WEP Warwick Evans Publishing

Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 7
TPCEE 2024

C-Simultaneous Localization and Mapping (C-SLAM) for Underwater Robotics Using Sonar Data

Zhaowen Gu

Boston University, Boston, 02215, USA

Abstract. Since the current methods for precise underwater navigation are either insufficient in terms of accuracy or environmentally unfriendly, C-SLAM aims to address the critical issue of underwater localization for Uncrewed Underwater Vehicles (UUVs) undertaking seabed surveys. To improve the accuracy of navigation and overall operational efficiency, we'll use Simultaneous Localization and Mapping (SLAM) techniques, which can achieve precise localization by looking at matching landmarks and Inertial Navigation System (INS) measurements from sonar data. While processing these sonar data, C-SLAM will adjust position estimation errors through loop corrections and pose-graph optimization methods. As a result, it can generate a detailed 2D/3D visual representation of the vehicle's trajectory in the seafloor environment. Considering the importance of exploring the seafloor for scientific research, resource identification, and environmental monitoring, it is essential to track the accurate trajectory of UUVs, and our final deliverable, C-SLAM, is dedicated to tackle this issue.

Keywords: C-Simultaneous; Localization; Mapping; Sonar Data.

1. Introduction

The ocean's vast and largely unexplored terrain presents significant challenges for precise underwater navigation, particularly for Uncrewed Underwater Vehicles (UUVs) engaged in seabed surveys. Numerous offshore industries require accurate sonar surveys of the ocean floor. However, existing methods that depend on acoustic beacons, often discarded into the seafloor without retrieval, present constraints on both accuracy and environmental sustainability. Using a GPS is also an inefficient method because it's unable to transmit signals underneath the ocean's surface. This challenge of underwater navigation prompted our team to devise a solution that not only addresses the accuracy issue but is more environmentally sustainable.

The purpose of our project is to develop an implementation of the C-SLAM algorithm tailored specifically for subaquatic exploration. The core approach involves leveraging Simultaneous Localization and Mapping (SLAM) techniques to enhance underwater vehicle localization. By utilizing the survey data, itself, the algorithm will accurately estimate the UUV's position based on correlated sightings of landmarks and geometric measurements from the sonar data. The proposed approach aims to provide a comprehensive solution for precise underwater navigation and mapping, reducing reliance on physical beacons and thus promoting a more sustainable subaquatic exploration method.

Despite the challenges posed by the underwater environment, the algorithm can identify and classify anomalies in sonar data as landmarks, resist the noise inherent in subsea data, and ensure a reliable and accurate mapping of the UUV's movements. Moreover, because the algorithm adapts to different levels of sonar data clarity, it can be applied to various underwater environments.

By eliminating the necessity for physical acoustic beacons, C-SLAM not only reduces ocean pollution but also stands as a more sustainable alternative compared to other methods. Additionally, the algorithm generates precise updated bathymetric charts and 2D/3D visual graphs of the seafloor environment, which helps us understand the underwater terrain more comprehensively.

Therefore, through this project, our team aims to optimize underwater navigation for UUVs, aiming to make subaquatic exploration more sustainable and precise, so that we can better understand and utilize the vast resources hidden beneath the ocean's surface.

2. Motivation

Offshore industries need accurate sonar systems but currently face the challenges of imprecise beacons. Additionally, UUVs operate in GPS-denied environments, making it difficult to navigate. To address this, our goal is to develop an algorithm that reduces error accumulation over time.

3. Methods

3.1. Conceptual Approach

Compared to the visual datasets or sensor-fusion datasets with Inertial Measurement Unit (IMU) data that most SLAM algorithms commonly use to estimate the vehicle's location above the seafloor, the side-scan sonar datasets collected from UUV present distinctive characteristics. These datasets include unique side-scan images and Inertial Navigation System (INS) data, comprising latitude, longitude, depth, heading, pitch, and roll values. During the pre-processing stage, the dead-reckoning method could be a potential mathematical approach to compute the vehicle's position relative to the seafloor at each time step, which involves the integration of the INS data and previously predicted positions. Also, C-SLAM is capable of identifying and extracting correct landmarks from the side-scan images, while matching the key points between paired side-scan images. Following the image processing stage, the identified landmarks can be used in optimizing the pose-graph of the vehicle through loop correction based on landmark correspondence at different time steps. Then, the vehicle's trajectory will be refined by reconciling the observed landmarks with those detected at various points in time. After updating every position of the vehicle on the pose graph through the loop correction process, a comprehensive 2D/3D visual map should form. This map can serve as a visual representation of the refined trajectory of the UUV during its exploration of the underwater environment.

3.2. Alternative Solutions

The two alternative solutions we considered are ORB-SLAM3 and LSD-SLAM. ORB-SLAM3 [1] is an advanced system that can perform either visual, visual-inertial, or multi-map SLAM with various types of cameras. Its initialization process relies heavily on Maximum-a-Posteriori (MAP) estimation, which can effectively estimate the true scale of the scene being reconstructed based on the visual and inertial sensor uncertainties. When an active map is initialized, the tracking thread focuses on localizing incoming frames, while the local mapping thread continually enhances and extends the map by incorporating new keyframes. Subsequently, the loop correction from the loop and map merging thread were used to minimize the errors in these keyframes to create a final map. In addition, by incorporating a multi-map system, ORB-SLAM3 can relocalize itself when it experiences tracking losses. Although the sonar dataset collected from UUV cannot be directly input into ORB-SLAM3 to generate an active map due to the unique characteristics of its INS data, it is still an accurate, effective SLAM algorithm that can serve as a strong reference to guide us in the development of C-SLAM.

LSD-SLAM [2] is a direct monocular SLAM algorithm that can reconstruct the 3D environment in real-time. In contrast to ORB-SLAM3, it uses filtering-based depth map estimation to update the current keyframe in the pose graph. The advantage of LSD-SLAM is its flexibility in accurately estimating both fine details and large-scale geometry through scale-aware formulation, which can detect scale drift to minimize alignment error. However, LSD-SLAM demonstrated lower accuracy and generalizability compared to ORB-SLAM3 [1], so we have shifted our focus away from LSD-SLAM, no longer considering it as our primary reference.

Please find the detailed engineering requirements listed in Appendix 1. These requirements specify the parameters that our C-SLAM algorithm needs to meet to successfully address the problem.

4. System Description

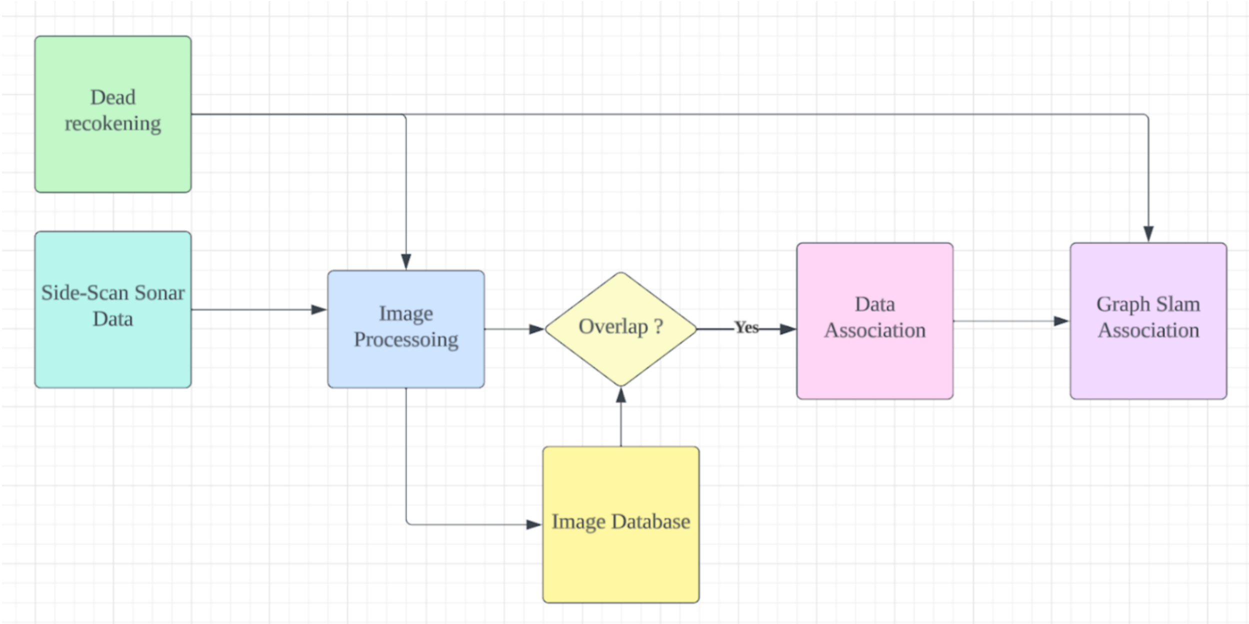

Figure 1. Block diagram of SLAM algorithm.

In a typical SLAM algorithm, six distinct sessions are involved, as outlined above. Our process involves processing images utilizing side-scan sonar data and comparing it with the global map. Subsequently, we examine whether the local image's landmarks align with any global landmarks. If there's a match, we link the global and local landmarks; otherwise, the algorithm progresses.

5. Methodology

5.1. Kalman Filter Update Equations

The Kalman Filter is an optimal estimator used for predicting the state of a dynamic system from a series of incomplete and noisy measurements. It works in two steps: prediction and update.

5.2. Prediction Step

In the prediction step, the filter estimates the state of the system at the next time step based on the current state and the system dynamics.

$$\hat{x}_{k|k-1} \footnote {predicted state estimate at time k given observations up to time $k-1$.} = A\footnote {state transition matrix} \hat{x}_{k-1|k-1} + B\footnote {control input matrix} u_{k-1} \footnote {control input at time $k-1$} $$

The predicted estimate covariance matrix is updated as follows:

$$P_{k|k-1} \footnote {predicted estimate covariance matrix} = A P_{k-1|k-1} \footnote {estimate covariance matrix at time $k-1$} A^\top + Q\footnote {process noise covariance matrix} $$

5.3. Update Step

In the update step, the filter incorporates the new measurement to refine the state estimate.

$$K_k\footnote {Kalman gain} = P_{k|k-1} H\footnote {measurement matrix} ^\top (H P_{k|k-1} H^\top + R\footnote {measurement noise covariance matrix}) ^ {-1} $$

The updated state estimate is:

$$\hat{x}_{k|k} \footnote {updated state estimate} = \hat{x}_{k|k-1} + K_k (z_k\footnote {measurement at time k} - H \hat{x}_{k|k-1}) $$

The updated estimate covariance matrix is:

$$P_{k|k} \footnote {updated estimate covariance matrix} = (I - K_k H) P_{k|k-1} $$

5.4. Pose-Graph Optimization

Pose-graph optimization is a technique used in robotics and computer vision to improve the accuracy of estimated trajectories and landmarks by refining the poses of the robot or camera. It is commonly used in simultaneous localization and mapping (SLAM) problems.

5.5. Pose-Graph Representation

In pose-graph optimization, the system is represented as a graph where nodes correspond to robot poses (or camera poses) and edges represent spatial constraints between these poses. Each node x_i in the graph represents a pose, and each edge e_{ij} represents a constraint between two poses.

5.6. Formulation

The goal of pose-graph optimization is to minimize the error in the graph. The optimization problem can be formulated as:

$$\hat{x} = \arg \min_{x\footnote{vector of all poses to be optimized}} \sum_{(i,j) \in E\footnote{set of edges in the graph, representing pairwise constraints}} \| z_{ij}\footnote{observed measurement or constraint between poses \(x_i \) and \(x_j \)} - h(x_i, x_j)\footnote{function that predicts the expected measurement based on the current poses} \|^2_{\Sigma_{ij}\footnote{ovariance matrix of the measurement \(z_{ij} \), representing the uncertainty}}$$

The error function to be minimized is the sum of squared residuals:

$$\text{Error}(x) = \sum_{(i,j)\in E} \| z_{ij} - h(x_i, x_j) \|^2_{\Sigma_{ij}}$$

5.7. Optimization Techniques

Various optimization techniques can be used to solve the pose-graph optimization problem, including:

- **Gauss-Newton Method:** An iterative optimization algorithm that approximates the error function using a quadratic model.
- **Levenberg-Marquardt Algorithm:** An extension of Gauss-Newton that incorporates a damping factor to improve convergence.
- **Gradient Descent:** A more general optimization technique that updates the poses in the direction of the negative gradient of the error function.

6. Algorithm

6.1. Sonar SLAM Algorithm

Read the file and store sonar pings Extract xy coordinates from sonar pings and store in set $S = \{(x_i, y_i)\}$ Isolate straight passes from the sonar data Extract xy coordinates of landmarks and store in set $L = \{(x'_j, y'_j)\}$

Apply Kalman Filter to estimate the current pose Update the pose estimate using the Kalman Filter equations

Merge landmarks into a new landmark $L_{\text{new}}=(x_\mu, y_\mu)$ where (x_μ, y_μ) is the mean coordinate

Apply pose-graph optimization to refine the pose estimates

6.2. Path Segmentation Algorithm

Input: List of sonar structures sonar_struct_list **Output:** List of linear segments

Update: $\text{prev\_headings} \leftarrow \text{current\_headings}$ Apply Kalman Filter to update heading estimate based on sonar data

Append: current_sublist to linear_segments Reset: $\text{current\_sublist} \leftarrow [\text{sonar\_struct})$ Append: to current_sublist

Append: current_sublist to linear_segments Filter: linear_segments to keep only segments with ≥ 1000 packets

Return: filtered_segments

7. Landmark Detection

To identify potential landmarks from sonar images, several image processing techniques are employed:

7.1. Grayscale Conversion

The sonar image is first converted to grayscale to simplify the processing. The grayscale intensity $I_{\text{gray}}(x, y)$ at each pixel (x, y) is computed as:

$$I_{\text{gray}}(x, y) = 0.299 \cdot R(x, y) + 0.587 \cdot G(x, y) + 0.114 \cdot B(x, y)$$

7.2. Gaussian Blurring

A Gaussian blur is applied to the grayscale image to reduce noise and emphasize larger structures. The blurred intensity $I_{\text{blur}}(x, y)$ is calculated using a Gaussian kernel $G(i, j)$:

$$I_{\text{blur}}(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{j=k} I_{\text{gray}}(x+i, y+j) \cdot G(i, j)$$

with the Gaussian kernel defined as:

$$G(i, j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right)$$

7.3. Image processing

In the image processing phase of our Sonar SLAM system, we began by adjusting the contrast and brightness of the captured images to enhance the visibility of key features. Various values were tested, and the optimal settings were selected based on visual inspection and performance metrics.

To further refine the images, we applied grayscale blurring, followed by Gaussian blurring, to smooth out the noise and improve edge detection. This preprocessing step was essential for achieving clearer contours in the subsequent stages.

After blurring, the Canny edge detector was used to filter out the contours present in the images. The Canny edge detector, with its multi-stage algorithm, effectively identified the edges of objects. This allowed us to accurately isolate the contours, which are critical for the landmark detection process.

Following contour detection, we computed the mean of these contours to establish a reference point for landmark identification. By averaging the contours, we defined a central location that represents the landmark. This approach ensured that the landmarks were marked with precision, contributing to the overall effectiveness of the SLAM system in recognizing and tracking environmental features.

Sonar Image processed after edge detection

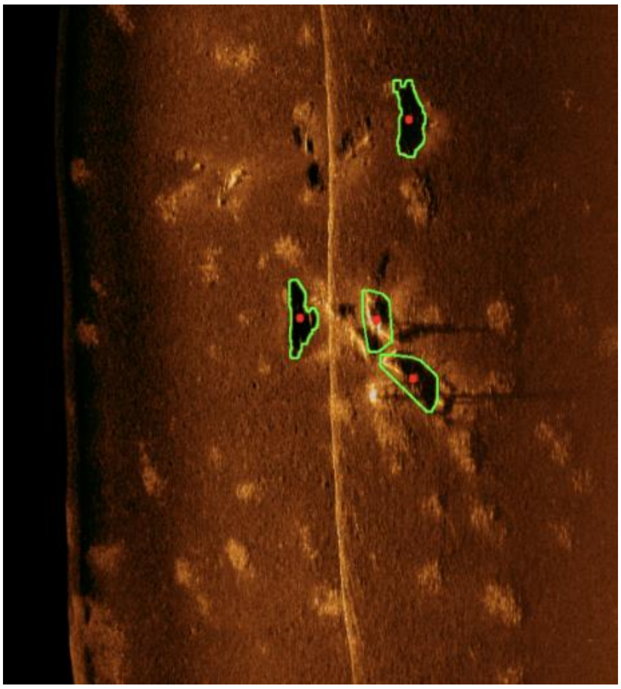

Figure 2. Sonar Image processed after edge detection

8. Results

8.1. Test Data

We tested our algorithm using both the simulated dataset from HoloOcean and the real-world datasets from the WWII torpedo dump and Bassurelle Sandbanks.

Underwater Survey of Basurelle Sandbanks, English Channel

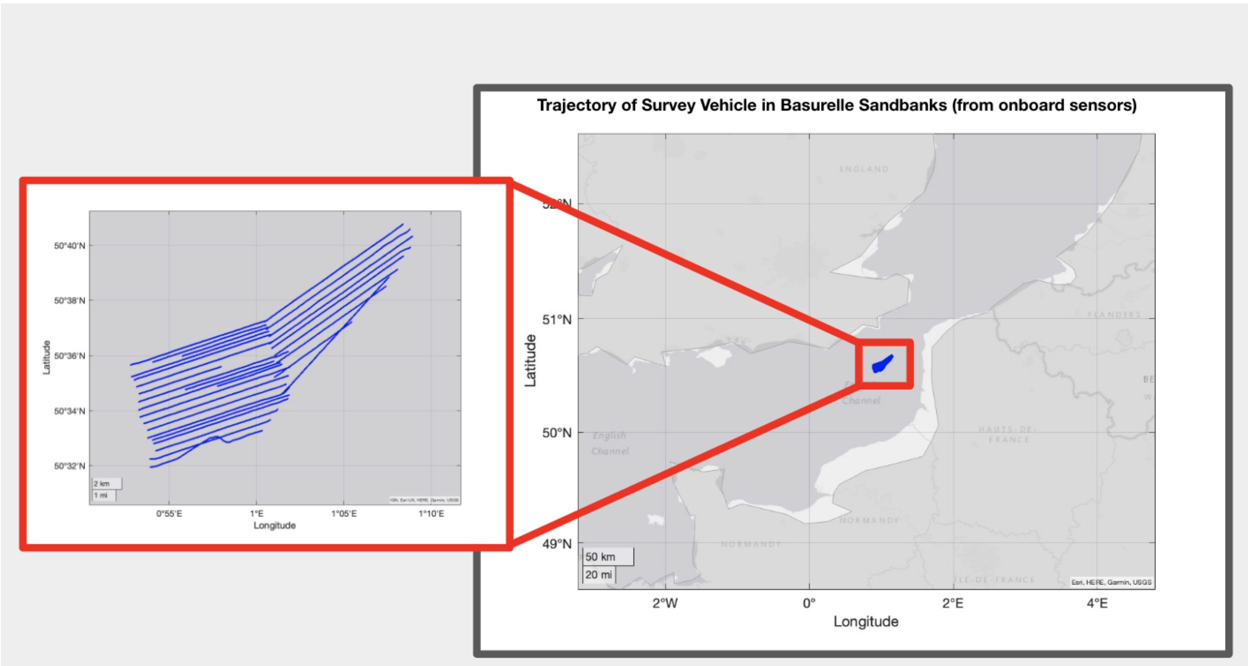

Figure 3. Underwater Survey of Basurelle Sandbanks, English Channel

(Underwater Survey of a WWII Torpedo Dump, Palau)

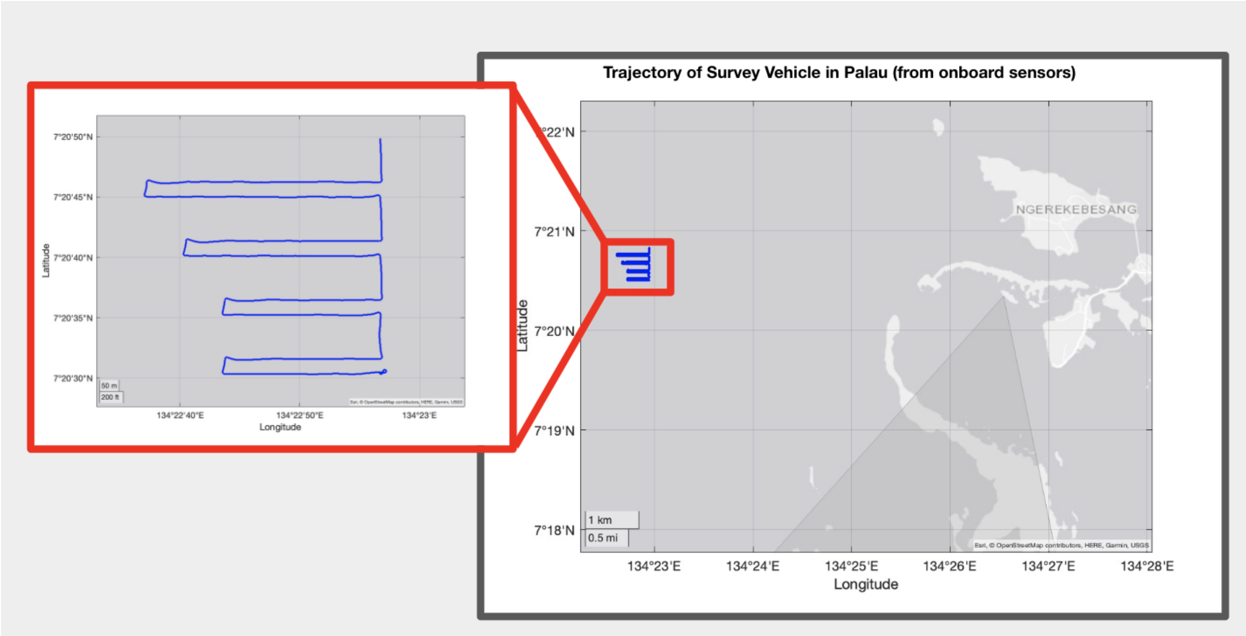

Figure 4. Underwater Survey of a WWII Torpedo Dump, Palau

8.2. 2D Trajectory and Landmark Plotting

We isolated the robot's straight paths and detected landmarks along those paths. Linear segments of the robot's path were plotted against the detected landmarks.

Segment paths to isolate straight passes of the robot and grouped landmarks

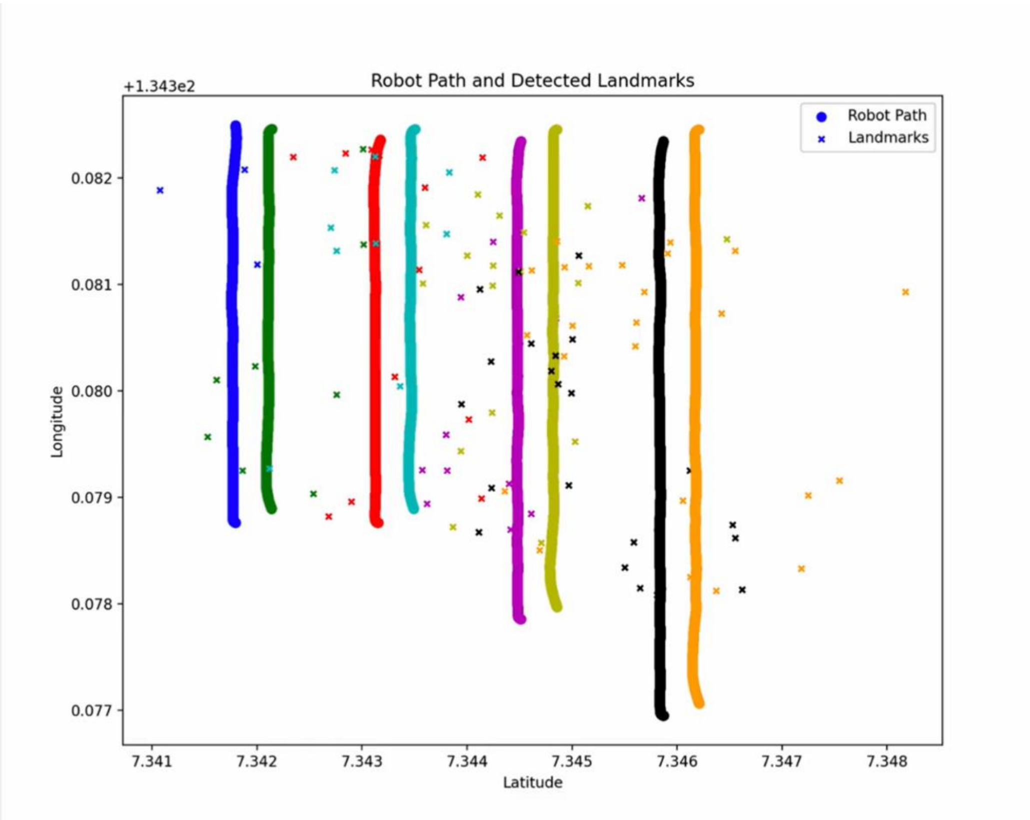

Figure 5. Segment paths to isolate straight passes of the robot and grouped landmarks

9. Conclusions

Our SLAM algorithm shows promise as a navigation solution for UUVs, successfully classifying underwater landmarks and plotting a 2D trajectory. While our implementation serves as a proof-of-concept, more refinement is required to prepare it for real-world applications. Due to the lack of real-world ground truth data to cross validate and limitation in funding and resources, there are many future implications left for exploration.

10. Future Research Directions

The presented Sonar SLAM algorithm offers a promising approach for underwater navigation, but several areas of improvement and further research remain. Future work will focus on the following directions:

10.1. Implementation of Pose-Graph Optimization

Incorporating pose-graph optimization into the SLAM framework can enhance accuracy by reducing cumulative errors in the estimated robot trajectory. This approach involves minimizing the error between the predicted and observed poses of the robot while leveraging loop closure to correct drift over time. Implementing pose-graph optimization will help refine the localization accuracy, particularly in long-duration missions or complex environments where traditional methods may struggle with error accumulation.

10.2. Performance Evaluation Using Underwater Simulation Data from HoloOcean

HoloOcean provides a realistic underwater simulation environment that allows for the testing and evaluation of SLAM algorithms in controlled scenarios. The next step involves evaluating the performance of the proposed Sonar SLAM algorithm within this virtual setting, using a variety of sonar datasets. Key metrics such as localization accuracy, landmark mapping consistency, and computational efficiency will be assessed. Simulations in HoloOcean offer a low-risk method to fine-tune the algorithm before real-world deployment.

10.3. Testing SLAM in Real-World Scenarios

After successful simulation trials, the algorithm will be tested in real-world underwater scenarios. These field tests will involve comparing the Sonar SLAM performance with existing underwater SLAM methods, such as the Blue-ROV SLAM system. Evaluating the algorithm under real environmental conditions, including sonar noise, limited visibility, and varying underwater terrain, will provide critical insights into its robustness, scalability, and adaptability to practical challenges faced in underwater robotics.

These research directions aim to progressively refine the Sonar SLAM algorithm, enhancing its accuracy, robustness, and practical applicability for underwater autonomous navigation and mapping.

11. Acknowledgments

We would like to thank Michael B. Gratton, Ph.D., and The Charles Stark Draper Laboratory for their guidance. We also appreciate the datasets provided by Eric Terrill from the Scripps Institution of Oceanography.

References

[1] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1–17, 2021. https://doi.org/10.1109/tro.2021.3075644

[2] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, Vol 8690, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer, 2014, pp. 834–849. https://doi.org/10.1007/978-3-319-10605-2_54

[3] J. Zhang et al., "A Fully-Automatic Side-Scan Sonar SLAM Framework," arXiv, Apr. 2023. https://arxiv.org/abs/2304.0185

[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2005.

[5] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[6] C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[7] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, 2006, pp. 1898–1903.

[8] S. Huang and G. Dissanayake, "Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1036–1049, 2007.

[9] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002, pp. 593–598.

[10] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[11] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II State of the Art," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[12] F. Lu and E. Milios, "Globally Consistent Range Scan Alignment for Environment Mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.