with necessary timing information for camera IMU synchronization thus making underwater state estimation feasible. In this paper we tested some of the most promising open-source Visual Inertial SLAM packages [10]–[14], in a variety of environments with very accurate results. Furthermore, the SVIn2 framework [10] is augmented with updating the 3D pose of the detected visual features after loop closure producing a consistent global map. The code is publicly available[1].

Experiments conducted over an artificial reef (refuelling barge wreck) off the coast of South Carolina; inside the Devil System cave, FL; at the spring waters (open water) of Troy Springs State Park, FL; and in the cavern of Ginnie Springs, FL. In particular inside the Ginnie Springs cavern, five fiducial markers were placed in different locations in order to estimate the accuracy of the estimated trajectory. These datasets are made publicly available together with calibration data and basic scripts for evaluating ground truth[2].

II. RELATED WORK

A stereo GoPro setup was used to map underwater caves [17], however, this technology is no longer available, and it is only recently that the IMU of the GoPro 9 allows scale-accurate results from a single camera. GoPro cameras have been studied in underwater settings [18], [19], and used, due to the high quality imagery in a variety of underwater tasks, for coral monitoring [19]–[22], underwater archaeology [23], and seafloor reconstruction [24], [25].

[1] https://github.com/AutonomousFieldRoboticsLab/gopro_ros
[2] https://afrl.cse.sc.edu/afrl/resources/datasets/

Wreck mapping has been studied using a variety of techniques all around the world. Photogrammetry of manually obtained images resulted in mosaics in Demesticha *et al.* [3], or from an ROV, see Nornes *et al.* [26]. While the Arrows EU project provides an overview of robotic technology used [27]. Menna *et al.* [28] provide a comprehensive review of techniques used. Mapping projects extend from Italy [29], Spain [30], Canada [31], Qatar [32], up to the arctic [33]. With the most famous wreck explorations of the Titanic [2] and the Antikythera [34] shipwrecks.

Coral reef mapping also utilizes vision. By creating specialized sensors [15], [35] or utilizing UAVs [36] there is a need for Underwater SLAM [37]. Due to the deteriorating health of the coral reefs, it is important to document the state of the different reefs and to measure the rate of deterioration. Of particular interest is to identify resilient species to assist re-population efforts.

There are few datasets from underwater experiments [8], [9], [38], however, obtaining ground truth is extremely challenging. The Aqualoc dataset [38] used the trajectory estimated using global optimization package Colmap [14] as ground truth. This work will contribute a novel collection of datasets, and in select cases a set of permanent landmarks to act as ground truth. This dataset contains high definition/resolution images and inertial data at 200 Hz. The image quality is far superior than existing datasets.

Mapping underwater caves is extremely challenging due to the total lack of ambient light. Wakulla Springs cave is one of the most well known and efforts to map it include the Wakulla 2 project [39], [40] utilizing mainly acoustic sensors, as was mapping a cenote [41]. Nocerino *et al.* [42] proposed the use of multiple cameras on a ROV for mapping caves, then use it to map caves in Sicily [43]. Malios *et al.* [44] proposed also a SLAM framework for confined spaces. The works of Rahman *et al.* [10], [45] has demonstrated accurate results over long trajectories in a variety of settings. On this work we utilize a subset of this work SVIn2 utilizing the initialization and loop closure extension over OKVIS [46]. Furthermore, a framework for denser reconstructions used the plethora of shadows in the cave environment [47]. We have augmented this framework to produce a consistent map, by updating the triangulated features after every loop closure.

III. PROPOSED APPROACH

A. Sensor Setup

The GoPro 9 consists of a color camera, an IMU, and a GPS. GPS does not work underwater; thus it was not used in this work. However, GPS information can be fused with Visual Inertial Navigation Systems (VINS) during above-water operations. For calibrating the camera intrinsic parameters and the extrinsic parameters of the sensor setup, we use a grid of AprilTags [48]. Table I presents the available sensors.

The GoPro 9 is equipped with Sony IMX677, a diagonal 7.85mm CMOS active pixel type image sensor with approximately 23.64M active pixels. GoPro 9 can run at 60 Hz at the maximum resolution of 4K. The Sony IMX677

Sensor	Type	Rate	Characteristics
Camera	Sony IMX677	60 Hz	max. 5599×4223, RGB color mosaic filters
IMU	Bosch BMI260	200 Hz	3D Accelerometer & 3D Gyroscope
GPS	UBlox UBX-M8030-CT	18 Hz	2 m CEP Accuracy

TABLE I: Overview of sensors in GoPro9 camera.

sensor has an inbuilt 12-bit A/D converter to shoot high-speed and high-definition videos using horizontal and vertical binning and subsampling readout. The sensor has on-chip R, G, and B primary color mosaic filters for better color capture. GoPro 9 has multiple settings for recording the video, while many can be used there are certain modes which are prohibitive to VIO operations due to the non-linear transformation of the image as detailed in [49]. We found that the SuperView mode generates non-linear distortions that thwart calibration of the camera intrinsic parameters during data collection. The videos were recorded at full High Definition (HD) resolution of 1960×1080 with wide lens setting: horizontal field-of-view (FOV) 118°, vertical FOV 69°, and hypersmooth level set to off. Hypersmooth levels control the electronic image stabilization that predicts camera motion and compensates for it by cropping the viewable image. Hypersmoothing can effectively crop up to 10% of the image frame and the amount of cropping depends on the amount of motion, rendering this mode extremely challenging for VIO applications.

GoPro 9 also includes a Bosch BMI260 IMU equipped with 16-bit 3-axis MEMS accelerometer and gyroscope. GoPro 9 inherently records IMU data at 200Hz. The timestamps of IMU and camera are synchronized using the timing information from metadata encoded inside the MP4 video.

GoPro 9 also includes a UBlox UBX-M8030 GNSS chip capable of concurrent reception of up to 3 GNSS (GPS, Galileo, GLONASS, BeiDou) and accuracy of 2 m horizontal circular error probable, meaning 50% of measurements fall inside circle of 2 m.

B. GoPro Telemetry Extraction

The GoPro 9 MP4 video file is divided into multiple streams namely video encoded with H.265 encoder, audio encoded in advanced audio coding (AAC) format, timecode (audio-video synchronization information), GoPro fdsc data stream for file repair and GoPro telemetry stream in GoPro Metadata Format referred as GPMF[50]. GPMF – is a modified Key, Length, Value solution, with a 32-bit aligned payload, that is both compact, fully extensible, and somewhat human readable in a hex editor. Please refer to [50] for more details on GPMF, here we focus on the camera-IMU synchronization.

GPMF is divided into payloads, extracted using *gpmf-parser* [50], with each payload containing sensor measurements for 1.01 seconds while recording at frame rate of 29.97 Hz as shown in Fig. 2. A particular sensor information is obtained from payload using FourCC–7-bit 4 character ASCII key, for instance 'ACCL' for accelerometer, 'GYRO' for gyroscope, and 'SHUT' for shutter exposure times. The payload also contains the starting time of each payload in microseconds relative to the start of the video capture. Since
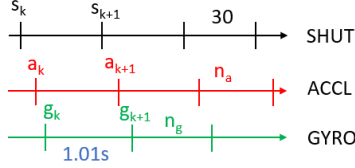
Fig. 2: Payload structure of GoPro metadata format with each payload containing 30 shutter exposure times, n_a accelerometer measurements and n_g gyroscope measurements in 1.01s at frame rate of 29.97Hz.

images are encoded in the video stream, we use the start of 'SHUT' payload to find relative timing with the accelerometer and gyroscope measurements. Using the start and end of payloads along with the number of measurements in that payload, we interpolate the timing of all measurements. We decode the video stream and extract images from MP4 file using the FFmpeg library and combine them with the IMU measurements using timing information from the GPMF payload.
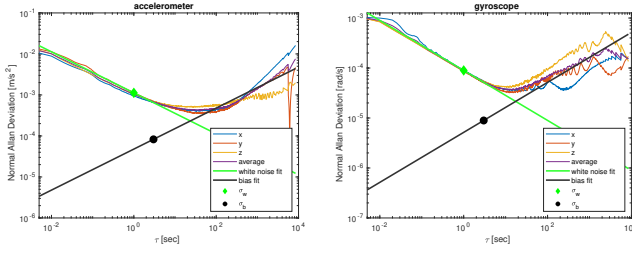

Fig. 3: Allan deviation plot of accelerometer (left) and gyroscope (right) where we fit line with slope -1/2 and 1/2 averaged over all dimensions obtained using [51]. The white noise σ_w can be found at $\tau = 1s$ on straight line with slope -1/2 and bias instability σ_b at $\tau = 3s$ on straight line with slope 1/2.

C. Calibration

Firstly, we calibrate the camera intrinsic parameters. We use the camera-calib sequences where we move the GoPro 9 camera in front of a calibration pattern placed in an indoor swimming pool from different viewing angles.

The intrinsic noise parameters of the IMU are required for the probabilistic modeling of the IMU measurements used in state estimation algorithms and the camera-IMU extrinsic parameters. We assume that IMU measurements (both linear accelerations and angular velocities) are perturbed by zero-mean uncorrelated white noise with standard deviation σ_w and random walk bias, which is the integration of white noise with standard deviation σ_b. To determine the characteristics of the IMU noise, the Allan deviation plot $\sigma_{Allan}(\tau)$ as a function of averaging time needs to be plotted. In a log-log plot white noise appears on Allan Deviation plot as a slope with gradient $-\frac{1}{2}$ and σ_w can be found at $\tau = 1$. Similarly, the bias random walk can be found by fitting a straight line with slope $\frac{1}{2}$ at $\tau = 3$; for details see [52]. Fig. 3 shows the Allan deviation plot for GoPro 9 camera along with the noise parameters.

D. Global Map

Most VIO packages ([10], [11], [13]), when applying loop closure they update only the pose graph, leaving the triangulated features in their original estimate. In contrast COLMAP [14] is a global optimization package and the final result optimizes both poses and feature 3D locations. We run COLMAP by using 2 images per second resulting in 2000 to 3000 images in cavern and cave sequences, which takes on average 7 to 10 hours. ORB-SLAM3 [12] performs global bundle adjustment after loop closure, however often diverges over large dataset as the local mapping is stopped during global optimization.

An enhancement is proposed for the SVIn2 framework to update the 3D pose of the tracked features every time loop closure occurs in order the 3D features to be consistent with the pose-graph optimization results. We maintain the pose graph with the keyframes as vertices for loop closure and the edges indicate the relative pose constraint between keyframes. For each keyframe f, the VIO module passes the following information to the loop closure module:

- $\boldsymbol{T}_{wf}$ pose of keyframe in world coordinate system
- 3D points visible in keyframe l_i with each point $l = [P_w, F, Q, I]$ has the following attributes: position in world frame $P_w \in \mathbb{R}^3$, index of keyframe $F \in \mathbb{N}$, landmark quality $Q \in \mathbb{R}$, and position of keypoint in image $I_l \in \mathbb{N}^2$.

SVIn2, provides a quality measure of a 3D point as the ratio of the square root of minimum and maximum eigenvalues of the Hessian block matrix associated with the 3D point. For each 3D point, we calculate its local position in keyframe as $P_f = \boldsymbol{T}_{wf}^{-1} P_w$, color C as RGB value at pixel location I. We collect all the observations of 3D landmark l from multiple keyframes as $O_f = [P_f, F, Q_f, C_f]$ hashed using keyframe index resulting in $O(1)$ lookup time for each observation. In the event of loop closure, we deform the global map so that the relative pose between each point and its attached keyframes remains unchanged. We fuse the multiple landmark observations from keyframe f = 1 to N to obtain global position P_w, color C and quality Q as shown in Eq. (1).

$$P_w = \frac{\sum_{f=1}^{N} T_{wf} * P_f * Q}{\sum_{f=1}^{N} Q}, C = \frac{\sum_{f=1}^{N} C_f * Q}{\sum_{f=1}^{N} Q},$$
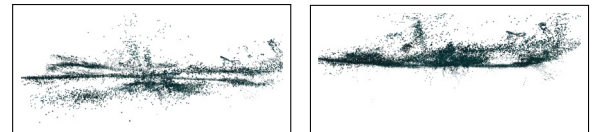$$Q = \frac{\sum_{f=1}^{N} Q}{N} \tag{1}$$

Fig. 4: Feature points generated from SVIn2 [10] and with updated positions after loop closure. View of the X-Z plane, from the shipwreck_1 sequence. Please note the three layers in the left image introduced by the gradual drift of the VIO process. Loop closure ensures that the z coordinates are consistent.

It should be noted that P_f is expressed as the relative position with respect to the keyframe and whenever the pose of keyframe T_{wf} changes due to loop closure updates,
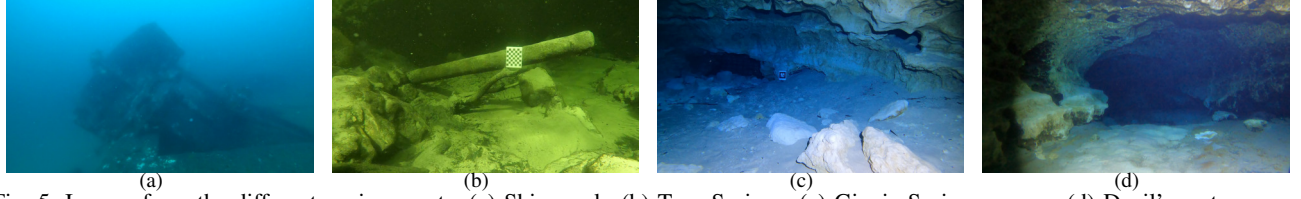
Fig. 5: Images from the different environments: (a) Shipwreck; (b) Troy Springs; (c) Ginnie Springs cavern; (d) Devil's system cave.

the location of the 3D landmarks in the global frame also changes accordingly; producing a globally consistent map as shown in Fig. 4. Realistically, the number of observations of a 3D point is quite small compared to the total 3D points in the scene; thus global map building scales linearly $O(n)$ with the number of points in the scene.

IV. Datasets

The GoPro9 Underwater VIO dataset consists of calibration sequences in addition to odometry evaluation sequences recorded using 2 GoPro 9 cameras. The two cameras are referred as g_i, $i \in [1, 2]$. Although we provide calibration results for all the datasets, the calibration sequences are made available to facilitate users wanting to preform their own calibration. Datasets can be categorized as:

- **camera-calib:** for calibrating the camera intrinsic parameters underwater. We provide calibration sequences using two calibration patterns: a grid of AprilTags and a checkerboard for each camera. The calibration patterns are recorded with slow camera motions at a swimming pool; making sure the calibration patterns are viewed from varying distances and orientations.
- **camera-imu-calib:** for calibrating the camera-IMU extrinsic parameters in order to determine the relative pose between the IMU and the camera. The camera is moved in front of the Apriltag grid exciting all 6 degrees of freedom. This sequence is recorded indoor just to expedite the calibration process as the extrinsic parameters are the same above and below water. Moreover, we also provide a **camera-calib** indoor sequence to assist with the camera-IMU calibration.
- **imu-static:** contains IMU data to estimate white noise and random walk bias parameters. These sequences are recorded with the camera stationary for at least 4 hours for each cameras.
- **shipwreck:** two sequences collected by handheld GoPro 9 on an artificial reef (refueling barge wreck) ∼ 55 Km outside of Charleston, SC, USA; see Fig. 5(a).
- **spring open water** one sequence collected by handheld GoPro 9 at the basin of Troy Springs State Park FL, USA; see Fig. 5(b). The settings had image stabilization on (hypersmooth) resulting into arbitrary cropping of the field of view.
- **cavern:** three trajectories traversed inside the ballroom cavern at Ginnie Springs, FL; see Fig. 5(c). Inside the cavern five markers (AR single tags [53]) were placed to establish ground truth measurements. Each trajectory consisted of several loops each observing

slightly different parts of the cavern but all ensuring the tag of that part of the cavern was visible.
- **cave:** two sequences were collected at the Devil's system, FL; see Fig. 5(d). For the first sequence the GoPro was mounted on the stereo rig and second sequence was using only the GoPro.

V. Experimental Results

Due to absence of GPS in underwater environments or motion capture systems, we use COLMAP [14] to generate baseline trajectories. COLMAP is a structure-from-motion (SfM) pipeline equipped with global bundle adjustment and loop closure capabilities; thus producing consistent camera trajectory and 3D reconstruction. Even though COLMAP provides good estimation of shape of trajectories, they can not be considered as ground truth. As monocular SfM inherently suffers from scale observability constraints and global optimization does not converge over large trajectories, we consider the estimated trajectories as accurate up to scale. We found that relative scale between COLMAP and all VIO trajectories was almost equal. Hence, COLMAP trajectories are scaled by scaling factor calculated from the average of all VIO trajectories in subsequent sections unless otherwise specified.

A. Tracking Evaluation Metrics

As COLMAP does not provide accurate scale, we evaluate the accuracy of the various tracking algorithms using the *absolute trajectory error (ATE)* metric after **Sim(3)** alignment [54]. ATE is calculated as the root mean squared difference between ground truth 3D positions obtained from COLMAP p_i and corresponding estimated 3D positions $\hat{p}_i$ aligned using optimal **Sim(3)** rotation matrix R, translation t, and scaling factor s:

$$e_{ATE} = \min_{(R,t,s) \in \mathbf{Sim(3)}} \sqrt{\frac{1}{n} \sum_{i=1}^{n} \| p_i - (sR\hat{p}_i + t) \|^2} \quad (2)$$

B. Tracking Results

We compare the performance of various open source visual-inertial odometry (VIO) methods on the above described datasets. Since, most of the VIO algorithms have parameters that are tuned at VGA resolution; their performance was found to be better at quarter resolution (960×540 pixels). However, we provide datasets at full high definition resolution (1920×1080 pixels) to further the research in underwater VIO and SfM. Unless specified otherwise, all the VIO algorithms use quarter resolution dataset.

We evaluate the performance of VINS-Mono [11], ORB-SLAM3[12], SVIn2[10] and OpenVINS[13] based on the RMSE of the ATE as shown in Table II. All the compared algorithms are equipped with loop closure resulting in low overall RMSE error. It should be noted that OpenVINS is based on the Mutiple-State Constraint Kalman Filter, where as all other methods are based on non-linear least squares optimization.

Sequence	length[m]	SVIn2	ORB-SLAM3	VINS-Mono	OpenVINS
g1_shipwreck1	71.60	0.148	**0.098**	0.143	0.167
g1_shipwreck2	79.07	0.238	0.207	**0.202**	0.337
g1_openwater	53.26	**0.263**	×	0.486	×
g1_cavern1	243.10	**0.090**	0.231	**0.090**	0.288
g1_cavern2	240.60	**0.074**	0.363	0.097	0.084
g2_cavern3	341.29	0.089	0.687	0.131	**0.081**
g2_cave1	219.58	**0.072**	0.150	0.105	0.081
g2_cave2	222.64	0.093	**0.089**	0.097	0.229

TABLE II: Performance of evaluated open source algorithms on various datasets based on root mean squared ATE in meters.

All system are able to track most of the sequences until the end. VINS-Mono and SVIn2 were able to track the complete trajectory consistently with good accuracy. In the cavern sequences, ORB-SLAM3 took too long during global bundle adjustment after loop closure and lost track as it disabled local mapping during global optimization. However, ORB-SLAM3 is equipped with map merging and was able to relocalize and merge the disjoint maps. This produced slightly inferior performance in the cavern sequences. OpenVINS required smooth motion for initialization as it only relies on IMU measurement for gravity alignment and orientation initialization. Hence, OpenVINS might diverge unless data collection is started from a static position or good initialization is found. Finally, it is worth noting that the open water sequence was recorded with the hypersmooth option activated resulting in failures in ORB-SLAM3 and OpenVINS.

C. AR-tag based Validation

As there is no continuous tracking for absolute ground truth, 3D landmark-based validation with AR tags is used to quantify the accuracy of the evaluated methods. As a part of the experimental setup in the cavern sequences with multiple loops, we placed 5 different AR-tags printed on waterproof paper at different locations inside the cavern. We observe the variance of the position of the AR-tags from their mean position over the whole length of trajectory. If the trajectories do not drift over time, the markers must be observed at the same location during multiple visits. Among the cavern sequences, we detected most AR-tags in the g1_cavern2 sequence; therefore, this sequence is used as reference for further analysis.

We determined the relative position between the camera and the tags in g1_cavern2 sequence using *ar_track_alvar*[3].

[3] http://wiki.ros.org/ar_track_alvar

By projecting a 3D cube over the tags using the pose estimate, we observed higher noise in the orientation estimate; hence, only the position of the AR-tags is used for the error analysis. Once the relative position from *ar_track_alvar* is found, the global position can be found as $T_{WM}^k = T_{WC}^k * P_{CM}$ where T_{WM}^k is the marker position in world coordinate frame W at time k, T_{WC}^k is the pose of the camera C in W at time k (produced by SLAM/odometry system), and P_{CM} is the relative position of marker M from camera.

Fig. 7 shows boxplots of the displacement from the mean position of the markers over the whole length of the trajectory for all the different methods, including COLMAP. Table III shows the summary of the standard deviation in translation and the average distance error. All the algorithms performed well with slightly inferior performance of ORB-SLAM3 due to tracking issues. Fig. 8 shows the position of the tags in g1_cavern2 sequence observed by different packages along with the COLMAP trajectory as reference.

	t_x(m)	t_y(m)	t_z(m)	Avg dist. error (m)
COLMAP	**0.057**	0.072	**0.034**	**0.099**
OpenVINS	0.063	**0.071**	**0.034**	0.104
ORB-SLAM3	0.246	0.222	0.068	0.268
SVIn2	0.058	0.075	0.049	0.106
VINS-Mono	0.073	0.086	0.045	0.122

TABLE III: Standard deviation in translation of detected tags and average distance error.

D. Global Mapping

We evaluate the global map produced by enhancing SVIn2 to update the triangulated feature positions after loop closures by comparing with COLMAP's sparse pointcloud. The pointclouds come from different sources and differ in size, so we align the pointclouds from COLMAP and SVIn2 as follows:

1) Perform voxel downsampling with voxel size of 10cm.
2) Compute FPFH [55] feature descriptor describing local geometric signature for each point.
3) Find correspondence between pointclouds by computing similarity score between FPFH descriptors.
4) Feed all putative correspondences to TEASER++ [56] to perform global registration finding transformation to align corresponding points.
5) Fine tune registration by running ICP over original point cloud with TEASER++ solution as initial guess.

The reconstruction results are compared based on registration accuracy using fitness and inlier_rmse metrics. More specifically, fitness is the ratio of number of inlier correspondences (distance less than voxel size) and number of points in SVIn2 pointcloud. Whereas, inlier_rmse is the root mean squared error of all inlier correspondences. Table IV shows the similarity between SVIn2 and COLMAP reconstruction based on fitness and inlier_rmse metrics. Fig. 10 shows aligned sparse reconstruction obtained from COLMAP and SVIn2 in g1_shipwreck1 and g1_cavern2 sequence.

VI. CONCLUSION

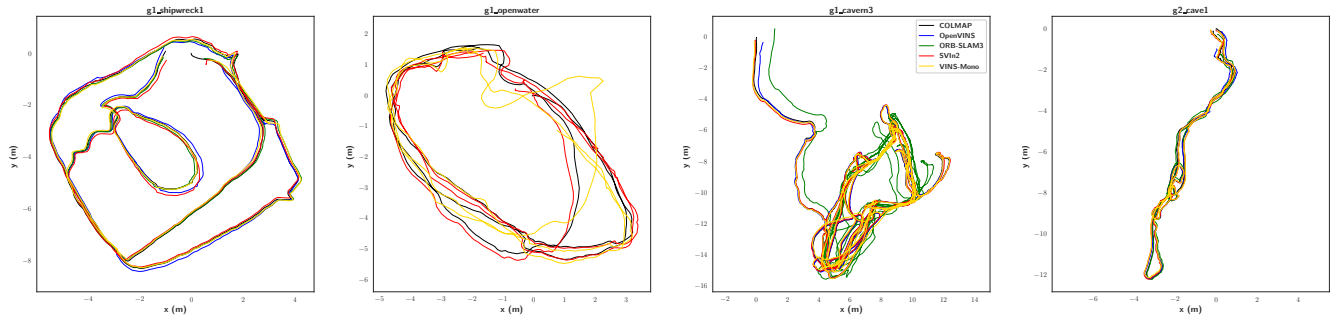In this work we presented a complete pipeline for underwater SLAM utilizing a commonly available, inexpensive,

Fig. 6: Trajectories of the evaluated algorithms on g1_shipwreck1, g1_openwater, g1_cavern3, and g2_cave1 sequences from our GoPro Underwater dataset. COLMAP (black) obtained using global bundle adjustment is used as ground truth up to scale. All other trajectories are obtained by enforcing real time constraints; even if serial processing modes are available for ORB-SLAM3 and OpenVINS.
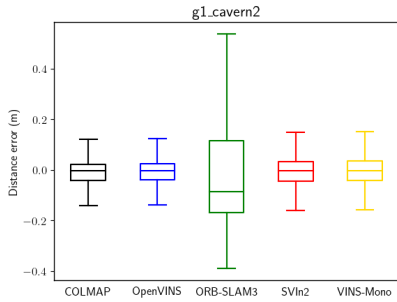

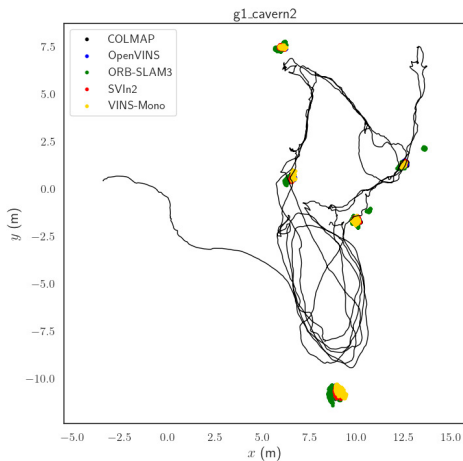Fig. 7: Boxplot of distance error for all tags visible in g1_cavern2.

Fig. 8: The position of observed tags (in total 5) placed at different places inside the cavern as calculated using evaluated algorithms. The COLMAP trajectory also shown as reference.

	fitness	inlier_rmse	# of correspondences
g1_shipwreck1	0.773	0.058	12,391
g1_shipwreck2	0.500	0.063	9959
g1_cavern1	0.738	0.058	34,481
g1_cavern2	0.727	0.060	44,797
g2_cavern3	0.698	0.059	46,875
g2_cave1	0.107	0.067	6302
g2_cave2	0.326	0.064	19,750

TABLE IV: Similarity between SVIn2 and COLMAP sparse reconstructions based on fitness and inlier_rmse metrics along with no. of correspondences.

action camera with superior performance. The proposed approach was tested in open and confined waters, with natural and artificial illumination, under challenging conditions. The
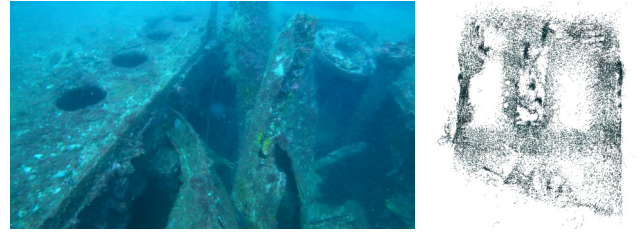

Fig. 9: A view from the GoPro showing part of the wreck's deck and the global map showing the 3D reconstruction of the mapped area. Please note the round in the reconstruction next to the middle opening with the broken beams.
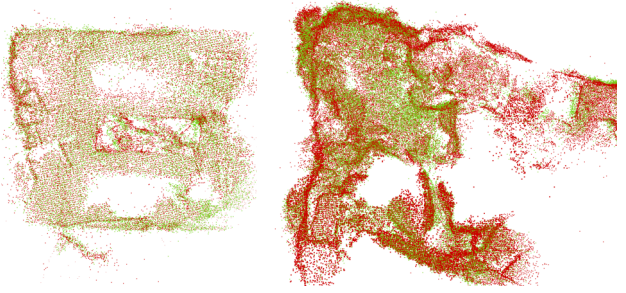

Fig. 10: Sparse reconstruction obtained using COLMAP (red) and SVIn2 (green) from the g1_shipwreck1 and g1_cavern2 sequences.

SVIn2 framework was augmented to correct the 3D features according to the updated pose graph after successful loop closure calculations. The resulting map demonstrated accuracy similar to the much slower global optimization COLMAP package. The experimental results verify that the specific camera is capable of producing accurate estimates of the trajectory together with consistent sparse representations of the environment. Future uses of the proposed framework would be in recording and documenting the surroundings and the trajectory of AUVs operating autonomously in challenging underwater environments such as caves and shipwrecks [57], [58].

Currently we are investigating synchronization methods between the GoPro camera and other devices such as Autonomous Underwater Vehicle and sensor suites. By introducing additional data streams such as water depth and magnetometer data, both providing absolute values, we expect to reduce the drift accumulating over long trajectories without loops and increase the overall accuracy.

REFERENCES

[1] J.-Y. Cousteau and F. Dumas, *The silent world*. Harper Collins, 1953.

[2] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter, "Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters," *The Int. Journal of Robotics Research*, vol. 25, no. 12, pp. 1223–1242, 2006.

[3] S. Demesticha, D. Skarlatos, and A. Neophytou, "The 4th-century bc shipwreck at mazotos, cyprus: New techniques and methodologies in the 3d mapping of shipwreck excavations," *Journal of Field Archaeology*, vol. 39, no. 2, pp. 134–150, 2014.

[4] M. González-Rivero, P. Bongaerts, O. Beijbom, O. Pizarro, A. Friedman, A. Rodriguez-Ramirez, B. Upcroft, D. Laffoley, D. Kline, C. Bailhache, *et al.*, "The catlin seaview survey–kilometre-scale seascape assessment, and monitoring of coral reef ecosystems," *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 24, no. S2, pp. 184–198, 2014.

[5] M. Modasshir and I. Rekleitis, "Augmenting coral reef monitoring with an enhanced detection system utilizing a deep semi-supervised learning approach," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 1874–1880.

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

[7] M. Roznere and A. Q. Li, "Real-time model-based image color correction for underwater robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 7191–7196.

[8] B. Joshi, S. Rahman, M. Kalaitzakis, *et al.*, "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)*, Nov. 2019, pp. 7221–7227.

[9] A. Quattrini Li, A. Coskun, S. M. Doherty, *et al.*, "Experimental comparison of open source vision based state estimation algorithms," in *International Symposium of Experimental Robotics (ISER)*, Mar. 2016.

[10] S. Rahman, A. Quattrini Li, and I. Rekleitis, "An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1861–1868.

[11] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, 2018.

[12] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, 2021.

[13] 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.

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

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

[16] R. Photo and Video, *KELDAN VIDEO 8X LIGHT (18,000 LUMEN, CRI 85)*, `https://reefphoto.com / collections / keldan / products / keldan-video-8x-light-18-000-lumen-cri-85-2020`, 2021.

[17] N. Weidner, S. Rahman, A. Quattrini Li, and I. Rekleitis, "Underwater cave mapping using stereo vision," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5709–5715.

[18] P. Helmholz and D. Lichti, "Assessment of chromatic aberrations for gopro 3 cameras in underwater environments," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2019.

[19] E. Nocerino, F. Neyer, A. Grün, M. Troyer, F. Menna, A. Brooks, A. Capra, C. Castagnetti, and P. Rossi, "Comparison of diver-operated underwater photogrammetric systems for coral reef monitoring," *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, 2019.

[20] F. Neyer, E. Nocerino, and A. Grün, "Image quality improvements in low-cost underwater photogrammetry," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, no. 2/W10, pp. 135–142, 2019.

[21] T. Guo, A. Capra, M. Troyer, A. Grün, A. J. Brooks, J. L. Hench, R. J. Schmitt, S. J. Holbrook, and M. Dubbini, "Accuracy assessment of underwater photogrammetric three dimensional modelling for coral reefs," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 41, no. B5, pp. 821–828, 2016.

[22] B. Gintert, A. Gleason, K. Cantwell, N. Gracias, M. Gonzalez, and R. Reid, "Third-generation underwater landscape mosaics for coral reef mapping and monitoring," in *Proceedings of the 12th international coral reef symposium, Cairns, Australia*, 2012, pp. 9–13.

[23] T. Van Damme, "Computer vision photogrammetry for underwater archaeological site recording in a low-visibility environment," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 5, p. 231, 2015.

[24] V. Raoult, P. A. David, S. F. Dupont, C. P. Mathewson, S. J. O'Neill, N. N. Powell, and J. E. Williamson,

"Gopros™ as an underwater photogrammetry tool for citizen science," *PeerJ*, vol. 4, e1960, 2016.

[25] V. E. Schmidt and Y. Rzhanov, "Measurement of micro-bathymetry with a gopro underwater stereo camera pair," in *2012 Oceans*, IEEE, 2012, pp. 1–6.

[26] S. M. Nornes, M. Ludvigsen, Ø. Ødegard, and A. J. SØrensen, "Underwater photogrammetric mapping of an intact standing steel wreck with rov," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 206–211, 2015.

[27] B. Allotta, R. Costanzi, A. Ridolfi, C. Colombo, F. Bellavia, M. Fanfani, F. Pazzaglia, O. Salvetti, D. Moroni, M. A. Pascali, *et al.*, "The arrows project: Adapting and developing robotics technologies for underwater archaeology," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 194–199, 2015.

[28] F. Menna, P. Agrafiotis, and A. Georgopoulos, "State of the art and applications in archaeological underwater 3d recording and mapping," *Journal of Cultural Heritage*, vol. 33, pp. 231–248, 2018.

[29] C. Balletti, C. Beltrame, E. Costa, F. Guerra, and P. Vernier, "Underwater photogrammetry and 3d reconstruction of marble cargos shipwreck.," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2015.

[30] N. Palomeras, N. Hurtós, M. Carreras, and P. Ridao, "Autonomous mapping of underwater 3-d structures: From view planning to execution," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1965–1971, 2018.

[31] D. F. Coleman, "Underwater archaeology in thunder bay national marine sanctuary, lake huron—preliminary results from a shipwreck mapping survey," *Marine Technology Society Journal*, vol. 36, no. 3, pp. 33–44, 2002.

[32] K. Bain, R. Cuttler, E. Kieran, and S. Al Naimi, "Wreck and ruin—mapping the shipwreck resource of qatar," in *Proceedings of the Seminar for Arabian Studies*, JSTOR, 2014, pp. 35–41.

[33] A. A. Mogstad, Ø. Ødegård, S. M. Nornes, M. Ludvigsen, G. Johnsen, A. J. Sørensen, and J. Berge, "Mapping the historical shipwreck figaro in the high arctic using underwater sensor-carrying robots," *Remote Sensing*, vol. 12, no. 6, p. 997, 2020.

[34] S. B. Williams, O. Pizarro, and B. Foley, "Return to antikythera: Multi-session slam based auv mapping of a first century bc wreck site," in *Field and Service Robotics*, Springer, 2016, pp. 45–59.

[35] A. Hogue and M. Jenkin, "Development of an underwater vision sensor for 3d reef mapping," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2006, pp. 5351–5356.

[36] M. D. Dunbabin and S. S. Allen, "Large-scale habitat mapping using vision-based auvs: Experiences, challenges & vehicle design," in *OCEANS 2007-Europe*, IEEE, 2007, pp. 1–6.

[37] S. Williams and I. Mahon, "Simultaneous localisation and mapping on the great barrier reef," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, IEEE, vol. 2, 2004, pp. 1771–1776.

[38] M. Ferrera, V. Creuze, J. Moras, and P. Trouvé-Peloux, "Aqualoc: An underwater dataset for visual–inertial–pressure localization," *The International Journal of Robotics Research*, vol. 38, no. 14, pp. 1549–1559, 2019.

[39] B. A. am Ende, "3d mapping of underwater caves," *IEEE Computer Graphics and Applications*, vol. 21, no. 2, pp. 14–20, 2001.

[40] W. C. Stone, B. A. a. Ende, F. L. Wefer, and N. A. Jones, "Automated 3d mapping of submarine tunnels," in *Robotics 2000*, 2000, pp. 148–157.

[41] M. Gary, N. Fairfield, W. C. Stone, D. Wettergreen, G. Kantor, and J. M. Sharp Jr, "3d mapping and characterization of sistema zacatón from depthx (deep phreatic thermal explorer)," in *Sinkholes and the Engineering and Environmental Impacts of Karst*, 2008, pp. 202–212.

[42] E. Nocerino, M. M. Nawaf, M. Saccone, M. B. Ellefi, J. Pasquet, J.-P. Royer, and P. Drap, "Multi-camera system calibration of a low-cost remotely operated vehicle for underwater cave exploration," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, no. 1, pp. 329–337, 2018.

[43] E. Nocerino, F. Menna, E. Farella, and F. Remondino, "3d virtualization of an underground semi-submerged cave system," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, no. 2/W15, pp. 857–864, 2019.

[44] A. Mallios, P. Ridao, D. Ribas, M. Carreras, and R. Camilli, "Toward autonomous exploration in confined underwater environments," *Journal of Field Robotics*, vol. 33, no. 7, pp. 994–1012, 2016.

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

[46] 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.

[47] S. Rahman, A. Quattrini Li, and I. Rekleitis, "Contour based reconstruction of underwater structures using sonar, visual, inertial, and depth sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 8048–8053.

[48] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.

[49] GoPro, *HERO9 Black - Video Settings and Resolutions*, https://community.gopro.com/t5/tkb/articleprintpage/tkb-id/kb-en/article-id/1597, 2021.

[50] ——, *Gpmf-parser*, `https : / / github . com / gopro/gpmf-parser`, 2021.

[51] P. Geneva, *Kalibr allan*, `https://github.com/ rpng/kalibr_allan`, 2021.

[52] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers, "The tum vi benchmark for evaluating visual-inertial odometry," in *International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[53] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.

[54] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 376–380, 1991.

[55] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009.

[56] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, 2021.

[57] M. Xanthidis, N. Karapetyan, H. Damron, S. Rahman, J. Johnson, A. O'Connell, J. O'Kane, and I. Rekleitis, "Navigation in the presence of obstacles for an agile autonomous underwater vehicle," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 892–899.

[58] M. Xanthidis, M. Kalaitzakis, N. Karapetyan, J. Johnson, N. Vitzilaios, J. O'Kane, and I. Rekleitis, "Aquavis: A perception-aware autonomous navigation framework for underwater vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, accepted.