2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Side-scan sonar based landmark detection for underwater vehicles

Simon Hoff[1] Vegard Haraldstad[1] Bjørnar Reitan Hogstad[1] Damiano Varagnolo[1]

***Abstract*— We propose and analyze a pipeline to transform raw sonar data from underwater vehicles into actionable information for Simultaneous Localization and Mapping (SLAM) in real time. The pipeline encompasses three sequential steps, each building upon state-of-the-art algorithms from the existing literature: swath processing to preprocess the raw sonar data, with a primary focus on eliminating blind zones and noise reduction; transformation of these swaths into probabilistic maps of the surroundings of the sonar; and finally, detection and classification of underwater landmarks from the probabilistic maps.**

Our work contributes by modifying algorithms from the literature to ensure computational efficiency and integrating them so that they operate in sequence, thereby furnishing valuable information for navigation purposes.

Through validation with field data, we then discuss which scenarios may prove difficult for the individual stages of the proposed pipeline. We provide indications on whether each step may encounter challenges and discuss how this occurrence may affect the overall quality of the final result. This empirical discussion is useful for discerning the practical applicability of the proposed pipeline in real-world settings.

***Index Terms*— Marine Robotics, Autonomous Vehicle Navigation, Mapping**

I. Introduction

Navigation for underwater vehicles is often poised by problems in obtaining accurate global position measurements. One may use long and ultra-short baseline (LBL, USBL) measurements, but these require dedicated hardware and installations. Another strategy is to use simultaneous localization and mapping (SLAM), i.e., build global representations of the environment through local observations [1]. Such local observations may be salient landmarks in the mapped environment, usable for example to construct sparse global representations of the surroundings. The accuracy such landmark-based SLAM algorithms depends though on the robustness of the landmark detectors at their heart. And since visual SLAM, i.e. SLAM using a camera, is easily affected by both poor visibility and lighting conditions underwater, another solution may be to use side scan sonars (SSS). These are usable in the context of SLAM due to their wide coverage area, relatively high resolution, and relatively low price [2], [3].

An SSS emits an acoustic pulse and measures the backscattered intensity from the surrounding environment. SSS systems mounted on AUVs typically consist of two transducers transmitting perpendicularly to the forward direction of the AUV, with a narrow beam in the along-travel direction and a wide one in the plane normal to the travel direction. Each pulse from the sonar effectively measures a swath of the seabed but is affected by the so-called blind zone (a low-intensity background noise found in the bins with the smallest slant ranges and due to the beam traveling the water column, causing no or little backscatter to the transducers).

Even though one may detect landmarks in individual SSS swaths, it may be better to do so in reconstructed 2D images where recorded swaths are stacked together to form so-called waterfall images. Here, landmarks typically correspond to low-intensity shadows or high-intensity echoes [3]–[5].

Recent SLAM strategies improve this detection step by using semantic information, especially to simplify the process of matching previous and current detections [1], [6], [7].

This paper contributes to this field by presenting a semantic landmark detection pipeline for side-scan sonar that does swath processing, builds 2D cartesian maps of the processed swaths, and performs semantic shadow landmark detection on the generated maps.

The paper starts with presenting related work in Section II. Next, it presents the 3 parts of the proposed pipeline in Section III, Section IV and Section V. Finally, results and discussion are presented in Section VI, and conclusions presented in Section VII.

II. Related work

A. Swath processing

The three operations typically performed when processing side-scan sonar swaths are: Removing blind zones from the raw data; normalizing the intensity of the swaths, to remove range-dependent anomalies, and; correcting the slant ranges, to transform their value from being equally spaced in the slant range to the corresponding one relative to the ground ranges. These operations can be performed single-swath-wise or in batch. In this paper, blind-zone removal and intensity normalization are performed swath-wise, whereas the slant range correction is omitted and rather performed as a part of the batch-wise map generation.

Starting with the blind zone, this can be found either by using geometrical considerations, or by analyzing the swaths directly. Both methods seek to find the first bottom return (FBR) of the SSS ping to remove the earlier returns, as these only contain background noise. In [8], a threshold together with a canny edge detector is used to find the FBR, whereas [9], [10] utilize machine learning for detecting

This work was partly supported by the Research Council of Norway through project No. 302435 and the Centres of Excellence funding scheme, project No. 223254.

[1]Simon Hoff, Vegard Haraldstad, Bjørnar Reitan Hogstad, and Damiano Varagnolo are with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway

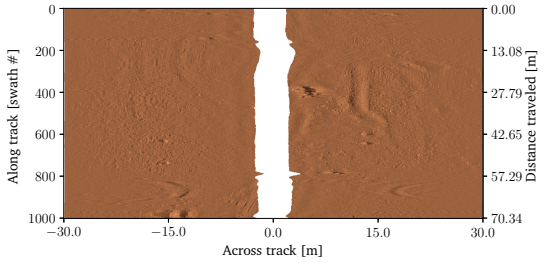


Fig. 1: A waterfall image from blind-zone removed, intensity normalized, slant range corrected swaths. The shades of brown represent the reflectivity (numbers between 0 and 1) of the seabed with respect to the transducer, meaning that perturbations from the assumed flat seabed, such as rocks or holes, will be the main contributor to variations in color.

the FBR. However, if an accurate estimate of the vehicle's altitude is available, the FBR can be found by geometrical considerations [11], which is the method utilized in this paper. Specifically, we utilize the measured altitude from the Doppler Velocity Log (DVL), which is accurate assuming that an accurate estimate of the sound speed is used.

Intensity normalization seeks to recover the reflectance map of the seabed from the recorded bins. Assuming that the measured swath consists of a luminance map $L(r_s)$ and a reflectance map $R(r_s)$, where r_s is the slant range, the measured swath is $I(r_s) = L(r_s) \cdot R(r_s)$. We can then estimate the luminance map by $\widehat{L}(r_s)$, before estimating the reflectance map by $\widehat{R}(r_s) = \dfrac{I(r_s)}{\widehat{L}(r_s)}$. Estimating the luminance map L can be performed in different ways. For instance, [12] models the seabed as a Lambertian surface and tests the Lambertian cosine law against a Lambertian $\cos^2$ law (with [13] doing the same against a $\sec$ Lambertian law). [14] tests different filters, such as a smoothing filter and a bilateral filter, to estimate L, whereas [15] utilizes smoothing cubic splines.

B. Map generation

The traditional way of reconstructing an image from the recorded swaths is to stack them one after another into waterfall images, as shown in Fig. 1. However, the quality of raw waterfall images depends on the vehicle navigating with constant speed, altitude, and attitude. This means that the produced images may contain frequent anomalies, and large portions of the data may even need to be discarded.

To generate more reliable maps from the SSS swaths one may start with generating a 2.5D map of the seabed by estimating its bathymetry, as in [16], [17], using shadows in the swaths to estimate this bathymetry, or as in [18], [19], using machine learning to do this estimation, or as in [17], using shape-from-shading techniques instead. All these approaches require computationally expensive steps, which in turn may prevent being able to implement real-time SLAM.

Another simpler solution is to map each recorded bin to their corresponding position on the seabed, constructing a 2D intensity map of the seabed. In general, such methods consist of three main steps. First, the AUV state estimation is utilized to determine the exact position on the seabed of each recorded bin (assuming a flat seafloor). Second, some type of interpolation or filtering is needed to determine the cell intensity of the map in a situation where several bins map to the same cell. Third, interpolation or filtering is also needed to fill the residual visual gaps in the map between the swaths where no bin covers the cells. [20] presents a kNN-based method that generates each map cell based on its k nearest bins if there is a bin closer than a distance threshold to each cell. In [11], a probabilistic sensor model is utilized to generate a probabilistic 2D map. This method does however suffer from large computational cost, making it challenging to deploy in a real-time system.

C. Landmark detection

Landmark detection, in the context of sparse landmark-based SLAM, is the process of detecting salient and distinct environment features. One key aspect is that the landmarks have to be reliably recognized. In addition, new research in SLAM utilizes landmark detectors that are able to estimate the semantic class of the landmarks.

Traditionally, landmark detectors such as SIFT [21], SURF [22], and ORB [23] have been utilized for visual SLAM. In [3], a combination of SIFT and FAST [24] corner features are used to find key points in a SLAM solution. On the other hand, in [4], SIFT and a Harrison detector are found to generate too many false positives, and a landmark detector based on intensity thresholding and geometric filtering is used instead. Intensity thresholding does unfortunately suffer from being challenging in a field context, where the acoustic properties of both the seabed and surrounding water may vary. Additionally, such thresholds are often dependent on the AUV altitude.

Shadows and sometimes echoes are commonly used landmarks in side-scan sonar maps. Such landmarks typically stem from elevated or lowered objects on the seabed, where an elevated object is recognized by an echo followed by a shadow, and the opposite for a lowered object.

In [4], shadow landmarks are detected in waterfall images using a low-intensity threshold followed by a set of geometrical filtering rules. [5] uses a high-intensity threshold, a low-intensity threshold, and a convolution filter to find landmarks in waterfall images. [2] utilizes a high-intensity threshold but also uses a template-matching detector and a low-intensity threshold in combination with a template-matching detector. In [25], a landmark detector uses a boosted cascade of Haar-like features that are tested against a combination of low and high-intensity thresholds to detect metallic objects on the seabed.

Machine learning has also been utilized to detect landmarks in the form of lowered or elevated objects on the seabed. In [26], a detector is trained to detect boulders in waterfall images, and in [10], a semantic detector is trained to distinguish between seafloor anomalies and rocks on the seafloor. [27] uses a CNN to segment individual swaths,

which are then stitched together into complete images. While this has the obvious drawback that additional steps must be taken to detect the location of each segmented landmark, such a step is reasonably trivial to perform.

Common to most existing works on Landmark detection is that the detection is performed on waterfall images. This usually precludes data collected during turning or abrupt changes to the vehicle's attitude, as the landmarks represented in the waterfall images will be severely distorted. In order to utilize data collected under abrupt attitude changes for landmark detection, landmark detection algorithms should instead be run on data mapped to the corresponding seabed locations.

III. Swath processing

The first step towards obtaining a 2D reflectance map of the seabed involves processing the raw sonar data, by computing the location of each bin on the seabed, as well as transforming the measured swath (received signal strength) into a reflectance map, as mentioned in Sec. II.

The proposed pipeline performs blind-zone removal, intensity normalization of the raw swaths, and incorporates slant range correction into the map generation algorithm rather than performing this as a separate step. The blind-zone removal step performs a pure geometric blind-zone removal, as in [11], but does include roll correction of the blind-zone. To do so, it first finds the slant range corresponding to the first bottom return for the starboard transducer as

$$r_{\text{fbr}} = \frac{h_{\text{stb}}}{\sin\left(\phi + \beta + \frac{\alpha_v}{2}\right)}, \tag{1}$$

where h_{stb} is the altitude of the starboard transducer, ϕ is the roll angle of the AUV, β is the mounting angle from the body y-axis of the AUV, and α_v is the vertical sensor opening of the sonar (see Fig. 2). Note that for finding the slant range of the port transducer's first bottom return, the sign of the roll angle ϕ is flipped. All bins with a slant range shorter than the first bottom return slant range are then considered the blind zone and removed from the measurement. Finally, we perform intensity normalization via smoothing cubic splines as in [15] and on the bins remaining after blind-zone removal with a chosen smoothing parameter p.

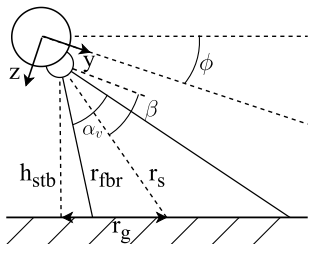


Fig. 2: A schematic representation of the geometry of the starboard transducer. The roll ϕ affects the transducer height h_{stb}, the slant range r_S, the ground range r_g, and the first bottom return range r_{fbr}.

IV. Computationally efficient probabilistic 2D map generation

The proposed probabilistic map generation algorithm builds on [11], bringing through algorithmic improvements to significantly improve computational time and incorporate slant range correction steps. The algorithm creates maps that consist of square map cells q, with four corners q_i, with sides equal to the map resolution δ_m, and with origin defined as M_0 (see Fig. 3).

The map generator models each swath with a deterministic slant range r_s and non-deterministic azimuth angle θ. For simplicity, we assume a Gaussian distribution centered around the acoustic axis of the sonar with a standard deviation equal to half of the horizontal transducer opening

$$p(\theta) = \frac{1}{\frac{\alpha_h}{2}\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{\theta}{(\frac{\alpha_h}{2})}\right)^2\right\} \tag{2}$$

leading to a probability of measurement m observing a cell q equal to

$$P^m(q) = \int_{\theta_{\min}}^{\theta_{\max}} p(\theta) d\theta, \tag{3}$$

where $\theta_{\min}$ and $\theta_{\max}$ are the minimum and maximum observation angles of the cell q by measurement m, as shown in Fig. 3.

Let S be the set of all measurements m that should be used to generate a map M, Q the set of all cells q in the map M, and Q_m the set of all cells q that has a probability of being observed by measurement m larger than ϵ, i.e.,

$$Q_m = \{q \in Q | P^m(q) \geq \epsilon\}. \tag{4}$$

To find the resulting intensity of the cells in Q_m, we propose to first linearly interpolate the measured intensity I^m in measurement m to each of the corners of the cells so that the resulting intensity is the mean of intensity of the four corners, i.e.,

$$V^m(q) = \frac{1}{4}\sum_{i=0}^{3}\left(w_1 I^m\left(\left\lceil \frac{r^m_{s,i}}{\delta_s} \right\rceil\right) + w_2 I^m\left(\left\lfloor \frac{r^m_{s,i}}{\delta_s} \right\rfloor\right)\right), \tag{5}$$

where δ_s is the slant range resolution of the measurements, $w_1 = \frac{r^m_{s,i}}{\delta_s} - \left\lfloor \frac{r^m_{s,i}}{\delta_s} \right\rfloor$, and $w_2 = 1 - w_1$. $r^m_{s,i}$ is the slant range for the i$^{\text{th}}$ corner of cell q for measurement m. This step differs from [11] in that it directly uses the slant range for interpolating, making it possible to skip the slant-range correction of the swaths.

To find the resulting intensity of a cell q, all the cell intensities from each of the measurements are weighted by their probability of observation and combined in

$$V(q) = \sum_{m \in S} \begin{cases} 0, & q \notin Q_m \\ P^m(q) V^m(q), & q \in Q_m. \end{cases} \tag{6}$$

Furthermore, $P(q)$ is used as a normalization term where

$$P(q) = \sum_{m \in S} \begin{cases} 0, & q \notin Q_m \\ P^m(q), & q \in Q_m, \end{cases} \tag{7}$$

such that the final intensity if cell q in the map M is given by

$$M(q) = P(q)^{-1}V(q). \tag{8}$$

The final step of the map generator is to fill the visual gaps that can be extrapolated from the already generated map. Only cells where one of the measurements $m \in S$ has observed it with a slant range shorter than the sonar range and longer than the slant range of the blind zone are considered. To do this, we use a k-NearestNeigbhour method, where the mean of the k nearest neighbors inside a range threshold of each candidate cell is used as the extrapolated intensity.

To make the algorithm more computationally efficient, we find the cells in Q_m by a flood-fill method to minimize the number of cells in Q that must be examined. This is done by starting the flood-fill from a cell known to be observed in the left portion of the swath and one in the right portion of the swath. The flood fill stops when no new neighboring cells are left to check if they belong in Q_m.

An example of a map generated through the proposed algorithm is shown in Fig. 4. Note that since the outer loop iterates over swaths, we reduce the number of calculations of the observation angles of a cell by realizing that neighboring cells share corners; hence, the observation angles will remain the same for these.

Algorithm 1 Probabilistic map generation

Input: S, M_0, δ_m
Output: M

$M, V, P \leftarrow \varnothing$
for $m \in S$ **do**
  $K \leftarrow$ GETINITIALCELLS(m, M_0, δ_m)
  while $q \leftarrow \text{pop}(K)$ **do**
    if $q \in Q_m$ **then**
      $V(q) \leftarrow V(q) + P^m(q)V^m(q)$
      $P(q) \leftarrow P(q) + P^m(q)$
      $K \leftarrow K \cup$ GETFOURCONNECTEDCELLS(q)
    end if
  end while
end for
for $q \in Q$ **do**
  $M(q) \leftarrow P(q)^{-1}V(q)$
end for
$M \leftarrow$ KNNFILL(M)

V. Semantic shadow landmark detection with height estimation

Rather than simply detecting features by their shadows, we aim to use the structure of the signal to determine whether the landmark observed is a hole in the seabed or a rise (e.g., a rock or mound of sediment). This detector executes four main steps: *1)* find landmark candidates by intensity thresholding; *2)* filter these candidates based on their geometric properties; *3)* classify the remaining candidates; *4)* estimate

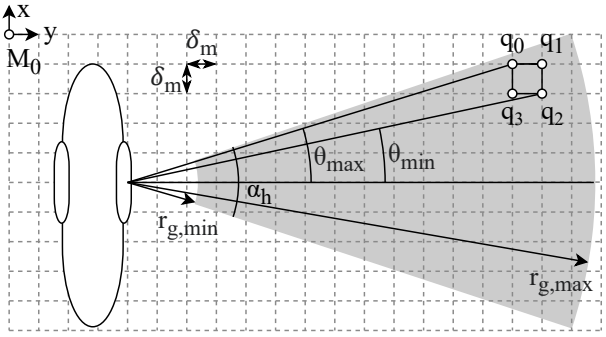


Fig. 3: The figure shows an AUV from above, how its starboard transducer maps to the seabed, and the definitions used when generating cartesian maps using the probabilistic map generation algorithms. Remade from [11].

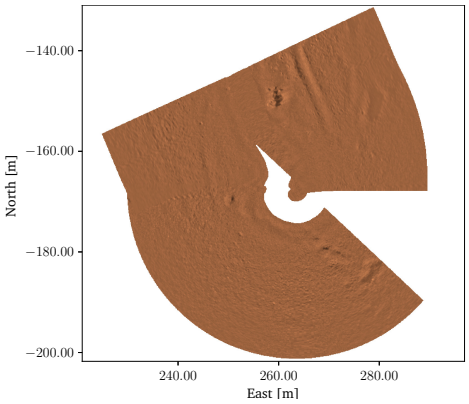


Fig. 4: The figure shows a map generated using the proposed map generation algorithm. A map resolution of $\delta_m = 0.1\,\text{m}$ and a probability threshold of $\epsilon = 0.1$ was used. As in Fig. 1, the colors represent the reflectance w.r.t. the AUV on a per-swath basis.

the height of each candidate. This process is conducted on maps generated from batches of 100 swaths. An overlap of $50\,\%$ is used between two consecutive batches to prevent skipping partially observed landmarks.

In more detail, step 1 above applies a low- and high-intensity threshold, $t_{i,l}$ $t_{i,h}$, on the map, generating two sets of landmark candidates L_l and L_h. The reason for using two different thresholds is to improve the robustness of the detector.

Step 2 geometrically filters the candidates based on their area A_l using a min and max area threshold $t_{A,l}$ $t_{A,h}$ and the amount of the square bounding box surrounding the landmark covered by the actual landmark ρ_{bb} using a min threshold $t_{\rho_{bb}}$, reducing the number of landmark candidates in L_l and L_h. The resulting landmark candidates L_c are found by taking the intersection of the landmarks in L_l and L_h. Here, two landmarks are considered the same if they share at least one map cell.

In step 3, the semantic class of the remaining landmark candidates is estimated by finding the order of the echo and shadow in the map. This is done by utilizing non-linear least

TABLE I: The tuning parameters used to generate the results in this paper.

Parameter	Value
p	$6.0 \cdot 10^{-6}$
δ_m	$0.1\,\mathrm{m}$
ϵ	0.1
$t_{i,l}$	0.96
$t_{i,h}$	0.98
$t_{A,l}$	$0.4\,\mathrm{m}^2$
$t_{A,h}$	$10.0\,\mathrm{m}^2$
$t_{\rho_{bb}}$	0.15
σ_{lc}	10.0
t_h	$0.15\,\mathrm{m}$

squares to fit a Gaussian derivative to several swaths that have observed the landmark and determine the order of echo and shadow by an ensemble score based on the sign of the fitted Gaussian derivative. For this, half of the swaths that have observed the landmarks are used, where the first and last quarters of swaths (sorted by measurement time) are not considered. In addition, if the number of swaths is even, one extra of the middle swaths is added to ensure a decision by the ensemble score. The function fitting is conducted only on the part of the swath containing the landmark and a distance before and after equal to twice the slant range of the landmark. Furthermore, this part of the swath is smoothed using a Gaussian filter with a standard deviation of σ_{lc} to improve performance. For the function fitting, the function used is defined as

$$f(x) = \frac{a}{\sqrt{2\pi}c^3}(x-b)\exp\left\{\frac{-(x-b)^2}{2c^2}\right\} + d, \qquad (9)$$

where a, b, c and d are parameters to be fitted. The order of the shadow and echo is determined by the sign of a, where $a < 0$ indicates an echo before a shadow, i.e., an elevated landmark. As the result of the function fitting is sensitive to the initialization, the function fitting is performed one time initialized with a negative a and one time with a positive one. The vote is given if both result in the same sign of a; if not, the result with the smallest sum of squared errors is chosen.

Lastly, in step 4, the height of the landmark candidates is estimated using geometrical consideration on the minimum and maximum slant range of the image before a final height filtering of the landmarks is performed to find the resulting landmarks using a minimum height threshold t_h. In the case of an elevated landmark, the landmark position is chosen as the smallest ground range of the landmark center. For lowered landmarks, the furthest ground range of the center is used.

VI. Results and discussion

We present the result of tests on real-world data collected with a Light-AUV from the Applied Underwater Robotics Laboratory at NTNU around the Svalbard islands and equipped with a dual frequency Deep Vision side scan sonar, used with a range of $30\,\mathrm{m}$ and a nominal frequency of $640\,\mathrm{kHz}$, and comprising a set of manually labeled landmarks. The code used can be found at `https://turlab.itk.ntnu.no/turlab/SSS_slam_landmark_detection`.

TABLE II: Average computation time over 10 runs used for generating a map of n_s swaths, using a map resolution of $\delta_m = 0.1\,\mathrm{m}$ and with all possible objects possible preallocated, as would be in real life implementations.

Algorithm	$n_s = 100$	$n_s = 500$	$n_s = 1000$
kNN [20]	236 ms	596 ms	842 ms
Original probabilistic [11]	3.92 s	30.9 s	91 s
Ours	67.1 ms	322 ms	908 ms

TABLE III: How often each part of the original map generator and the proposed one are performed, using a map resolution of $\delta_m = 0.1\,\mathrm{m}$ and 100 swaths. This results in a map of $617 \cdot 655$ cells with many empty cells.

Algorithm	(3)	(5)	(8)
Orig. probabilistic [11]	$404 \cdot 10^5$	$248 \cdot 10^3$	$404 \cdot 10^3$
Ours	$959 \cdot 10^3$	$239 \cdot 10^3$	$404 \cdot 10^3$

A. Performance of the probabilistic map generation

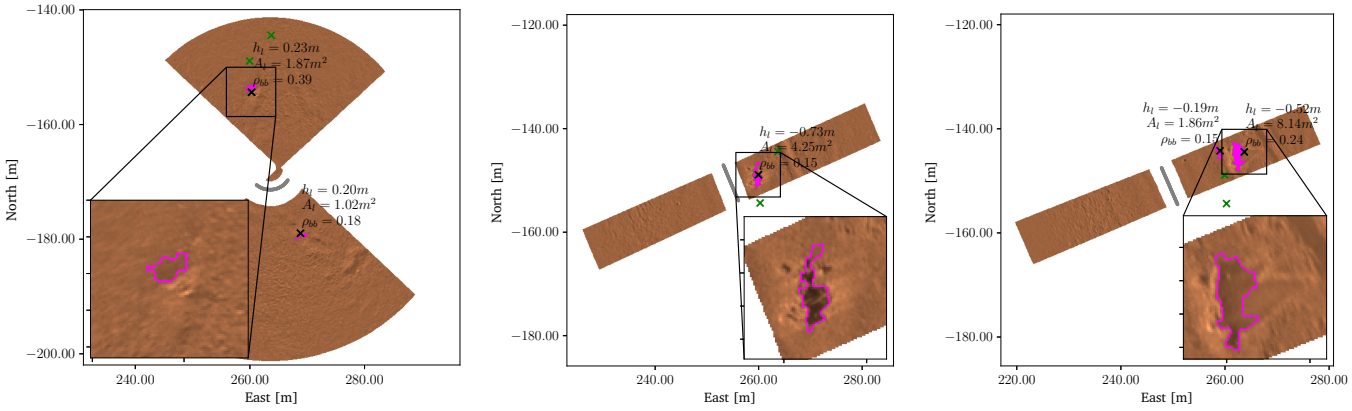
We test the proposed algorithm against the original one in [11] and the kNN-based map generation in [20] via implementation in the Julia programming language and test on an Intel i5 CPU with Ubuntu 22. Note that no additional information was calculated during these tests, and that the kNN-algorithm filling the visual gaps was used for both map generators. The results, summarized in Table II, show how the proposed algorithm is considerably faster than the original one even when integrating slant range correction steps.

The proposed map generator was also tested against the one in [11] on the task of generating a map of 100 swaths in terms of how many times (3), (5), and (8) were executed. The results, shown in Table III, highlight a significant difference in the number of times a cell's observation probability (3) was calculated.

B. How well does the pipeline work?

Normalizing swaths using smoothing cubic splines is also simple but does, in contrast to the blind zone correction, rely on the data itself. Despite this, it is robust and easy to tune, as only one tunable parameter exists. However, we have not verified the effect of actual variations in the intensity over a large portion of swaths. Note that level but slanted seabeds across the entire swath may lead to errors in the reported slant, but this may be potentially beneficial when performing landmark detection and SLAM, as the landmarks observation would be less affected by the roll of the AUV and such seabed inclines.

Fig. 5 shows the overall results of the proposed pipeline at time steps 1, 8 and 26. The pipeline detects all larger landmarks and their extent, except in time step 26. We observe that when the observation angle is the same in time step 8 and 26, the landmark is classified as a lowered object,



(a) The landmarks in time step 1. (b) The landmarks in time step 8. (c) The landmarks in time step 26.

Fig. 5: The figure displays the same landmark detected over three distinct time steps. The black ”x” marks the current landmark, and the green ones are the other two landmarks. The path of the LAUV is displayed in grey.

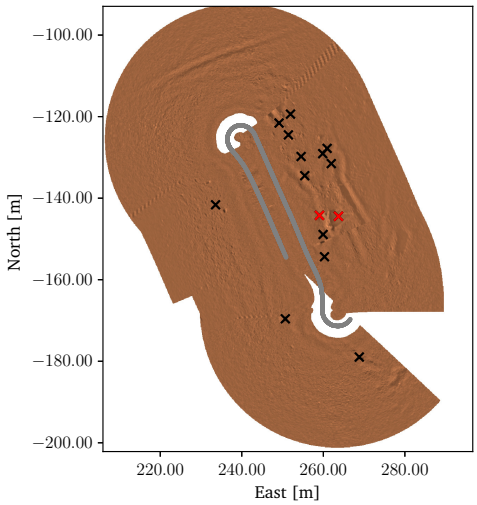


Fig. 6: The entire map with all detected landmarks up to time step 26. The two red ‘x‘ displays the two new landmarks. The path of the LAUV is displayed in grey.

but with a different viewpoint in time step 1, it is classified as an elevated object.

The detectors capabilities in detecting the landmark’s border, choosing suitable slant ranges to estimate the landmark’s height, and detecting the same landmark from different observation angles and ranges, are more brittle (e.g., as in Fig. 5c, where one part of the landmark is missed due to the low dynamical range of the image).

The landmark classification then works for elevated landmarks, but due to an unbalanced dataset, it is difficult to assess the classification’s performance on other types of landmarks. In any case, as shown in Fig. 5, different observation angles and ranges greatly affect how the same landmark is detected.

Fig. 6 visualizes the map estimated by the proposed pipeline up to time step 26. Here, all the detected landmarks are highlighted, thus also the landmarks shown in Fig. 5 (around the red ”x’s”). Fig. 6 shows how the drift of the state estimation influences the estimated position of the same landmark over different time steps - an example of how semantic information may aid in reducing navigation uncertainty.

TABLE IV: Positive annotations (P), false positives (FP), false negatives (FN), and true positives (TP) for the used dataset.

	P	**TP**	**FP**	**FN**
Dataset 1	183	129	34	54

C. *Quantative results*

Table IV shows that the landmark detector correctly identifies a substantial amount of manually annotated landmarks. In more details it works best on clear round, elevated, or lowered objects, and struggles to fully detect the whole extent of the landmark the farther we move from this working condition. Two distinct landmarks close to each other can also be wrongfully detected as one, andlandmarks in the far range are also challenging to detect due to a lower dynamical range in the far bins. In this situation the detector may hallucinate in the presence of sand ripples. In addition, landmarks could be wrongly detected when the LAUV had an acoustic transmission with the surface vessel, as the acoustic transmission affected the SSS measurement.

As for the landmark classification performance, the normalized confusion matrix in Table V shows a good performance in classifying elevated objects, and a poor one in classifying lowered objects. The classifier inherits the same struggles of the detector, with landmarks not round or nicely defined and in the far range being the most difficult case. We note that for these cases also manual labelling is actually difficult.

VII. Conclusion

The proposed pipeline is shown to reliably identify and classify landmarks when close to the sonar. The blind zone

TABLE V: Confusion matrix of the semantic landmark classifier, on a dataset with 121 elevated landmarks and 8 lowered ones.

		Ground thruth	
		Elevated	Lowered
Predicted	Elevated	0.760	0.625
	Lowered	0.240	0.375

remover is simple and robust as it does not directly rely on the sonar data and does not require tuning.

The swath normalization is robust and easy to tune, as only one tunable parameter exists. We do however note that we have not verified the effect actual variations in the intensity over a large portion of a swaths. The map generator is robust and produces usable maps at a computational cost that is compatible with real-time computing requirements, thanks to the impact of the proposed algorithmic improvements on the overall computation complexity. Overall, the landmark detector delivers good performance in detecting landmarks.

The pipeline thus provides complete information usable in semantic SLAM, e.g., estimating the height and class of the landmarks. The pipeline seems to have good detection performance in "easy" cases, but requires further work to improve its performance and robustness otherwise. Despite this, the pipeline in any case lays the groundwork for a robust and accurate landmark detection pipeline and is already usable to improve navigation performance for underwater vehicles.

REFERENCES

[1] D. M. Rosen, K. J. Doherty, A. Terán Espinoza, and J. J. Leonard, "Advances in Inference and Representation for Simultaneous Localization and Mapping," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, 2021.

[2] K. Siantidis, "Side scan sonar based onboard SLAM system for autonomous underwater vehicles," in *Proc. Autonomous Underwater Vehicles Symposium*, 2016.

[3] J. Zhang, Y. Xie, L. Ling, and J. Folkesson, "A fully-automatic side-scan sonar simultaneous localization and mapping framework," *IET Radar, Sonar & Navigation*, vol. 18, 2024.

[4] I. Leblond, S. Tauvry, and M. Pinto, "Sonar image registration for swarm AUVs navigation: Results from SWARMs project," *Journal of Computational Science*, vol. 36, 2019.

[5] J. Petrich, M. F. Brown, J. L. Pentzer, and J. P. Sustersic, "Side scan sonar based self-localization for small Autonomous Underwater Vehicles," *Ocean Engineering*, vol. 161, 2018.

[6] K. Doherty, D. Fourie, and J. Leonard, "Multimodal Semantic SLAM with Probabilistic Data Association," in *Proc. 2019 International Conference on Robotics and Automation*, 2019.

[7] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based Semantic SLAM," in *2019 International Conference on Intelligent Robots and Systems*, 11 2019.

[8] C.-C. Shih, M.-F. Horng, Y.-R. Tseng, C.-F. Su, and C.-Y. Chen, "An Adaptive Bottom Tracking Algorithm for Side-Scan Sonar Seabed Mapping," in *Proc. 2019 IEEE Underwater Technology*, 2019.

[9] G. Zheng, H. Zhang, Y. Li, and J. Zhao, "A Universal Automatic Bottom Tracking Method of Side Scan Sonar Data Based on Semantic Segmentation," *Remote Sensing*, vol. 13, 2021.

[10] X. Qin, X. Luo, Z. Wu, J. Shang, and D. Zhao, "Deep Learning-Based High Accuracy Bottom Tracking on 1-D Side-Scan Sonar Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, 2022.

[11] A. Burguera and G. Oliver, "High-Resolution Underwater Mapping Using Side-Scan Sonar," *PLOS ONE*, vol. 11, 2016.

[12] A. Burguera and G. Oliver, "Intensity correction of Side-Scan Sonar images," in *Proc. 2014 IEEE Emerging Technology and Factory Automation*, 2014.

[13] W. Xu, L. Ling, Y. Xie, J. Zhang, and J. Folkesson, "Evaluation of a canonical image representation for sidescan sonar," in *Proc. OCEANS 2023 - Limerick*, 2023.

[14] X. Ye, H. Yang, C. Li, Y. Jia, and P. Li, "A Gray Scale Correction Method for Side-Scan Sonar Images Based on Retinex," *Remote Sensing*, vol. 11, 2019.

[15] M. Al-Rawi, A. Galdran, A. Isasi, F. Elmgren, G. Carbonara, E. Falotico, D. A. Real-Arce, J. Rodriguez, J. Bastos, and M. Pinto, "Cubic spline regression based enhancement of side-scan sonar imagery," in *Proc. OCEANS 2017 - Aberdeen*, 2017.

[16] J. M. Cuschieri and M. Hebert, "Three-Dimensional Map Generation From Side-Scan Sonar Images," *Journal of Energy Resources Technology*, vol. 112, 1990.

[17] K. R. Jones and P. Traykovski, "A Method to Quantify Bedform Height and Asymmetry from a Low-Mounted Sidescan Sonar," *Journal of Atmospheric and Oceanic Technology*, vol. 35, 2018.

[18] N. Bore and J. Folkesson, "Neural Shape-From-Shading for Survey-Scale Self-Consistent Bathymetry From Sidescan," *IEEE Journal of Oceanic Engineering*, 2023.

[19] Y. Xie, N. Bore, and J. Folkesson, "Bathymetric Reconstruction From Sidescan Sonar With Deep Neural Networks," *IEEE Journal of Oceanic Engineering*, vol. 48, 2023.

[20] B. Reitan Hogstad, "Side-Scan Sonar Imaging and Error-State Kalman Filter Aiding Unmanned Underwater Vehicle (UUV) to Autonomy," 2022.

[21] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, 2004.

[22] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, 2008.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. 2011 International Conference on Computer Vision*, 2011.

[24] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006.

[25] J. Aulinas, X. Llado, J. Salvi, and Y. R. Petillot, "Feature based slam using side-scan salient objects," in *Proc. OCEANS 2010 - Seattle*, 2010.

[26] J. H. Christensen, L. V. Mogensen, and O. Ravn, "Side-Scan Sonar Imaging: Automatic Boulder Identification," in *Proc. OCEANS 2021*, 2021.

[27] A. Burguera and F. Bonin-Font, "On-line multi-class segmentation of side-scan sonar imagery using an autonomous underwater vehicle," *Journal of Marine Science and Engineering*, vol. 8, 2020.