Benchmarking Visual Localization and Mapping Algorithms on Underwater Cave Dataset

Max Rucker
University of Michigan
mruck@umich.edu

***Abstract*—Autonomous Underwater Vehicles (AUVs) are extremely limited by their sensor capabilities, leading to unique and creative methods to be used for underwater localization. One of which is using new advancements in visual feature detection and SLAM to get more precise localization when navigating enclosed or tight-knit areas such as underwater caves or sunken shipwrecks. This paper analyzes two state-of-the-art visual SLAM algorithms on an underwater cave dataset taken from an autonomous underwater vehicle traversing the environment. This environment provides a challenge for visual SLAM systems due to the attenuation and backscattering of water, along with bland environments that feature-based methods may struggle with. The two visual SLAM algorithms used are ORBSLAM 3, an open-source visual-inertial SLAM system commonly used in in-air applications, and SVIn2, an acoustic, visual, inertial SLAM algorithm specifically designed for underwater applications. Our results show that monocular ORBSLAM has very inconsistent results over a long trajectory with issues in tracking due to underwater effects such as attenuation, backscatter, and bland environments. Along with this, ORBSLAM mono-inertial and SVIn2 fail when IMU data can not be correctly calibrated initially. This paper hopes to add valuable new evaluations of these visual SLAM systems in a unique environment and provide insight into potential further studies that could be done for better SLAM for autonomous underwater vehicles and their respective datasets.**

I. Introduction

Autonomous underwater vehicles (AUVs) are becoming a strong tool for exploring underwater areas and conducting research without the need for human divers or operators. AUVs provide a unique opportunity to explore unexplored areas, study marine ecosystems, map underwater geological features and more [1]. Within marine robotics, one of the biggest challenges has been the mapping of underwater environments. Whether it be sunken shipwrecks, underwater caves, or coral reefs, there is an effort to explore more of these areas. Autonomous underwater vehicles (AUVs) offer great opportunities to explore these areas safely and collect data on many unreachable areas [2]. While AUVs do provide great potential for exploring these underwater environments, they are heavily limited by what sensors can sufficiently work in underwater settings. Common methods such as GPS and LIDAR fail in underwater settings and more expensive methods are utilized such as SONAR and Doppler velocity logs [3] [4]. This provides a great issue for autonomous path planning and localization through these extreme areas. One option for SLAM in terrestrial applications is Visual SLAM which uses camera imaging and feature detection to map and localize in real time. Visual SLAM poses an inexpensive process for localization that can be applied to underwater settings.

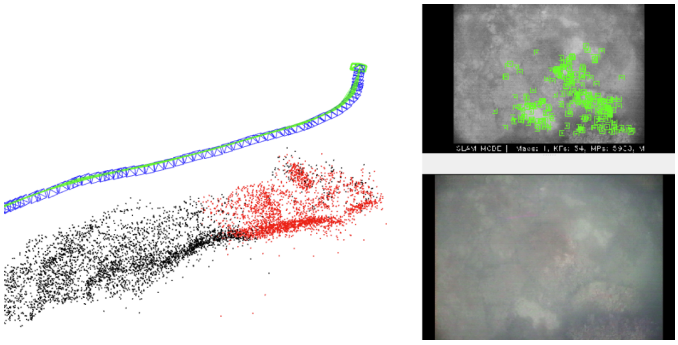

Fig. 1. ORBSLAM Monocular Visual SLAM running on underwater caves sonar and vision dataset [10]. Camera Trajectory and ORB feature detection can be seen.

Visual localization and mapping algorithms have seen a great deal of innovation with new developments in feature-based detection. With these innovations, feature-based algorithms have become a reliable approach for localization methods. Algorithms such as ORBSLAM [5] have been seen to give reliable results in many applications. While visual SLAM has great results in terrestrial applications, the underwater setting provides a new challenge for visual SLAM systems [6]. Issues not found in air settings such as light attenuation and backscattering have drastic effects on camera imaging [1] [7] [8] [9].

With this said, there could be cases where visual localization and mapping could provide further benefits to the localization of AUVs such as small caves and sunken shipwrecks [6]. This paper displays results from state-of-the-art feature-based visual localization and mapping algorithms tested on monocular underwater camera data to showcase their effectiveness at localizing underwater. These results can give insight into what next steps need to be taken for underwater imaging and SLAM to be improved for AUVs.

II. Related Works

A. Feature Detection

The main driver behind visual localization and mapping for monocular camera data is feature detection. The key idea behind feature detection is taking an image and noting key

points within the image. The way these key points are selected within images depends on the algorithm used and could vary greatly, but it usually involves differences in hue, saturation, or more [11]. Using these features in an image, they can be compared against features in future images and matched together to determine shifts within the camera. The main state-of-the-art algorithms within feature detection include ORB [12], SIFT, and SURF. These algorithms have been tested extensively on various applications and have seen great results with various data [13]. The main areas that they fail in are when it comes to repetitive or bland images without lots of differences [12]. In these scenarios, it becomes hard for the algorithms to spot various features within an image or determine matches in further images.

Recently, there have been some further developments in feature-based detection with deep learning. While these methods have been seen as successful, there are external problems that arise with them [14]. This includes needing to have extensive data to train networks on. In environments such as underwater, there is a lack of data with ground truths to train models on. For the scope of this project, this paper will not be evaluating or testing any deep learning methods.

B. ORBSLAM 3

One state-of-the-art visual localization and mapping system is based on ORB feature detection [12] and is known as ORBSLAM [5]. ORB feature detection is one of the few open-source feature detection methods and was made as a competition to other patented algorithms such as SIFT and SURF. It has been shown to outperform these methods consistently [12]. ORB feature detection is based on using FAST keypoint detection to gather initial feature points on a pyramid scale and then passes those feature points through BRIEF which encodes them in binary for fast processing [15]. Utilizing these ORB features, ORBSLAM creates keyframes that are used to create a sparse map that can be referenced by new keyframes to localize the camera. ORBSLAM also allows for loop closure, where when it detects a loop it can fix a previous trajectory based on the previously created map. Overall, ORBSLAM is a strong visual SLAM algorithm that has been shown to outperform many other classical feature-based SLAM approaches.

New developments have been made in ORBSLAM leading to the latest version being ORBSLAM 3 [16] which introduces a new mapping Atlas to hold multiple maps when tracking is lost, as well as visual-inertial SLAM that utilizes inertial measurement unit data for better tracking accuracy. This paper leverages ORBSLAM 3 as an open-source tool to evaluate its performance in an underwater setting. This paper will push ORBSLAM 3 to its limits with the challenging environment that comes with underwater camera data where feature detection may be minimal. This paper will test ORBSLAM 3 with its monocular and mono-inertial settings.

C. SVIn2

One newly developed SLAM system for underwater applications is SVIn2 [6] which fuses acoustic, visual, and inertial data for SLAM. SVIn2 uses the OKVIS [17] package for visual-inertial SLAM and fuses it with SONAR data for better tracking within the low-feature environment in underwater settings. Along with using this SONAR data, SVIn2 uses image enhancement through Contrast Limited Adaptive Histogram Equalization (CLAHE) for better feature detection [6]. This is a preprocessing method that creates several histograms for each section of an image and redistributes contrast using these histograms while capping them to keep the entire image contrast consistent. This is done to aid the feature detection of OKVIS which is a keyframe-based SLAM algorithm similar to ORBSLAM. For the scope of this project, this paper will only focus on testing the visual-inertial aspect of SVIn2.

D. Underwater Imaging

The two biggest issues that underwater imaging faces are backscattering and attenuation. Backscattering is when light is reflected off particles floating in a medium and directed back at the camera. For imaging, this results in bright hazes appearing in the image at random moments throughout a video. With the amount of particles commonly floating in water in underwater environments, backscattering largely affects the clarity of video imaging taken from an AUV. Backscattering can cause key parts of an image to become blurry or covered due to this intense light being reflected into the camera [7] [8] [9]. The other issue imaging faces is attenuation where light is absorbed or diffracted by the medium's particles. Water has high attenuation and can absorb a large amount of light. This causes high-frequency light such as red colors to be absorbed leaving mostly low-frequency light such as blue or greenish colors to travel further [7] [8] [9]. This is why images underwater have a blueish hue. The main issue caused by this for imaging is that the distance cameras can see can vary extremely depending on the quality of the water. This paper will look at how these imaging limitations affect the quality of visual SLAM systems.

III. Technical Approach

A. Underwater Caves Sonar Dataset

For this paper, the main dataset being used to evaluate visual SLAM is the underwater cave sonar dataset from Girona Underwater Vision and Robotics lab [10]. This dataset comprises data gathered by an autonomous underwater vehicle (AUV) within the intricate and unstructured confines of an underwater cave complex. The AUV used in this dataset is the Sparus AUV. It has six main sensors attached, including two inertial measurement units, two sonar sensors, one Doppler velocity log, and a downward-facing analog camera. For our purposes, we will be focusing on using the downward-facing analog camera for visual data for feature detection and SLAM. The camera used has a resolution of 384x288 pixels. An example of the image quality is seen in Figure 2.

Fig. 2. Camera view from Sparus AUV, adapted from [10]

This dataset also provides a ground truth in the form of traffic cones placed throughout the AUV's path. For our purposes. we can use the placement of these cones and their absolute distances to compare the predicted trajectory length against the length between each cone pairing. The limiting factor behind this ground truth is that the measurements are not precise, only being taken by a tape measurer throughout the cave by divers. Along with this, only the absolute distances are provided and not the direction. This does give a somewhat general estimate of the distance, but it can be used to get a gauge of how accurate the predicted distance is. The path the AUV took and the placement of the cones can be seen in Figure 3.

For utilizing this dataset, ROSBAG files are provided that contain information for the camera and the sensor data. One ROSBAG contains pure camera data of the images with a framerate of 5 fps. It also contains information for the calibration of the camera. The other ROSBAG contains information on the IMU, Sonar, and DVL data along with each transformation from sensor frame to body frame. Along with this, estimated odometry based on all these sensors is provided from evaluation from the lab. This information on the trajectory can be used as a sudo ground truth for our localization to be compared against.

There are two main reasons this dataset was chosen. The first reason is that this dataset is a real-world cave dataset taken from an AUV. There are very few datasets that have a monocular camera navigating through an underwater cave where the AUV is close to the floor or walls for a clean image feed to be taken for evaluation. This is a great test example of how a real AUV could leverage visual SLAM in these scenarios where it is forced to be close to the cave walls for navigation. Another reason is that this is one of the few datasets that has somewhat of a ground truth for evaluation. Many underwater datasets lack ground truth which makes it extremely hard to work with them and evaluate them quantitatively. The cones along with the implemented sensor fusion odometry provide a great base for this paper to evaluate visual trajectories.

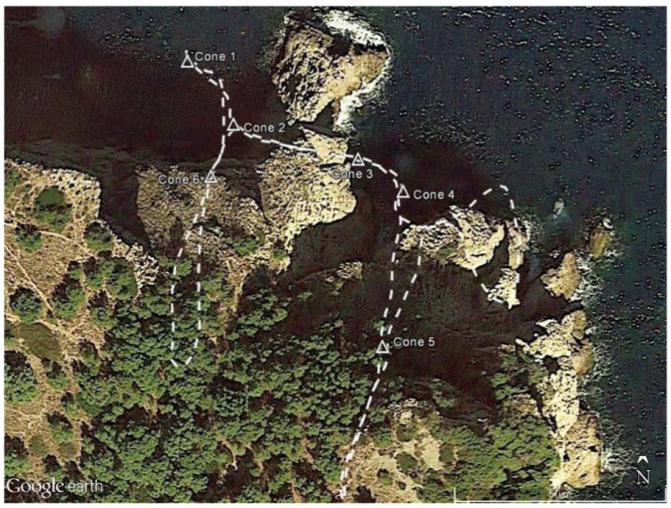

Fig. 3. Path of Sparus AUV through underwater cave along with cone placement and labeling, adapted from [10]

B. ORBSLAM 3 Monocular Approach

For the implementation of Monocular ORBSLAM on the dataset, parameters had to be tuned to allow the algorithm to run effectively for the environment. For calibration, the number of features was set to 2000, the pyramid scale set to 1.2, and the number of levels to seven. These values are a bit higher than normal due to the low contrast of the underwater environment presented by the dataset. Another key implementation was using the Contrast limited adaptive histogram equalization (CLAHE) to increase the range of contrast in the video feed. The difference in feature detection can be seen in figure 4

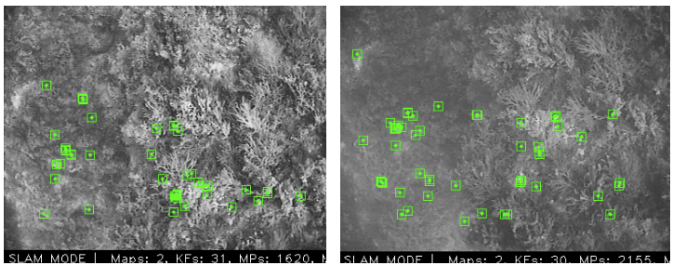

Fig. 4. CLAHE effects on image quality and feature detection. Left is CLAHE corrected image and right is initial grayscale image.

C. ORBSLAM 3 Mono-Inertial Approach

For ORBSLAM 3 mono-inertial SLAM, the same ORB parameters were used as in the ORBSLAM Monocular SLAM and CLAHE was also applied before feature detection. For the inertial aspect, the rotation and translation matrix was applied to take the camera pose and adjust it to the IMU (body) frame.

Along with this, the IMU parameters such as acceleration walk, gyroscopic walk, noise, and frequency.

D. SVIn2 Approach

For SVIn2, the initial OKVIS parameters were used since they have already been adjusted to account for underwater datasets through SVIn2. Along with this, the images are already processed with CLAHE for image correction. For the IMU data, the same process for IMU frame was done as in the ORBSLAM 3 mono-inertial system.

IV. Results

A. Evaluation

To evaluate the trajectory against the ground truth, the absolute pose error (APE) and the relative pose error (RPE) are used to compare the ORBSLAM trajectory against the sudo ground truth provided in the caves dataset [12].

The absolute pose error is the error between the predicted trajectory and the ground truth trajectory at a certain time. The average APE is defined as:

$$APE_{\mu} = \frac{1}{n}\sum_{i=1}^{n} ||trans(E_i)||^2 \tag{1}$$

E_i is the absolute trajectory error that comes from the Horn method which finds the rigid body transformation to line up the trajectories and have them overlap.

The relative pose error is similar to the absolute pose error except instead of taking the pose error over the entire trajectory, the relative pose error only takes the error for small snippets of the predicted trajectory. This allows for analysis over small areas to evaluate trajectories incase there is a drift in the predicted trajectory. The average RPE is defined as:

$$RPE_{\mu} = \frac{1}{m}\sum_{i=1}^{m} ||trans(F_i)||^2 \tag{2}$$

F_i is the relative trajectory error that is similar to E_i except that F_i is only calculated over a small step of Δ to avoid one bit of error changing the absolute error of the entire trajectory.

For evaluation, the mean APE and RPE will be calculated for trajectories between pairs of cones within the dataset. By running ORBSLAM only between cone pairings, it will not run into issues of relocalization when tracking is lost in cases where attenuation and backscattering is too extreme for ORBSLAM to handle.

Evo is used to calculate and match trajectories from the predicted trajectory and the ground truth as well as calculate the APE and RPE. It is also used to graph the trajectories in a 3D scene. Example graphs can be seen in figures 5, 6, and 7.

B. ORBSLAM 3 Monocular Only

The results from ORBSLAM 3 Monocular can be seen in Table I. Some key notes before getting into the evaluation are the missing data points for the ground truth distance. The biggest issue was that the underwater dataset does not include the distance from cone 4 to cone 5 and vice versa. There was no explanation for why this distance was missing, so we have no ground truth distance for pairings with cone 5. Another thing is the distance where it loops back to a cone such as 6 - 6. This looping makes it so no ground truth can be gathered during that pathing sequence. For this reason, we also included the sudo ground truth distance calculated from the provided odometry in the dataset.

With that, looking at the results from ORBSLAM, one main takeaway is the inconsistency with tracking. Only four out of the eleven paths could localize and map for their respective distances. Overall, ORBSLAM could only hold tracking for an average of 21 meters. While this seems good, there are two outliers which are the cone paths of 4-5 and 6-1 where the predicted path distance is much longer than reality. These outliers are due to the APE being high (¿ 2) for those paths meaning localization for those paths was bad. Removing those two points from the data gives an average of 11.50 meters. This is a very short tracking distance for ORBSLAM, but with inconsistencies in the environment with bland scenes, it is hard for feature detection to stay effective.

Looking at the APE and RPE, it is interesting to see that the values are low for most of the datasets. There is an overall average of 1.88 m for APE and an average of 0.393 m for RPE. These low errors show that when ORBSLAM can keep consistent tracking underwater, the error is very minimal. While the APE is a bit larger, this is because one spike in the

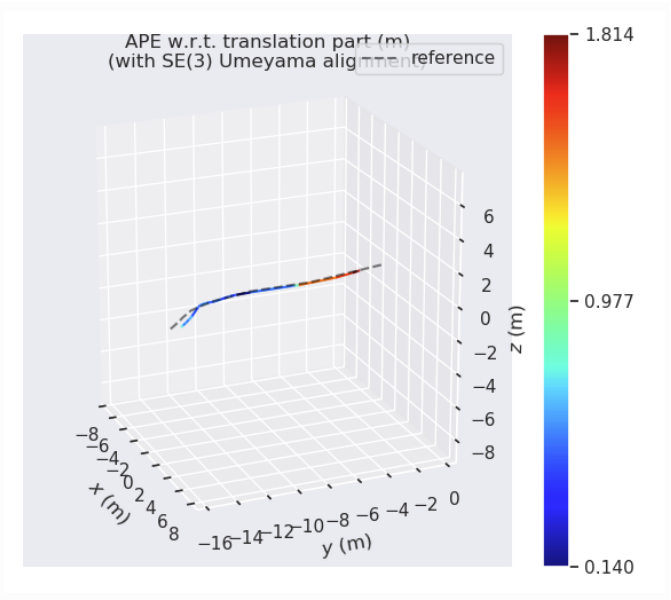

Fig. 5. Overlaid trajectories for cone pairings 1-2 compared against sudo ground truth.

TABLE I
ORBSLAM RESULTS FROM UNDERWATER CAVES DATASET. INCLUDES CONE PAIRINGS, CONE DISTANCE, SUDO GROUND TRUTH DISTANCE, % OF TRAJECTORY COMPLETED, MEAN APE, MEAN RPE, AND PREDICTED TRAJECTORY DISTANCE.

ORBSLAM 3 Testing						
Cone Pairings	Cone Dist (m)	Sudo GT Dist (m)	%	APE	RPE	Dist (m)
1 - 2	19	19.43	100	0.736	0.608	15.761
2 - 3	32	32.24	30	0.732	0.323	8.52
3 - 4	16	13.38	100	0.229	0.366	11.98
4 - 5	-	39.08	47.5	1.243	0.340	9.98
5 - 5	-	181.21	1.5	-	-	-
5 - 4	-	35.12	100	11.71	0.596	88.7
4 - 3	16	18.52	35	0.096	0.219	4.04
3 - 2	32	32.05	39.1	1.001	0.417	11.81
2 - 6	11	15.14	100	0.233	0.337	13.425
6 - 6	-	104.03	14.9	0.231	0.341	16.526
6 - 1	30	32.49	68	2.653	0.389	33.60

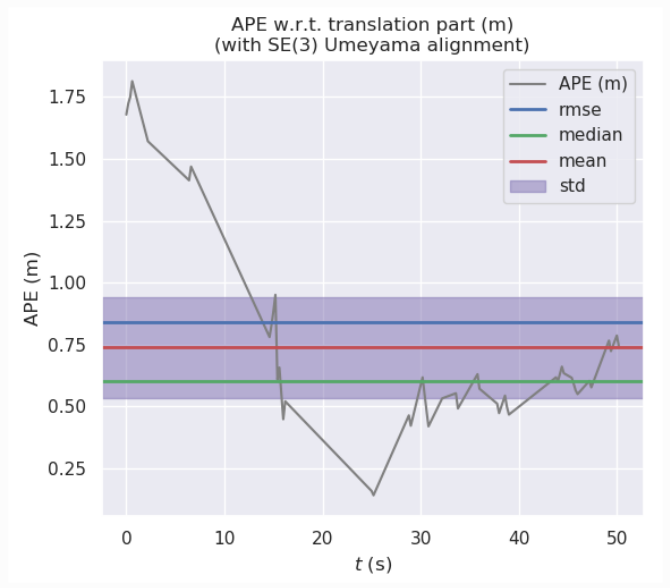

Fig. 6. APE graph for cone pairings 1-2.

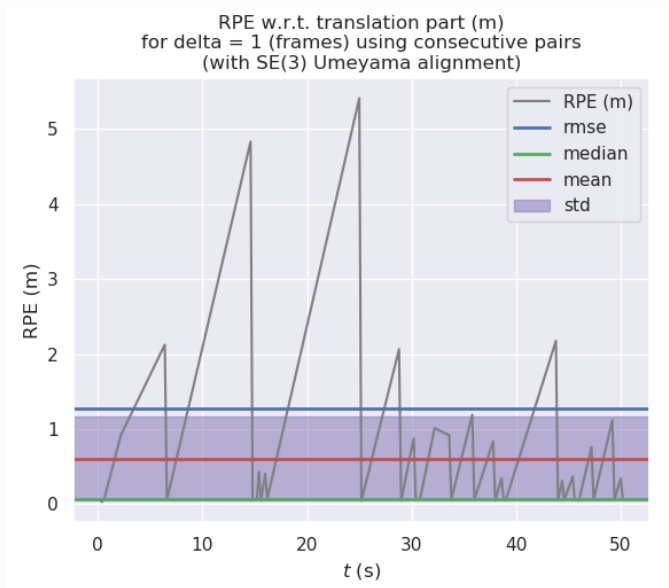

Fig. 7. RPE graph for cone pairings 1-2.

error anywhere along the path will cause the APE to be much higher for the rest of the path than compared to the RPE.

Lastly, looking at the predicted distances they are mostly inconsistent. This comes from many of the paths not being able to finish meaning that the distance was cut short. It is also interesting to see that the APE and predicted distance seem to have a positive correlation, but with this limited amount of data, it can not be directly proven.

C. ORBSLAM3 Mono-inertial

When testing ORBSLAM3 Mono-inertial, many issues came up with the fusion of IMU and visual data. The main issue was with initializing the IMU. When ORBSLAM starts up, it needs time to initialize the IMU data to account for IMU drift. For real situations, this is not an issue since the camera can be held steady while ORBSLAM initializes the IMU data. When testing on the underwater dataset, the AUV is already moving at the start of the rosbag provided, not giving enough time for ORBSLAM to initialize the IMU.

Along with this, the dataset did not provide any calibration information from camera to IMU reference frame. They do provide the axis' and translation specs, however, there may be slight inconsistencies with this and the real-world rotations and distances. The dataset also does not provide any information on the specs of the IMU used, resulting in having to use the data from the datasheet of the IMU. These issues could have been avoided if IMU to vision calibration was provided, such as Kalibr [18].

D. SVIn2

When testing with SVIn2, the same issues that ORBSLAM Mono-inertial suffered from were seen. These two main issues of initialization and lack of calibration made inertial SLAM data unusable for this dataset, so evaluation could not be done with these methods. In figure 8, you can see the exploding trajectory resulting from SVIn2 running on the dataset.

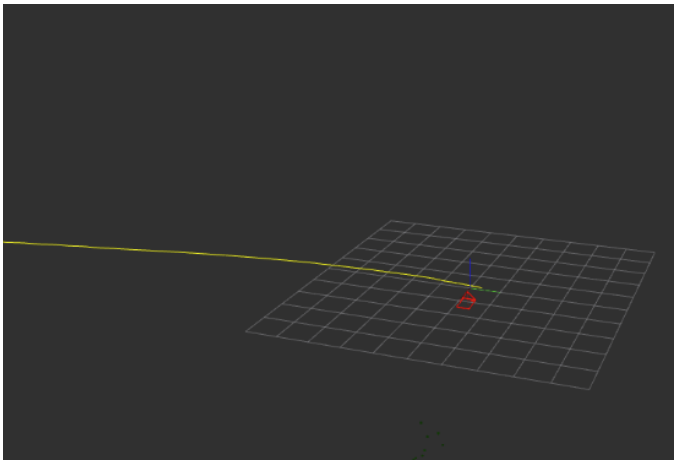

Fig. 8. Trajectory from SVIn2 running on underwater caves dataset.

E. Limitations

The main areas ORBSLAM failed were areas where attenuation made it extremely hard to get a clear image of the ground, and when there was little objects in the scene to provide features. The lack of features and similarities of the scene caused ORBSLAM to struggle to localize and get an accurate map leading to drifts or loss of tracking. Once loss of tracking occurred ORBSLAM struggled to relocalize with the lack of uniqueness in the underwater caves. Some example photos of images ORBSLAM struggled with can be seen in 9.

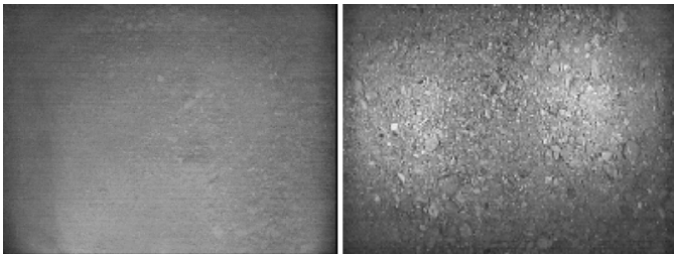

Fig. 9. Two images showcasing where ORBSLAM failed. Both images lack contrast which makes ORB feature detection extremely hard.

Another big issue with underwater caves is the lack of looping in the dataset and the size of the dataset. While some cone pairings were close together, some were extremely far away. The issue with this is that a loss of tracking anywhere along the path would ruin the tracking for the entire trajectory. These large gaps gave more time for a loss of tracking, making it difficult to evaluate these trajectories along the underwater cave.

V. Conclusions & Further Work

Throughout this paper, benchmarking of ORBSLAM 3 and SVIn2 was performed on a real-world AUV dataset against the sudo ground truth provided by the authors of the dataset. From the results shown, it is clear that current applications of visual SLAM struggle when faced with an underwater dataset. This is due to the lack of features provided in the underwater environment along with the attenuation and backscattering. Along with this, inconsistencies with calibration from the dataset can have drastic impacts on the result of visual-inertial SLAM methods.

There are several further directions for future work to expand upon this paper. One of which would be to display more visual SLAM methods for seeing the differences in feature detection. Another would be to use other datasets. This dataset had many limitations such as the lack of ground truth, low resolution camera, and issues with the calibration of the inertial sensors. Lastly, it would be useful to test stereo cameras since they provide more information than a single monocular camera.

Even though visual SLAM struggles in the underwater environment, there are key takeaways that could provide better tracking to aid visual SLAM in an underwater environment. The first takeaway is image processing for feature detection. The underwater effects drastically decrease the quality of camera imaging, so new developments need to be made in underwater imaging before feature detection becomes a reliable system for visual SLAM. One concept that could be used is deep learning, which is seen in systems such as UGAN [19] and WaterGAN [20]. These new methods could support feature detection but do fail in temporal conditions. New developments need to be made to have consistent image enhancement throughout a video stream to keep feature detection effective. Along with this, some work could be done with deep learning feature-detection, however, there are many issues with training these networks with the lack of data on the underwater environment.

References

[1] R. S. R, A. Sungheetha, and D. C. R, "Revolutionizing underwater exploration of autonomous underwater vehicles (auvs) and seabed image processing techniques," Nov. 2023. arXiv:2402.00004 [cs].

[2] B. Joshi, M. Xanthidis, M. Roznere, N. J. Burgdorfer, P. Mordohai, A. Q. Li, and I. Rekleitis, "Underwater exploration and mapping," in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, p. 1–7, Sept. 2022.

[3] M. R. Bariq, R. Darmakusuma, and R. Yusuf, "Underwater scanning using lidar," in *2023 International Conference on Electrical Engineering and Informatics (ICEEI)*, p. 1–5, Oct. 2023.

[4] R. Meyer, "Gps doesn't work underwater," June 2016.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, p. 1147–1163, Oct. 2015.

[6] S. Rahman, A. Q. Li, and I. Rekleitis, "Svin2: An underwater slam system using sonar, visual, inertial, and depth sensor," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 1861–1868, Nov. 2019. arXiv:1810.03200 [cs].

[7] M. Bryson, M. Johnson-Roberson, O. Pizarro, and S. B. Williams, "True color correction of autonomous underwater vehicle imagery," *Journal of Field Robotics*, vol. 33, p. 853–874, Sept. 2016.

[8] R. Kaneko, H. Higashi, and Y. Tanaka, "Physics-inspired synthesized underwater image dataset," Apr. 2024. arXiv:2404.03998 [cs, eess].

[9] Y. Wang, J. Guo, W. He, H. Gao, H. Yue, Z. Zhang, and C. Li, "Is underwater image enhancement all object detectors need?," Nov. 2023. arXiv:2311.18814 [cs].

[10] "Underwater caves sonar data set - Angelos Mallios, Eduard Vidal, Ricard Campos, Marc Carreras, 2017." https://journals-sagepub-com.proxy.lib.umich.edu/doi/10.1177/0278364917732838.

[11] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2d2: Repeatable and reliable detector and descriptor," June 2019. arXiv:1906.06195 [cs].

[12] D. Prokhorov, D. Zhukov, O. Barinova, K. Anton, and A. Vorontsova, "Measuring robustness of visual slam," in *2019 16th International Conference on Machine Vision Applications (MVA)*, p. 1–6, May 2019.

[13] D. Tyagi, "Introduction to feature detection and matching," Apr. 2020.

[14] H. M. S. Bruno and E. L. Colombini, "Lift-slam: a deep-learning feature-based monocular visual slam method," *Neurocomputing*, vol. 455, p. 97–110, Sept. 2021. arXiv:2104.00099 [cs].

[15] D. Tyagi, "Introduction to orb (oriented fast and rotated brief)," Apr. 2020.

[16] 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 multi-map slam," *IEEE Transactions on Robotics*, vol. 37, p. 1874–1890, Dec. 2021. arXiv:2007.11898 [cs].

[17] 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, p. 314–334, Mar. 2015.

[18] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, p. 4304–4311, May 2016.

[19] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing underwater imagery using generative adversarial networks," Jan. 2018. arXiv:1801.04011 [cs].

[20] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, "Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images," *IEEE Robotics and Automation Letters*, p. 1–1, 2017. arXiv:1702.07392 [cs].