2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Data-driven Loop Closure Detection in Bathymetric Point Clouds for Underwater SLAM

Jiarui Tan[1], Ignacio Torroba[1], Yiping Xie[1], and John Folkesson[1]

Abstract— **Simultaneous localization and mapping (SLAM) frameworks for autonomous navigation rely on robust data association to identify loop closures for back-end trajectory optimization. In the case of autonomous underwater vehicles (AUVs) equipped with multibeam echosounders (MBES), data association is particularly challenging due to the scarcity of identifiable landmarks in the seabed, the large drift in dead-reckoning navigation estimates to which AUVs are prone and the low resolution characteristic of MBES data. Deep learning solutions to loop closure detection have shown excellent performance on data from more structured environments. However, their transfer to the seabed domain is not immediate and efforts to port them are hindered by the lack of bathymetric datasets. Thus, in this paper we propose a neural network architecture aimed to showcase the potential of adapting such techniques to correspondence matching in bathymetric data. We train our framework on real bathymetry from an AUV mission and evaluate its performance on the tasks of loop closure detection and coarse point cloud alignment. Finally, we show its potential against a more traditional method and release both its implementation and the dataset used.**

I. INTRODUCTION

There is an increasing demand from both the scientific community and industry to create high resolution terrain models of the ocean floor. Tasks such as laying of pipelines and cables or deep sea oceanographic and environmental research necessitate accurate seabed maps for safe execution and high-fidelity geo-referencing of the data collected [1]. Autonomous underwater vehicles (AUVs) equipped with multibeam echosounders (MBES) are becoming essential tools for these missions, thanks to the high-resolution 3D reconstructions of the underwater environment that they can provide, regardless of the water conditions. This is due to their capacity to get closer to the seafloor and to reach inaccessible areas compared to surface vessels. However, their ability to travel long distances underwater is hindered by the lack of a global navigation satellite system (such as GPS) underwater that could bound the navigation drift to which AUVs are particularly sensitive. Although similar localization systems exist for underwater vehicles, such as long baseline (LBL) or ultra-short baseline (USBL) [2], they are limited in their operational capabilities by the propagation of sound in water and the need to deploy the transponders close enough to the the survey site.

In this context, simultaneous localization and mapping (SLAM) techniques can provide an alternative method for the vehicle to autonomously correct its trajectory estimate based only on its own proprioceptive sensors and the measurements collected from the environment [3]. Current state-of-the-art bathymetric SLAM frameworks fuse dead reckoning (DR) estimates and trajectory corrections from the registration of overlapping submaps in a factor graph optimization [4]. Although well-studied, these techniques rely critically on the successful detection of the overlap between submaps upon loop closure (LC), which remains an open problem for bathymetric data in large, unstructured environments [5]. This is mainly due to the scarcity of distinguishable landmarks on the seabed, together with the low resolution and the noise levels characteristic of MBES data.

Thus, this work focuses on the problem of autonomous data association in bathymetric point clouds. Given the complexity of deriving principled solutions to this problem in unstructured environments, we present a data-driven approach built upon the latest advancements in keypoint selection and descriptor learning achieved by neural network (NN) architectures [6], [7]. We train and evaluate our method on MBES data from an AUV mission under the Thwaites glacier in Antarctica. Finally, although our results are limited by the amount of data available, we show the potential for our approach to be readily applied to loop closure detection and coarse alignment for submap-based bathymetric SLAM.

II. RELATED WORK

Autonomous place recognition and point cloud alignment are two key components of SLAM solutions as they provide loop closure constraints for bounding the vehicle's DR drift. Both processes rely on keypoint selection and descriptor embedding to detect and describe salient points that can be easily disambiguated.

Regarding the selection of keypoints and the construction of their descriptors, the existing approaches can be divided into handcrafted and learning-based. As examples of the former, in [5] the bathymetric point clouds are converted to "pseudo-images" to extract SIFT descriptors [8] for the matching. Working directly on the point clouds, Suresh *et al.* [9] extracts keypoints with a Harris 3D detector [10]. These are then encoded in SHOT descriptors [11], which are clustered to create bathymetric submap dictionaries for loop closure detection. This method will be applied later in this paper as a baseline. Data-driven methods go beyond manual design of detectors and descriptors and instead aim to learn them directly from the data. This was made possible by the seminal PointNet from [12], which allowed neural networks to work directly with geometric point clouds. This architecture and its extension to hierarchical learning, PointNet++

The authors are with the Division of Robotics, Perception and Learning at KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. {jiaruit, torroba, yipingx, johnf}@kth.se

[13], enabled further research on learning local descriptors directly from point clouds [14], [6], [7]. Deng *et al.* [14] proposed a novel N-tuple loss based on the contrastive loss from [15], combining point pair features and global context to learn a globally-aware feature descriptor. Lu *et al.* [7] presented a successful end-to-end point cloud registration framework, although constrained by the need of accurate ground truth poses in the training stage. To overcome this, in [6], a weakly-supervised framework is designed to learn salient local features from 3D point clouds without precise knowledge of the ground truth point-to-point correspondence matching. Moving beyond neural networks, in [16] Gaussian processes (GP) [17] are used to find salient keypoints by comparing the smooth GP posterior mean trained on a patch of bathymetry against the raw data.

Finding correspondences across point clouds in the feature space then allows to perform place recognition. Random sample consensus (RANSAC) [18] has been widely employed for this due to its simplicity and robustness against outliers [19]. In [20], a random forest is trained to learn the most representative metric to perform matching in their highly-complex feature space. In [16], a target and a source graph are constructed from the keypoints and the matching is carried out as a weighted network alignment problem. The correspondences found can then be used to define a coarse alignment between point clouds that will be used as the initial pose in the fine registration step. The iterative closest point (ICP) [21] and its variants have been widely used in SLAM frameworks [4], [9]. Data-driven approaches exist that seek to learn the full registration process from a pair of point clouds [7], [22]. However, following [23] in this work we employ the generalized ICP (GICP) [24] method for fine registration of the bathymetric point clouds in the experimental section.

From the works presented above, only [5], [9] and [16] target bathymetric data, which already indicates the complexity of the problem. However, in [9] the data is collected in an structured environment and in [16] a high-resolution subsea optical laser scanner is used. In our work, we have focused on the problem of keypoint selection and feature descriptors construction in bathymetry collected with an MBES in a large, unstructured environment. Our approach builds upon deep learning techniques, as opposed to the standard computer vision methodology employed in [5].

III. METHODOLOGY

Given a set of N V-dimensional points, $P_i \in \mathbb{R}^{N\times V}$, the goal is to characterize their W-dimensional descriptors $\xi_i \in \mathbb{R}^{N\times W}$ such that the most representative geometric features common across overlapping point clouds result in similar descriptors that can be then matched for loop closure detection. More specifically, we aim to create point descriptors that encapsulate local information and that are robust against noise. Furthermore, in order to quantify their saliency, we seek to define a set of scalars $w_i \in \mathbb{R}^N$ that weight the descriptors' prominence within their environment.

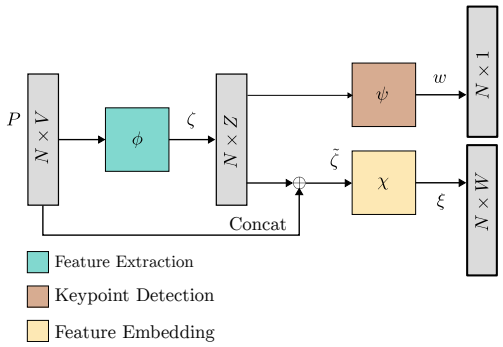

Fig. 1: Our inference network architecture. Point clouds are fed into the network and weights w and descriptors ξ are output. w indicate the saliency of the features and ξ are used for down-stream data association.

Specifically, we make use of the Siamese architecture [25] to learn these descriptors and their weights from P_i.

A. Metric learning

Formally, we aim to regress a mapping of the form $f : P_i \mapsto \xi_i, w_i$ parameterized by a set of variables θ so that the common features between ξ_i and ξ_j, within an overlapping area from a second set of points P_j, are close to each other according to some metric. Additionally, we intend the value of w_i to directly correlate with the saliency of the descriptors within ξ_i. Since the mapping f is intractable, learning such function can be formulated as a metric learning problem [26]. The triplet loss approach [27] to metric learning aims to construct a latent space where similar descriptors are brought closer and different ones farther, thus defining a distance metric that encodes the similarity of the geometric features within the space. This is achieved by working on sets of inputs or $triplets$, which consist of an anchor input (a) randomly sampled, a positive one (p) known to be similar to the anchor and a negative one (n), with nothing in common with the anchor. Eq. 1 shows the standard triplet loss for a triplet $\{a, p, n\}_t$.

$$\mathcal{L}_t = \left[\|\xi_a - \xi_p\|_2 - \|\xi_a - \xi_n\|_2 + \gamma \right]_+, \tag{1}$$

where $[z]_+ = \max(z, 0)$ denotes the hinge loss, $\|\cdot\|_2$ is the L2-norm and γ models the margin enforced between the distances of positive and negative pairs. In our case, the triplets comprise sets of points P_i and the similarity between sets is measured in terms of physical overlap.

B. Proposed architecture

Learning f can be achieved by maximizing the likelihood of its parameters θ for a given dataset of triplets. To this end, we divide the process into three learning modules denoted: i) feature extraction ϕ, ii) keypoint detection ψ and iii) feature embedding χ, as shown in Fig 1.

As a first step, for a given input point set $P_i = \{p_k\}_{k=1}^N$, the deep feature extraction network learns a Z-dimensional feature $\zeta_k \in \mathbb{R}^Z$ for each of the input points. Using the resulting ζ_k the keypoint detector module ψ further

maps each descriptor to its scalar weight $w_k \in \mathbb{R}$, which models the salience of the original point. In parallel, each ζ_k descriptor is then concatenated with the relative position of its corresponding point (with respect to the axis of the point cloud), that is, $\tilde{\zeta}_k = \zeta_k \oplus p_k$, to include geometric information. Using $\tilde{\zeta}_k$, the feature embedding network χ seeks to embed each $\tilde{\zeta}_k$ into a more detailed W-dimensional feature descriptor $\xi_k \in \mathbb{R}^W$ that includes information from the neighbouring points.

The original θ is now composed of three sets of learnable parameters $\theta = \{\theta_\phi, \theta_\psi, \theta_\chi\}$ parameterizing the three neural networks, ϕ, ψ and χ respectively. The original function $f : \mathbb{R}^{N\times V} \rightarrow \mathbb{R}^{N\times W} \times \mathbb{R}^N$ can now be described as follows:

$$f(P_i, \theta_\phi, \theta_\psi, \theta_\chi) = (\pi_1(P_i, \theta_\phi, \theta_\chi), \pi_2(P_i, \theta_\phi, \theta_\psi)) \quad (2)$$

$$\pi_1(P_i, \theta_\phi, \theta_\chi) = \chi(\phi(P_i), P_i) : \mathbb{R}^{N\times V} \rightarrow \mathbb{R}^{N\times W}, \quad (3)$$

$$\pi_2(P_i, \theta_\phi, \theta_\psi) = \psi(\phi(P_i)) : \mathbb{R}^{N\times V} \rightarrow \mathbb{R}^{N}. \quad (4)$$

The details of each module are described below.

1) Deep feature extraction: The hierarchical propagation strategy adopted in [13], including the across level skip links, allows PointNet++ to encode features at different scales of locality in a hierarchical manner. We follow this approach in our feature extraction network and build ϕ with a segmentation module consisting of a sequence of three set abstraction layers and three feature propagation layers. The output of ϕ are the sets of deep features ζ_i, which are concatenated with the positions of the input points in P_i, named $\tilde{\zeta}_i$, for further feature embedding later.

2) Keypoint detection: Inspired by [6] and [7] we use a weighting layer as the keypoint detection network to learn the salience of the features extracted by ϕ. In this work we use the multi-layer perceptron (MLP) applied in DeepVCP, consisting of 3 fully-connected layers, followed by the *softplus* activation function. The inputs to the keypoint detection layer are the sets of deep features extracted above ζ_i and the outputs are sets of scalar w_i that will weight the points in the triplet loss.

3) Deep feature embedding: Following the design of DeepVCP once more, we employ a second deep feature embedding network to learn a more detailed descriptor of local patches for the triplet learning, which effectively helps improve the correspondence matching [7]. Thus, χ is formed by one grouping layer and one PointNet layer (as described in [13]) followed by the normalization in the last step. Local patches are constructed by aggregating neighbouring points using the grouping layer and used to learn a further embedding ξ_i of local areas from the point-wise features $\tilde{\zeta}_i$.

IV. BATHYMETRY LEARNING

In this section we present the methodology followed to train the architecture presented on bathymetric data. Both the NN implementation and the dataset can be found here [1].

[1] https://github.com/tjr16/bathy_nn_learning

A. The bathymetric dataset

The data used in the experiments has been collected as part of a larger mission with a Hugin 3000 AUV equipped with a MBES Kongsberg EM 2040 and without any external navigation aid. The survey was carried out beneath the Thwaites glacier in west Antarctica and the segment used for these experiments covers approximately $10\,\mathrm{km}^2$, surveyed over 6 hours. However, the vehicle had already navigated for approximately 5 hours before reaching this area, which has resulted in accumulated DR drift present in the data.

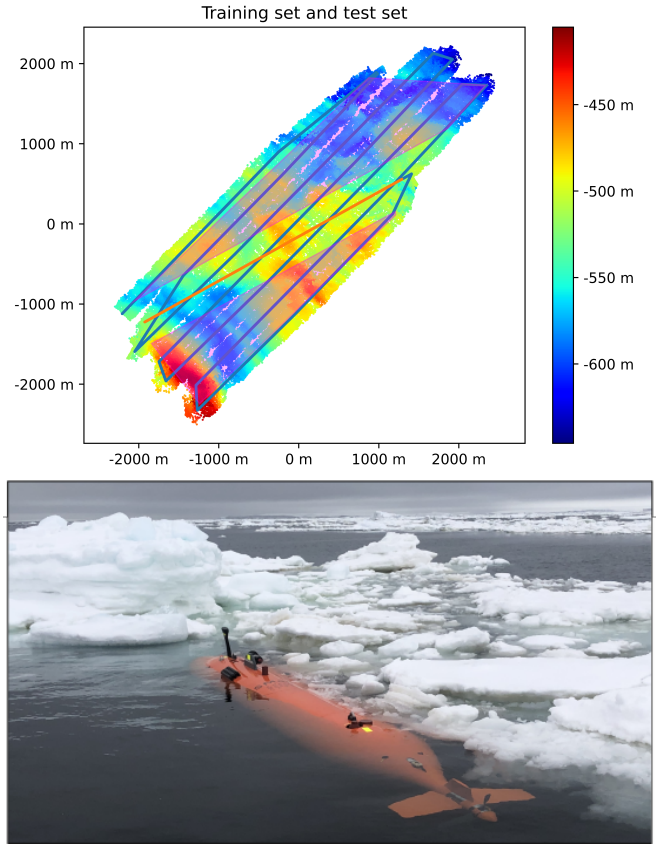

Fig. 2: Bathymetry of the surveyed area used for training and testing (top) and the Hugin AUV after the survey (bottom).

The resulting bathymetry can be seen in the top of Fig. 2. The AUV's DR trajectory estimate has been plotted in blue for the section of the lawn mowing pattern and orange for the revisiting swath. The areas shaded in red roughly outline the sections of the data used in the training set. The swath corresponding to the orange trajectory has been used for the test set. Finally, the validation set is composed of bathymetry collected during the same mission right outside that area. The bottom image in Fig. 2 shows the Hugin after the mission.

B. Generation of triplets

In the ideal case, metric learning for data association should be trained on point clouds collected from two crossing trajectories. However, since we only have one revisiting swath, which we keep for the test set, we have to artificially generate overlapping point clouds for model training and

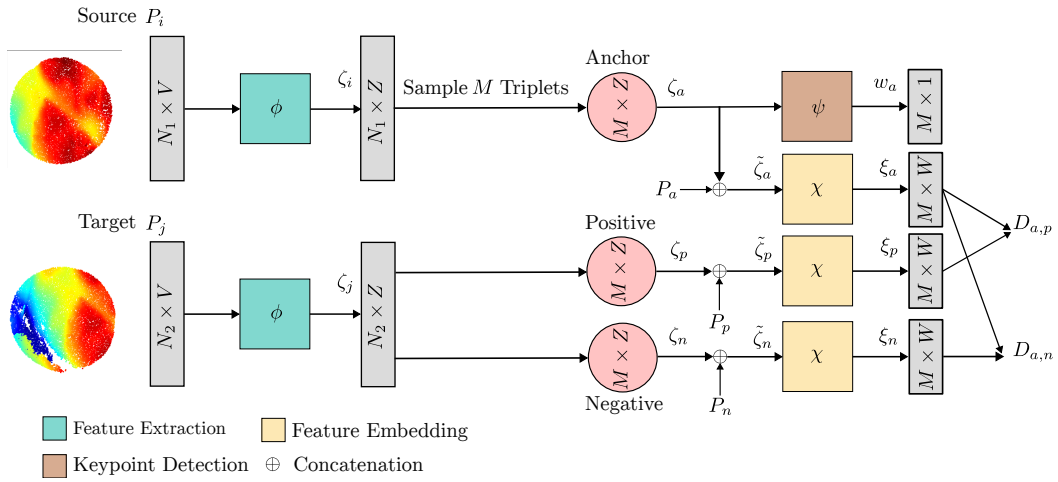

Fig. 3: The Siamese architecture for training. The feature extraction network takes two point clouds (a source point cloud and a target point cloud) as input and extracts informative features ζ, from which we randomly sample M triplets (anchor P_a, positive P_p and negative P_n). The features ζ from the anchor of the sampled triplets is fed into the keypoint detector to learn the saliency. In parallel, we concatenate the relative positions with ζ to get $\tilde{\zeta}$, which is further fed into the feature embedding network to learn the final descriptor ξ. For a triplet, we can compute their corresponding feature descriptors ξ_a, ξ_p, ξ_n, which are used to construct the triplet loss.

validation. We do so by randomly sampling cylindrical point clouds, as in [20], of radius $100\,\mathrm{m}$ from the red-shaded survey areas. The resulting point clouds are grouped into positive and negative pairs based on their overlap, which can only be inferred based on the AUV's DR estimates, since ground truth is not available. A pair is considered positive if their Intersection Over Union (IoU) $\in [0.4, 0.8]$ and negative if they present no overlap. We apply preprocessing and data augmentation steps to the resulting pairs in order to prepare and maximize our limited dataset. First, the point clouds are demeaned, and their absolute positions are kept as labels. We then downsample the point clouds to reduce computational requirements and obtain more uniformly distributed sets of points. Finally, to make our model more robust to errors arising from the vehicle's depth and heading, we perform data augmentation applying the following steps:

1. Each point cloud is disrupted by a random rotation $\alpha \sim \mathcal{U}(-3.0, 3.0)^\circ$ around the z-axis.
2. It is followed by a random translation across the z-axis: $\Delta z \sim \mathcal{U}(-0.2, 0.2)$ m.
3. Finally, each point's coordinates are corrupted by i.i.d Gaussian noise $\delta \sim \mathcal{N}(0, 0.05)$ m.

The test set, however, is created in a different manner to maximize the data available. Following the orange trajectory in Fig. 2, we crop cylindrical point clouds every 10 pings as target point clouds. We also sample source point clouds from nearby blue trajectories in the same way. In total, 209 point clouds are generated following this procedure, from which 123 positive pairs can be extracted. For comparison, we also randomly search for 123 negative pairs among these point clouds. The size and composition of the final dataset are summarized in Table I.

Since we seek to learn local descriptors of the geometric features within the point clouds, we do not use the created point clouds directly to construct the triplets. Instead, we crop local patches of $15\,\mathrm{m}$ radius within the point clouds. Given a positive pair of point clouds in the training set or validation set, the farthest point sampling (FPS) [28] strategy is used to uniformly draw 512 candidate anchor points from the source point cloud. From the 512 samples, up to M anchor points within the overlapping region are randomly selected and a local patch is cropped. This number is constrained by the hardware capacity and can be increased with more powerful resources. For each anchor patch, a positive patch and a negative patch are sampled from the target point cloud. The positive patch is the closest one to the anchor point and their distance is below $1\,\mathrm{m}$. The negative one that does not overlap with the anchor is randomly selected. To avoid cropping incomplete patches on the boundary of point clouds, the sampling is carried out in their inner area.

TABLE I: Datasets

Dataset	Point clouds	Positive pairs	Negative pairs
Training set	1500	5752	-
Validation set	400	1614	-
Test set	209	123	123

C. Training of the network

The training configuration of the architecture presented in Fig. 1 is shown in Fig. 3. It consists of two Siamese networks, namely, both feature extraction networks shown in Fig. 3 have identical weights for the source and target point clouds. The same reasoning applies to the three feature embedding networks for the anchor, positive and negative samples in the triplets. During training, instead of using every point to construct the triplets, we randomly sample $M = 5$ triplets from the source point clouds every epoch

and calculate the triplet loss based on these M triplets. Specifically, we uniformly sample $N_1 = N_2 = 8192$ points as input, whose $V = 4$ dimensions are the combination of the relative positions and the absolute depth encoding. Since PointNet++ ignores the absolute position information of a point cloud, the absolute depth of each point is encoded as an additional feature vector. For every point $p_k \in P$ we concatenate its relative position with $\sin(\frac{\pi}{50} d_k)$, where d_k is the absolute depth. In this manner, the depth information will be encoded into learned feature descriptors while avoiding directly learning absolute depths. The dimensions of the descriptors we use are $Z = W = 32$. Note that we only concatenate the learned features ζ with the relative positions of point p, so the dimension of $\hat{\zeta}$ is $32 + 3 = 35$.

To find the optimal θ, we seek to optimize the loss in Eq. 1 by applying stochastic gradient descent (SGD) to handle large amounts of triplets efficiently during the training. The minibatch size has been set to $|\mathcal{B}| = 16$ for a fair balance between computation constraints and performance. Similarly to [6], we weight each individual loss $\mathcal{L}_t$ by the normalized weight w'_t across the triplets within a minibatch $\mathcal{B}$. Thus, the loss becomes $\mathcal{L} = \sum_{t \in \mathcal{B}} w'_t \mathcal{L}_t$. We use Adam [29] to optimize this loss, with a learning rate of 10^{-5} and a 5% decay per epoch. The final model was trained in a single Nvidia GEFORCE RTX 2080 Ti GPU. It took approximately 2 hours (20 epochs) for the validation loss to converge.

V. EXPERIMENTS

In this section we assess the performance of our solution in the tasks of place recognition and correspondence matching.

A. Autonomous place recognition

We have tested our network on the task of place recognition for loop closure detection on bathymetric point clouds. The keypoint detector is used to select some interest points with high weights, and the corresponding descriptors from the feature embedding layer are used for data association. The matching between the descriptors is carried out applying a brute-force (BF) matcher with cross-check for a large number of candidates [30]. More in detail, the similarity between features is encoded by the Euclidean distance in the latent space learned. Therefore, the BF enforces that either of the two candidate features must be the best match for the other, according to that metric, for a proposed match to be accepted. Afterwards, the correspondences with an absolute depth difference greater than 2 meters are discarded. Finally, loop closure candidates are accepted based on a minimum threshold on the number of matches.

TABLE II: Confusion matrix with LC detections

		Predicted	
		Positive	Negative
Real	Positive	27	96
	Negative	0	123

Fig 4 showcases an instance of correct correspondence matching using the descriptors from our network. For loop

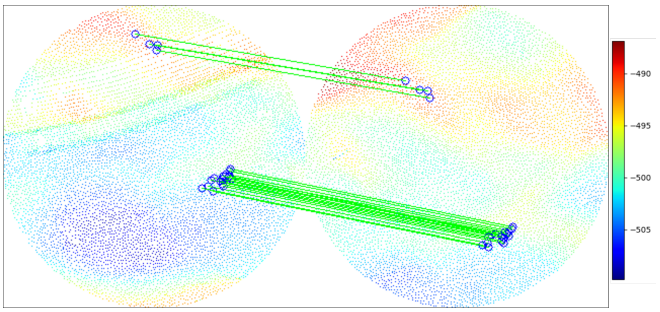

Fig. 4: Instance of correspondence matching using the descriptors from our network. Depth in meters.

closure candidates to be accepted, a minimum threshold of 3 matches has been empirically found to work well, filtering out all false positives. In the presence of more than one LC candidate for a given point cloud, the one with most matches is selected. Table II compiles a confusion matrix with the LC detection results for our method.

To show the complexity of the task under study, we implement as a baseline method the approach presented in [9]. Their pipeline consists in a keypoint detection step based on Harris 3D, followed by the extraction of SHOT descriptors and their clustering via k-means in order to build a bag of words (BoW) per point cloud. The BoW is then used to detect loop closure candidates for point clouds registration. For a fair comparison, the k-means model has been trained on the training set beforehand. Fig 5 summarizes the results from applying the baseline to our bathymetric test set. The cosine similarity has been computed across every BoW for positive and negative matches, resulting in a mean value of 0.4712 for positive matches and 0.4733 for negative ones. This implies that no information has been encoded in the dictionaries and therefore they have not been further used for the LC detection experiment. This result highlights the need for specific methods for MBES point clouds from large, unstructured seabed regions.

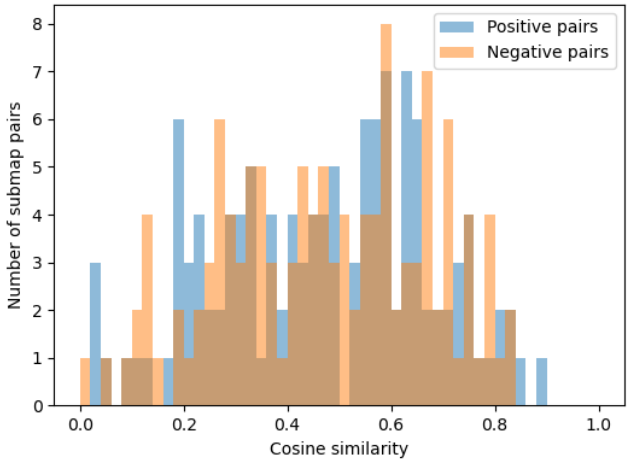

Fig. 5: Cosine similarity of BoW using the approach in [9] for positive and negative pairs of submaps.

B. Coarse and fine registration of bathymetric submaps

We now define a submap as a tuple of the form $S_i = \{P_i, T_i\}$, where $T_i \in SE(3)$ is the absolute pose of the AUV when collecting the data, rigidly attached to the set of points P_i. For a full view of the submap construction process from MBES pings, see [4]. Submap-based SLAM frameworks rely on the registration of overlapping submaps to correct the AUV poses attached to the point clouds upon revisiting an area. In this context, the correspondence matches obtained from a successful loop closure detection can be further utilized to perform a coarse registration of the target and source point clouds, thus correcting the vehicle trajectory. To this end, the singular value decomposition (SVD) of the cross-covariance matrix of the best 20 matches obtained is computed to find the rigid transformation $T_{i,j}^{SVD}$ that coarsely aligns the submaps S_i and S_j [31]. This approach is faster than iterative methods such as least-squares solutions and is robust under reasonable levels of noise. Afterwards, a fine registration is carried out iteratively via GICP in order to maximize the consistency of the final point cloud reconstruction of the target area. Due to the lack of ground truth observations of the seabed areas whose bathymetry we aim to reconstruct, we choose to employ the error metric presented in [32] to assess the performance of our network in a full registration pipeline. This metric gauges the misalignment between overlapping point clouds evaluating the vertical distances between points in the overlapping areas (which should be zero in the case of perfect alignment). After the fine registration, the resulting, corrected relative transformation between AUV poses $T_{i,j}^{GICP}$ can be added as a constraint in a SLAM back-end [33].

TABLE III: Average RMS consistency errors from the DR and the submaps registration experiments

Method	DR	Baseline	Ours
Bathymetric RMS (m)	3.6405	3.2151	1.8861

Table III compiles the results from the registration process introduced above. It presents the average root mean square (RMS) consistency errors across the 27 recognized positive pairs of submaps from the place recognition experiment. It can be seen how the original misalignment between overlapping point clouds, due to the drift in the AUV's DR estimate, has been reduced by approximately 50% with our method, outperforming the results with the correspondences from the baseline method in [9].

Figure 6 depicts the steps in the registration process and the associated bathymetric error being minimized for a pair of submaps from the experiment. On the left, two submaps whose overlap has been detected by the network are shown with their original misalignment caused by the AUV's DR error. On the center, the submaps after a coarse registration with the features detected by the network. On the right, the final relative pose between the submaps after registration.

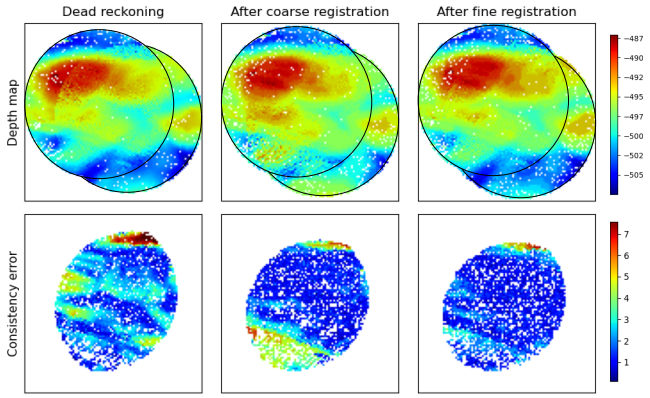

Fig. 6: Depth maps (top) and consistency errors (bottom) before and after registration. Colorbars in meters.

VI. CONCLUSIONS

We have presented a learning framework for detection and description of keypoints that targets the specific challenges of bathymetric raw data for AUV SLAM. Our architecture is built upon [13] and [7] and applies a triplet loss [27] to learn the optimal metric for correspondence matching across point clouds. It has been designed to take the saliency of the points into account and to hierarchically encode local information of the geometric features within the point clouds while remaining resistant to noise. The network has been trained on bathymetric data collected with an AUV equipped with a MBES on a real under-ice mission, deprived of any external positioning system. It has been tested on the tasks of global loop closure detection and coarse submap alignment for AUV trajectory correction. Due to the lack of ground truth positioning of the vehicle, we have employed the consistency metric defined in [32] to assess its performance.

We show how the results presented, although limited to one dataset, outperform the baseline selected [9] on both tasks. In fact, the baseline method is quickly rendered entirely inadequate for loop closure detection in this type of terrain, which illustrates the difficulty of the problem. This reinforces our believe that specific solutions are required for data association in bathymetric data from unstructured seabed environments. Thus, to help advance research on such solutions, we make publicly available both the implementation of our architecture and the bathymetric dataset used. Additionally, the average operation runtime of the network is 0.274 seconds/submap, 2 orders of magnitude faster than applying the method from [16], thus making our framework a more suitable module for a real time SLAM front-end.

ACKNOWLEDGMENT

The authors thank the Knut and Alice Wallenberg foundation for funding MUST, Mobile Underwater System Tools, project that provided the Hugin AUV for these tests. This work was supported by Stiftelsen för Strategisk Forskning (SSF) through the Swedish Maritime Robotics Centre (SMaRC) (IRC15-0046) and by the Wallenberg AI, Autonomous Systems and Software Program (WASP).

REFERENCES

[1] A. G. Graham, A. Wåhlin, K. A. Hogan, F. O. Nitsche, K. J. Heywood, R. L. Totten, J. A. Smith, C.-D. Hillenbrand, L. M. Simkins, J. B. Anderson *et al.*, "Rapid retreat of thwaites glacier in the pre-satellite era," *Nature Geoscience*, pp. 1–8, 2022.

[2] P. Rigby, O. Pizarro, and S. B. Williams, "Towards geo-referenced auv navigation through fusion of usbl and dvl measurements," in *OCEANS 2006*. IEEE, 2006, pp. 1–6.

[3] J. J. Leonard and A. Bahr, "Autonomous underwater vehicle navigation," *Springer handbook of ocean engineering*, pp. 341–358, 2016.

[4] I. Torroba, N. Bore, and J. Folkesson, "Towards autonomous industrial-scale bathymetric surveying," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2019.

[5] M. Hammond, A. Clark, A. Mahajan, S. Sharma, and S. Rock, "Automated point cloud correspondence detection for underwater mapping using auvs," in *OCEANS 2015-MTS/IEEE Washington*. IEEE, 2015, pp. 1–7.

[6] Z. J. Yew and G. H. Lee, "3dfeat-net: Weakly supervised local 3d features for point cloud registration," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 607–623.

[7] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepvcp: An end-to-end deep neural network for point cloud registration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 12–21.

[8] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[9] S. Suresh, P. Sodhi, J. G. Mangelson, D. Wettergreen, and M. Kaess, "Active slam using 3d submap saliency for underwater volumetric exploration," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3132–3138.

[10] I. Sipiran and B. Bustos, "Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes," *The Visual Computer*, vol. 27, no. 11, pp. 963–976, 2011.

[11] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[12] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[14] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 195–205.

[15] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1735–1742.

[16] T. Hitchcox and J. R. Forbes, "A point cloud registration pipeline using gaussian process regression for bathymetric slam," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4615–4622.

[17] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[19] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus," in *European conference on computer vision*. Springer, 2008, pp. 500–513.

[20] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5266–5272.

[21] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[22] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7163–7172.

[23] I. Torroba, N. Bore, and J. Folkesson, "A comparison of submap registration methods for multibeam bathymetric mapping," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE, 2018, pp. 1–6.

[24] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP." in *Robotics: science and systems*, vol. 2, no. 4, 2009, p. 435.

[25] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 539–546 vol. 1.

[26] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification." *Journal of machine learning research*, vol. 10, no. 2, 2009.

[27] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[28] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] F. K. Noble, "Comparison of opencv's feature detectors and feature matchers," in *2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE, 2016, pp. 1–6.

[31] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[32] C. Roman and H. Singh, "Consistency based error evaluation for deep sea bathymetric mapping with robotic vehicles," in *Robotics and automation, 2006. ICRA 2006. Proceedings 2006 IEEE international conference on*. Ieee, 2006, pp. 3568–3574.

[33] I. Torroba, C. I. Sprague, N. Bore, and J. Folkesson, "Pointnetkl: Deep inference for gicp covariance estimation in bathymetric slam," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4078–4085, 2020.