2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

An Open-Source Solution for Fast and Accurate Underwater Mapping with a Low-Cost Mechanical Scanning Sonar

Tim Hansen and Andreas Birk[1]

Abstract— **An open-source software framework is presented that allows real-time underwater mapping with popular marine robotics components, namely a BlueRobotics BlueROV2 with its standard Ping360 Mechanical Scanning Sonar (MSS) and a A50 Doppler Velocity Log (DVL), which are low-cost devices for their respective types - if not even the most affordable ones on the market. The software runs with low computational power on a Raspberry Pi4. The framework builds upon Synthetic Scan Formation (SSF) where single MSS beams or scan-lines are embedded into a pose-graph. The rendering of scans is not only based on navigation, but based on the graph itself. Scans formed from scan-lines can be optimized by online Simultaneous Localization and Mapping (SLAM) and result in improved scans, based on the current state of the graph. In subsequent steps this leads to improved registration results. To this end, a combination of two different types of loop-closures is presented. Namely a consecutive loop closure, and a proximity based loop closure, which reduces the overall drift. The framework is validated in three different test-environments, namely a pool, a test-tank with a gantry for ground truth motion, and the flooded basement of a WW-II submarine bunker. Among others, it is shown that there is an increased accuracy compared to conventional SLAM and that the software is usable in real-time during a mission with the low-cost hardware.**

I. INTRODUCTION

There has always been a strong tradition and interest in low-cost solutions in robotics. For marine robotics, this has even substantially increased in recent years, among others due to the availability of commercial open-source devices, e.g., in form of the BlueRobotics BlueROV2 and its related components [1]. Also in academic research, there is a body of work on hardware related aspects of low-cost marine robots [2], [3], [4], [5], [6], [7], [8], [9], [10].

With respect to the software side of low-cost marine robotics, there is for example research on navigation [11], [12], [13], [14], [15], [16], control [17], [18], [19], or diver tracking [20], [21], [22]. Vision plays an important role in many of these approaches. This includes also generic research on low-cost underwater vision like the estimation of depth-information from monocular images [23], which can be used in a variety of applications.

But while cameras tend to be relatively low-cost, there are many use-cases for marine robots where underwater visibility is poor. Sonar [24], [25], is a well-established alternative, but many devices, respectively device types are not only much more expensive than vision, but they also require payload- and energy-capabilities that are beyond typical low-cost marine robots. Somewhat an exception are Mechanically Scanning Sonars (MSS), which as the name suggests mechanically rotate a single sonar beam. Despite their simplicity, they can be very useful in low visibility environments like harbors or marinas [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38]. Our interest in the topic stems from the exploration of the flooded basement of a WW-II submarine bunker (Fig.1) [39], [40] in the context of the digitization of cultural heritage [41].

The work presented here uses a BlueRobotics Ping360 MSS, a WaterLinked A50 Doppler Velocity Log (DVL), and a BlueROV2, which are among the - if not the - most affordable devices of their respective type on the market. Like with low-cost robotics in general, there are also two main aspects to be considered. First, there are the sensors themselves with their limitations in terms of, e.g., spatial and temporal resolution as well as with respect to the noise level compared to high-end devices. Second, there are computational resource limitations, namely those of a Raspberry Pi 4 that can be easily integrated on a BlueROV2. Note in this context that onboard processing is not only essential for Autonomous Underwater Vehicles (AUV). It is also of interest for Unmanned Underwater Vehicles (UUV) in general, as there can be limitations in communications bandwidths for even very high-end Remotely Operated Vehicles (ROV) [42], [43].

The Ping360 is used to generate maps with Synthetic Scan Formation (SSF) [44], which operates on a level between conventional Synthetic Aperture Sonar [45], [46], [47], [48] and Simultaneous Localization and Mapping (SLAM) with MSS [39], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38]. The core idea of SSF is to use pose-estimates for each single beam or scan-line to form synthetic scans. Instead of just using core navigation to combine multiple scan-lines into each scan once, SSF uses online SLAM to update the pose-estimates of the scan-lines and new, improved scans can be (re-)rendered. Improved scans lead to improved registration results in the subsequent SLAM processing and they hence lead to an overall improved map quality.

The full code of the low-cost implementation of underwater mapping with SSF, in ROS2, is freely available at `https://github.com/constructor-robotics/underwaterSLAM_2D_lowCost`.

The rest of this paper is structured as follows. Sec.II provides a short overview of Synthetic Scan Formation

*This work was supported by the Deutsche Forschung Gemeinschaft (DFG) in the project "Unconstrained Synthetic Aperture Sonar (U-SAS)"

[1]Tim Hansen and Andreas Birk are with the School of Computer Science and Engineering, Constructor University, 28759 Bremen, Germany thansen@constructor.university, abirk@constructor.university

Fig. 1: The WW-II submarine bunker Valentin that is investigated within the context of the digitization of cultural heritage (top). In the flooded basement that is of historical interest, there is very low to no visibility, which makes the use of sonar necessary for orientation during a mission (bottom).

(SSF). The implementation of our software framework is described in Sec.III. This includes the specific methods for implementing SSF (Sec.III-A), the loop closures strategies that play an important role from the perspective of this paper (Sec.III-B), and a short overview of the structure of the software package (Sec.III-C). Experiments and results from three different test environments are presented in Sec.IV. Sec.V concludes the paper and shortly discusses use-cases as well as possible modifications for future work.

II. Synthetic Scan Formation

Our open source code for MSS scan integration and mapping builds upon Synthetic Scan Formation (SSF), which is shortly described here. A detailed presentation can be found in [44].

The following notations are used. A scan-line sl is a vector $I[i_{tof}]$ of the return intensities measured by a single beam. The index i_{tof} corresponds to the time-of-flight, i.e., it determines the distance d to the point where the back-scatter originated. A MSS mechanically rotates the beam, i.e., the scan-line $sl(\gamma)$ from a minimum angle γ_{min} to a maximum angle γ_{max} with a step-width $\Delta\gamma$. The three parameters γ_{min}, γ_{max}, $\Delta\gamma$ can typically be set by the user.

A scan sc is simply a collection of k consecutive scan-lines sl_i. Given a sonar at a fixed location, a scan sc forms a polar image $I[\gamma][d]$ of the sonar's surroundings. But on a moving platform, motion-compensation has to be used to avoid distortions. This means that the 2D pose $p_s(t) = [x_s(t), y_s(t), \phi_s(t)]^T$ of the sensor at time-step t needs to be taken into account, which depends in a fixed and known way on the 2D vehicle pose $p(t) = [x(t), y(t), \phi(t)]^T$. For the sake of simplicity, we just refer to the vehicle pose $p(t)$ in the following.

The state-of-the-art is to use the pose-estimate $p(t)$ from the core navigation with dead-reckoning to estimate the frame $\mathcal{F}_t^{sl}$ of each sl_t with $\mathcal{F}_t^{sl} = [x(t), y(t), \phi(t) + \gamma(t)]^T$, where the rotation is the addition of rotation from the robot and the sonar. To form a scan $sc(t_i)$, each sl_t with $t_i \leq t \leq t_{i+k}$ is projected into the frame $\mathcal{F}_{t_{i+k}}^{sl}$ at the end of the scan. Given scan $sc(t_i)$, it can be entered into a pose graph at node v_i and processed for a loop closure with the help of a registration.

Contrary to the state-of-the-art, in SSF, each scan-line sl_t is entered into a node v_j together with its frame F_t^{sl}. At first glance, there is no benefit in this as the individual scan-lines do not contain enough information for registration and a scan formation still needs to be used. The major advantage is that the frames F_i^{sl} associated with each sl_i can get optimized by online SLAM, i.e., the quality of scans formed from them improves.

The trade-off is that the computation effort increases. Before a scan can be used for registration in a loop-closure, it needs to be newly rendered when one of the pose-estimates of its scan-lines has been optimized. But this leads to improved results and the increase in time is minor [44].

Note that SSF generalizes to any form of graph-SLAM, i.e., it is independent of the back-end, and to any registration method, i.e., it can be used with other forms of registration than the one used here; this also includes methods using, e.g., point-clouds or features extracted from the scans.

III. Implementation

A. Methods for the Scan Formation

The implementation of the synthetic scan formation is based on factor graphs [49] that are optimized with iSAM2 [50] from the GTsam library [51]. In the front-end, Fourier-SOFT in 2D (FS2D) is used, which is a spectral registration method that is well suited for low-quality, noisy sonar scans [52]. A grid size of $N = 128$ is chosen for the registration with FS2D. For SLAM, FS2D needs to be augmented with an uncertainty estimation. FS2D operates in the frequency domain where the different signal processing methods used in there generate in theory Dirac pulses for each degree of freedom (dof). In reality with imperfect data, a peak detection is needed to determine the registration results. Here, the 0-th dimensional persistent homology for 2D images [53] is used to this end. As in [54], a Principal Components Analysis (PCA) in a circle around each peak is used to determine a co-variance matrix.

The core navigation is based on the standard approach for underwater dead-reckoning [55], [56], [57]. An Extended Kalman Filter (EKF) estimates the current pose of the robot $\boldsymbol{x}_{ekf} = [x, y, z, \psi, \theta, \phi]^\top$ using the input from a Doppler Velocity Log (DVL) and an Inertial Measurement Unit (IMU), namely an A50 [58] and an Xsens MTi-300 [59].

Due to the relatively slow speed of the BlueROV2, the g vector can be used for the estimation of roll and pitch with the IMU. The state variables $[x, y, \phi]$ are open loop estimations, while the others can be measured by the available sensors. The z value is provided by the default pressure sensor of the BlueROV2.

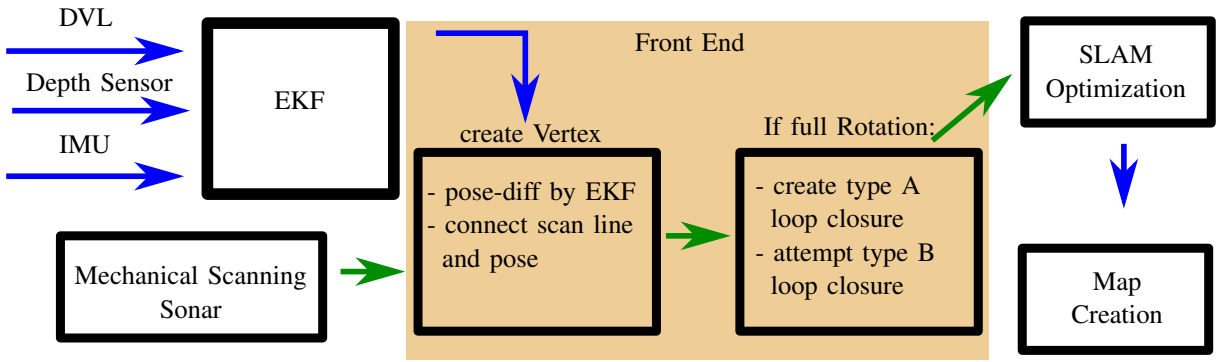


Fig. 2: An overview of the components of the software framework. The blue arrows indicate an asynchronous data stream, while the green arrows synchronizes the computation in the framework.

B. Loop Closures

Loop closures are essential for SLAM in general and for SSF in particular. Many different ways of constructing a loop closure with the graph formulation described above are possible. We use here two different kinds of loop closures.

First, a loop closure based on the recently recorded scan lines is used, which is denoted here as type A loop closure. Second, a proximity based loop closure with previously generated synthetic scans is used, which we denote as type B loop closure.

Type A loop closures are designed to reduce the error of the dead-reckoning system as motivated as follows. Assume a new scan line is recorded and we want to create a loop closure. The goal is to cover a large area with the scan lines and to have a good quality 2D sonar image rendered from the scan-lines. Hence, the rotation of the scan lines combined with the rotation of the nodes should cover a full circle, to maximize the area. For the type A loop closure, the strategy is used that the node rotation and the scan line rotation together should reach 2π to trigger the rendering of a new scan (image), which is registered with the previous one. This means that k scan-lines sl_i are used for rendering a scan whenever the sl_i cover a 360^o field of view. Therefore, the type A loop closure is a kind of local loop closure. The second scan for registration is here also generated every 2π. The frequency, how often the type A loop closure is computed can be chosen. For example, we can generate a type A loop closure each half scan of the environment, which results in two for each full 2π scan. Depending on the speed of the vehicle, or the registration algorithm used and its robustness, etc., various other options that generate more or less overlap between scans could be used.

The type B loop closure is simply a proximity based strategy on the so to say global map level. When the robot is close to an earlier visited location in the graph, the scan-lines that were last entered into the graph and that together cover 2π are used to render a scan sc^{now}. The second scan sc^{past} is rendered based on the vertex v^{past} in the proximity of the currently visited location. To this end, all scan-lines sl_i^{past-} before, respectively sl_i^{past+} after v_{past} are used that respectively cover 1π. Every time a loop closure of type A is added to a graph, a type B loop closure is also attempted. In order to create a type B loop closure, the earlier visited location has to be nearby, which is defined by a constant radius r_{lc}, i.e., based on proximity.

C. Framework Structure

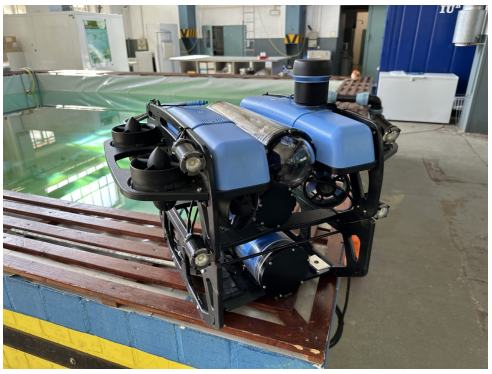
The software framework is using the Robotic Operating System (ROS2) for the communication between each component. RViz is used for visualization. The software is split into three main parts. First, there is the Extended Kalman Filter (EKF) state estimation. Second, the front-end including Fourier-SOFT in 2D (FS2D) registration. Third, there is the back-end with the iSAM2 optimization and map rendering.

In Fig. 2 an overview of the software package is depicted. The blue arrows indicate that the signal is received asynchronously while the green arrow indicates a synchronized data stream. The design of the software is straight forward. Every time a new measurement by the EKF arrives, the measurement is appended to a list, which is used when a new scan line arrives. A new node is created, and the scan line is added. Thereafter, if the observed rotation is greater than 2π, the loop closures are attempted, and the optimization is performed in the end.

IV. Experiments and Results

Three different test environments (Fig.3) are used to evaluate our software framework. Fig.3a shows the BlueROV2 vehicle and first test environment in form of a simple pool in the Oceanlab of Constructor University.

The second environment is quite interesting as it provides ground truth motion data, which is a major challenge for underwater robotics in general. Fig.3b shows a test-tank at the Institute of Mechanics and Ocean Engineering of the Technical University of Hamburg(TUHH), which has a gantry mechanism that can be used as a so-to-say propulsion system. The Computerized Numerical Control (CNC) of the gantry provides accurate motions and highly precise localization, thus allowing a quantitative analysis. The downside is that the tank is only 2 m wide and 4 m long.



(a) Constructor Ocean Lab

(b) TUHH Test-Tank with Gantry

(c) Access to Valentin Basement

Fig. 3: An overview of the three test environments.

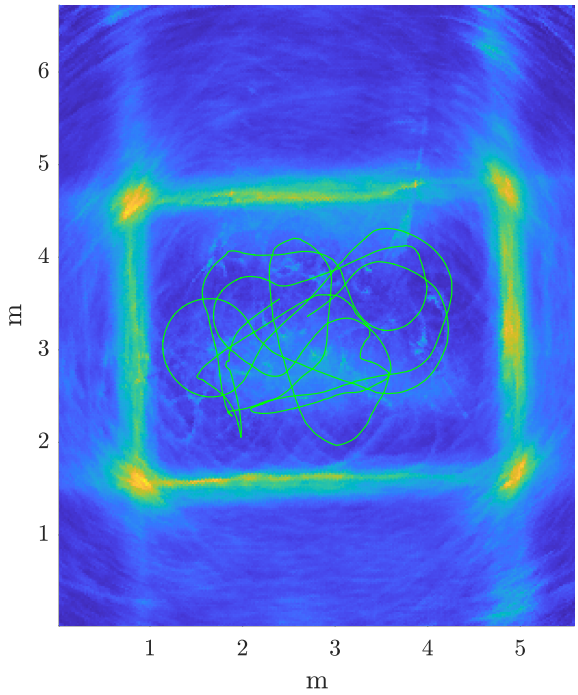


Fig. 4: Map generated in the pool in the Constructor Ocean Lab. The estimated path is shown in green.

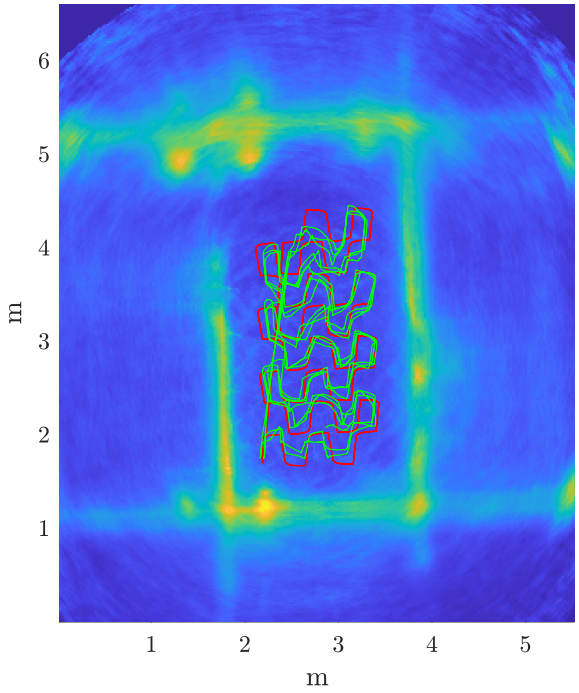


Fig. 5: Map generated in the TUHH test-tank with a gantry for ground truth motion. The very challenging path with abrupt changes in direction is shown in red.

The entrance to the basement of the WW-II submarine bunker Valentin is shown in Fig.3c. This environment is a typical real use-case where the generation of maps with sonar is important for exploration due to the low visibility (Fig.1).

As can be seen in Fig.'s 4, 5 and 6, our software produces reasonable maps in all three environments. The Oceanlab pool is simply a rectangular basin that is correctly represented. For the TUHH test-tank, a window (upper left corner) and some structures inside the tank are well captured. The map of the bunker basement shows exits and corridors at both ends as well as some brick-made wall-fragments inside. In addition to the maps as renderings of the accumulated scan images, the estimated paths are shown in green.

As the ground truth motion is available for the TUHH test-tank, the related true path is also shown in red in Fig. 5. The motion in this environment is quite challenging as it forms a rectangular zig-zag pattern with very abrupt changes in the direction of the motion. The core navigation with the EKF as part of our software smooths these corners, which leads to the estimated path shown in green.

TABLE I: Accuracy of our framework with Synthetic Scan Formation (SSF) in comparison to core navigation with the EKF (nav) and a state-of-the-art SLAM (sota SLAM).

error (mean $\pm$std) as L2-norm (m)		
nav	sota SLAM	SSF (ours)
0.3836 ± 0.2155	0.1449 ± 0.1002	0.0959 ± 0.0698

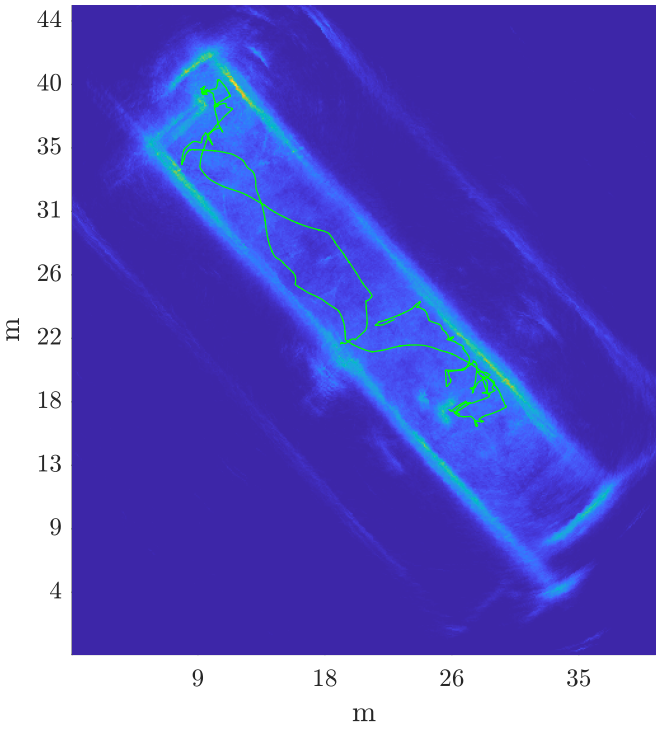


Fig. 6: Map generated in the basement of the submarine bunker Valentin.

More importantly, the ground truth allows a quantitative analysis of the accuracy of our framework. Standard core navigation with the EKF and a state-of-the-art SLAM [39] without synthetic scan formation are used for comparison. All three methods are provided with the same input data. The results are shown in Tab.I. The errors are computed via the Euclidean distances of the estimated locations to the accurate ground truth for each pose.

As can be expected, navigation with just the EKF performs worst with substantial errors that render it useless for mapping. SLAM substantially improves the situation. A further clear improvement is achieved with our approach using synthetic scan formation.

Further important results from the perspective of this paper are shown in Fig.7. The graphs in the sub-figures show the time it takes (y-axis, in sec) to acquire the sonar data for a full sonar scan (blue line) and to process a scan (brown line) over the duration of each mission (x-axis, in min).

While the speed of sound is slow and the maximum range has hence a strong influence on the time it takes to acquire a single scan-line, other settings of the sonar also influence the scanning speed. One way to acquire a faster scanning speed is to increase the mechanical stepping angle of each measured scan line. Furthermore, the size of the intensity array measured by the sonar can be set, which also also influences the scanning speed.

This is illustrated in Fig.'s 7a and 7b where two different stepping angle settings are used in the Oceanlab pool, namely 0.9° and 4.5° combined with two different intensity array sizes of 500 and 200, respectively. With the fast scanning speed setting, the average time to acquire the data for a scan is about 9 sec while it is about 32 sec for the slow scanning speed setting. The 9 sec are an example for a kind of lower bound for the scan acquisition time if a coarse coverage of a small environment is sufficient, while the 32 sec are an example for the typical settings for exploring a medium sized-environment.

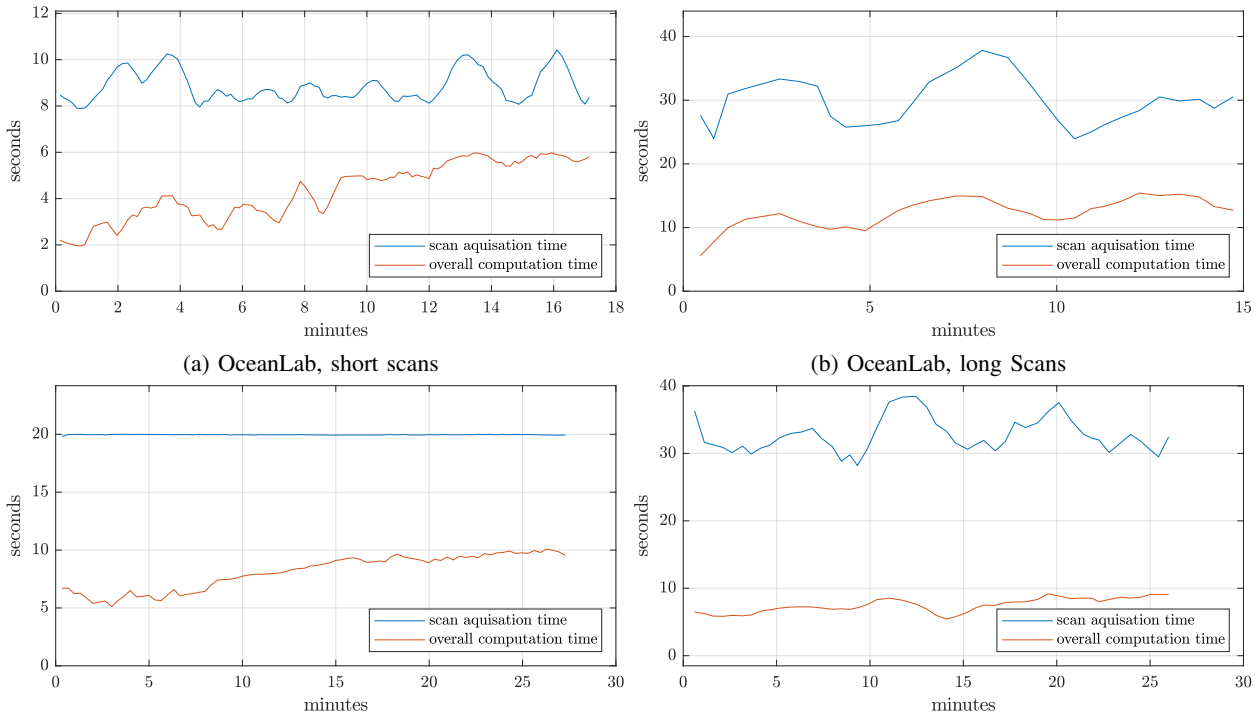
Note that the time to acquire a full scan fluctuates due to the strategy used here to render a scan based on a collection of scan-lines that cover 360^o, i.e., the combined rotation of the mechanically actuated sonar beam and of the vehicle on which the sensor is mounted on should be 2π. Therefore it depends directly of the movement of the robot during the mission. The time to acquire a scan is hence constant in the experiment in the TUHH test-tank (Fig.7c) as the gantry only provides translations as ground truth motions, i.e., there is no rotation from the "vehicle" in this case.

The most important insight is in all four cases shown in Fig.7 is that the time for processing (brown line) is substantially shorter than the update rate with which the sensor can provide new data (blue line). So, the processing can be done in real-time despite the computational limitations of a Raspberry Pi 4. The difference in computation times in different scenarios also depend on the other settings of the sonar. When larger amounts of data is returned by the sonar measurement, namely the intensity array size, the rendering process of a full sonar scan increases. This can be seen between the two different figures of the OceanLab computation times.

Nonetheless, there is also an increase in computation time over the mission duration. This is due to the fact that over time the graph size increases and more type A and type B loop closures are created. This increases both the optimization time from iSAM2 and the rendering computational overload of different parts in the algorithm.

Fig.8 provides a bit of insight into the computational requirements of the different parts of the software. The average computation time is differentiated into four parts, namely the optimization time of iSAM2, the two types of loop closures, and the rendering process with its computational overhead. The respective processing times are averaged over four minute time windows in the mission time and their distribution at the end of each window is shown. It can be seen that the average time of the type B loop closure depends heavily on the trajectory of the robot. For example, the average computation time between minute 13 and 17 for that software part is much smaller than for the other times of the mission. This window corresponds to a part of the mission where the robot is moving into areas it has not visited before and where it therefore can not successfully attempt any type B loop closures.

This can for example be exploited in a exploration-exploitation trade-off strategy when using the software framework in third party experiments or missions that are intended to run over multiple hours or even days. In general, there are vast opportunities to use the presented code as a basis for third party experiments and systems.



(a) OceanLab, short scans

(b) OceanLab, long Scans

(c) TUHH test-tank

(d) Bunker Valentin basement

Fig. 7: The time it takes (y-axis, in *sec*) to acquire the sonar data for a full sonar scan (blue line) and to fully process a scan (brown line) over the duration of each mission (x-axis, in *min*) for the three scenarios, respectively two different parameter settings in the Oceanlab pool. In all cases, the processing can be done in real-time.

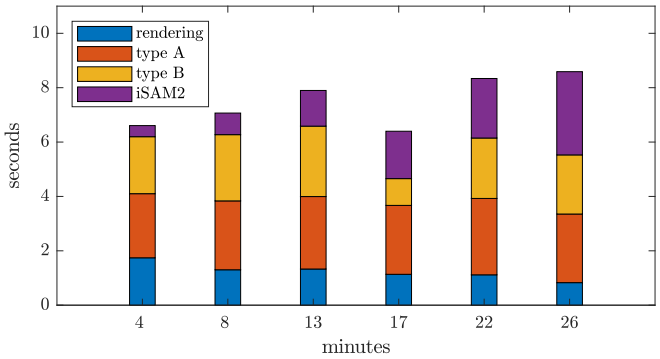


Fig. 8: The development of the distribution of the processing time (y-axis, in *sec*) over mission time-windows (y-axis, in *min*) for the different software components in the bunker Valentin experiment.

V. Conclusion and Future Work

We presented an open-source framework for synthetic scan formation on popular low-cost hardware in form of a BlueROV2 with a Raspberry Pi4 and its standard sonar and doppler velocity log (DVL), i.e., a Ping360 and a A50, which are all also popular components for the development of low-cost AUVs. Despite the resource-constraints and the limitations in sensor quality, the open-source code can be used as it is for a wide-range of functions for operator-assistance and semi- as well as full-autonomy. More precisely, the synthetic scan formation performs in real-time and it can be used for tasks including navigation, obstacle-avoidance, object-recognition, and environment classification. Furthermore, the code provides as it is high quality mapping in real-time over extended time periods well above mission durations of 20 to 30 minutes.

In addition to using the code in the variety of use-cases mentioned above, there are further options for future work. Though being likely the least costly available DVL on the market, the A50 is nonetheless the most expensive single component of our set-up. It can hence be of interest to study the option to replace it with a hydrodynamic model of the vehicle. Furthermore, the synthetic scan formation as used now provides high quality mapping in real-time, but with increasing overall computation time due to the increasing optimization effort in the iSAM2 back-end. There are multiple options to address this for extended mission times. These include the use of more sophisticated loop-closure strategies or the use of hierarchical mapping. Last but not least, synthetic scan formation is agnostic to the SLAM back-end as well as to the registration method used within the front-end. It is hence an option to investigate the performance of alternatives to iSAM2 or SF2D.

Acknowledgment

We thank the Institute of Mechanics and Ocean Engineering of the Technical University of Hamburg for giving us access to their research pool with the gantry.

REFERENCES

[1] BlueRobotics, "Bluerov2," 2023. [Online]. Available: https://bluerobotics.com/store/rov/bluerov2/

[2] K. Xue, X. Ji, D. Qu, Y. Peng, and H. Qian, "Oboat: An agile omnidirectional robotic platform for unmanned surface vehicle tasks," *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2023.

[3] Y. Zhang, R. Zhu, J. Wu, and H. Wang, "Simobot: An underactuated miniature robot driven by a single motor," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5748–5759, 2022.

[4] Z. Yang, D. Chen, D. J. Levine, and C. Sung, "Origami-inspired robot that swims via jet propulsion," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7145–7152, 2021.

[5] G. Knizhnik, P. deZonia, and M. Yim, "Pauses provide effective control for an underactuated oscillating swimming robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5075–5080, 2020.

[6] F. Berlinger, J. Dusek, M. Gauci, and R. Nagpal, "Robust maneuverability of a miniature, low-cost underwater robot using multiple fin actuation," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 140–147, 2018.

[7] B. R. Page, S. Ziaeefard, A. J. Pinar, and N. Mahmoudian, "Highly maneuverable low-cost underwater glider: Design and development," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 344–349, 2017.

[8] J. Higgins and C. Detweiler, "The waterbug sub-surface sampler: Design, control and analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, Conference Proceedings, pp. 330–337.

[9] M. Makrodimitris, I. Aliprantis, and E. Papadopoulos, "Design and implementation of a low cost, pump-based, depth control of a small robotic fish," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, Conference Proceedings, pp. 1127–1132.

[10] A. Caffaz, A. Caiti, G. Casalino, and A. Turetta, "The hybrid glider/auv folaga," *IEEE Robotics and Automation Magazine*, vol. 17, no. 1, pp. 31–44, 2010.

[11] Y. Wang, R. Hu, S. H. Huang, Z. Wang, P. Du, W. Yang, and Y. Chen, "Passive inverted ultra-short baseline positioning for a disc-shaped autonomous underwater vehicle: Design and field experiments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6942–6949, 2022.

[12] S. Yan, Z. Wu, J. Wang, M. Tan, and J. Yu, "Marine autonomous navigation for biomimetic underwater robots based on deep stereo attention network," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, Conference Proceedings, pp. 8418–8423.

[13] E. M. Fischell, N. R. Rypkema, and H. Schmidt, "Relative autonomy and navigation for command and control of low-cost autonomous underwater vehicles," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1800–1806, 2019.

[14] J. Eisele, Z. Song, K. Nelson, and K. Mohseni, "Visual-inertial guidance with a plenoptic camera for autonomous underwater vehicles," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2777–2784, 2019.

[15] N. R. Rypkema, E. M. Fischel, and H. Schmidt, "Closed-loop single-beacon passive acoustic navigation for low-cost autonomous underwater vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, Conference Proceedings, pp. 641–648.

[16] S. Wirth, P. L. N. Carrasco, and G. O. Codina, "Visual odometry for autonomous underwater vehicles," in *MTS/IEEE OCEANS*. IEEE, 2013, Conference Proceedings, pp. 1–6.

[17] G. Knizhnik, P. Li, X. Yu, and M. A. Hsieh, "Flow-based control of marine robots in gyre-like environments," in *International Conference on Robotics and Automation (ICRA)*, 2022, Conference Proceedings, pp. 3047–3053.

[18] D. A. Duecker, N. Bauschmann, T. Hansen, E. Kreuzer, and R. Seifried, "Towards micro robot hydrobatics: Vision-based guidance, navigation, and control for agile underwater vehicles in confined environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, Conference Proceedings, pp. 1819–1826.

[19] E. M. Fischell, A. R. Kroo, and B. W. O'Neill, "Single-hydrophone low-cost underwater vehicle swarming," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 354–361, 2020.

[20] A. G. Chavez, A. Ranieri, D. Chiarella, and A. Birk, "Underwater vision-based gesture recognition: A robustness validation for safe human-robot interaction," *IEEE Robotics and Automation Magazine (RAM)*, vol. 28, no. 3, pp. 67–78, 2021.

[21] H. M. Chou, Y. C. Chou, and H. H. Chen, "Development of a monocular vision deep learning-based auv diver-following control system," in *IEEE/MTS Global Oceans*. IEEE, 2020, Conference Proceedings, pp. 1–4.

[22] W. Remmas, A. Chemori, and M. Kruusmaa, "Diver tracking in open waters: A low-cost approach based on visual and acoustic sensor fusion," *Journal of Field Robotics*, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21999

[23] B. Yu, J. Wu, and M. J. Islam, "Udepth: Fast monocular depth estimation for visually-guided underwater robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, Conference Proceedings, pp. 3116–3123.

[24] J. P. Marage and Y. Mori, *Sonar and Underwater Acoustics*. Wiley, 2013.

[25] R. P. Hodges, *Underwater Acoustics: Analysis, Design and Performance of Sonar*. Wiley, 2010.

[26] Z. Xu, H. Qiu, M. Dong, H. Wang, and C. Wang, "Underwater simultaneous localization and mapping based on 2d-slam framework," in *IEEE International Conference on Unmanned Systems (ICUS)*, 2022, Conference Proceedings, pp. 184–189.

[27] L. Chen, A. Yang, H. Hu, and W. Naeem, "Rbpf-msis: Toward rao-blackwellized particle filter slam for autonomous underwater vehicle with slow mechanical scanning imaging sonar," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3301–3312, 2020.

[28] M. Jiang, S. Song, J. M. Herrmann, J. H. Li, Y. Li, Z. Hu, Z. Li, J. Liu, S. Li, and X. Feng, "Underwater loop-closure detection for mechanical scanning imaging sonar by filtering the similarity matrix with probability hypothesis density filter," *IEEE Access*, vol. 7, pp. 166 614–166 628, 2019.

[29] M. Jiang, S. Song, F. Tang, Y. Li, J. Liu, and X. Feng, "Scan registration for underwater mechanical scanning imaging sonar using symmetrical kullback-leibler divergence," *Journal of Electronic Imaging*, vol. 28, no. 1, 2019.

[30] A. Mallios, P. Ridao, D. Ribas, and E. Hernández, "Scan matching slam in underwater environments," *Autonomous Robots*, vol. 36, no. 3, pp. 181–198, 2014. [Online]. Available: https://doi.org/10.1007/s10514-013-9345-0

[31] A. Burguera, Y. González, and G. Oliver, "Underwater slam with robocentric trajectory using a mechanically scanned imaging sonar," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, Conference Proceedings, pp. 3577–3582.

[32] A. Mallios, P. Ridao, M. Carreras, and E. Hernandez, "Navigating and mapping with the sparus auv in a natural and unstructured underwater environment," in *OCEANS 2011*, 2011, Conference Proceedings, pp. 1–7.

[33] H. Bülow, M. Pfingsthorn, and A. Birk, "Using robust spectral registration for scan matching of sonar range data," in *7th Symposium on Intelligent Autonomous Vehicles (IAV), IFAC*. IFAC, 2010, Conference Proceedings.

[34] A. Burguera, G. Oliver, and Y. Gonzàlez, "Scan-based slam with trajectory correction in underwater environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, Conference Proceedings, pp. 2546–2551.

[35] A. Mallios, P. Ridao, E. Hernandez, D. Ribas, F. Maurelli, and Y. Petillot, "Pose-based slam with probabilistic scan matching algorithm using a mechanical scanned imaging sonar," in *OCEANS 2009*, 2009, Conference Proceedings, pp. 1–6. [Online]. Available: 10.1109/OCEANSE.2009.5278219

[36] E. Hernandez, P. Ridao, D. Ribas, and A. Mallios, "Probabilistic sonar scan matching for an auv," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, Conference Proceedings, pp. 255–260. [Online]. Available: 10.1109/IROS.2009.5354656

[37] D. Ribas, P. Ridao, J. D. Tardos, and J. Neira, "Underwater slam in man-made structured environments," *Journal of Field Robotics*, vol. 25, no. 11/12, pp. 898–921, 2008.

[38] D. Ribas, P. Ridao, J. Domingo Tardos, and J. Neira, "Underwater slam in a marina environment," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, P. Ridao, Ed., 2007, Conference Proceedings, pp. 1455–1460.

[39] T. Hansen, F. Buda, and A. Birk, "Underwater exploration with sonar of the flooded basement of a ww-ii submarine bunker in the context

of digitization of cultural heritage," in *MTS/IEEE OCEANS*, 2023, Conference Proceedings, pp. 1–5.
[40] A. Birk, F. Buda, and T. Hansen, "Dating wall constructions from brick sizes in the flooded basement of a ww-ii submarine bunker for the digitization of cultural heritage," in *MTS/IEEE OCEANS*, 2023, Conference Proceedings.
[41] A. Birk, F. Buda, H. Bülow, A. G. Chavez, C. A. Müller, and J. Timpe, *Digitizing a Gigantic Nazi Construction: 3D-Mapping of Bunker Valentin in Bremen*. Berlin, Boston: De Gruyter, 2022, pp. 133–168. [Online]. Available: https://doi.org/10.1515/9783110714692-006
[42] T. Luczynski, P. Luczynski, LukasPehle, M. Wirsum, and A. Birk, "Model based design of a stereo vision system for intelligent deep-sea operations," *Measurement*, vol. 144, pp. 298–310, 2019.
[43] A. Birk, T. Doernbach, C. Mueller, T. Luczynski, A. G. Chavez, D. Köhntopp, A. Kupcsik, S. Calinon, A. Tanwani, G. Antonelli, P. d. Lillo, E. Simetti, G. Casalino, G. Indiveri, L. Ostuni, A. Turetta, A. Caffaz, P. Weiss, T. Gobert, B. Chemisky, J. Gancet, T. Siedel, S. Govindaraj, X. Martinez, and P. Letier, "Dexterous underwater manipulation from distant onshore locations," *IEEE Robotics and Automation Magazine (RAM)*, vol. 25, no. 4, pp. 24–33, 2018.
[44] T. Hansen and A. Birk, "Synthetic scan formation for underwater mapping with low-cost mechanical scanning sonars (mss)," *IEEE Access*, 2023.
[45] P. Vouras, K. V. Mishra, A. Artusio-Glimpse, S. Pinilla, A. Xenaki, D. W. Griffith, and K. Egiazarian, "An overview of advances in signal processing techniques for classical and quantum wideband synthetic apertures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 2, pp. 317–369, 2023.
[46] M. Amani, F. Mohseni, N. F. Layegh, M. E. Nazari, F. Fatolazadeh, A. Salehi, S. A. Ahmadi, H. Ebrahimy, A. Ghorbanian, S. Jin, S. Mahdavi, and A. Moghimi, "Remote sensing systems for ocean: A review (part 2: Active systems)," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 1421–1453, 2022.
[47] R. E. Hansen, *Introduction to Synthetic Aperture Sonar*. InTech, 2011.
[48] M. P. Hayes and P. T. Gough, "Synthetic aperture sonar: A review of current status," *Oceanic Engineering, IEEE Journal of*, vol. 34, no. 3, pp. 207–224, 2009.
[49] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.
[50] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, Conference Proceedings, pp. 3281–3288.
[51] D. Frank and K. Michael, *Factor Graphs for Robot Perception*. now, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8187520
[52] T. Hansen and A. Birk, "Using registration with fourier-soft in 2d (fs2d) for robust scan matching of sonar range data," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, Conference Proceedings.
[53] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*. American Mathematical Society, 2010.
[54] M. Pfingsthorn, S. Schwertfeger, H. Bülow, and A. Birk, "Maximum likelihood mapping with spectral image registration," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2010, Conference Proceedings.
[55] F. Maurelli, S. Krupinski, X. Xiang, and Y. Petillot, "Auv localisation: a review of passive and active techniques," *International Journal of Intelligent Robotics and Applications*, vol. 6, no. 2, pp. 246–269, 2022. [Online]. Available: https://doi.org/10.1007/s41315-021-00215-x
[56] L. Paull, M. Seto, S. Saeedi, and J. J. Leonard, *Navigation for Underwater Vehicles*. Berlin, Heidelberg: Springer, 2018, pp. 1–15. [Online]. Available: https://doi.org/10.1007/978-3-642-41610-1_15-1
[57] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.
[58] WaterLinked, "A50 dvl," 2023. [Online]. Available: https://waterlinked.com/dvl-a50
[59] Xsense, "Mti-300 ahrs," 2023. [Online]. Available: https://www.movella.com/products/sensor-modules/xsens-mti-300-ahrs